Dr. Yahia Fares University of Médéa



STUDY AND DESIGN OF A 2D CNC PLOTTER.

BENAISSA SIDALI.

YAHIAOUI MOHAMED.

EOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Dr. Yahia Fares University of Médéa



Faculty of Technology
Department of Mechanical Engineering

Dissertation

Submitted in fulfillment of the requirements for a Master's degree in

Mechanical Engineering

Option: Mechatronics

Presented by

* BENAISSA Sid Ali
* YAHIAOUI Mohamed

Theme

STUDY AND DESIGN OF A 2D CNC PLOTTER.

Jury

Promoter Dr.ZEMIRLINE Adel.

Co-promoter Pr.ABUDURA Salam.

Jury President Dr.AMOURA Nasreddine.

First examiner Dr.BOUGHEDAOUI Rachid.


بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Acknowledgments

Praise be to the Almighty God who has given us faith, courage, and patience to carry out this work.

Our deepest appreciation goes to our supervisor, Dr. Zemirline Adel from Medea University, for the unwavering confidence he has placed in us. His constant presence, guidance, humility, advice, and constructive feedback have significantly contributed to the successful progression of this work.

We also thank Pr. Abudura Salam for his co-supervision, offering invaluable guidance, availability, attentive listening, and patience throughout the realization of this project.

Our sincere gratitude is extended to the esteemed examination committee members, Dr. Amoura Nasreddine and Dr. Boughedaoui Rachid. Your dedicated efforts and valuable guidance have had a substantial impact on the success of this research. We appreciate your time and commitment.

We are grateful to Dr. Abdullah Al-Haj for granting us access to his 3D printer. Our thanks also go to the "Schneider Tabilac Berrouaghia" Foundation for providing access to their workshop and equipment, as well as to the workshop supervisor, Youssef Abdul Rahim.

We acknowledge and thank everyone who has contributed to the improvement of our work through their feedback and suggestions.

Our heartfelt thanks are also directed to our parents, brothers, sisters, and entire family for their unwavering encouragement and prayers, which have enabled us to achieve this endeavor. We are deeply grateful for the confidence they have placed in us.

Lastly, we express our appreciation to all those who have contributed in any capacity to the development of this work.

O Allah, send your blessings upon your noble messenger, his family, and companions, and bestow your blessings upon us in our lives.

الإهداء

من قال أنا لها نالها وأنا لها وإن أبت رغما عنها أتيت بها نلتها وعانقت اليوم مجدا عظيما فعلتها بعد أن كانت مستحيلة كانت دروبا قاسية و طرقا خسرت بها الكثير لكني و صلت.

الى العزيز الذي حملت إسمه فخرا يردد اسمي عاليا في عنان السماء حاملا شرف لقبك و بكل اعتزاز أنا لهذا الرجل إبنا الى والدي رحمه الله.

الى من كانت الداعمة الأولى و الأبدية ملاكي الطاهر من كان وجودها يمدني بالسعي دون ملل التي ظلت دعواتها تضم إسمي دائما معلمتي الأولى دكتورتي الأولى، أمي أهديك هذا الإنجاز الذي لولاك لم يكن أهديك مراحلي و إنجازاتي كلها فالفضل و الثناء للمولى ثم لكفاحك لأجلي هنا أنا اليوم أهديك علما و شهادة تخليتي عنها في سبيل رعايتي و تعليمي ممتن لأن الله اصطفاك من بين البشر أما لي.

الى خيرة أيامي و صفوتها الى حافزي الأزلي الى مصدر الشغف و الإرادة الى من آمنوا بي الى إخوتي رعاهم الله بحفظه.

الى كل غالى أردت ذكره اليوم الى الذين يبهجهم نجاحي الى العائلة الكريمة الى كل صديق عزيز الى كل من أعانني يوما ولو بنية صالحة، إلى زملائي في الدراسة.

تحية خالصة الي زميلي محمد يحياوي .

الى صاحب الفضل في كل خطواتنا لإنجاز هذا العمل الى الدكتور الفاضل زميرلين عادل.

BENAISSA SIDALI.♥

الإهداء

إلى من كانوا دوماً سندي ودعمي في مسيرتي العلمية والحياتية، أهدي هذه المذكرة.

إلى زوجتي العزيزة، التي كانت مصدر قوتي وإلهامي، شكراً لكِ على صبركِ ودعمكِ المستمر وحبكِ الذي لا ينضب. إلى أطفالي الأعزاء، الذين هم فرحة حياتي ودافع إصراري، أتمنى أن تكونوا فخورين بوالدكم.

إلى والدي ووالدتي، اللذين لم يتوانيا يوماً عن تقديم النصائح والدعم والتشجيع، لكم كل الحب والتقدير.

إلى إخوتي وأخواتي، الذين كانوا دوماً بجانبي في كل خطوة من خطواتي، شكراً لكم على كل ما قدمتموه لي.

إلى أصدقائي الأعزاء، الذين كانوا معي في اللحظات الصعبة والجميلة، أشكركم على صداقتكم الحقيقية وتشجيعكم المستمر.

تحية خالصة الى من شاركني عملي سيدعلي بن عيسى.

إلى زملائي في الدراسة، الذين شاركوني هذه الرحلة الأكاديمية، شكراً لكم على الدعم والمساعدة التي لم تتوقف.

أهدي هذا العمل لكم جميعاً، فقد كنتم جزءاً لا يتجزأ من هذا الإنجاز.

بكل الحب والتقدير.

YAHYAOUI MOHAMED. ♥

الملخص

يهدف هذا المشروع البحثي إلى تطوير آلة رسم صغيرة تعمل بتقنية CNC لتطبيقات الرسم ثنائي الأبعاد باستخدام تقنية Arduino. تم تصميم هذه الآلة لتحريك القلم بدقة عبر سطح مستوٍ لإنشاء رسومات وتصاميم معقدة بناءً على مدخلات رقمية. يتضمن المشروع جوانب التصميم الميكانيكي والدوائرة الإلكترونية والبرمجة اللازمة لبناء آلة الرسم.

ركز التصميم الميكانيكي على إنشاء إطار مستقر، ودمج محركات الخطوة لتحقيق حركة دقيقة، وتنفيذ آلية رفع القلم. تشمل الدائرة الإلكترونية توصيل لوحة Arduino بمحرك خطوة بخطوة، ومشغل المحرك، ومصدر الطاقة لتحقيق حركة متحكم فيها لآلة الرسم. تتضمن البرمجة كتابة كود لتفسير أوامر G-code وتوليد إشارات التحكم في المحرك لمحركات الخطوة. يهدف هذا المشروع إلى تقديم دليل شامل لبناء آلة رسم صغيرة تعمل بتقنية CNC لتطبيقات الرسم ثنائي الأبعاد.

الكلمات المفتاحية : آلة رسم ثنائية الأبعاد "CNC"، Arduino، التصميم الميكانيكي، محرك خطوة خطوة ، مشغل المحرك.

Abstract

The purpose of this research project is to develop a mini CNC machine plotter for 2D drawing applications using Arduino technology. This plotter is designed to move the pen precisely across a flat surface to create complex drawings and designs based on digital input. The project includes the mechanical design, electronic circuitry, and programming aspects needed to build a plotter.

The mechanical design focused on creating a stable frame, integrating stepper motors for precise movement, and implementing a pin-lifting mechanism. The electronic circuit involves connecting an Arduino board to a stepper motor, motor driver, and power supply to achieve controlled movement of the plotter. Programming involves writing code to interpret G-code commands and generate motor control signals for stepper motors. also. The purpose of this project is to provide a comprehensive guide to building a mini CNC machine plotter for 2D drawing applications.

Keywords: CNC plotter 2D machine, Arduino, mechanical design, stepper motor, motor driver.

Résumé

Le but de ce projet de recherche est de développer un mini traceur CNC pour des applications de dessin 2D en utilisant la technologie Arduino. Ce traceur est conçu pour déplacer le stylo avec précision sur une surface plane afin de créer des dessins et des conceptions complexes basés sur des entrées numériques. Le projet comprend la conception mécanique, les circuits électroniques et les aspects de programmation nécessaires pour construire un traceur.

La conception mécanique s'est concentrée sur la création d'un cadre stable, l'intégration de moteurs pas à pas pour un mouvement précis, et la mise en œuvre d'un mécanisme de levée du stylo. Le circuit électronique implique la connexion d'une carte Arduino à un moteur pas à pas, un pilote de moteur et une alimentation électrique pour obtenir un mouvement contrôlé du traceur. La programmation consiste à écrire du code pour interpréter les commandes G-code et générer des signaux de contrôle moteur pour les moteurs pas à pas. Le but de ce projet est de fournir un guide complet pour construire un mini traceur CNC pour des applications de dessin 2D.

Mots clés : Traceur CNC 2D, Arduino, conception mécanique, moteur pas à pas, pilote de moteur.

Contents

General Introduction……1

1 Chapter I Comprehensive Overview of CNC Machines and Plotters. 4

1.1 Introduction....4

1.2 Part One: CNC Machines. 4

1.2.1 Introduction....4

1.2.2 Definition....4

1.2.3 The Benefits of CNC Machining....5

1.2.4 The Different Types of CNC Machines....5

1.3 Part Two: CNC Plotters. 10

1.3.1 Definition of CNC plotter....10

1.3.2 The main types of CNC plotters....10

1.4 Why We Chose This Project....14

1.5 Conclusion....14

2 Chapter II Design and Building. 16

2.1 Overview....17

2.2 Components....17

2.2.1 Mechanics....17

2.2.2 Electronics....17

2.2.3 Other....17

2.3 Software Setup....17

2.3.1 Arduino IDE....17

2.3.2 Inkscape....18

2.3.3 Processing....18

2.4 Mechanical design....19

2.4.1 SOLIDWORKS....19

2.4.2 Frame....20

2.4.3 Linear Actuators (X and Y Axis)....24

2.4.4 Pen Holder (with optional pen up/down mechanism)....26

2.4.5 Drawing surface....29

2.4.6 Mechanical Assembly....30

2.5 Electronics and Control Part....30

2.5.1 Control part....30

2.5.2 Actuators....40

2.5.3 Electrical Assembly 50
3 Chapter III Operation and use 52
3.1 How To Operate This Machine 52
3.1.1 The First Step 53
3.1.2 The second step 54
3.1.3 The Third step 55
3.1.4 Upload The Code: 58
3.1.5 The Fourth Step 60
3.2 Testing and Troubleshooting 63
3.2.1 Test Runs 63
3.2.2 Troubleshooting 64
General Conclusion 67
Bibliography 69
Appendix I
Appendix A II
Appendix B XVI

Table of Figures

Figure 1:machine operators.[1.1] 5
Figure 2: CNC Milling Machine.[2.2] 6
Figure 3: CNC Turning.[3.3] 6
Figure 4: CNC Router.[4.4] 7
Figure 5: CNC Plasma Cutter.[5.5] 8
Figure 6: CNC Electric Discharge Machine.[6.6] 8
Figure 7: CNC Laser Cutter.[7.7] 9
Figure 8: Electrostatic Plotters. 10
Figure 9: Pen plotters.[8.8] 12
Figure 10: creating printed circuit boards PCB.[9.9] 14
Figure 11: Model CNC Plotter.[10.10] 16
Figure 12: Model CNC Plotter.[11.11] 16
Figure 13: Arduino IDE. 18
Figure 14: Inkscape Logo. 18
Figure 15:Processing Logo. 19
Figure 16: SolidWorks Logo. [11] 19
Figure 17: The first plate of the structure. 20

Figure 18: The second plate.....21
Figure 19: assemble the frame parts. 22
Figure 20: The supports that will hold the installation of the stepper motors. 23
Figure 21: The frame after installing the supports.....23
Figure 22: DVD, CD drives -stepper Motor.[12] 24
Figure 23:DVD drives, Stepper Motor. 25
Figure 24: DVD drives, Stepper Motor on the frame.....26
Figure 25: Pen holder base. 27
Figure 26: Pen holder support.....28
Figure 27:Servo Motor. 29
Figure 28: the pen holder.[13] 29
Figure 29: Drawing surface. 30
Figure 30:Final mechanical assembly. 30
Figure 31: Arduino Logo. 31
Figure 32: Arduino Uno Board.....32
Figure 33: Arduino Uno Pin.[14].....33
Figure 34: The Atmega 328P Pin.[15].....35
Figure 35: The Atmega 328P .[15] 35
Figure 36: Power jack cable. 36
Figure 37: L293D Board.....37
Figure 38: L293D pinout.[16].....38
Figure 39: Electric Motors.....41
Figure 40: Servo motor.....42
Figure 41: Servo motor pulses.....43
Figure 42: Servo motor Pinout. 44
Figure 43: Stepper Motors.....45
Figure 44: Cross-Section of a Stepper Motor.....45
Figure 45: Stepper Motor Steps.....46
Figure 46: Stepper Motors.....46
Figure 47: Linear Stepper Motors in DVD Players. 47
Figure 48: Rotary Stepper Motors in DVD Players.....47
Figure 49:Conceptual Model of Bipolar stepper Motor. 48
Figure 50: Performance curve comparison with approximately same volume.[21].....49
Figure 51: Full circle. 50
Figure 52: Arduino Uno & L293D shield board. 53
Figure 53: Connecting stepper motors and servo motors to the L293D shield. 53
Figure 54: Drawing Template.....54
Figure 55: Define the Shape According to the Template. 54
Figure 56: Save the File as G-code.....55
Figure 57: The G-code File.....55
Figure 58: Determine the type of Arduino board. 58
Figure 59: Determine the connected port. 59
Figure 60: Enter the code.....59
Figure 61: upload the code. 59
Figure 62: Processing interface. 60
Figure 63: Enter the code.....60
Figure 64: Run the CNC machine. 61

Figure 65: Select serial port. 62
Figure 66: Select the G-Code file. 62
Figure 67:Control UI messages. 63
Figure 68: dragon logo. 63
Figure 69: dragon logo drawing our CNC plotter. 64
Figure 70: An image showing the perpendicularity of the axle holders. 64
Figure 71: Ensure that the poles of the stepper motors are soldered to the connecting wires. 65
Figure 72: AFmotor library. 66
Figure 73: The pin must be removed. 66

Table 1: L293D Pin Configuration Details.[16] 39

Diagram 1: Block Diagram for the system based on Arduino UNO and L293D shield. 52

General Introduction

In modern manufacturing and digital fabrication, CNC (Computer Numerical Control) technology has revolutionized how we approach precision and efficiency. This project focuses on developing a mini CNC plotter for 2D drawing applications. By leveraging components from old DVD drives and integrating Arduino technology, this plotter aims to provide a cost-effective yet highly functional solution for creating complex designs with precision.

Background and Motivation

The evolution of manufacturing from manual craftsmanship to automated production lines has significantly increased efficiency and reduced costs. The 1970s and 1980s marked a pivotal shift towards flexible manufacturing systems, capable of handling diverse product ranges and small batch production. CNC technology emerged as a cornerstone of this revolution, offering unparalleled precision, consistency, and versatility across various industries, including aerospace, automotive, and electronics.

CNC Technology Overview

CNC machines automate the control of machining tools through pre-programmed software, translating digital designs into precise physical movements. This process enables the creation of intricate and highly accurate parts, which would be challenging to achieve manually. Common types of CNC machines include milling machines, lathes, routers, plasma cutters, and plotters, each serving specific functions and applications.

Project Scope and Objectives

This research project aims to build a mini CNC plotter for 2D drawings, using stepper motors for the X and Y axes and a servo motor for the Z-axis (pen holder). The project encompasses the following key components:

- Mechanical Design: Constructing a stable and precise frame, integrating stepper motors, and designing a pen-lifting mechanism.

- Electronic Circuitry: Connecting an Arduino board to the stepper motors and motor driver, ensuring controlled movement.

- Programming: Writing code to interpret G-code commands and generate motor control signals, enabling the plotter to accurately reproduce digital designs.

Significance and Applications

The mini CNC plotter developed in this project serves as a comprehensive educational tool, demonstrating the integration of mechanical, electronic, and programming disciplines. It provides a foundation for various applications, such as:

- Educational Purposes: Teaching the fundamentals of CNC technology, robotics, and automation.

- Prototyping: Creating precise prototypes for design and engineering projects.

- Artwork: Producing intricate and detailed drawings for artistic endeavors.

- PCB Manufacturing: Assisting in fabricating printed circuit boards by precisely plotting circuit layouts.

By harnessing readily available components and open-source software, this project not only showcases the potential of CNC machines but also contributes to the broader understanding and accessibility of CNC technology. The mini CNC plotter shows how innovation in mechatronics can lead to practical and impactful solutions in various fields, from education to industrial applications.

Chapter I

Comprehensive Overview of CNC Machines and Plotters.

1 Chapter Ⅰ Comprehensive Overview of CNC Machines and Plotters.

1.1 Introduction

The chapter is split into two sections. The first part will provide a detailed overview of CNC machines, including different types, their applications, and their significance.

In the second part, we will delve into CNC Plotters and explore the popular variations. Additionally, we will highlight the key areas where they are utilized and their intended functions.

Finally, we will mention the reasons that led us to choose this project.

1.2 Part One: CNC Machines.

1.2.1 Introduction

During the transition from handicraft production in the 19th century to early 20th-century mass production, significant changes in manufacturing systems emerged. A defining feature of this era was the adoption of dedicated conveyor assembly lines, which enhanced efficiency and reduced labor costs in mass production.

The industrial landscape of the 1970s and 1980s witnessed a shift towards broader production capabilities to accommodate diverse part ranges. This demand led to the development of "flexible" production systems capable of manufacturing small batches of varied products. These systems utilize groups of reprogrammable computer numerical control (CNC) machines, integrating them with automated transport and storage systems.

CNC technology, short for computer numerical control, has become indispensable in modern manufacturing. These advanced machines are computer-driven, delivering unmatched levels of efficiency, precision, and consistency that surpass manual processes.

CNC machining has enabled operations that were previously considered unachievable, fundamentally transforming manufacturing capabilities. Moreover, the diversity of parts producible by CNC machines is complemented by a variety of machine types designed to suit specific manufacturing tasks, each distinguished by its unique design, operational methodology, and production capabilities.[1]

1.2.2 Definition

A computer numerical control (CNC) machine is a manufacturing tool that uses preprogrammed computer software to control machine and tool motion. Essentially, it automates the operations of machine tools such as Turning and milling machines, and grinders.

CNC machines interpret specific instructions encoded in computer-aided design (CAD) files and convert them into precise movements of machine parts. This enables high-precision and complex cuts, shapes, and operations that would be difficult or impossible to achieve manually. CNC machine tools are widely used in various industries such as manufacturing, aerospace, automotive, and woodworking to streamline production processes and increase efficiency.[2]

1.2.3 The Benefits of CNC Machining

The use of CNC machine tools guarantees the uniformity of product quality due to their high precision. This method surpasses manual editing in terms of accuracy and can be replicated consistently. It also leads to enhanced production speed and efficiency.

By employing computers to control machines and automate processes, production speed and quality are significantly increased. Moreover, the advantage of machines not experiencing fatigue or requiring breaks makes them more efficient compared to manual processing.

One of the main benefits for CNC machine operators is safety. Using CNC machines is much safer than manual machines because they work behind guards or even enclosed transparent safety doors.

CNC machining is more cost-effective. Traditional machining requires one operator per machine. This type of processing requires fewer machine operators because an experienced operator can operate multiple machines simultaneously. Because CNC is so precise, it reduces errors in the manufacturing process and eliminates unnecessary waste.[3]



Figure 1:machine operators.[1.1]

1.2.4 The Different Types of CNC Machines

These impressive machines are designed to produce a wide variety of items. Therefore, different types of CNC machine tools are commonly used.

All of the following machines use G-code, which is a language that CNC machines understand. Each type of CNC machine tool serves a specific purpose.

1.2.4.1 CNC Milling Machine

CNC milling machines are one of the most common types of CNC machine tools, which use computer control to cut a variety of materials. Milling machines use specific numerical and alphabetic programs to move the spindle in different ways.

Multiple programming languages are recognized by most CNC machines. These machines can perform a variety of functions such as face milling, shoulder milling, thread cutting, drilling, and turning. Most of them are available in three- to six-axis configurations.

CNC milling machines are very large compared to other tools and can be very expensive. CNC milling machine manufacturers include Okuma, HAAS, and DMG Mori. The factory uses so-called G-code, which, as already mentioned, is standardized.[2]



Figure 2: CNC Milling Machine.[2.2]

1.2.4.2 CNC Lathe (Turning)

A lathe is a CNC machine tool that cuts a workpiece during the turning process. CNC lathes can make precise cuts quickly using a variety of cutting tools.

These CNC machines have very high precision compared to manual lathes. They typically have fewer axes than CNC mills, making them smaller and more compact.

CNC lathes have similar controls to CNC mills and can read G-code and other proprietary programming languages. [2]



Figure 3: CNC Turning.[3.3]

1.2.4.3 CNC Router

A computer numerical control (CNC) router is a machine that bears resemblance to the handheld router commonly used for cutting different materials. This type of CNC machine is capable of cutting steel, wood, aluminum, composites, plastic, and foam.

CNC routers are similar to CNC mills as they utilize computer numerical control to create tool paths that allow the machine to operate. By producing various items in a shorter period, CNC routers help reduce waste and increase productivity significantly.

Most routers can work on a specific material in all three dimensions, making them ideal for smaller projects, prototype models, and intricate designs. Additionally, there are three-axis, four-axis, five-axis, and six-axis routers available.[2]



Figure 4: CNC Router.[4.4]

1.2.4.4 CNC Plasma Cutter

Plasma cutting utilizes a plasma torch to effectively cut through various materials, with a primary focus on heavy materials like steel and metal shapes.

In a CNC plasma cutting machine, gas is ejected from a nozzle at very high speeds. The gas ejected from the nozzle creates an arc on the surface to be cut.

This converts some of the gas into plasma, which has temperatures ranging from 10,000 to 50,000 degrees. The plasma is hot enough to melt the material being cut and blow the molten metal away from the interface.

In terms of shape and size, CNC plasma cutters are very similar to CNC milling machines. These machines only work with two-dimensional shapes.[2]



Figure 5: CNC Plasma Cutter.[5.5]

1.2.4.5 CNC Electric Discharge Machine

The CNC Electric Discharge Machine, also known as EDM, is a method used to shape materials by utilizing electrical discharges or sparks.

This process involves the removal of material from a workpiece through a series of electrical discharges between two electrodes. These electrodes are positioned in a dielectric fluid, which is often charged with electricity. The machine calculates the necessary amount of electrical discharge for each electrode to achieve the desired outcome.[2]



Figure 6: CNC Electric Discharge Machine.[6.6]

1.2.4.6 CNC Laser Cutter

The CNC laser cutter is similar to CNC plasma machines, with the primary distinction being the use of lasers predominantly for cutting various materials such as metals, plastics, or hardwood.

Laser cutting offers exceptional precision and is highly effective for intricate designs and fine details.

The laser's intensity can be precisely adjusted based on the material's density and strength, allowing for optimal cutting results across a range of applications and materials.[2]



Figure 7: CNC Laser Cutter.[7.7]

These are some of the most common CNC machines in various manufacturing fields.

There is also another type that is popular in several fields, which is the CNC plotter.

1.3 Part Two: CNC Plotters.

In this work, we will learn specifically about its types and the most prominent fields in which it is used. to accomplish a study of how to build a CNC plotter, from design to manufacturing and completion.

1.3.1 Definition of CNC plotter

A plotter represents a sophisticated computer output device designed specifically for producing precise, high-quality graphics and drawings at a large scale.

This device operates by handling a pen or similar drawing implement to render lines, curves, and shapes on paper or other media. Unlike standard printers that generate raster images, plotters produce vector-based output, ensuring the creation of sharp and exceptionally accurate drawings.[4]

1.3.2 The main types of CNC plotters

1.3.2.1 Electrostatic Plotters

Electrostatic plotters, also called drum plotters, use charged toner particles to create images on paper. The charged drum attracts toner particles and transfers them to the paper. These plotters produce high-speed output and are commonly used for architectural and engineering drawings.[4]



Figure 8: Electrostatic Plotters.

As we mentioned before, an Electrostatic plotter is a type of plotter that draws images on paper using an electrostatic process. These plotters are most frequently used for “Computer-Aided Engineering (CAE)”, producing raster images via either a “Liquid Toner” or a “Dry Toner” model. Here's how they work:

Liquid Toner Models

-These models use a toner that is "positively charged" and becomes attracted to the paper that is negatively charged.

-The toner particles pass through a line of electrodes in the form of tiny wires or nibs.

- The spacing of the wires controls the resolution of the plotter (e.g., 100 or 400 wires per inch).

- Liquid Toner models can print in "black and white" or "color".

Dry Toner Models

- Dry Toner models use a process similar to "xerography" in photocopiers.

- Unlike laser printers or photocopiers, there is "no transfer drum" used in most electrostatic plotters.

- The imaging paper is directly exposed to the charging electrode array.

- These models can also be printed in black and white or color.

1.3.2.1.1 Image Quality and Speed

-The image quality produced by some electrostatic plotters was "lower" than that of contemporary pen plotters.

- However, the increased speed and economy made them useful.

- Unlike pen plotters, the plot time of a rasterized electrostatic plotter was "independent of the level of detail" of the image.

1.3.2.1.2 Modern Applications

- Modern electrostatic color plotters are found in the "short-run graphics industry".

- They can print on a variety of paper or plastic film surfaces.

1.3.2.1.3 Historical Context

Electrostatic plotters were known in the early days of computer graphics, and by "1967", several manufacturers commercially supplied them.

In summary, electrostatic plotters play a crucial role in producing visual representations for engineering and graphics applications. Whether it's creating detailed schematics or colorful designs, these plotters continue to be relevant in various fields. [5]

In our research, we will focus more on pen plotters.

1.3.2.2 Pen Plotters

Pen plotters use a pen or a set of pens to draw continuous lines in the drawing area. The plotter mechanism moves the pen along the x and y axes to create precise shapes and curves. These plotters are known for their accuracy and the ability to produce detailed drawings.[6]

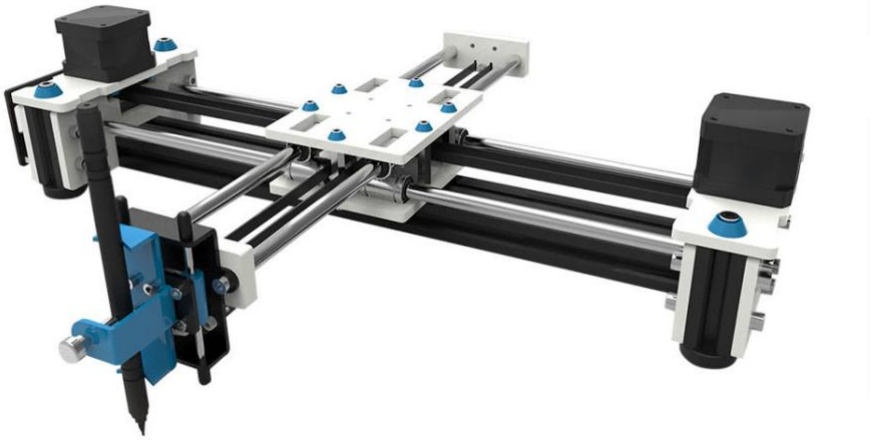


Figure 9: Pen plotters.[8.8]

1.3.2.2.1 The advantages of using a plotter

Plotter usage provides numerous benefits compared to alternative printing methods:

Precision

Plotters are capable of producing highly accurate and detailed drawings with precise line weights, making them well-suited for technical and engineering purposes.

Large-format capability

Plotters are specifically designed to accommodate large paper sizes, enabling the creation of drawings and graphics on a larger scale.

Versatility

Plotters can work with a variety of media types, such as paper, vellum, film, and even fabrics, offering flexibility for different design needs.

Longevity

Plotters typically utilize high-quality pens or toner that are resistant to fading and can maintain their quality over time, ensuring the durability of your drawings.

Vector-based output

Plotters produce vector-based output, allowing drawings to be scaled without losing resolution or quality, making them ideal for architectural and engineering projects.[6]

1.3.2.2.2 Some common applications of plotters

Plotters have a wide range of applications in various industries, such as:

Architecture and Engineering

Plotters are commonly utilized in architecture and engineering firms to generate detailed blueprints, floor plans, and construction drawings.

Computer-aided design (CAD)

Plotters are essential components of CAD systems, allowing designers and engineers to convert digital designs into physical drawings.

Geographic information systems (GIS)

Plotters are integral to GIS applications for creating maps and cartographic visualizations with accurate geographical information.

Fashion and Textile Design

Plotters are employed in the fashion industry to produce patterns, textile designs, and templates for clothing and fabric manufacturing.

Signage and Graphics

Plotters are utilized for producing large-scale signage, banners, posters, and vehicle wraps that require high-quality and precise graphics.[6]

creating printed circuit boards (PCBs)

A Printed Circuit Board (PCB) is an electronic component that utilizes copper conductors to establish electrical connections among various parts. PCBs also offer structural support for electronic components, enabling them to be installed within a casing.

PCBs are constructed using layers of conductive copper alternating with layers of electrically insulating material. The conductive elements on PCBs consist of copper traces, pads, and conductive planes. The mechanical framework is formed by the insulating material sandwiched between the conductor layers. The entire structure is plated and coated with a non-conductive solder mask, with a silk screen printed on top of the mask to indicate the placement of electronic components. Following these manufacturing processes, the bare board is subjected to printed circuit board assembly, during which components are soldered onto the board and the PCBA undergoes testing.

Using a CNC plotter for creating PCBs (Printed Circuit Boards) can be a very effective and precise method.[7]



Figure 10: creating printed circuit boards PCB.[9.9]

During our research for a graduation project and thesis, we discovered that this is one of the most important areas of using the CNC plotter.

1.4 Why We Chose This Project

Based on our field of study (Mechatronics) and in the context of searching for a project idea to complete our graduation thesis, we aimed to choose a research topic that encompasses the disciplines of our specialization, including mechanical, electronic, and programming fields.

Therefore, we initially chose a CNC Machine, which involves conducting a mechanical study of the machine and also developing electronic circuits. Additionally, we often perform a programming study for it using various programming languages based on the project's plan.

After extensive research, we settled on the Mini CNC Plotter 2D as the project idea. This choice was made based on thorough investigation and following a clear methodology with specific criteria. we concluded that building and completing this project would be feasible in terms of equipment and software availability, as well as within a carefully planned budget. Furthermore, this project serves as a foundation for several innovative applications that are expected to see increased demand in the market in the future.

These are the key considerations that led us to finalize this project idea for our graduation thesis.

1.5 Conclusion

After we have explained the purpose of the CNC Plotter, and as we mentioned at the beginning, we will design a prototype of this device to achieve the maximum possible effectiveness with the lowest cost.

In the next chapter, we will learn how to build a CNC Plotter machine.

Chapter II

Design and Building.

2 Chapter Ⅱ Design and Building.

CNC plotter 2D machines vary based on their mechanical design, external structure construction, electrical design, and control of axle motors. The connection between the motors is also a distinguishing factor. Various programming methods can be used to control the machine.



Figure 11: Model CNC Plotter.[10.10]

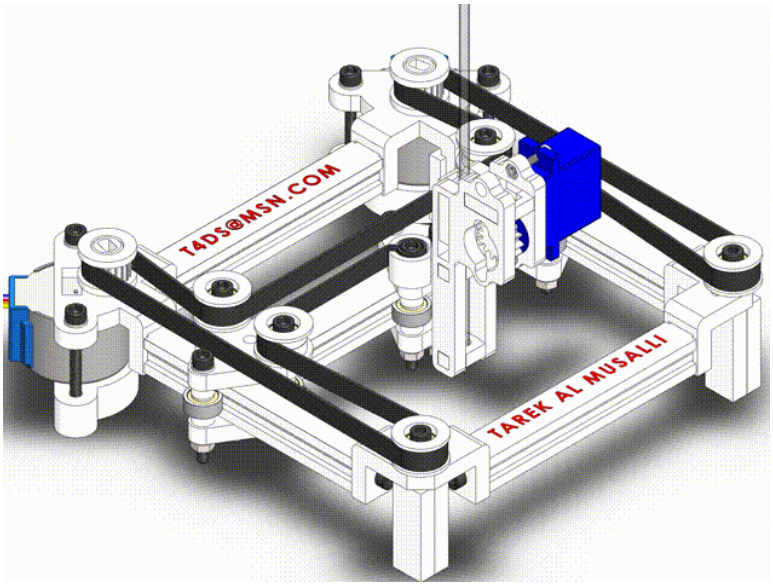


Figure 12: Model CNC Plotter.[11.11]

We will use stepper motors from a DVD player for the CNC plotter depending on the timeframe and capabilities.

We will design the mechanics to accommodate these motors and assess the drawing space.

Furthermore, we will explore the control and programming techniques required to achieve the goals of CNC plotters.

2.1 Overview

This (mini CNC plotter) is made from an old DVD drive and an L293D motor shield.

It's a low-cost solution that can draw any image uploaded from a PC.

The plotter operates in the X-Y plane, allowing you to create intricate designs.

2.2 Components

2.2.1 Mechanics

Frame: We can use pre-cut sheet metal, MDF wood, or 3D-printed parts.

Pen Holder (with optional pen up/down mechanism).

Drawing Surface (paper, cardboard, etc.).

2.2.2 Electronics

Arduino Uno board.

Motor Driver (L293D shield or similar).

Servo Motor (for pen lift mechanism).

Linear Actuators (X and Y Axis): Repurposed DVD Drives (stepper motors and lead screws).

2.2.3 Other

Jumper wires.

Various fasteners (screws, nuts, etc.).

2.3 Software Setup

2.3.1 Arduino IDE

The Arduino IDE, an 'Open Source' software application, is utilized for writing and uploading code to Arduino boards. It offers a user-friendly interface for compiling, writing, and uploading code to Arduino microcontrollers. The IDE is compatible with the Arduino programming language, which is derived from C/C++. Additionally, it features a text editor, message area, toolbar with common function buttons, and various menus. This application is accessible for Windows, Mac, and Linux operating systems and can be obtained from the Arduino website.[8]



Figure 13: Arduino IDE.

2.3.2 Inkscape

Inkscape is a vector graphic editor that is available as open-source software. It is capable of creating and editing various types of vector graphics, including models, diagrams, line arts, charts, logos, and intricate paintings. One of its key features is the ability to generate ‘g-code’ for printing images.

Despite its powerful capabilities, Inkscape does not demand a modern, high-performance computer to function effectively. It offers various drawing tools, filters, effects, and patterns.

Inkscape is a popular choice among professionals for editing vector images, seamlessly integrating into design workflows alongside open-source and proprietary tools.[9]



Figure 14: Inkscape Logo.

2.3.3 Processing

The software used in this project is open-source. The Processor software is utilized to transmit the G-code file to the plotter or Arduino via serial communication.[10]

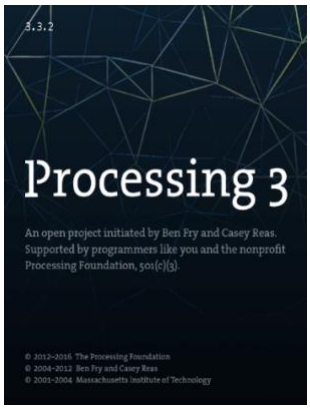


Figure 15:Processing Logo.

2.4 Mechanical design

In this section, we shall elucidate the mechanical blueprint of the plotter. Our primary emphasis will be on formulating a suitable framework for the selected motors (specifically stepper motors extracted from a DVD Player). In addition, we will systematically delineate the X and Y axes, while also devising a pen holder.

Additionally, we use the Solidworks software for modeling of the chassis components and other elements of the CNC plotter.

2.4.1 SOLIDWORKS

SolidWorks, a brand under Dassault Systèmes, specializes in the development and distribution of software for solid modeling computer-aided design (CAD), computer-aided engineering (CAE), 3D CAD design, collaboration, analysis, and product data management (PDM).

Engineers and designers commonly utilize SolidWorks to model finalized 2D and 3D engineering drawings and models through a parametric feature-based method.

With capabilities such as Finite Element Analysis for simulations and expedited modeling, SolidWorks is an essential tool in the field of product design and development.[11]



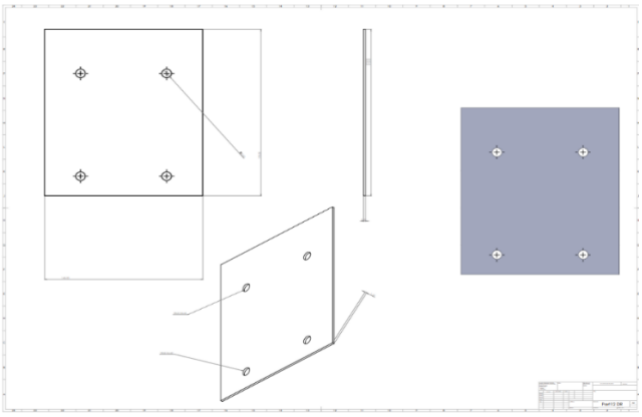
Figure 16: SolidWorks Logo. [11]

2.4.2 Frame

It is best to use lightweight and solid materials such as aluminum or wood to design a simple structure that accommodates the X and Y axes.

In our structure, solid and lightweight metal sheets are used.

The first plate will hold the y-axis.





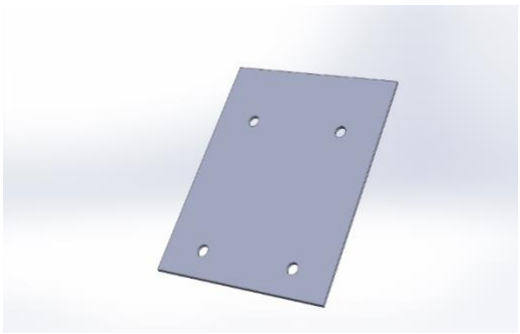
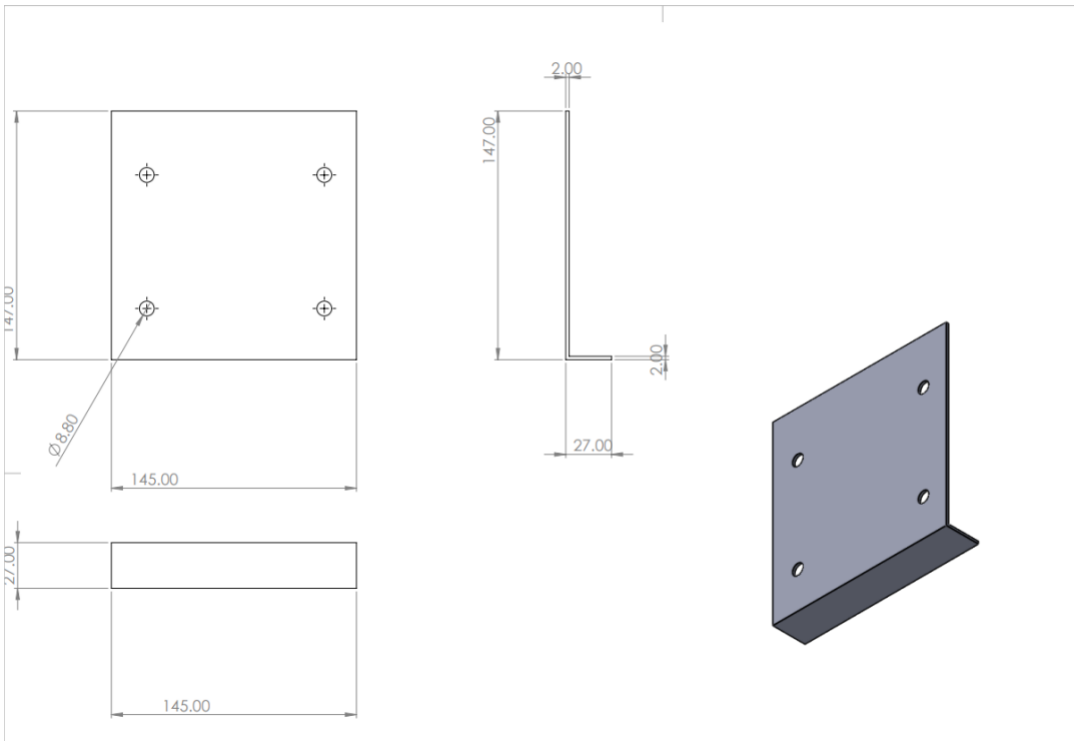


Figure 17: The first plate of the structure.

The second plate:

The second plate will hold the x-axis.





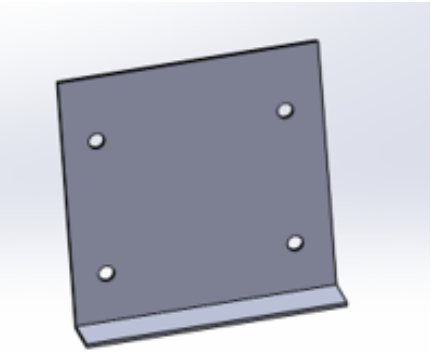


Figure 18: The second plate.

Now we will assemble the frame parts:

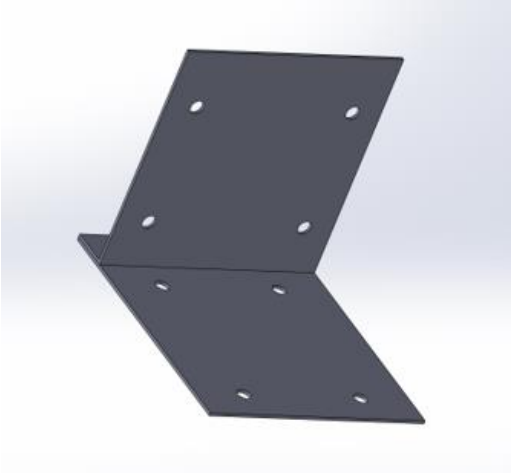
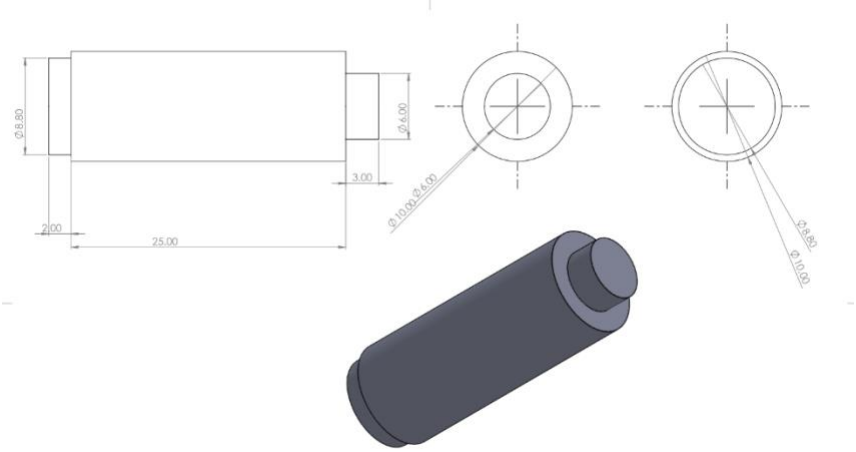


Figure 19: assemble the frame parts.

The supports that will hold the installation of the stepper motors.



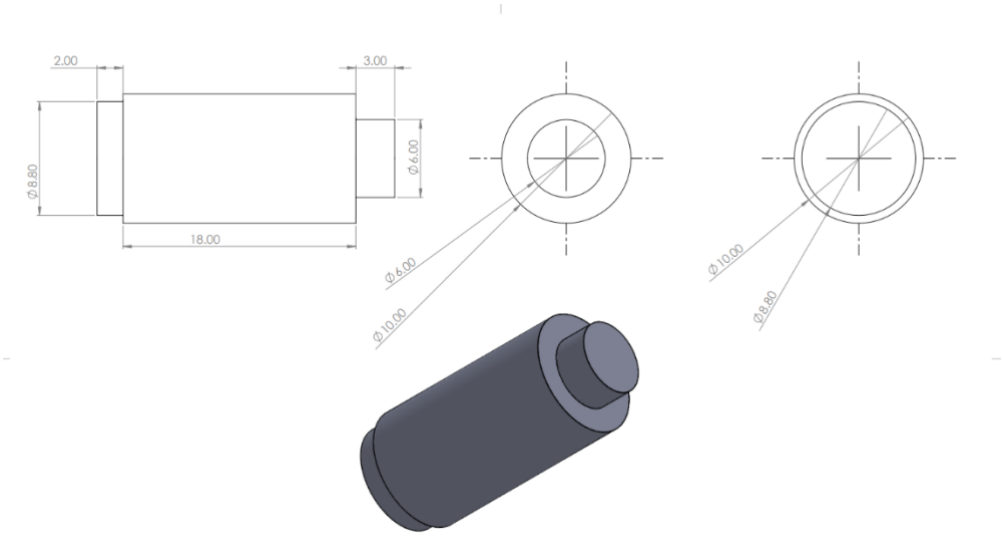


Figure 20: The supports that will hold the installation of the stepper motors.

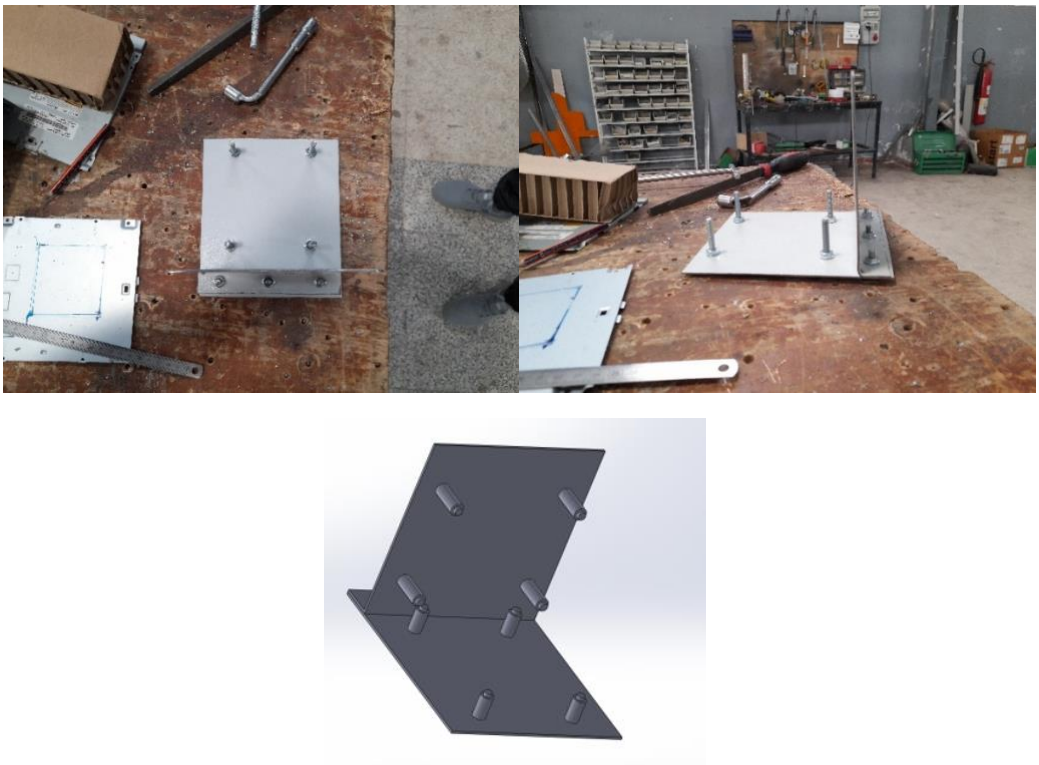


Figure 21: The frame after installing the supports.

2.4.3 Linear Actuators (X and Y Axis)

As previously stated, we utilized stepper motors sourced from a DVD player. We made alterations to the dimensions of the original model to meet our specific requirements.

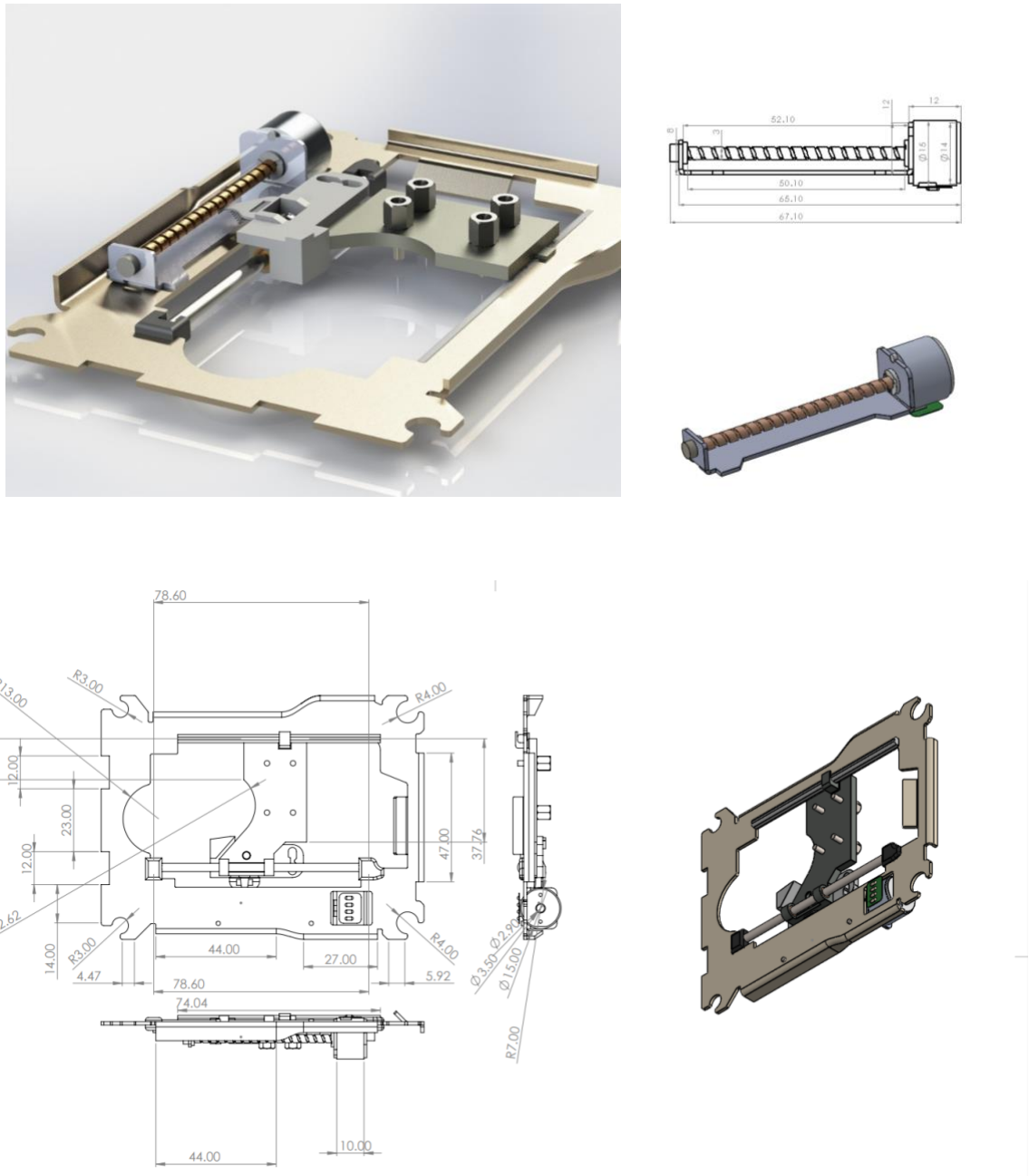
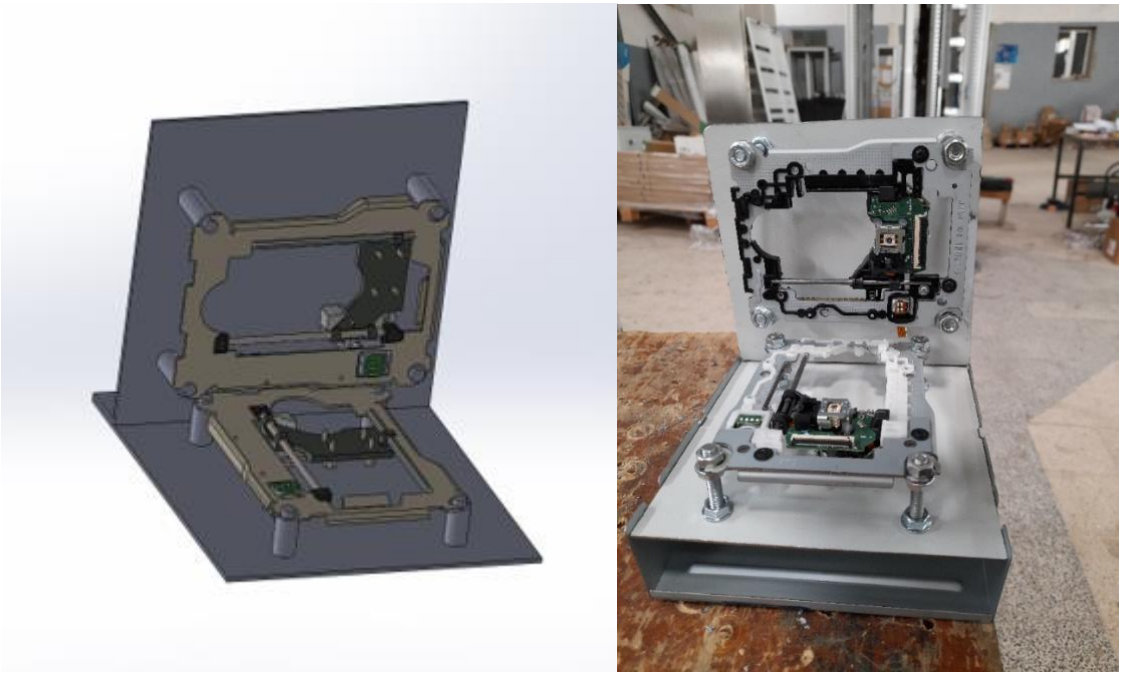


Figure 22: DVD, CD drives -stepper Motor.[12]



Figure 23:DVD drives, Stepper Motor.

Considering the alignment of the x and y axes, along with the initial position, we have perpendicularly positioned the motors to ensure precise maintenance of the reference point.



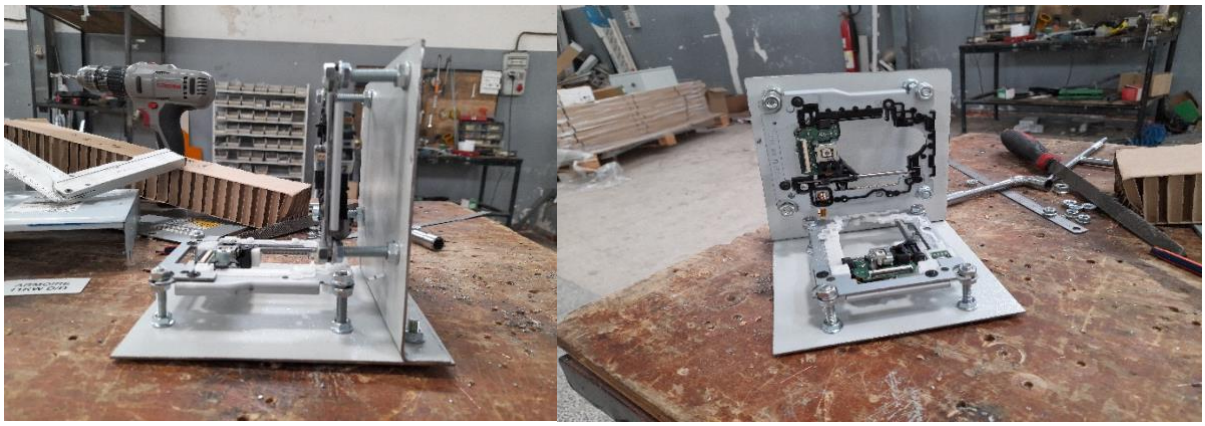
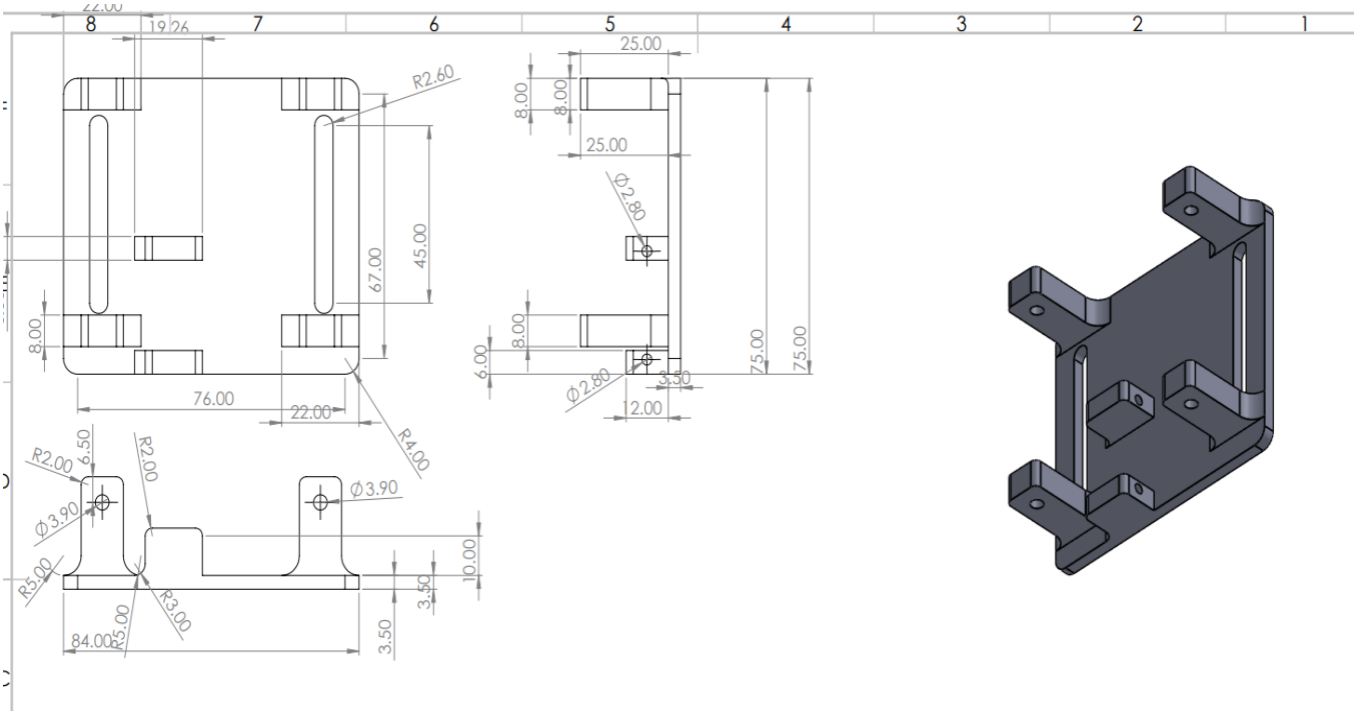


Figure 24: DVD drives, Stepper Motor on the frame.

2.4.4 Pen Holder (with optional pen up/down mechanism)

This section focuses on the seamless and accurate operation of the drawing pen, ensuring the prevention of technical errors and enhancing the overall professionalism in project creation.

2.4.4.1 Pen holder base



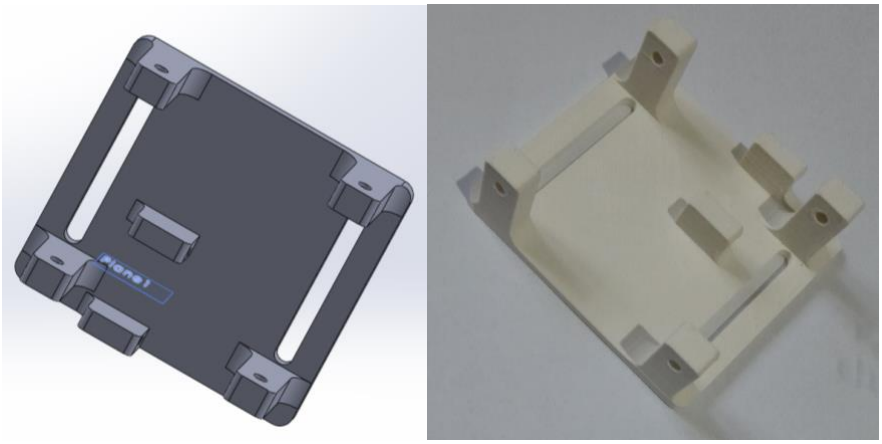
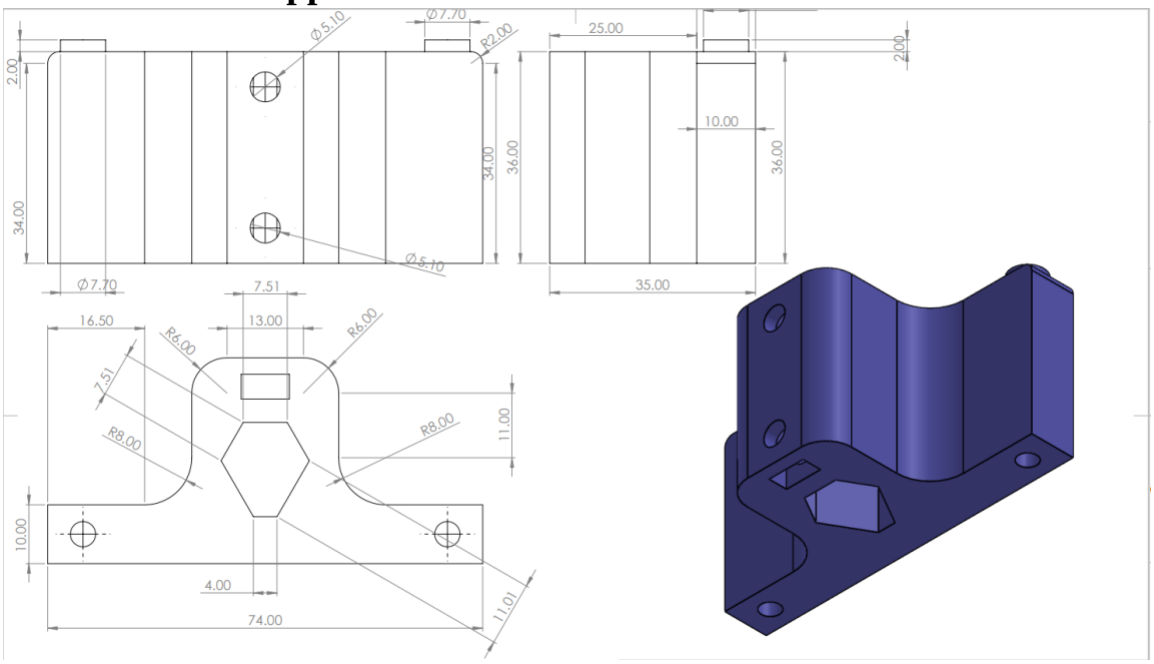


Figure 25: Pen holder base.

2.4.4.2 Pen holder support



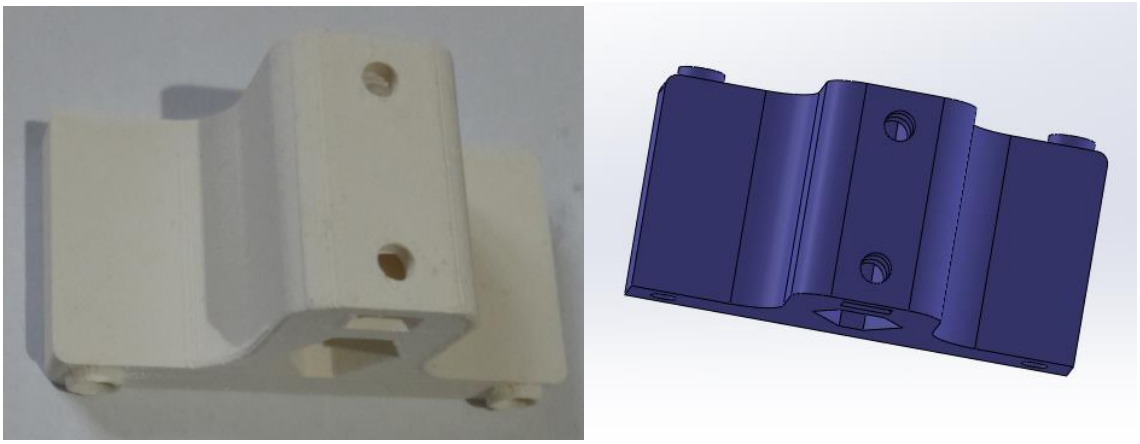
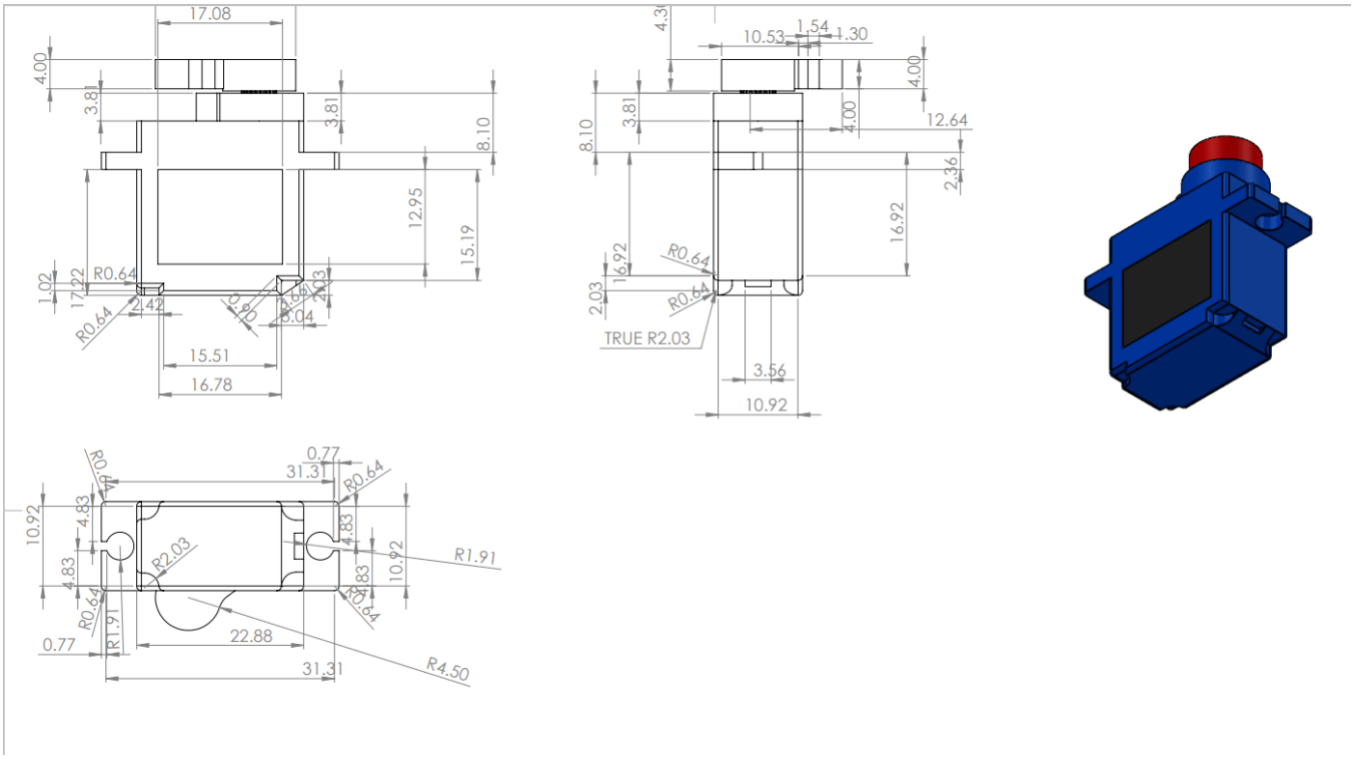


Figure 26: Pen holder support.

2.4.4.3 Servo Motor



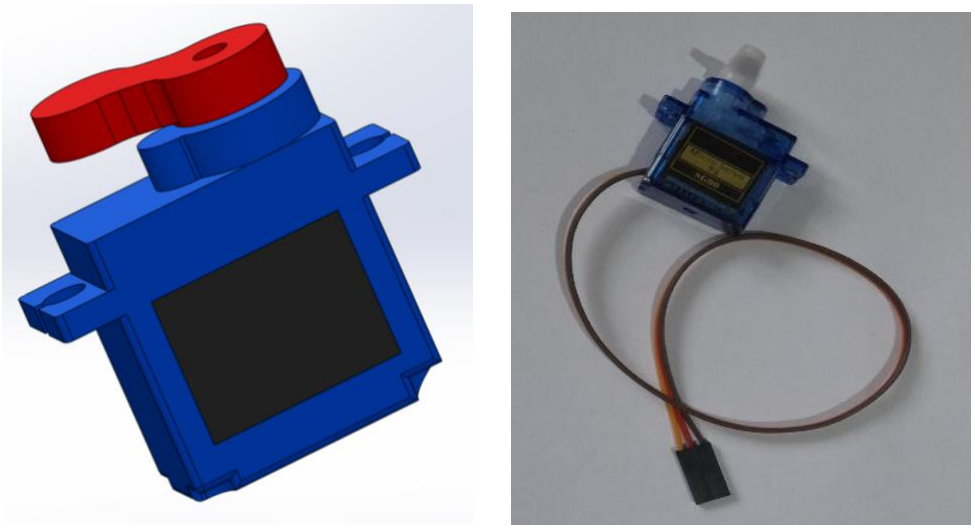


Figure 27:Servo Motor.

These were the basic elements of the pen holder.

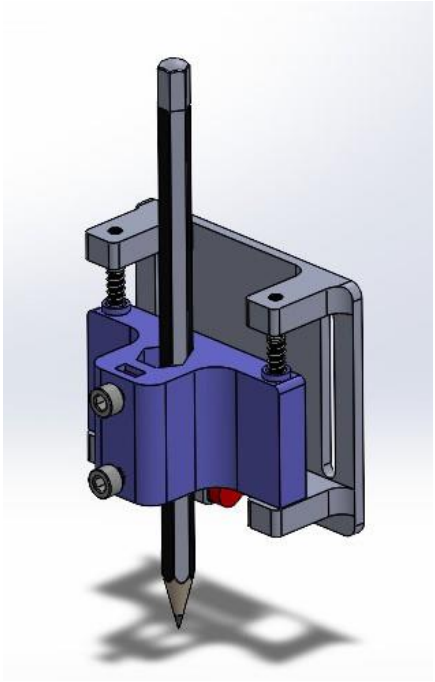


Figure 28: the pen holder.[13]

2.4.5 Drawing surface

The available drawing space is determined based on the stepper motors' range of motion. Consequently, a metal plate was created to secure the drawing paper.

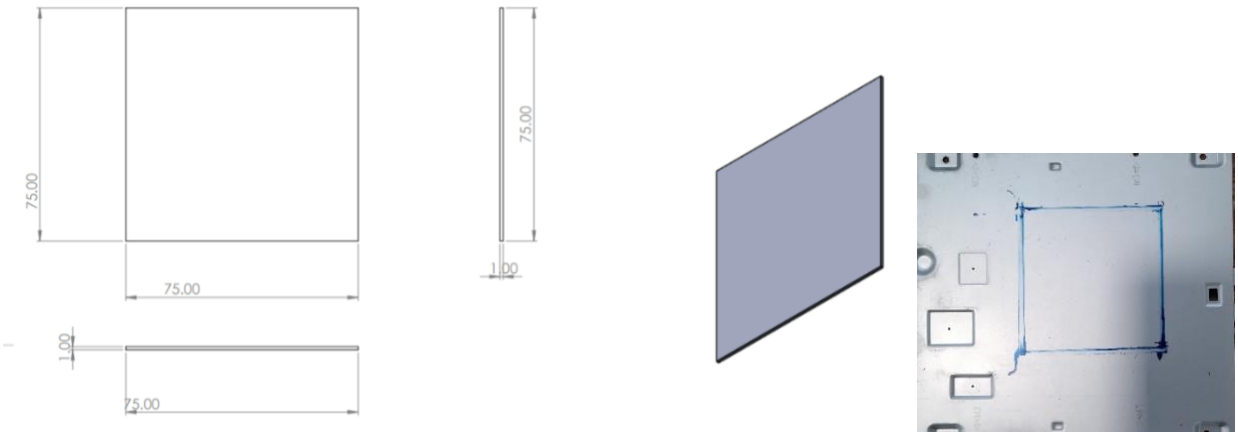


Figure 29: Drawing surface.

2.4.6 Mechanical Assembly

Following the design and fabrication of each component for our project machine, the next step involves the mechanical assembly of these parts to complete the final product.

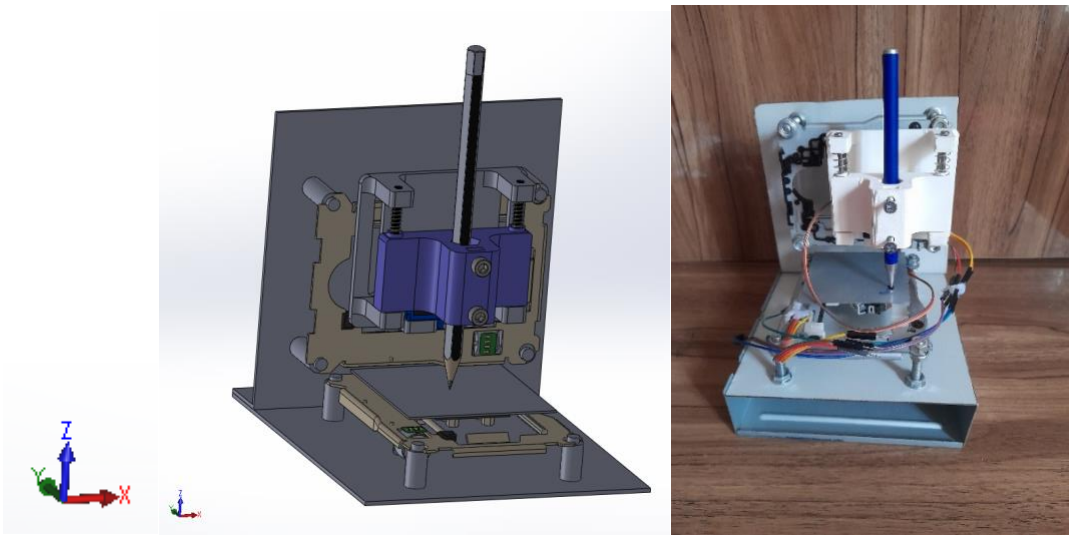


Figure 30:Final mechanical assembly.

2.5 Electronics and Control Part

In this section, we will explore the process of operating a CNC Plotter, along with understanding the various electrical components required for constructing this project.

2.5.1 Control part

Various mechanisms are employed for controlling CNC machines in general. Given the constraints of our project, namely being a small-scale "CNC Plotter" with a limited budget and time, we have

carefully considered and determined that the Arduino platform is the most suitable choice. This decision has been validated through extensive testing and evaluation, ensuring optimal control and performance for this type of mini-project.

2.5.1.1 Arduino



Figure 31: Arduino Logo.

2.5.1.1.1 Introduction

When an idea strikes, we grab a pen and jot it down on a sheet of paper. Now, envision a world where we can effortlessly create objects that seamlessly interact with the surroundings. This is precisely where the Arduino platform steps in.

Arduino is a collection of electronic microprocessor boards that are open source and can be programmed with great ease to comprehend and engage with the environment. Throughout the years, Arduino has established itself as the go-to choice for constructing electronic projects. It has even been utilized in space missions to operate microsatellites and in underwater expeditions to command miniature robotic submersibles.[14]

2.5.1.1.2 The Board

The Arduino Uno is a microcontroller board that utilizes the ATmega328P (datasheet) as its core component. It features 14 digital input/output pins (6 of which can function as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button.

All necessary components to support the microcontroller are integrated into the board; simply connect it to a computer using a USB cable or power it with an AC-to-DC adapter or battery to begin. With the Uno, you can experiment freely without the fear of making mistakes. In the worst-case scenario, you can easily replace the chip for a minimal cost and start anew.[14]

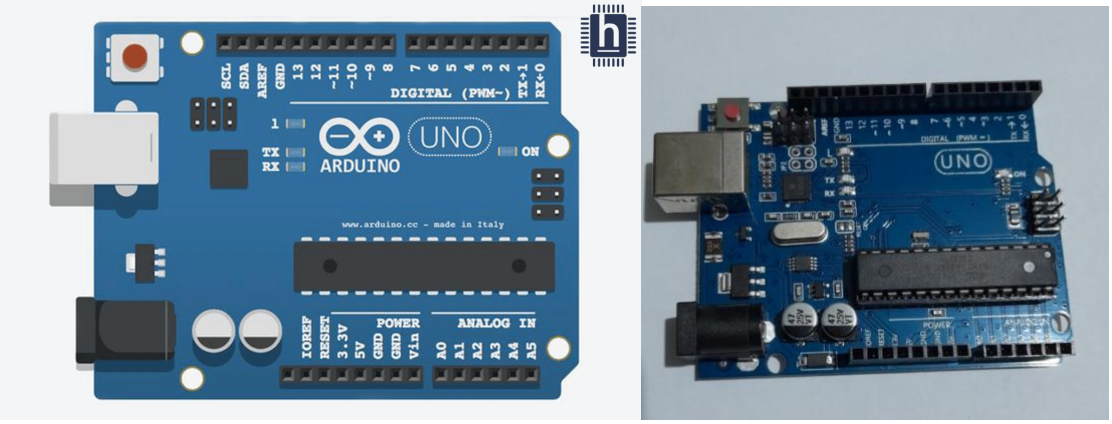


Figure 32: Arduino Uno Board.

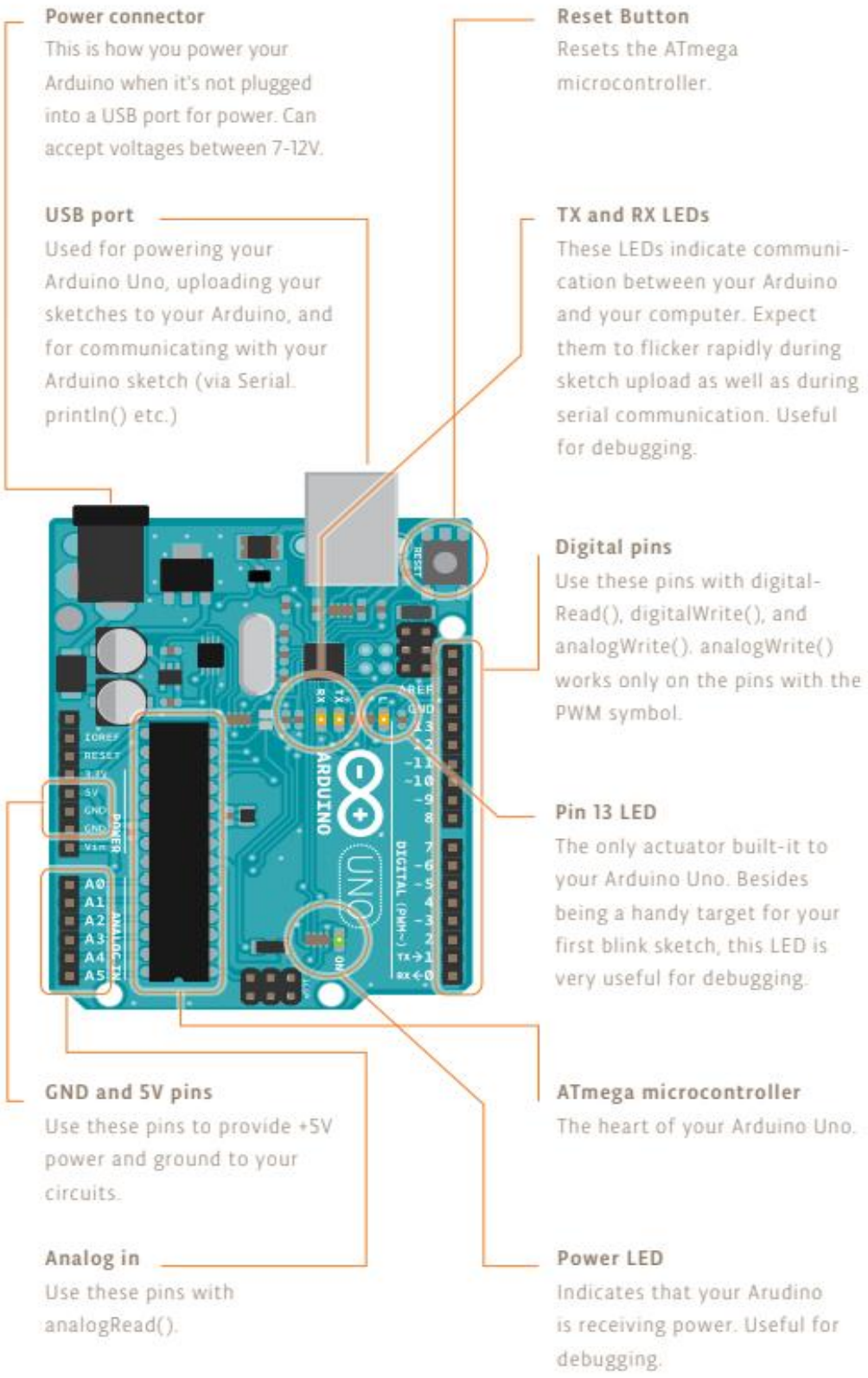


Figure 33: Arduino Uno Pin.[14]

2.5.1.1.3 Analog Input

The Arduino ATmega-328 microcontroller board is equipped with 6 analog input pins, which are labeled from A0 to A5. These analog inputs allow for processing within the range of 0 to 5V. Analog signals are characterized as continuous-time signals, making them suitable for specific applications, and are also known as non-discrete time signals.

Signals such as voltage and current can be classified as either analog or digital based on their time signal properties. In certain applications, the Arduino microcontroller may require the use of analog inputs instead of digital inputs, utilizing the analog input ports or pins available.[14]

2.5.1.1.4 Digital Inputs

Digital inputs refer to signals that are non-continuous in nature, characterized by discrete input pulses represented as 0's and 1's. These inputs can exist in either an on or off-state.

The Arduino ATmega328 microcontroller is equipped with 12 digital input pins labeled D0 to D11, allowing for up to 12 inputs to be utilized for digital input/output applications. The functionality of digital input ports involves receiving discrete input pulses and transmitting them to the ports, enabling the ports to be utilized for both input and output processes.

It is important to note that these digital pins are specifically designed to handle digital inputs only.[14]

2.5.1.1.5 The ATMEGA-328

integrated circuit is comprised of 28 pins, with 6 analog inputs denoted as PC0 to PC5 in the pin diagram. These analog input pins carry continuous time signals that serve as analog inputs for the system.

Additionally, there are 12 digital inputs represented as PD1 to PD11, functioning as digital input ports for pulse width modulation (PWM). PWM allows for the transmission of signals in a discrete form. Both analog and digital input ports have various applications, with power supply input utilizing VCC and GND pins.

Pins PB6 and PB7 serve as a crystal for generating a clock signal. Utilizing this crystal, clock signals can be generated and used for input sources. The PC6 pin is designated for the reset option, enabling program reset by utilizing this pin.

The pin diagram of the ATMEGA-328 microcontroller is illustrated below.[15]

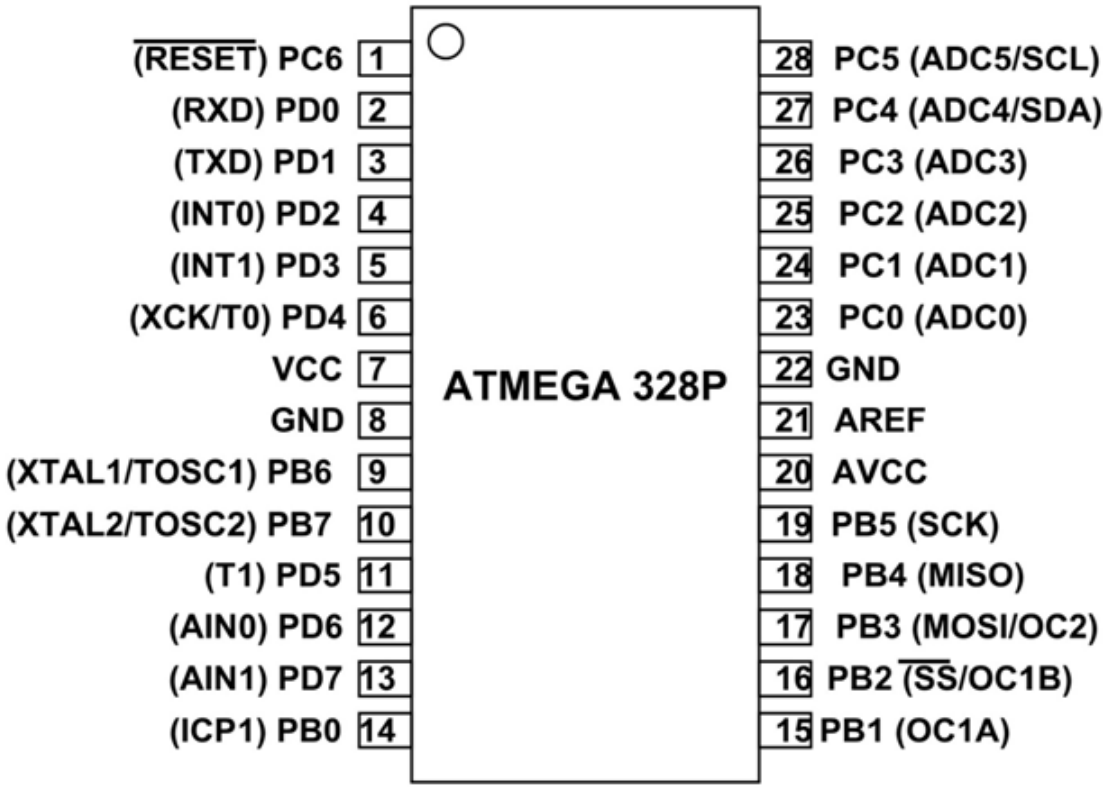


Figure 34: The Atmega 328P Pin.[15]

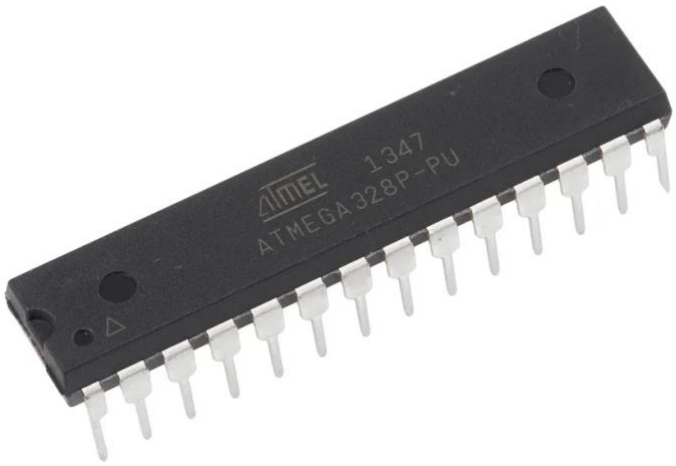


Figure 35: The Atmega 328P .[15]

2.5.1.1.6 Power jack cable / USB Port

The Arduino atmega-328 microcontroller can connect with various electronic devices, including computers, through the use of a USB port or a power jack cable.



Figure 36: Power jack cable.

By utilizing these power jack cables, we can upload Arduino programs for different applications.

Initially, the program can be initialized or modified using Arduino software tools. Subsequently, these programs can be transferred to the Arduino microcontroller board via either a USB cable or a power jack cable.[14]

2.5.1.1.7 Power supply

The Arduino microcontroller is equipped with an additional power supply source, which can be accessed through a power supply port located at the corner of the device. This power supply port allows for connection to an external power supply, such as an AC power supply, or a DC power supply through input pins.

The power supplies generate an active form of power for the Arduino microcontroller, which is capable of accepting a range of power supply voltages. It is important to note that exceeding the specified power supply voltage range can result in damage to the microcontroller. Therefore, it is crucial to provide only the designated range of power supply to the Arduino microcontroller.[14]

2.5.1.1.8 Working Principle

The operational principle of the Arduino microcontroller involves establishing proper connections. This includes checking all input ports and ensuring a stable power supply connection. The output pins can then be linked to external devices based on their intended applications.

The programming required for these applications can be carried out using Arduino software. This software allows for customization according to the specific requirements of the applications. It operates using the C and C++ programming languages, which are considered high-level languages. By defining the operational conditions, a program can be developed to facilitate the desired applications. Subsequently, these programs can be uploaded to the Arduino microcontroller using a power jack cable. Once uploaded, the microcontroller is ready to execute the program.

The ATMEGA-328 microcontroller is capable of storing programs and functions as a processor to execute tasks accurately. By providing analog or digital inputs to the system, processes can be carried out in accordance with the applications.

The program within the Arduino software can be modified to control the application process, and then re-uploaded to the Arduino microcontroller via the power jack cable. A reset button is

available for resetting the program, effectively clearing previous programs to enable the microcontroller to be used for other applications.

The Arduino ATMEGA-328 microcontrollers are versatile and can be utilized for numerous applications. They are commonly employed in automation industries to regulate processes and operate systems in an automated manner.[14][15]

2.5.1.2 L293D Motor Driver IC

The L293D is a well-known integrated circuit (IC) that is widely used for the purpose of operating small DC motors and bipolar stepper motors. Its primary function is to efficiently control inductive loads, including motors, relays, and solenoids.

The L293D is specifically designed as a quadruple half-H driver IC, enabling it to drive two DC motors bidirectionally (forward and reverse) or a single stepper motor. It is capable of operating within a broad range of voltages, typically up to 36V, and can handle currents of up to 600 mA per channel (with a peak of 1.2 A).

To ensure the protection of the circuit, the L293D is equipped with built-in diodes that safeguard against back EMF (Back electromotive force). Due to its versatility and reliability, this IC is commonly utilized in various applications such as hobby projects, robotic platforms, and any other scenario that requires motor control in conjunction with a microcontroller or other control circuitry.

The L293D simplifies the process of motor control by providing an integrated solution that incorporates straightforward control logic.[16]

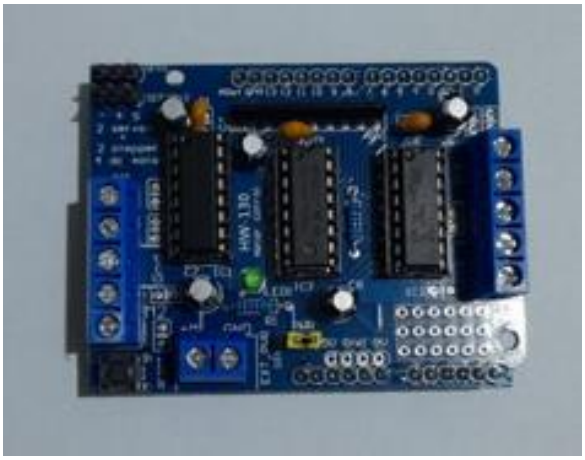


Figure 37: L293D Board.

2.5.1.2.1 L293D pinout

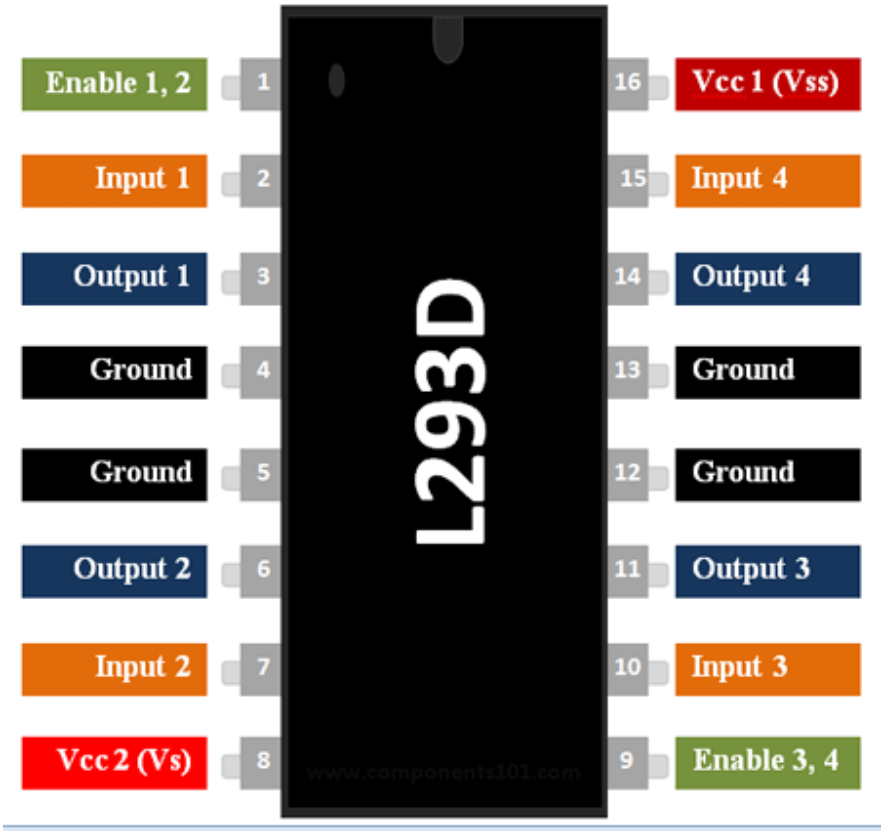


Figure 38: L293D pinout.[16]

2.5.1.2.2 L293D Pin Configuration Details

Pin1 (Enable)	Pin 1 is known as the enable pin. It has a major effect on Input and output. If there is High logical signal on enable pin (EN) then there will be input and output between pin 2,3,6 & 7 (Input 1, Output 1, Input 2 & Output 2)
Pin2 (Input 1)	Mostly input means where we provide the input to give the output. But here Input 1 means which polarity we want to give at Output 1.
Pin3 (Output 1)	Output 1 is the input of the first motor/Motor 1. It attaches to its one end.
Pin4 (Ground)	The ground pin will attach to the ground of the circuit.
Pin5 (Ground)	The ground pin will be attached to the ground, and it will remain common with the previous ground.
Pin6 (Output 2)	Output 2 will attach to the input of the first motor/Motor 1. It will attach to its second end.
Pin7 (Input 2)	Input 2 will attach to the control button or device to control the Output 2 just like Input 1.
Pin8 (V_{cc})	Pin8 is the voltage pin for Motor. It will device how much power we are going to attach the Motor. This Power should not be more than 36 volts and should not be less than 4.5 volts.
Pin9 (Enable)	Pin 9 is also the same as Pin 1. It controls the input and output signals. Pin 9 Controls the connection between Input 3, Input 4, Output 3 and Output 4. It also enables the connection when the logic signal will be High (1).
Pin10 (Input 3)	Input 3 will control the output polarity of the Pin 11 (Output 3) by logic signals.
Pin11 (Output 3)	Output 3 will be connected to the one end of the second motor.
Pin 12 (Ground)	Pin 12 will attach to the common ground with all other grounds.
Pin13 (Ground)	Pin 13 will also be attached to the common ground with all other grounds.
Pin14 (Output 4)	Pin 14 will attach to the second end of the second motor.
Pin15 (Input 4)	Pin 15 will control the output polarity of the Pin 14 (Output 4) by logic signals.
Pin16 (V_{cc})	Pin 16 will the Power we will provide to the L293D to activate it or to turn it on. The power level of Pin 16 should be 4.5 – 7Volts. Voltage more than 7 will burn the IC

Table 1: L293D Pin Configuration Details.[16]

2.5.1.2.3 The Importance of Motor Drivers in Electrical Systems

The significance of motor drivers in electrical systems cannot be overstated. In the realm of modern electrical systems, motors, whether AC or DC, play a crucial role by providing the ability to rotate in multiple directions. This versatility is essential in various fields such as robotics, automotive applications, and beyond. While AC motors operate within specific parameters, the directionality of DC motors can be easily modified by reversing the current polarity.

DC motors have gained popularity due to their efficiency, especially in applications that require rotational motion. However, there are often situations where bidirectional motor rotation is necessary. In the past, achieving this required complex and time-consuming methods involving diodes and transistors.

To simplify the complexity of motor control, the integrated circuit (IC) L293D was developed. This IC not only allows for bidirectional motor rotation but also offers a range of additional functions with remarkable ease and efficiency. By utilizing the L293D IC, the construction and control of motors become more streamlined, saving time and effort in various applications.

2.5.2 Actuators

Actuators are essential components in devices and machines, facilitating physical movements by converting various forms of energy, such as electrical, air, or hydraulic, into mechanical force. In essence, actuators play a crucial role in enabling movement within any machine.

It is crucial for machine designers to carefully evaluate the technical pros and cons of stepper and servo technologies to choose the most suitable motor-drive system for a specific application, as each technology caters to different niches and can impact the overall success of the project.

As previously stated, electric motors are required to represent the x and y axes of movement, as well as the z-axis for the pen holder.

Considering the characteristics of our project and the fact that it involves a small-scale prototype machine, we opted for stepper motors sourced from a DVD Player to control the x and y axes, along with a servo motor to operate the pen holder z-axis.

In this part, we will present a study of these two types of electric motors.

2.5.2.1 Electric Motors

An electric motor is a device that transforms electrical energy into mechanical energy by utilizing the interaction between magnetic fields and conductors carrying current. It functions based on the principles of electromagnetism, primarily harnessing the force generated by magnetic fields on electric currents flowing through a coil.

Electric motors are of utmost importance in modern industry and everyday life, as they supply power to a wide range of equipment, from household appliances to industrial machinery. These motors come in different types, each designed for specific purposes, such as "DC Motors" and "AC Motors". Furthermore, they are further categorized based on their particular requirements and

applications. Additionally, there are specialized motors like "Stepper Motors" and "Servo Motors" that cater to specific needs and functions. [17]

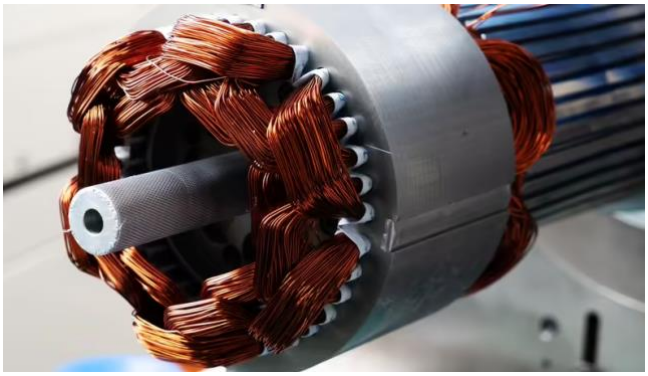


Figure 39: Electric Motors.

2.5.2.2 Servo Motor

2.5.2.2.1 Servo

Servo refers to a closed-loop control system that encompasses various applications. In this system, the feedback signal plays a crucial role in regulating the motor's speed and direction, ultimately leading to the attainment of the desired outcome.[18]

2.5.2.2.2 Servo motor

A servo motor is a compact device equipped with an output shaft. By transmitting a coded signal, the servo motor can accurately position the shaft to specific angular positions. As long as the input line carries the coded signal, the servo motor will steadfastly maintain the desired angular position of the shaft.

There are four main components inside a hobby servo, a motor, a gearbox, a potentiometer and a control circuit.

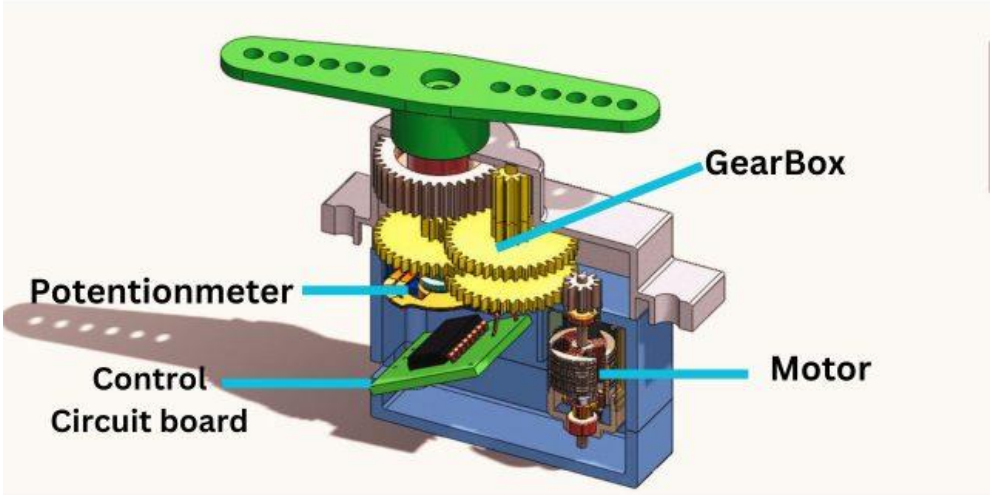


Figure 40: Servo motor.

A servo gear unit is responsible for decreasing the speed of a motor. The potentiometer offers position feedback to the servo control unit, where the current motor position is compared to the desired position. Based on any discrepancies, the control unit adjusts the motor's actual position to align with the target position. If the motor being controlled is a DC motor, it is commonly referred to as a DC Servo Motor. Conversely, if an AC motor is being controlled, it is known as an AC Servo Motor.

The primary distinction between the two types of motors lies in their power source. AC servo motors rely on an AC power source, while DC Servo motors require a DC power source, such as batteries. In an AC servo motor, speed is directly linked to the frequency of the supply voltage and the number of magnetic poles, whereas in a DC servo motor, speed is directly proportional to the applied voltage.[18]

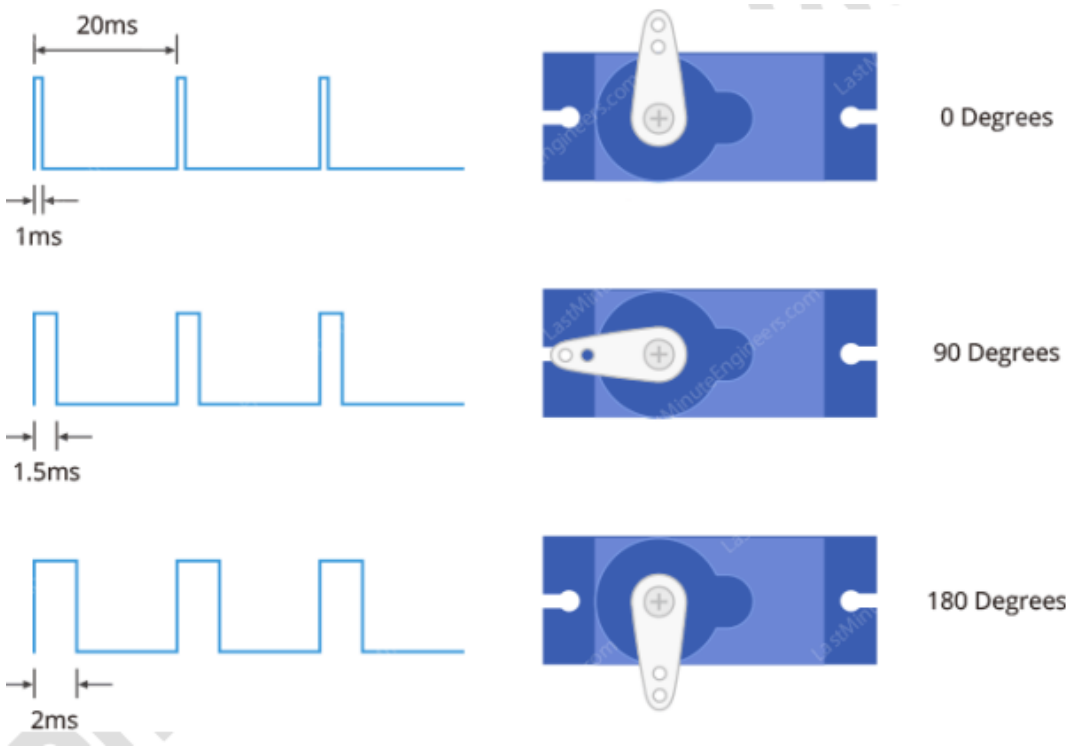


Figure 41: Servo motor pulses.

- If the pulse is high for 1 ms, then the servo angle will be zero.

-If the pulse is high for 1.5ms, then the servo will be at its center position.

-If the pulse is high for 2ms, then the servo will be at 180 degrees.

-Pulses ranging between 1ms and 2ms will move the servo shaft through the full 180 degrees of its travel.[18]

Note

The minimum and maximum duration of the pulses can sometimes vary with different brands and they can be 0.5ms for 0 degrees and 2.5ms for 180 degrees position.

2.5.2.2.3 Servo Motor Pinout

Servo motors typically have three connections and are as follows:



Figure 42: Servo motor Pinout.

GND is a common ground for both the motor and logic.

5V is a positive voltage that powers the servo.

Control is input for the control system.[18]

The servo motors exhibit a range of wire colors, however, the red wire consistently represents a voltage of 5V, while the ground wire can either be black or brown. On the other hand, the control wire is typically identified by the colors orange or yellow.

2.5.2.3 Stepper Motors

Stepper motors are a type of DC motor that operate by moving in discrete steps. These motors consist of multiple coils organized into phases. By energizing each phase in a specific sequence, the motor rotates incrementally, one step at a time. This precise control over positioning and speed makes stepper motors the preferred choice for various applications requiring accurate motion control, with no need for a sensor.

Additionally, stepper motors are available in various sizes, styles, and electrical characteristics, making them versatile for different needs.[19]

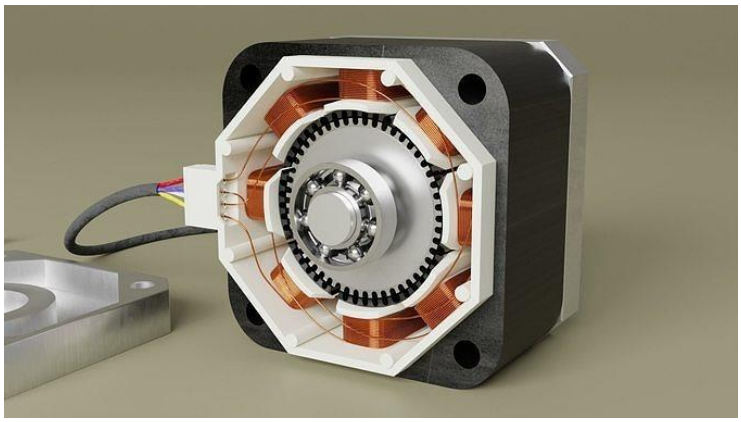


Figure 43: Stepper Motors.

2.5.2.3.1 Stepper Motor Working Principles

The stepper motor consists of a stator and a rotor, like all electric motors. The stator contains teeth with coils, while the rotor can be a permanent magnet or a variable reluctance iron core. Further details on rotor structures will be discussed later. The diagram in the figure illustrates a cross-section of the motor, with the rotor being a variable-reluctance iron core. [19]

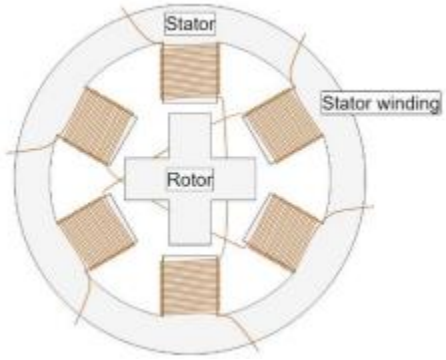


Figure 44: Cross-Section of a Stepper Motor.

The fundamental operation of the stepper motor involves energizing stator phases to create a magnetic field that aligns the rotor. By activating different phases sequentially, the rotor can be rotated to achieve a specific position.

The figure demonstrates this operational principle. Initially, coil A is energized, aligning the rotor with the magnetic field. Subsequently, when coil B is energized, the rotor rotates 60° clockwise to align with the new magnetic field. The same rotation occurs when coil C is energized. The colors of the stator teeth in the images indicate the direction of the magnetic field produced by the stator winding.[19]

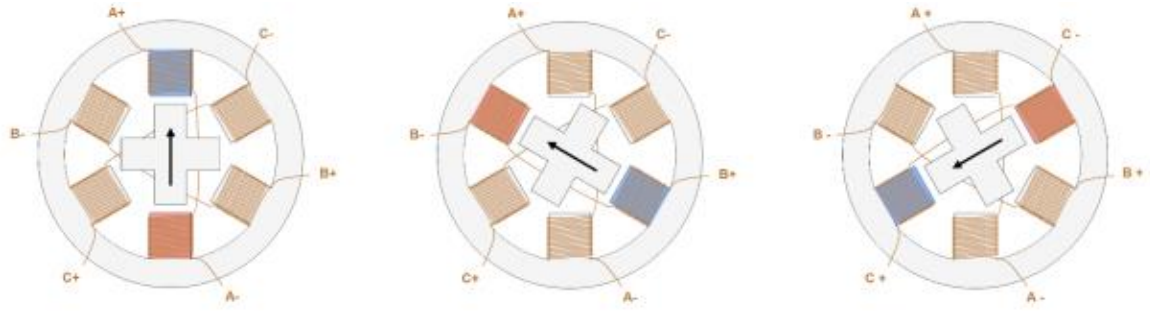


Figure 45: Stepper Motor Steps

2.5.2.3.2 The applications of stepper motors

Stepper motors are ideal for various applications due to their unique characteristics. They are particularly well-suited for precise positioning tasks, making them perfect for use in 3D printers, CNC machines, camera platforms, X,Y plotters, and even in some disk drives.
Additionally, stepper motors offer excellent speed control, allowing for precise increments of movement which is essential for process automation and robotics.
Moreover, these motors provide maximum torque at low speeds, making them a preferred choice for applications that require both low speed and high precision.[19]



Figure 46: Stepper Motors.

2.5.2.3.3 Stepper Motors in DVD Players

Stepper motors found in DVD players are utilized as precise control devices to accurately move the laser assembly for reading disc data. Here are some important points to note:

Key Characteristics

Accuracy: Stepper motors ensure precise control over the laser assembly's position to accurately read disc data.

Compact Size: These motors are small and compact to fit within the limited space of a DVD player.

Low Torque Requirement: Due to the light mechanical load of the laser assembly, high torque is not necessary for these motors.

High Resolution: Stepper motors often have a high step count per revolution, enabling fine control over the laser's position.[20]

2.5.2.3.4 Types of Stepper Motors in DVD Players

Linear Stepper Motors: Directly move the laser assembly along a linear path.



Figure 47: Linear Stepper Motors in DVD Players.

In our project, we will use this type to represent movement along the x and y axes.

Rotary Stepper Motors: Used for rotating the disc spindle or adjusting the laser assembly's position through gears or linkages.[20]



Figure 48: Rotary Stepper Motors in DVD Players.

2.5.2.3.5 Applications

Laser Assembly Movement: Primarily used to move the laser assembly radially across the disc to access different data tracks.

Spindle Control: Stepper motors may also control the speed and position of the disc spindle.

Reuse

Stepper motors from DVD players can be repurposed for various DIY electronics projects, such as:

Robotics: Ideal for small-scale robotics projects requiring precise movements.

3D Printers: Used for fine control of the print head or other components.

CNC Machines: Control the tool or workpiece position.

Arduino Projects: Easily controlled using Arduino boards for custom projects.[20]

2.5.2.3.6 Pins Stepper motors in DVD players

Stepper motors found in DVD players are commonly bipolar, having two coils and therefore four pins. The pins are typically grouped in pairs, with each pair corresponding to a specific coil. To determine which pins are paired together for each coil, a simple method involving a small battery, such as an AA battery, can be used. By connecting the battery to different pairs of pins, you can observe which pairs cause the motor shaft to move or twitch; these pairs are the ones belonging to the same coil.

or To identify the pins of a stepper motor, you can use a multimeter to measure the resistance between wires.[20]

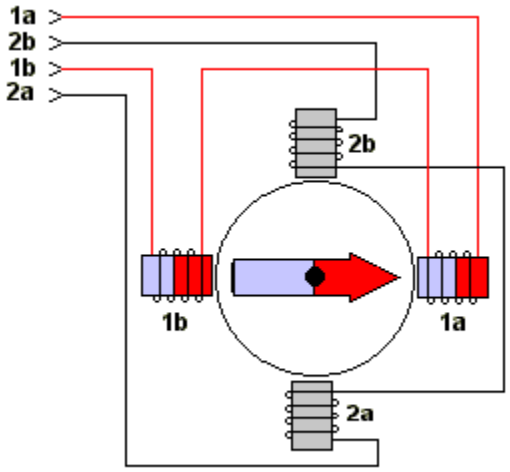


Figure 49:Conceptual Model of Bipolar stepper Motor.

2.5.2.4 Torque and Speed Considerations

The decision to use a stepper or servo motor is usually determined by the specific application. Stepper motors are typically sized with double the continuous requirement to accommodate additional acceleration and deceleration torque, or to meet a specific peak torque demand.

On the other hand, servo motors are sized based on the application's specific speed and torque requirements, maximum intermittent acceleration/deceleration, holding torque (if needed), and the equivalent RMS continuous torque throughout the entire motion profile.

Generally, for applications needing high-throughput, high-speed, and high-bandwidth capabilities for disturbance correction, as well as high rpm with or without tight coordination between axes, servo motors are the preferred choice. If the application requires modest point-to-

point position performance and rpm, based on process loads and expected disturbances, steppers may be more suitable.

Moreover, when the loads are reasonable, the ability of a stepper motor to maintain position using holding torque with power and detent torque without power can be advantageous. Speed-torque curves illustrate the differences between stepper and AC permanent magnet servo motors of the same size (see Figure).

Stepper motors typically deliver higher continuous torque at lower speeds compared to servo motors. However, servo motors can generate intermittent peak torques at low speeds and provide peak and continuous torques across a broader and higher speed range.[21]

Stepper and Servo Capability.

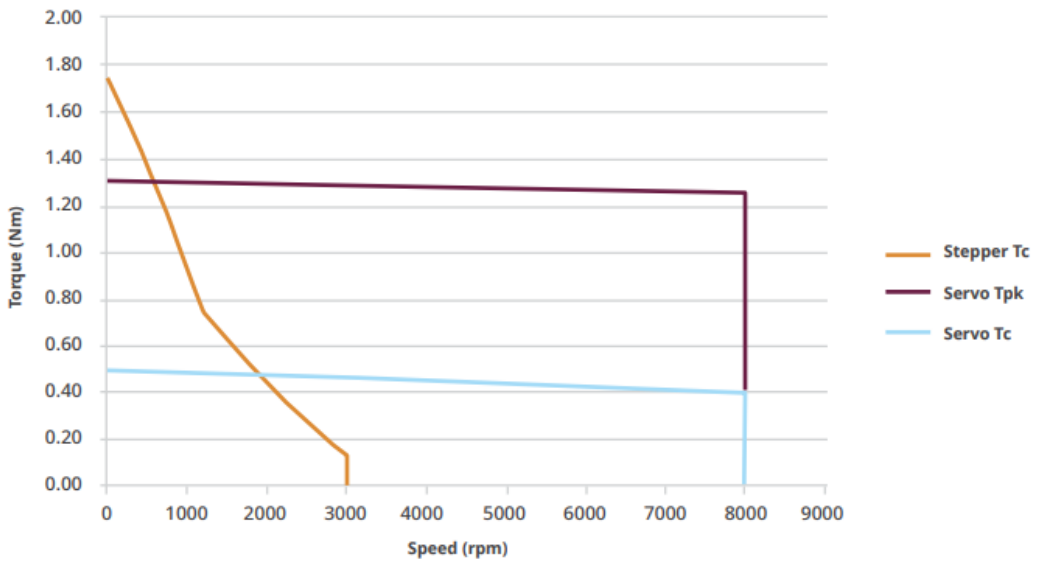


Figure 50: Performance curve comparison with approximately same volume.[21]

2.5.3 Electrical Assembly

After identifying the electrical and electronic components of our project, we must assemble them to achieve the electrical design.

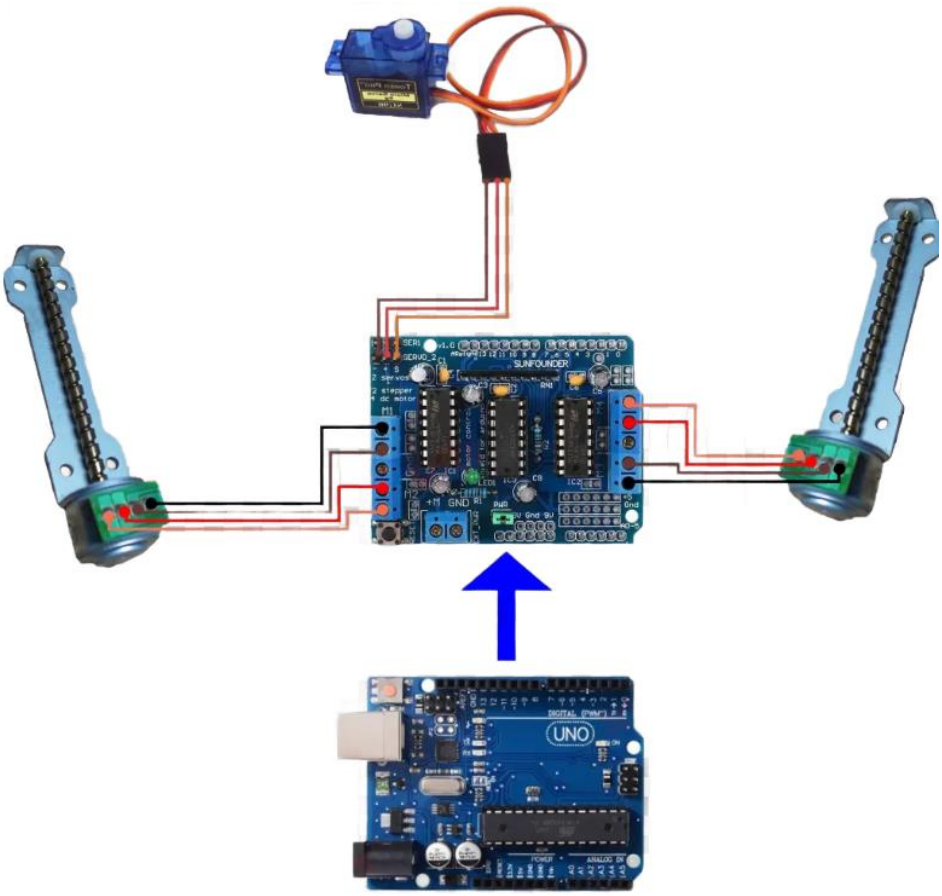


Figure 51: Full circle.

Chapter III

Operation and use.

3 Chapter Ⅲ Operation and use.

3.1 How To Operate This Machine

In this section, we will explore the initial procedures for activating the machine. Commencing from the ultimate assembly and establishing the link between the electrical and mechanical configurations, we will also delve into the significance of computer programs and their practical utilization.

To properly understand how our device works, we have designed this diagram.

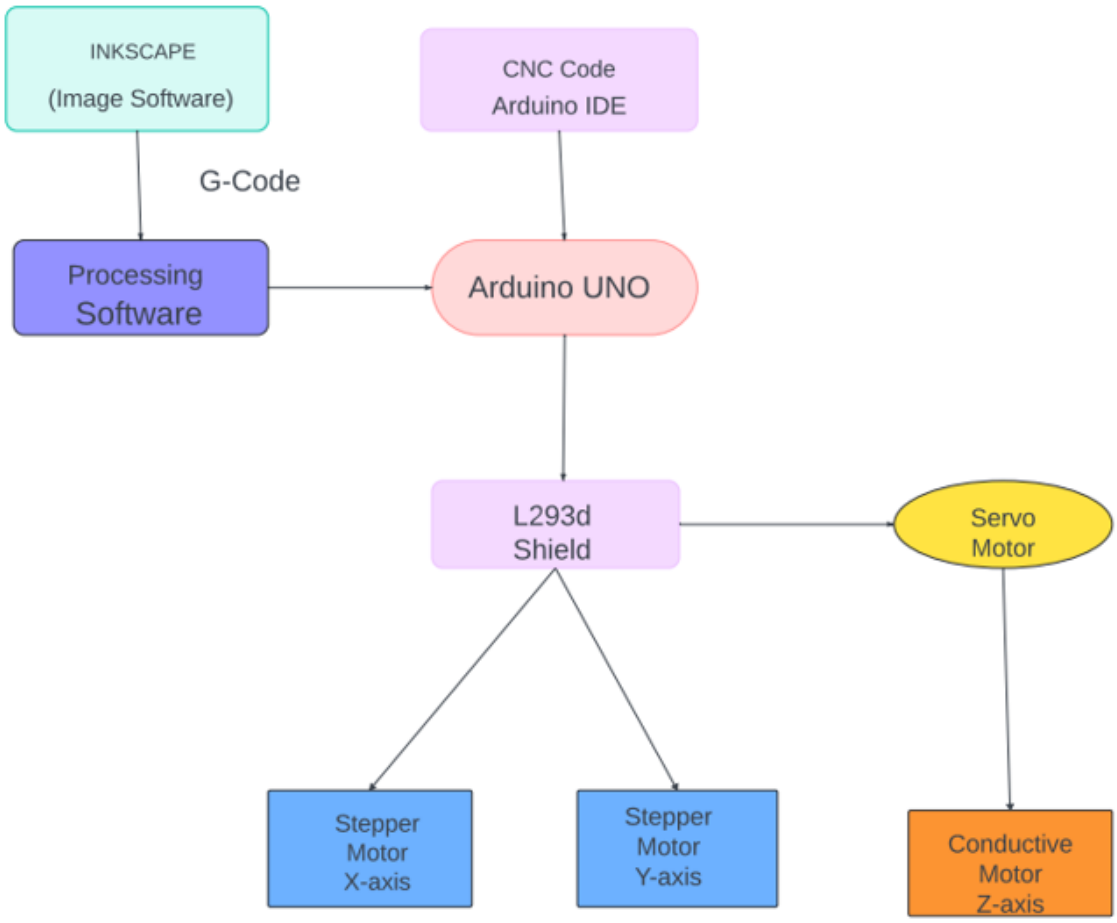


Diagram 1: Block Diagram for the system based on Arduino UNO and L293D shield.

3.1.1 The First Step

Connecting the Arduino Uno board to the L293D shield board.



Figure 52: Arduino Uno & L293D shield board.

Determine the poles of stepper motors COIL 1 and COIL2 using a multimeter.

Connecting stepper motors and servo motors to the L293D shield.



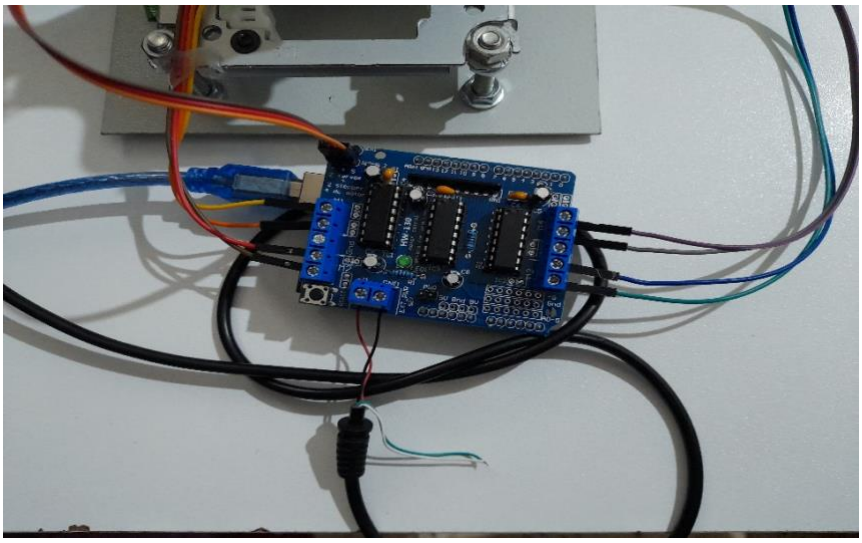


Figure 53: Connecting stepper motors and servo motors to the L293D shield.

The power connection (12V) can be provided by a battery or via the computer's USB port.

In addition to connecting the Arduino board to the computer using a Power Jack cable / USB port.

3.1.2 The second step

After determining the shape to be drawn, we must use the Inkscape program to determine the dimensions of the shape according to the drawing space available to us, then save the file as g-code.

The dimensions of the drawing area were measured, then we designed a template that matches them.

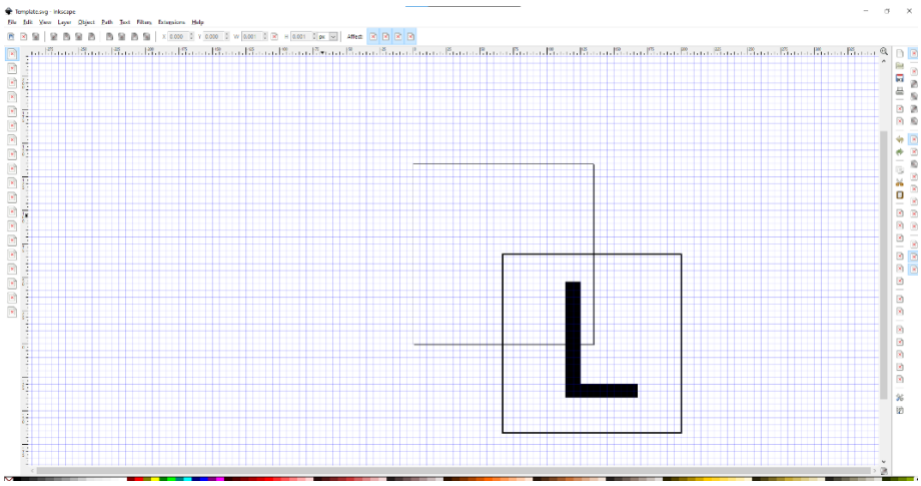


Figure 54: Drawing Template.

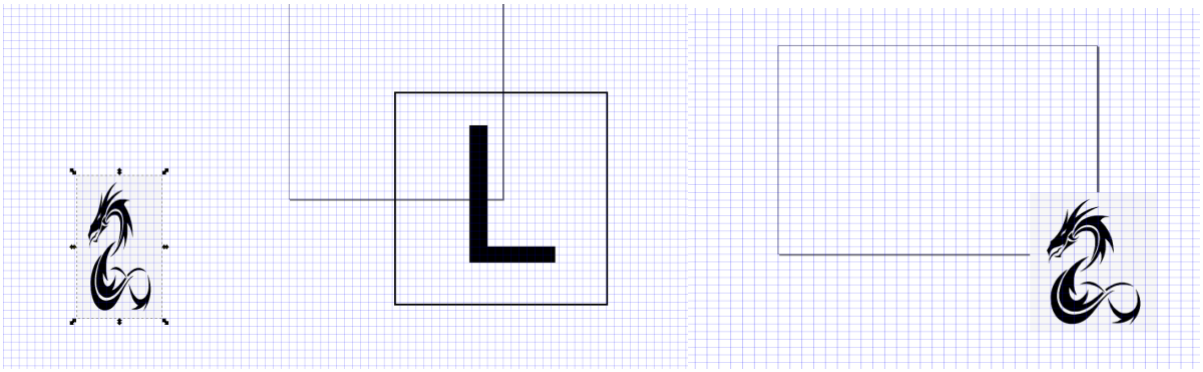


Figure 55: Define the Shape According to the Template.

G-code tools is a freely available Inkscape extension that enables the export of G-code for utilization with CNC machines.

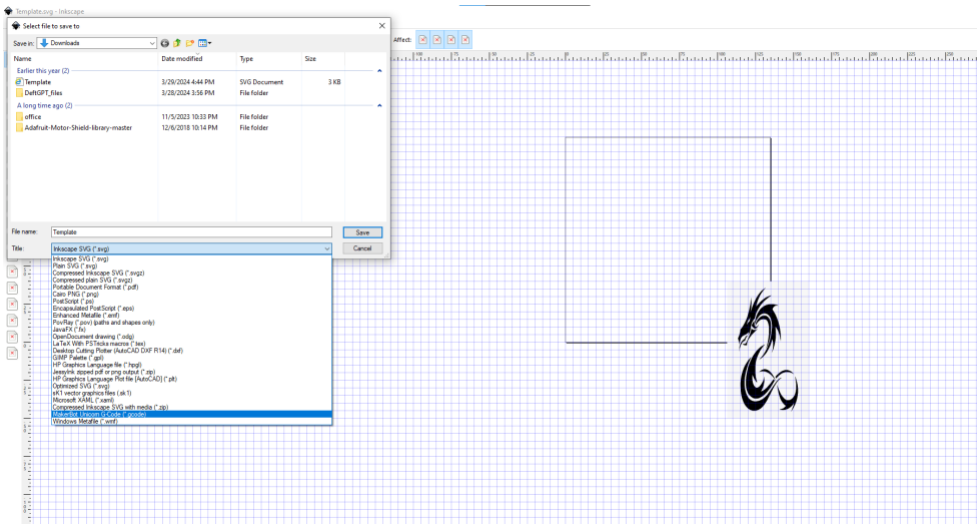


Figure 56: Save the File as G-code.



Figure 57: The G-code File.

3.1.3 The Third step

Following the connection of the Arduino to the computer, it is essential to upload the appropriate code compatible with the specific machine and the intended task.

The Arduino code can be obtained from different platforms and specialized websites that serve CNC designers and users.

Once the code is ready, it must be uploaded to the Arduino board using the Arduino IDE program.

Here's a breakdown of the key components and their roles in the code:

Libraries and Global Variables:

`#include <Servo.h>` and `#include <AFMotor.h>`: Include the necessary libraries for controlling servos and motors.

`AF_Stepper motory(48, 1);` and `AF_Stepper motorx(48, 2);`: Define two stepper motors for the X and Y axes with 48 steps per revolution.

`Servo servo;`: Define a servo motor to control the pen's up and down movements.

Configuration Constants:

`#define penup 115` and `#define pendown 160`: Define the servo positions for the pen up and down states.

`#define Spin 9`: Define the pin number for the servo.

Setup Function:

Initializes the serial communication and attaches the servo.

Sets the speed of the stepper motors.

Outputs a ready message to the serial monitor.

Main Loop:

Continuously reads commands from the serial input.

Processes incoming lines of G-code or similar commands to control the machine's movements.

Handles commands to move the pen up (`penUp()`) and down (`penDown()`).

Handles commands to draw lines by moving the motors.

Command Processing:

`processIncomingLine()`: Parses and executes commands to control the machine's movements.

`drawLine(float x1, float y1)`: Draws a line to the specified coordinates by calculating the required steps for the motors.

`penUp()` and `penDown()`: Move up and down the pen using the servo motor.

Roles of Specific Parts:

Reading Commands:

The main loop reads lines of commands from the serial interface.

Commands are parsed to determine whether they are movement commands (`G-code` commands like `G0`, `G1`) or other instructions.

Processing Movement Commands:

`processIncomingLine()`: Processes commands and determines new positions for the machine's actuators.

Commands like `G0 X10 Y10` would move the actuator to (10, 10).

Drawing Lines:

The `drawLine()` function calculates the steps needed to move the motors to the new coordinates.

Uses Bresenham's line algorithm to determine the path and drive the stepper motors accordingly.

(Bresenham's line algorithm is a computer graphics algorithm used for drawing straight lines on a grid or raster display, such as a computer screen. The algorithm determines which points on the grid should be plotted to form the closest approximation to a straight line between two given points.)

Pen Control:

`penUp()` and `penDown()` use the servo motor to Move the pen up and down .

These functions ensure the pen is in the correct position for drawing or moving without drawing.

Overall Function:

The primary purpose of the code is to interpret commands that are transmitted through the serial interface and convert these commands into specific movements for the CNC machine. It manages the positioning of the pen or tool in both the X and Y axes through stepper motors, while also controlling the Z axis (pen up/down) with a servo motor. This functionality enables the machine to create drawings based on the commands it receives.

NOTE

The code and an explanation of its most important parts are attached in the appendix to this thesis.[22]

3.1.4 Upload The Code:

Steps to upload the code using the Arduino IDE.

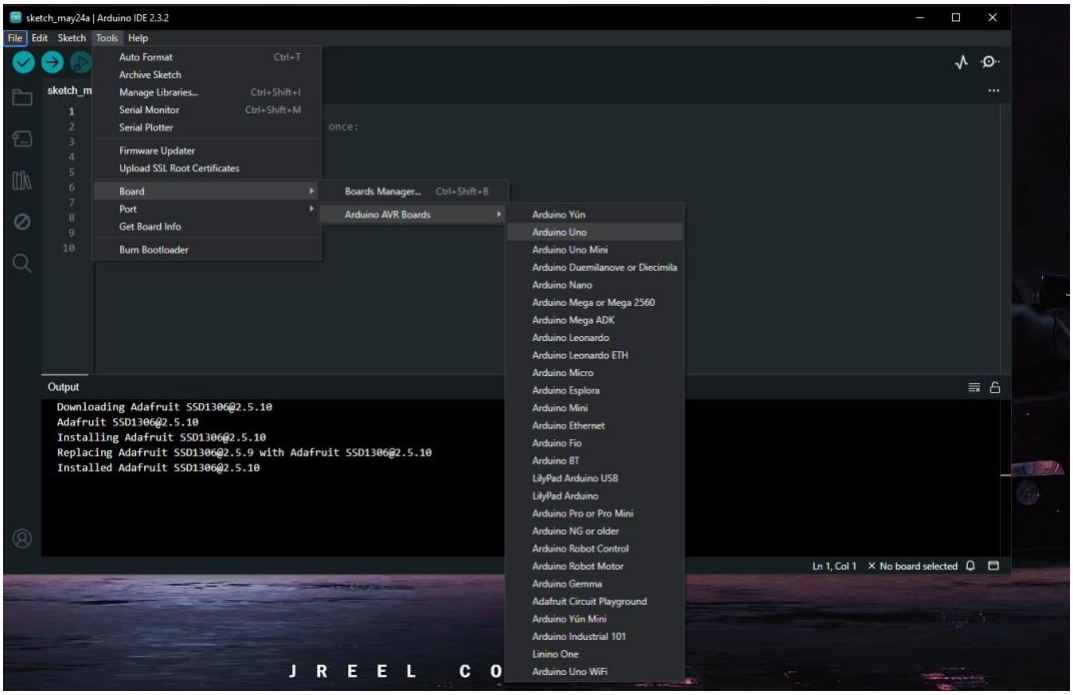


Figure 58: Determine the type of Arduino board.

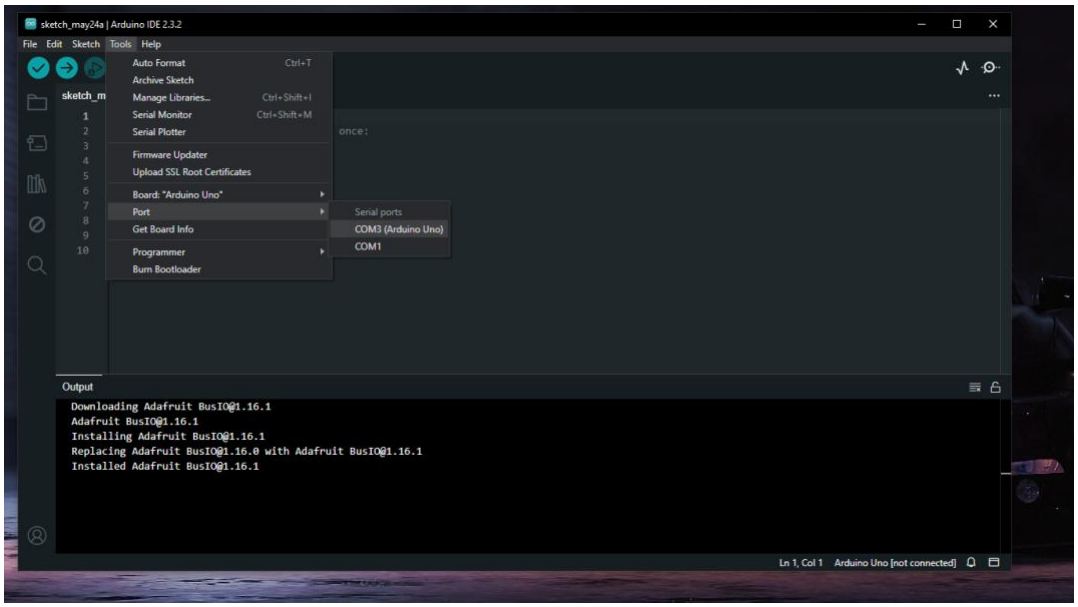


Figure 59: Determine the connected port.

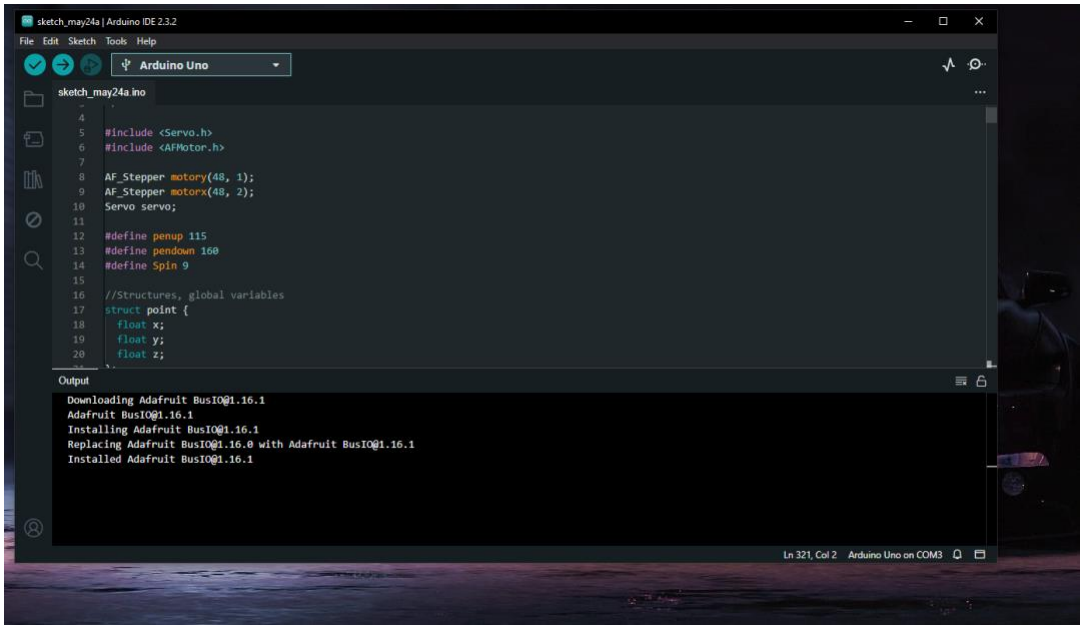


Figure 60: Enter the code.

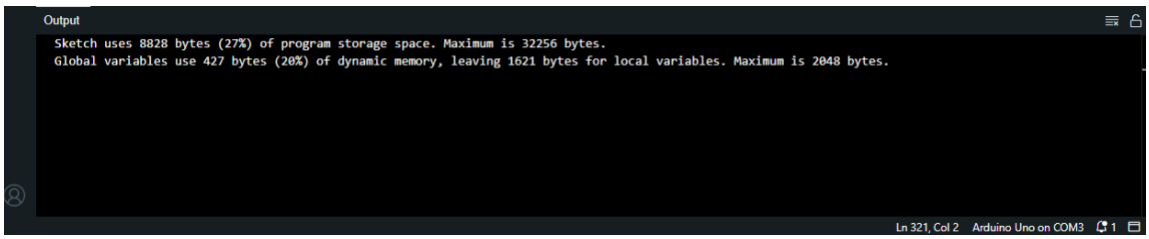


Figure 61: upload the code.

3.1.5 The Fourth Step

Processing is programming language software that is used for electronic drawings. The GTCRL processing program sends G-code files from the user interface to the CNC plotter.

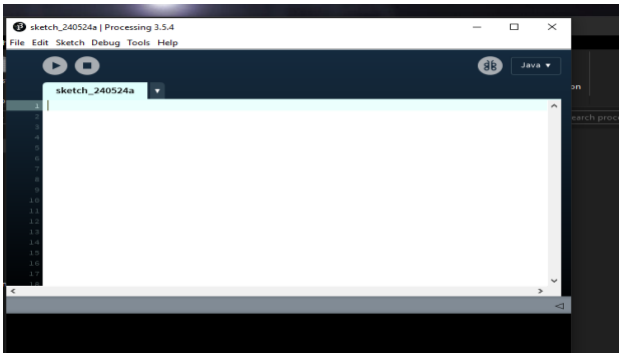


Figure 62: Processing interface.

Then the code must be entered, which is a Processing sketch designed to interface with a CNC machine via serial communication.

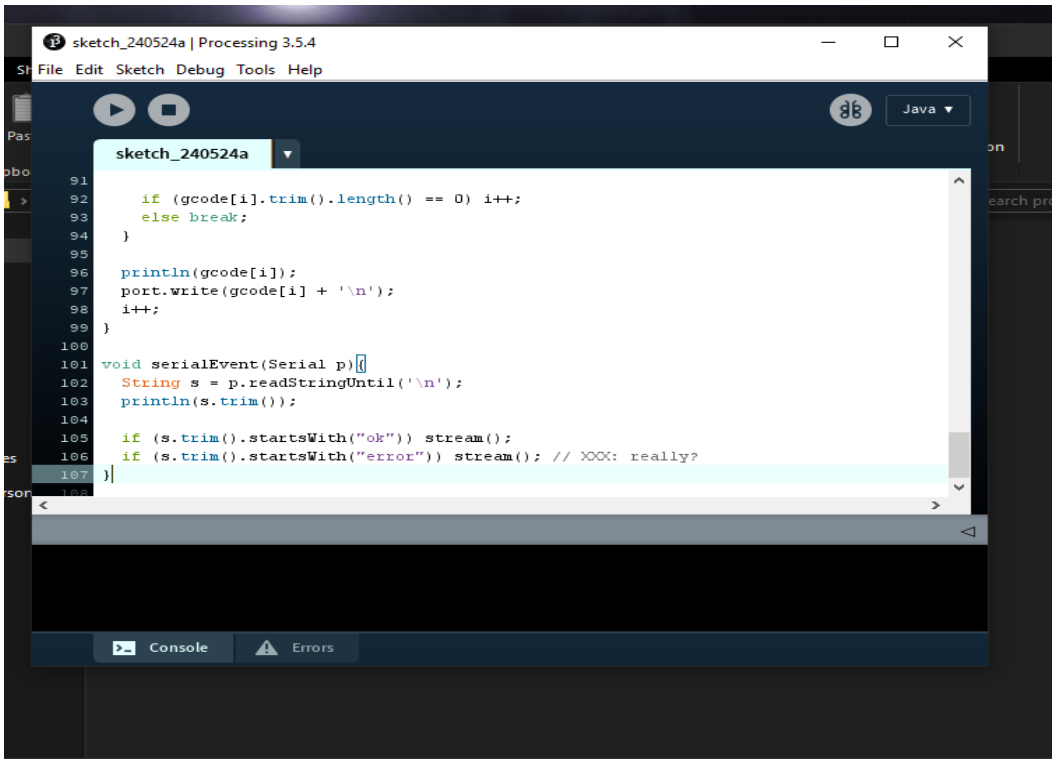


Figure 63: Enter the code.

This code is divided into main parts:

Import Statements, Global Variables, Function to Open Serial Port, Function to Select Serial Port, Setup Function, Draw Function, Key Press Event Handler, File Selected Callback, Stream Function, Serial Event Handler.

Overall, this code sets up a user interface to control a CNC machine, allowing the user to select a serial port, load G-code files, and stream those commands to the machine while handling serial communication events.

NOTE

The code and an explanation of its most important parts are attached in the appendix to this thesis.[22]

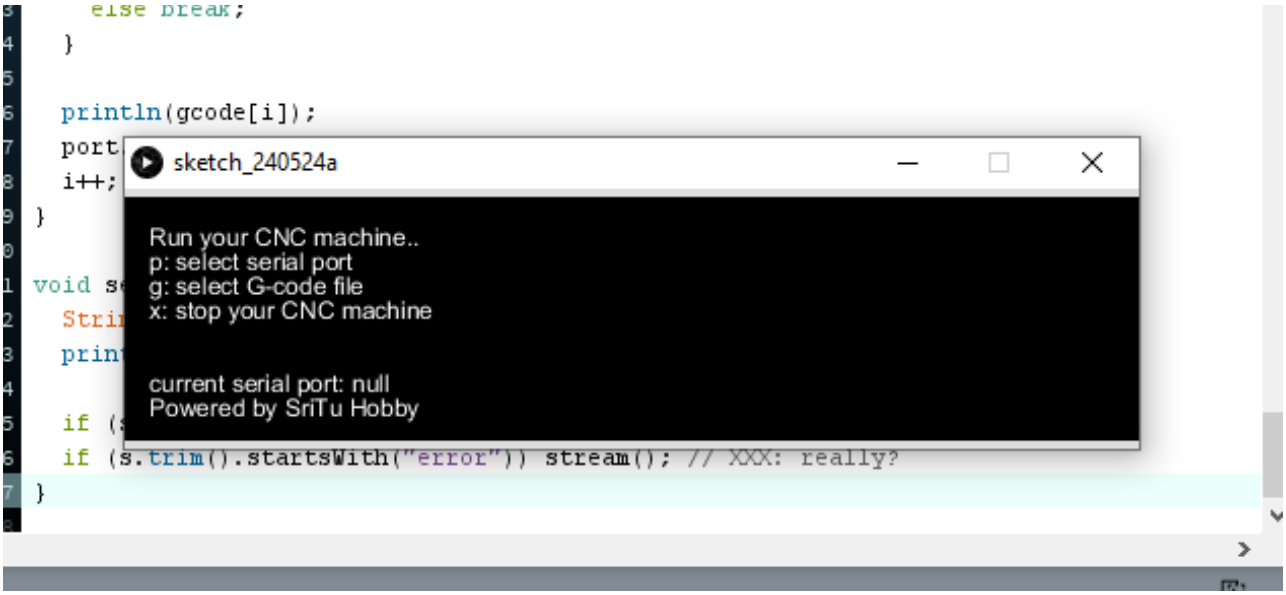


Figure 64: Run the CNC machine.

The port of Arduino Uno is selected by pressing ‘P’ button on the keyboard.

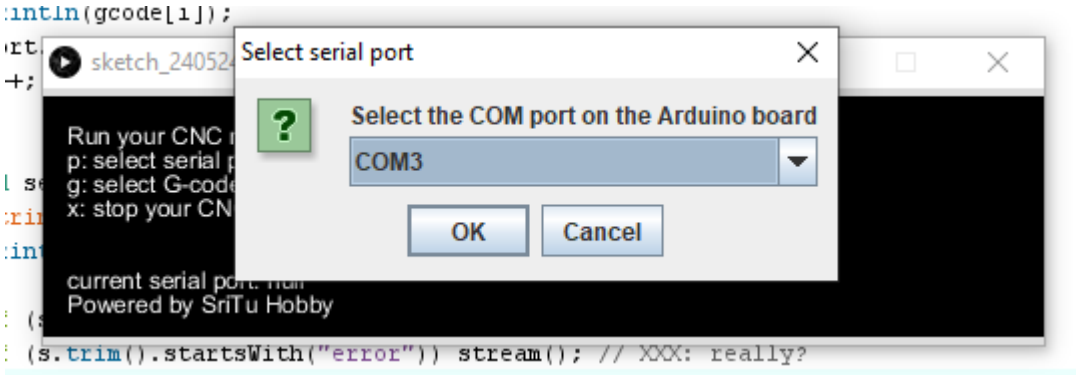


Figure 65: Select serial port.

The complete CNC code has been uploaded to the Arduino, after which the Arduino will be ready to receive the G-code file from the processing software. Upon pressing the 'G' button, the G-Code file will be selected.

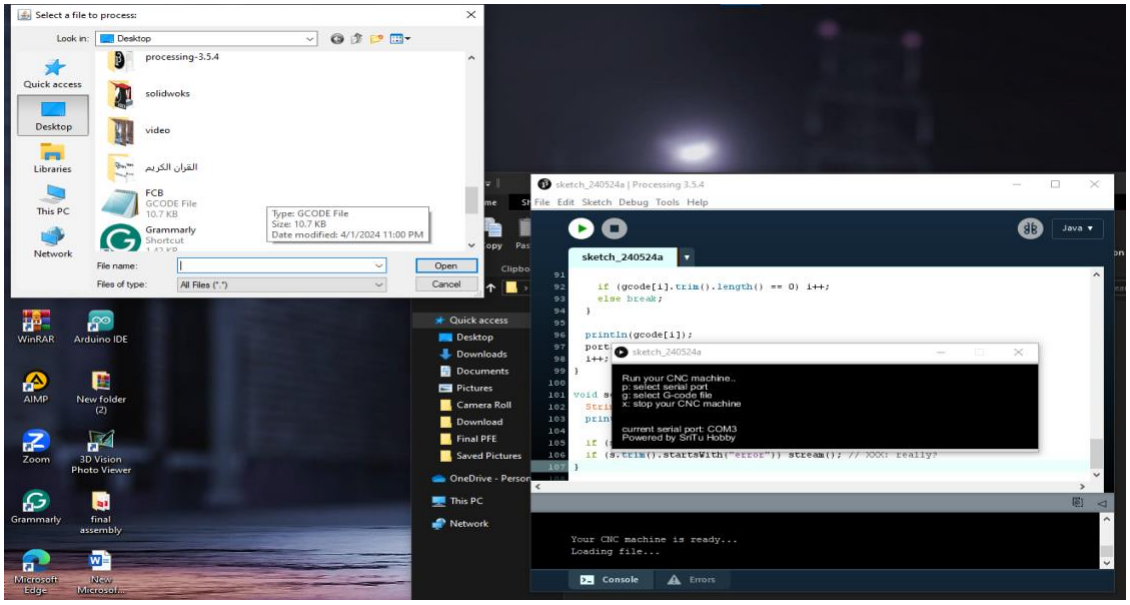


Figure 66: Select the G-Code file.

After this step, the machine will start working

NOTE

The instructions appearing on the user screen can be modified, whether by adding instructions or changing the control buttons, and this is done through the instructions included in the code that represent the messages that appear on the screen.

```
void draw(){
  background(0);
  fill(255);
  int y = 24, dy = 12;
  text("Run your CNC machine..", 12, y); y += dy;
  text("p: select serial port", 12, y); y += dy;
  text("g: select G-code file", 12, y); y += dy;
  text("x: stop your CNC machine", 12, y); y += dy;
```

Figure 67:Control UI messages.

3.2 Testing and Troubleshooting

3.2.1 Test Runs

After carrying out the various assembly operations (mechanical, electrical), we must first ensure the correct operation of the used motors.

After that, we will enter the stage of practical testing of the machine's performance.

We will conduct a simple test run to ensure proper functionality.We will follow the operating steps previously mentioned.

Operating example:



Figure 68: dragon logo.

The result of machine drawing:



Figure 69: dragon logo drawing our CNC plotter.

As the figure shows, the machine works smoothly, and the drawing is clear and identical to the example.

3.2.2 Troubleshooting

3.2.2.1 Firstly

Fixed errors during the mechanical assembly process, where we had to place the axle holders parallel to the body plates.

It is very important to ensure that the x and y axes are perpendicular to each other and to ensure that there is a principle that represents the zero point for each of them.

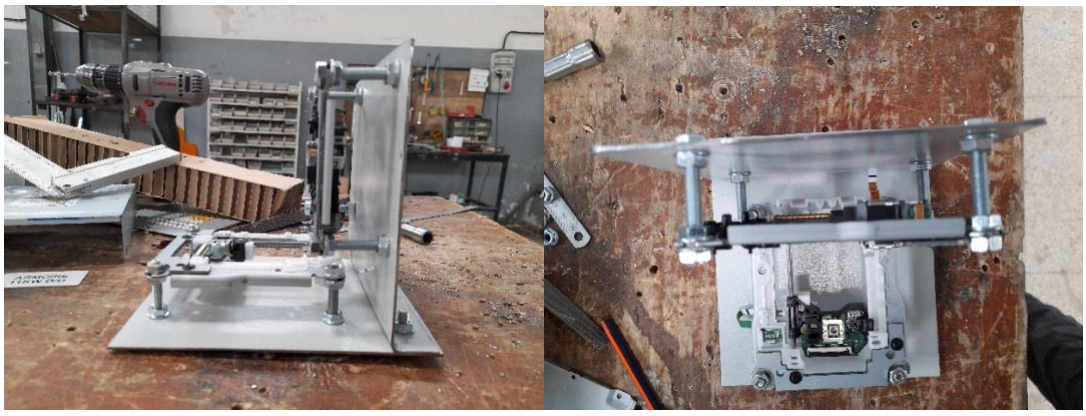


Figure 70: An image showing the perpendicularity of the axle holders.

Ensure that the motor poles are connected using a multimeter. As the following Figure explains.



Figure 71: Ensure that the poles of the stepper motors are soldered to the connecting wires.

These were the errors that were fixed in terms of compilation and construction.

3.2.2.2 Secondly

In terms of control and programming, we will talk about the Arduino IDE program, where the definition that matches the type of motors that we used must be downloaded. As the following Figure explains.

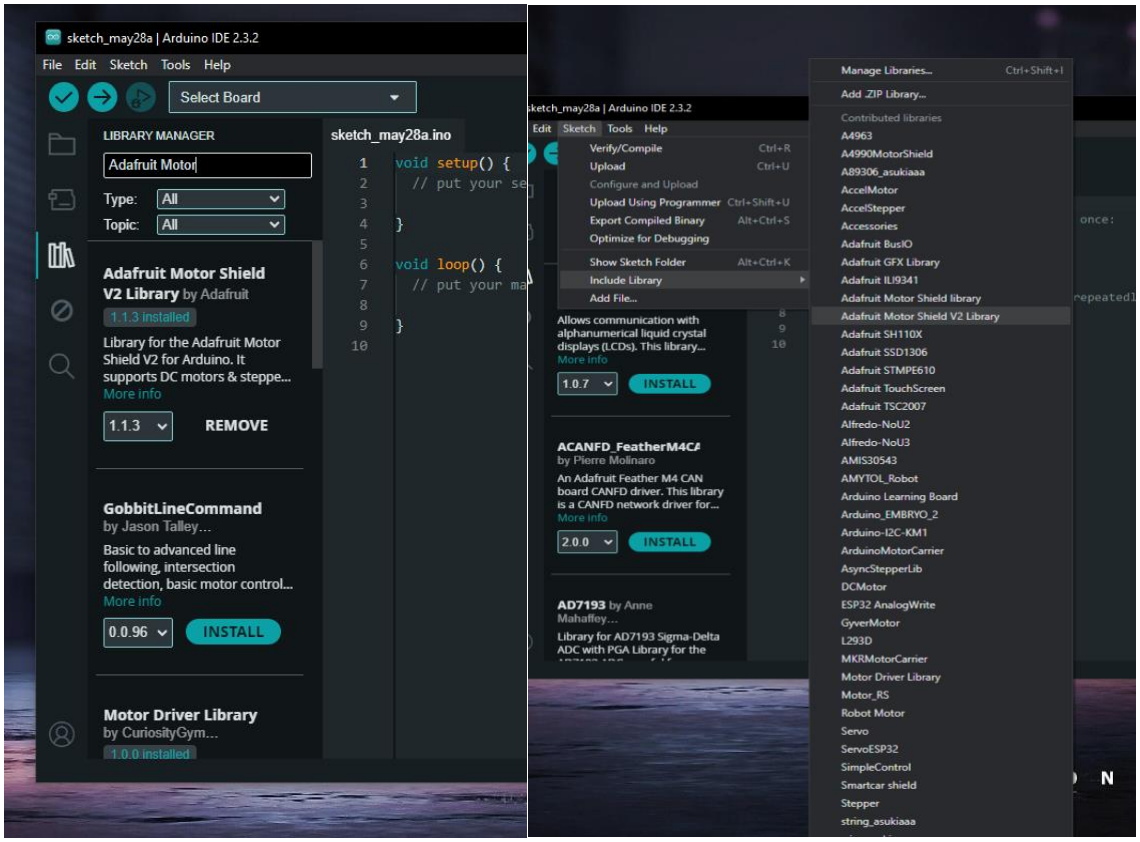


Figure 72: AFmotor library.

Before connecting the electrical supply, this port must be deleted.

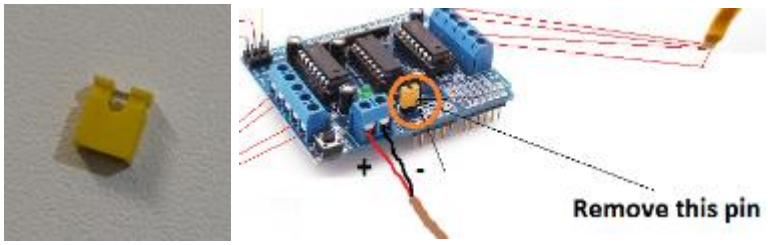


Figure 73: The pin must be removed.

General Conclusion

This project demonstrates the development of a mini CNC plotter designed for 2D drawing tasks, utilizing Arduino technology and repurposed components from old DVD drives. The project encompassed mechanical design, electronic circuitry, and programming, resulting in an affordable tool capable of producing precise and detailed drawings.

The mini CNC plotter serves as a versatile educational tool, integrating principles from mechatronics, electronics, and software engineering. Its applications range from academic projects and prototyping to artistic endeavors and PCB manufacturing, highlighting the flexibility and utility of CNC technology.

By employing readily available materials and open-source software, this project underscores the potential of CNC machines and aids in the broader dissemination and understanding of CNC technology. The successful completion of this plotter underscores the impact of innovation in mechatronics, offering practical solutions and encouraging further advancements in both educational and industrial contexts.

In summary, the mini CNC plotter exemplifies how contemporary technology can be harnessed to create practical, efficient, and educational tools, paving the way for future innovations and applications in CNC technology. The use of recycled materials, such as disk drives, simplifies the process by eliminating the need for complex calculations and extensive mathematical equations, thereby saving time and effort.

Potential Enhancements

One key feature of our device is its ability to undergo significant modifications, which can be explored as future projects, including but not limited to:

1. Modifying the Pen Holder

- Expanding the pen holder to accommodate more pens of different specifications. The design can be adjusted to hold various pen sizes, colors, or different types of drawing tools (e.g., markers, and pencils).

- Considering weight distribution and stability when adding more pens, potentially reinforcing the holder to prevent wobbling during operation.

2. Developing Work Tools

- Creating additional tools for tasks such as PCB board etching, sculpting, or cutting. Some ideas include:

- PCB Etching: Designing a custom attachment for holding an etching tool (like a small rotary tool) for precise PCB etching.

- Sculpting/Cutting: Attaching a mini rotary tool or laser module for detailed sculpting or cutting tasks.

3. Adding Controls and Sensors

- Implementing limit switch detectors to prevent the pen from exceeding its intended range, possibly using optical sensors.

- Adding homing sensors to ensure accurate positioning at the start of each drawing session.

- Incorporating safety features to stop the plotter if it encounters an obstacle or reaches its limits.

4. Display Interface

- Adding a display screen to show real-time status and information, such as:

  - Current position (X, Y coordinates).

  - Pen status (up/down).

  - Error messages (e.g., paper jam, pen out of ink).

  - Progress during long drawings.

- Utilizing small OLED displays or repurposing an old smartphone screen.

Our project machine serves as a foundation for other machine installations, which can be customized and developed to meet specific purposes and requirements.

Bibliography

[1] X. W. Xu and S. T. Newman, “Making CNC machine tools more open, interoperable and intelligent—a review of the technologies,” Computers in Industry, vol. 57, no. 2, pp. 141–152, Feb. 2006, doi: https://doi.org/10.1016/j.compind.2005.06.002.

[2] Universal Technical Institute , “6 of the Most Common CNC Machines,” UTI Corporate, Jun. 21, 2020. https://www.uti.edu/blog/cnc/6-cnc-machines.

[3] “What are the Benefits of CNC Machining Service vs. Manual Machining?,” Eagle Stainless, Jan. 24, 2019. https://eagletube.com/about-us/news/what-are-the-advantages-of-cnc-machining-vs-manual-machining/.

[4] “What is a plotter? | What are the main types of plotters? | Lenovo US,” www.lenovo.com. https://www.lenovo.com/us/en/glossary/plotter/?orgRef=https%253A%252F%252Fwww.google.com%252F (accessed May 10, 2024).

[5] “About Electrostatic plotter,” dbpedia.org. https://dbpedia.org/page/Electrostatic_plotter (accessed May 10, 2024).

[6] “What is a plotter? | What are the main types of plotters? | Lenovo US,” www.lenovo.com. https://www.lenovo.com/us/en/glossary/plotter/?orgRef=https%253A%252F%252Fwww.google.com%252F.

[7] “What is a PCB?,” www.youtube.com. https://www.youtube.com/watch?v=YJr-kHy6STg.

[8] Arduino, “Software,” www.arduino.cc, 2023. https://www.arduino.cc/en/software.

[9] “Welcome to the Inkscape Beginners’ Guide! — Inkscape Beginners’ Guide 1.0 documentation,” inkscape-manuals.readthedocs.io. https://inkscape-manuals.readthedocs.io/en/latest/index.html (accessed May 12, 2024).

[10] “Overview \ Processing.org,” processing.org. https://processing.org/overview/ .

[11] “What Is SolidWorks? – Simply Explained,” All3DP, Oct. 28, 2020. https://all3dp.com/2/what-is-solidworks-simply-explained/.

[12] “Free CAD Designs, Files & 3D Models | The GrabCAD Community Library,” grabcad.com. https://grabcad.com/library/dvd-cd-drives-stepper-motor-1 (accessed May 14, 2024).

[13] “Free CAD Designs, Files & 3D Models | The GrabCAD Community Library,” grabcad.com. https://grabcad.com/library/cnc-pen-holder-2 (accessed May 14, 2024).

[14] “ARDUINO PROJECTS BOOK.” Available: https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf.

[15] R. Hari. Sudhan, M. Ganesh. Kumar, A. Udhaya. Prakash, S. Anu. R. Devi, and S. P., “ARDUINO ATMEGA-328 MICROCONTROLLER,” IJIREEICE, vol. 3, no. 4, pp. 27–29, Apr. 2015, doi: https://doi.org/10.17148/ijireeice.2015.3406.

[16] “L293D Motor Driver IC introduction, pinouts and how to use,” Microcontrollers Lab, Nov. 12, 2019. https://microcontrollerslab.com/l293d-motor-driver-ic-introduction-pinouts-example/.

[17] G. Slemon, “Electric motor,” Encyclopædia Britannica. 2019. Available: https://www.britannica.com/technology/electric-motor.

[18] S. Yousif, “Microcontroller Design Lecture 6: Servo Motor.” Accessed: May 21, 2024. [Online]. Available: https://uomus.edu.iq/img/lectures21/MUCLecture_2022_51835998.pdf.

[19] “Stepper Motors Basics: Types, Uses, and Working Principles.” Available: https://media.monolithicpower.com/mps_cms_document/2/0/2020-stepper-motors-basics-types-uses-_and-working-principles_r1.0.pdf.

[20] “Wiring up a 4-pin DVD drive stepper motor?,” Electrical Engineering Stack Exchange. https://electronics.stackexchange.com/questions/134930/wiring-up-a-4-pin-dvd-drive-stepper-motor (accessed May 23, 2024).

[21] “Stepper Motor or Servo Motor: Which Should It Be?” Available: https://www.kollmorgen.com/sites/default/files/Servo-or-Stepper-Guide_KM_WP_000286_RevC_EN.pdf.

[22] S. Hobby, “How to make a DIY Arduino CNC drawing machine at home,” SriTu Hobby, Jul. 16, 2021. https://srituhobby.com/how-to-make-a-diy-arduino-cnc-drawing-machine-at-home/#google_vignette (accessed May 28, 2024).

Internet link

[1.1] “Precision Engineering Hampshire | Berkshire | Surrey | Oxford & UK,” Atlantic Precision. https://www.atlantic-precision.co.uk (accessed May 10, 2024).

[2.2]“DIAM 1670 - 3-axis CNC milling machine by SOMAB | DirectIndustry,” www.directindustry.com. https://www.directindustry.com/prod/somab/product-8100-1083951.html (accessed May 10, 2024).

[3.3]“CNC Lathes and Turning Centres,” Mills CNC. https://www.millscnc.co.uk/cnc-machines/lathes-turning-centres/ (accessed May 10, 2024).

[4.4]“Top Rated 4 Axis CNC Router 1325 with 4x8 Rotary Table,” STYLECNC. https://www.stylecnc.com/cnc-wood-router/STYLECNC-1325-CNC-router-with-4-axis-rotary.html (accessed May 10, 2024).

[5.5]“2024 Top Rated 4x8 CNC Plasma Cutting Table for Sale,” STYLECNC. https://www.stylecnc.com/plasma-cutter/cnc-plasma-cutting-table.html (accessed May 09, 2024).

[6.6] “AR65-MA - Wire-cut EDM by Langfang Novick Digital Equipment Co.,Ltd. | DirectIndustry,” www.directindustry.com. https://www.directindustry.com/prod/langfang-novick-digital-equipment-co-ltd/product-206445-2478994.html (accessed May 09, 2024).

[7.7]“Laser Cutting Machine | CNC Laser Cutter | ACCURL,” accurlcnc.com. http://accurlcnc.com/5-1-laser-cutting-machine/205111/ (accessed May 09, 2024).

[8.8] “Pen plotter,” www.basjacobs.com. https://www.basjacobs.com/post/2018-06-28-pen-plotter/ (accessed May 10, 2024).

[9.9] LPKF, “LPKF ProtoMat E44,” www.lpkf.com. https://www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/products/lpkf-protomat-e44. (accessed May 10, 2024).

[10.10] “EasyDraw V2 Writing and Drawing Machine (FULLY ASSEMBLED) – shopmakerq.com,” Shopmakerq.com, Mar. 03, 2021. https://shopmakerq.com/product/2d-cnc-plotter-a-machine-for-writing-and-drawing-fully-assembled/.(accessed May 12, 2024).

[11.11] “My Plotter CNC Machine - download free 3D model by TarekAlMusalli - Cad Crowd,” www.cadcrowd.com. https://www.cadcrowd.com/3d-models/my-plotter-cnc-machine (accessed May 12, 2024).

APPENDIX

Appendix

Appendix A: Documents showing the dimensions of parts and pieces used in mechanical assembly.

Frame parts

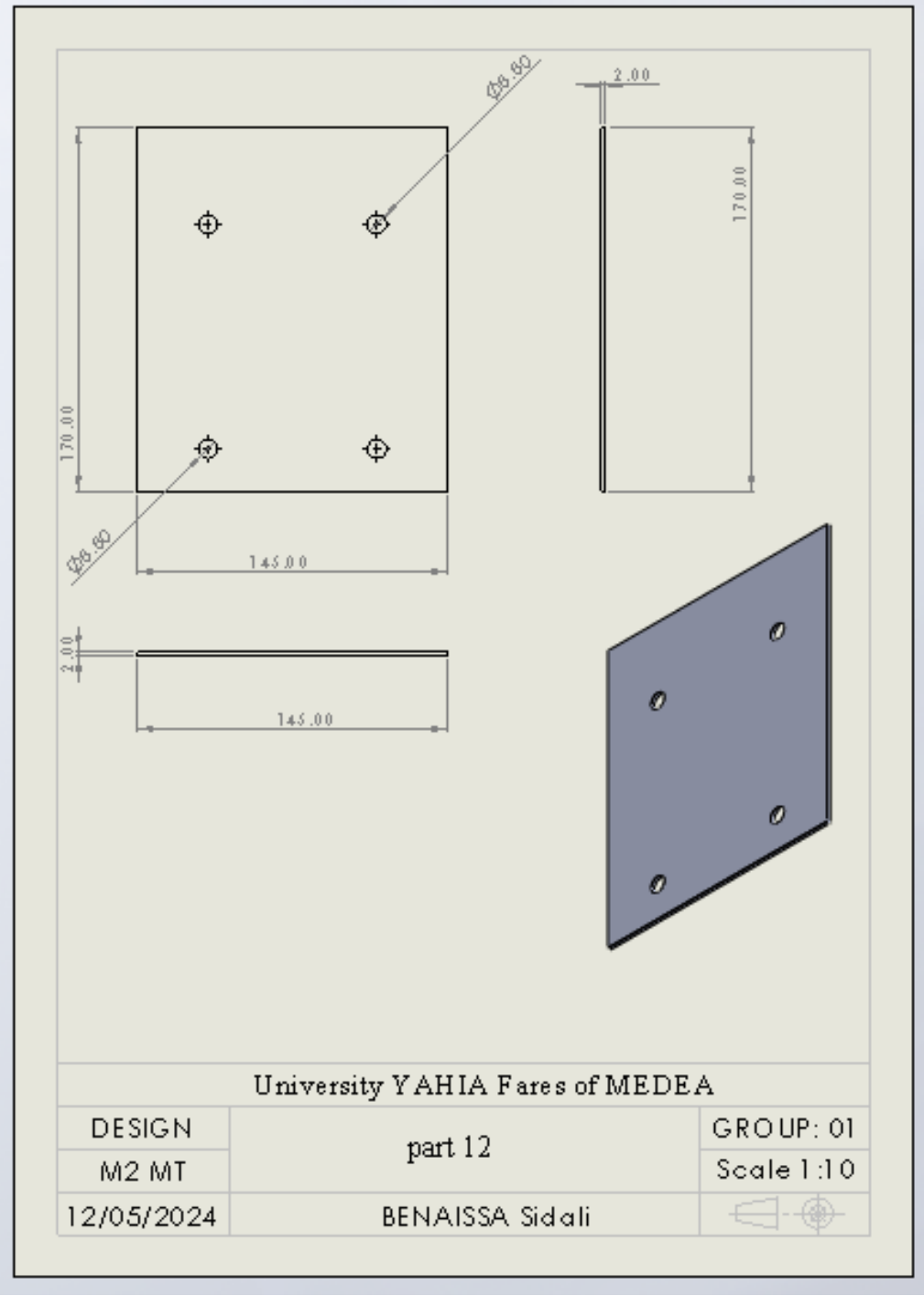


Figure 74 : Part 1.

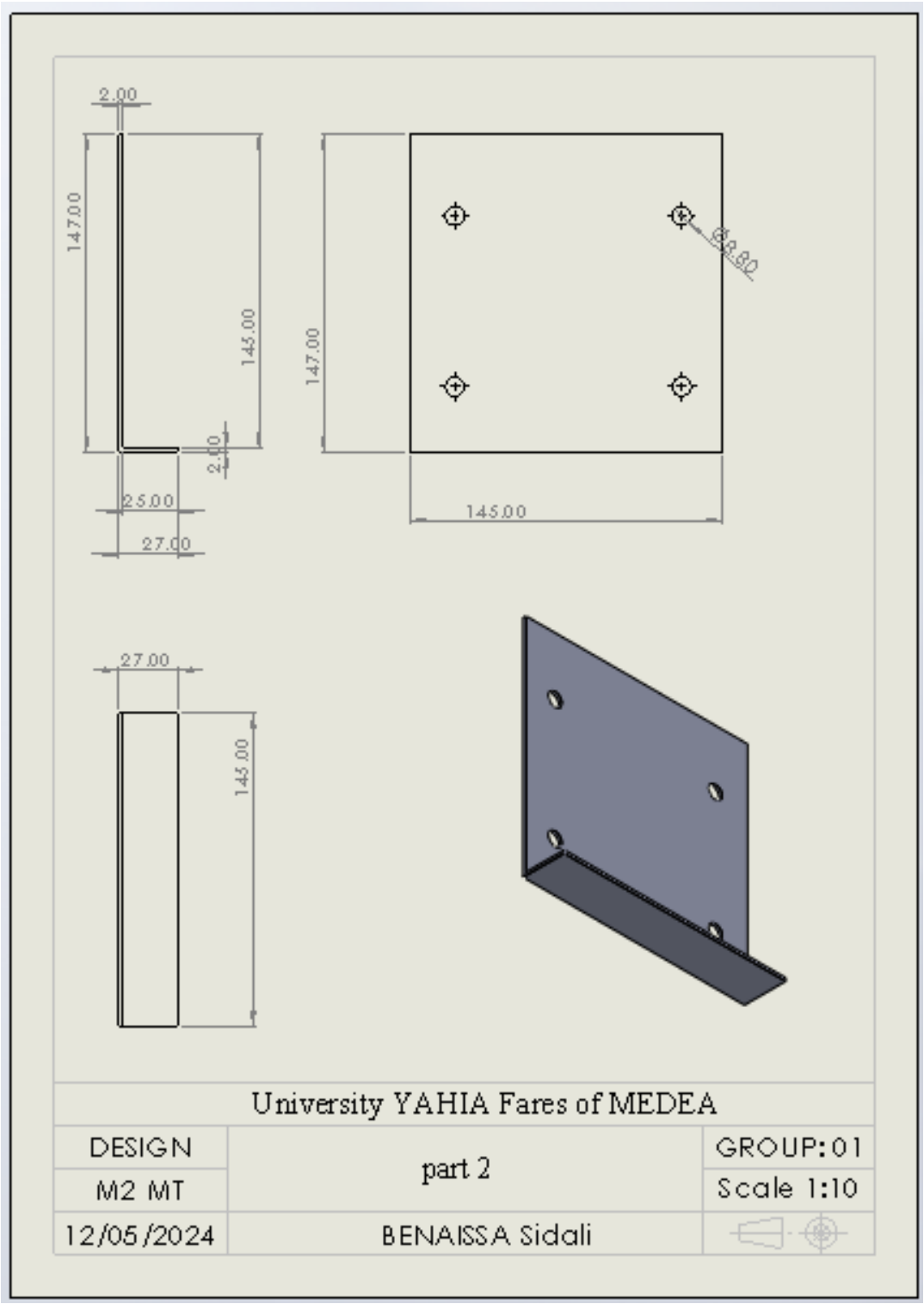


Figure 75: Part 2.

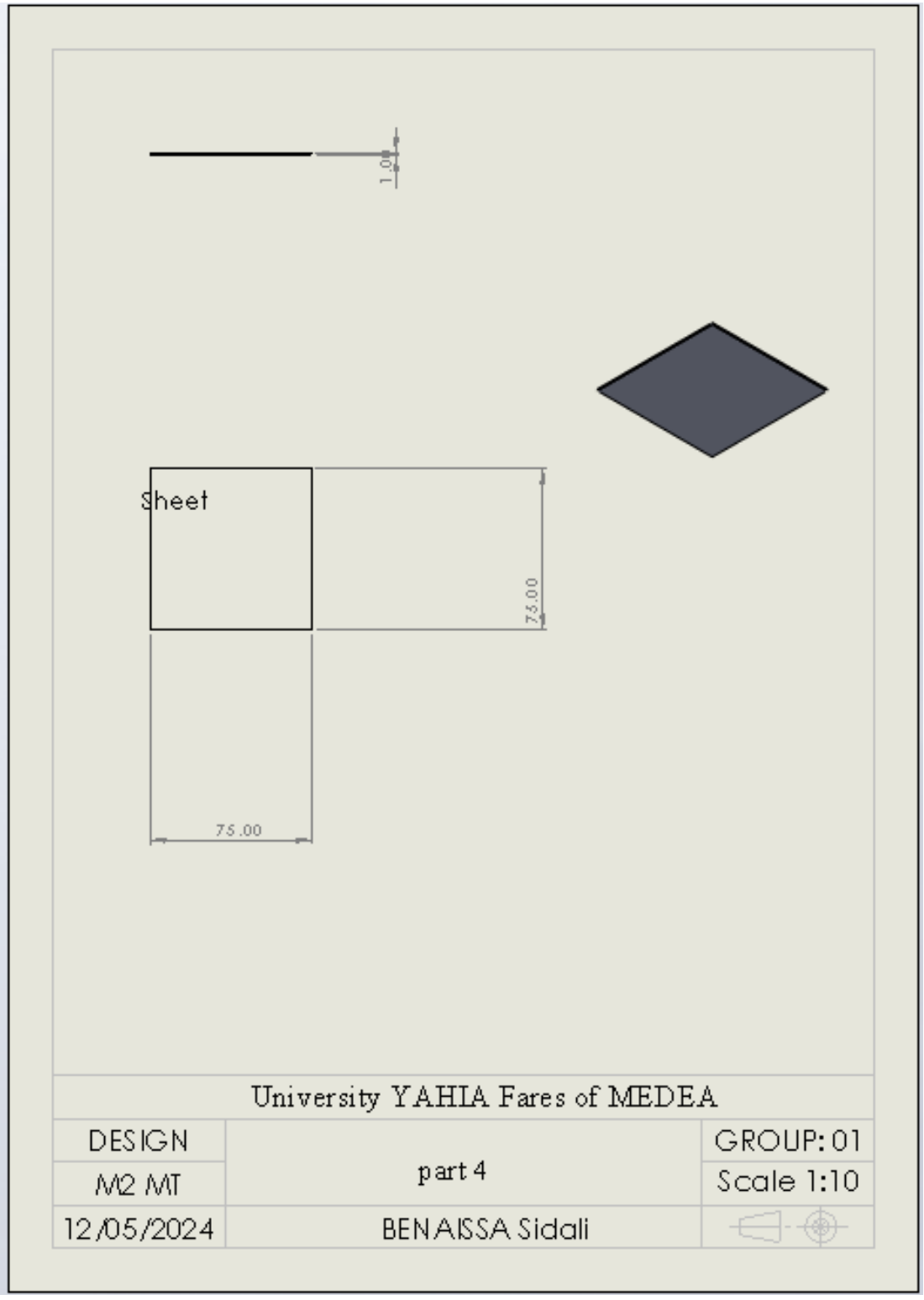


Figure 76:Part 4.

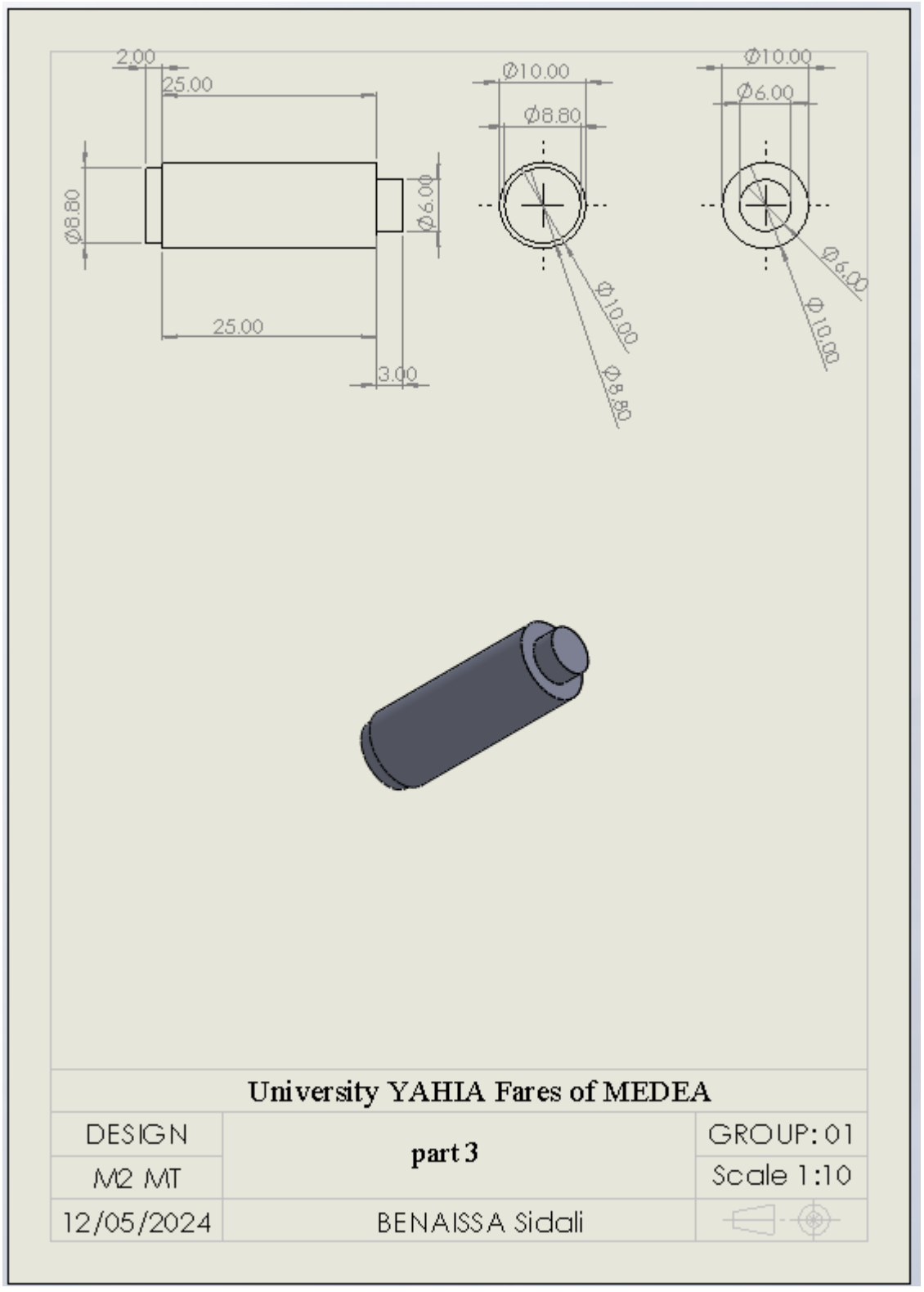


Figure 77: Part3.

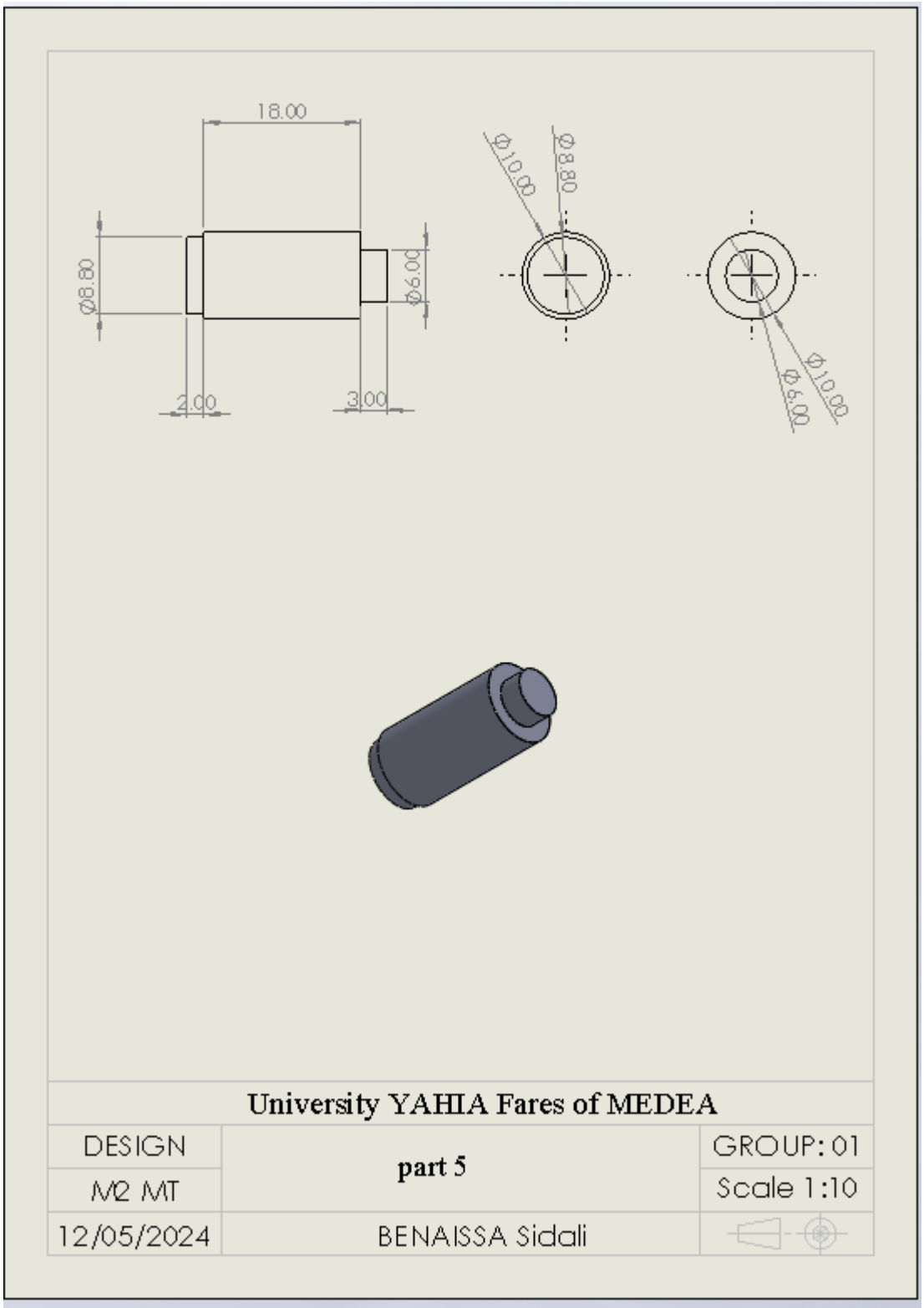


Figure 78:Part 5.

Pen holder

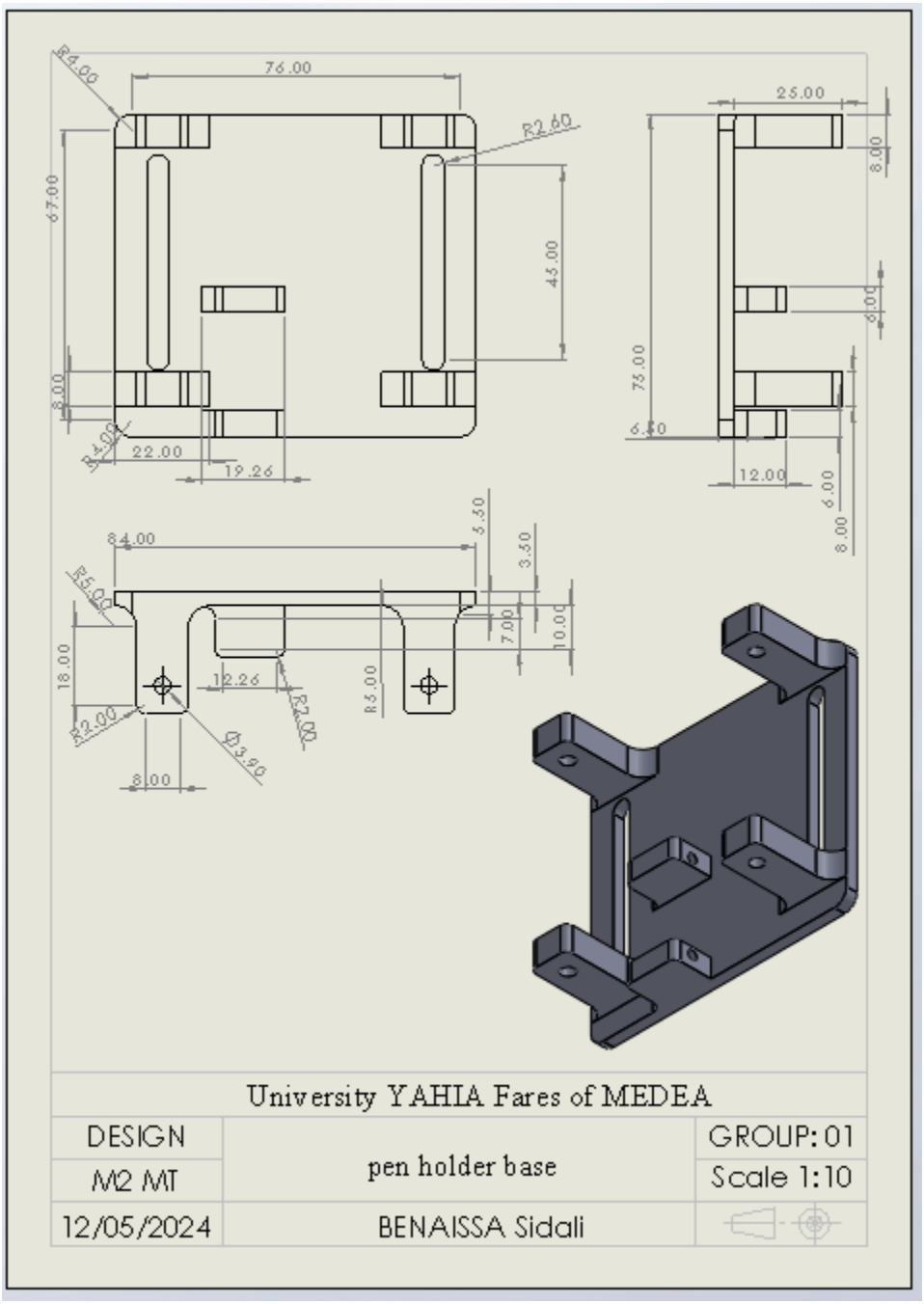


Figure 79:Pen holder base.

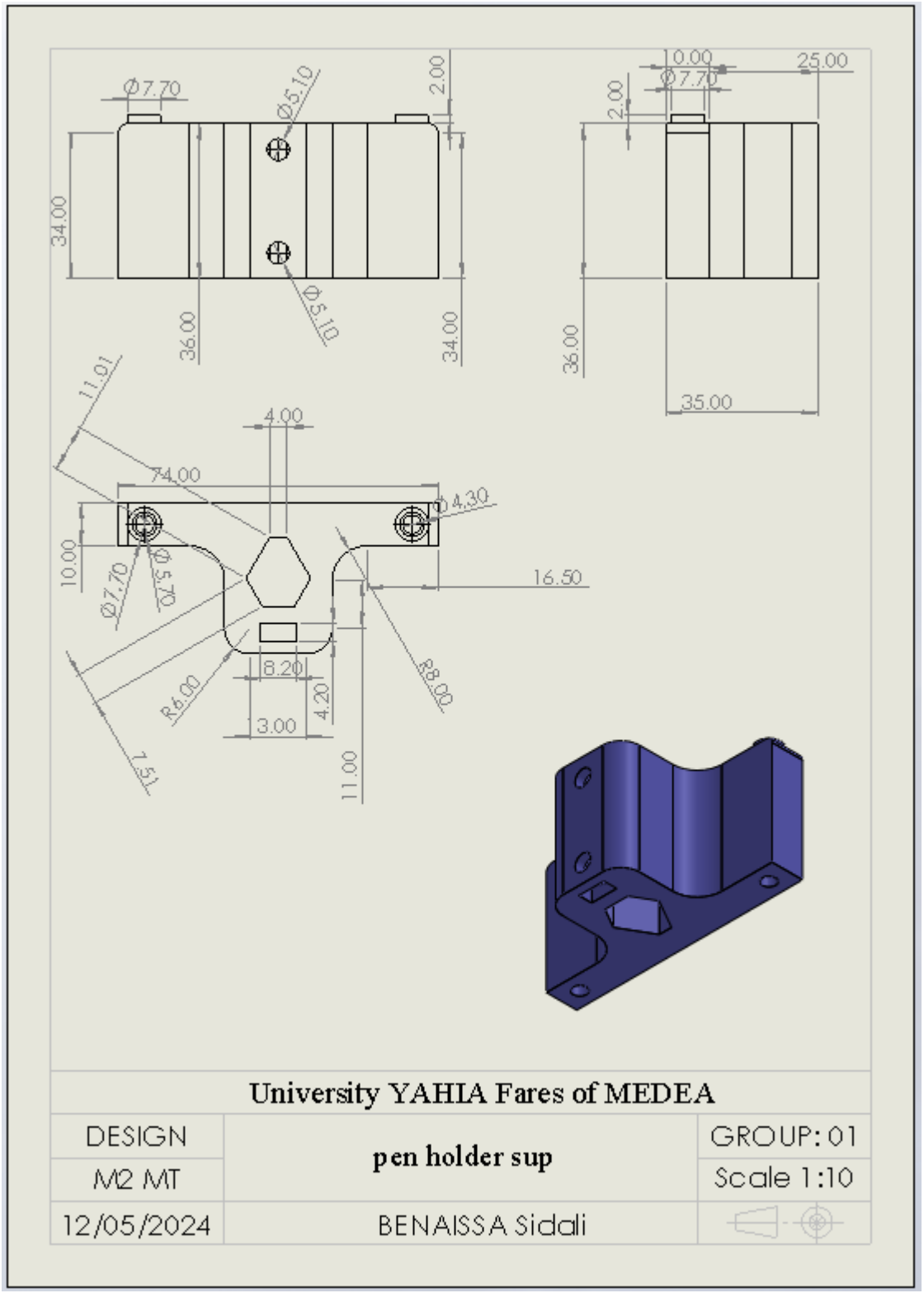


Figure 80:Pen holder sup.

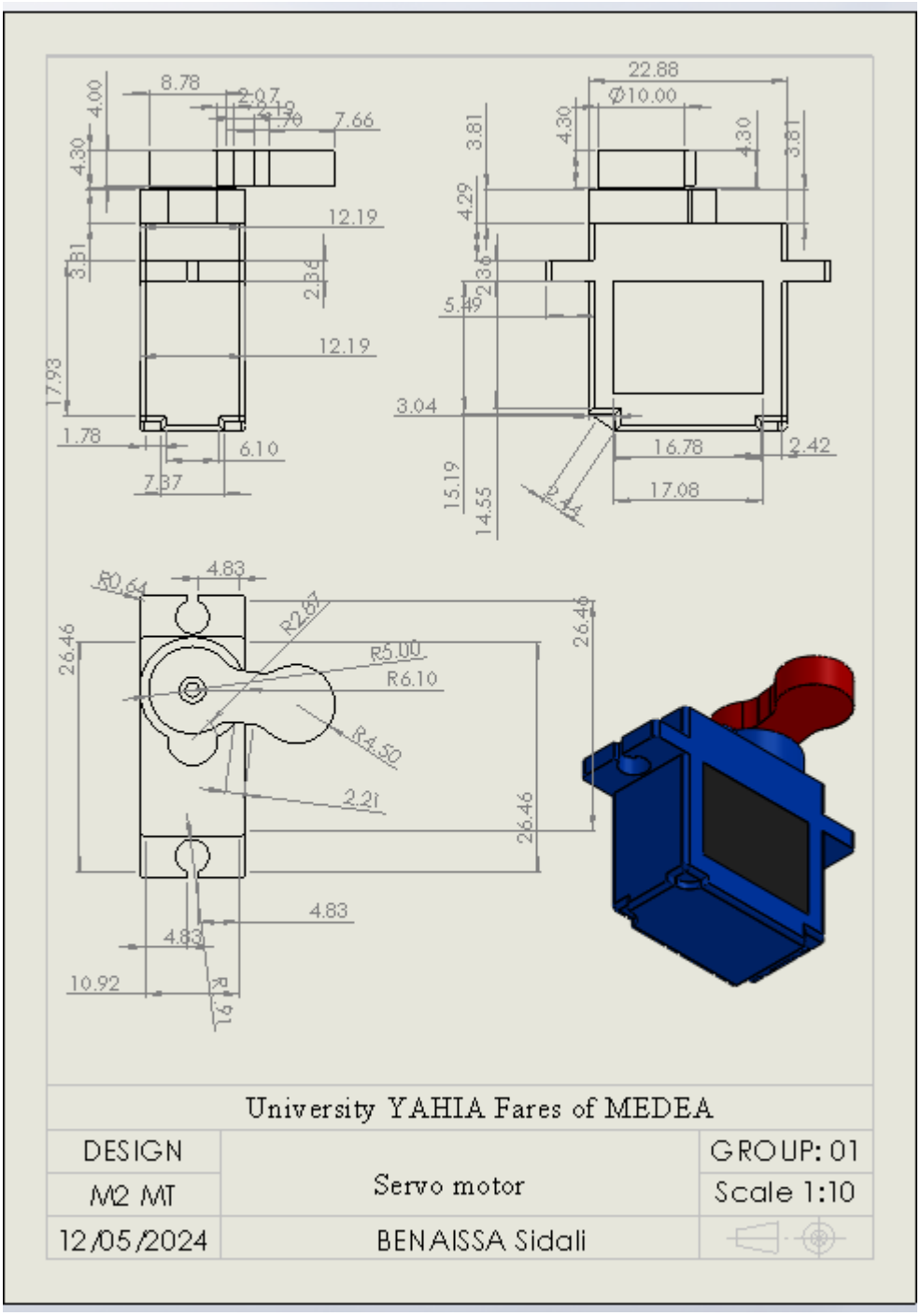


Figure 81:Servo Motor.

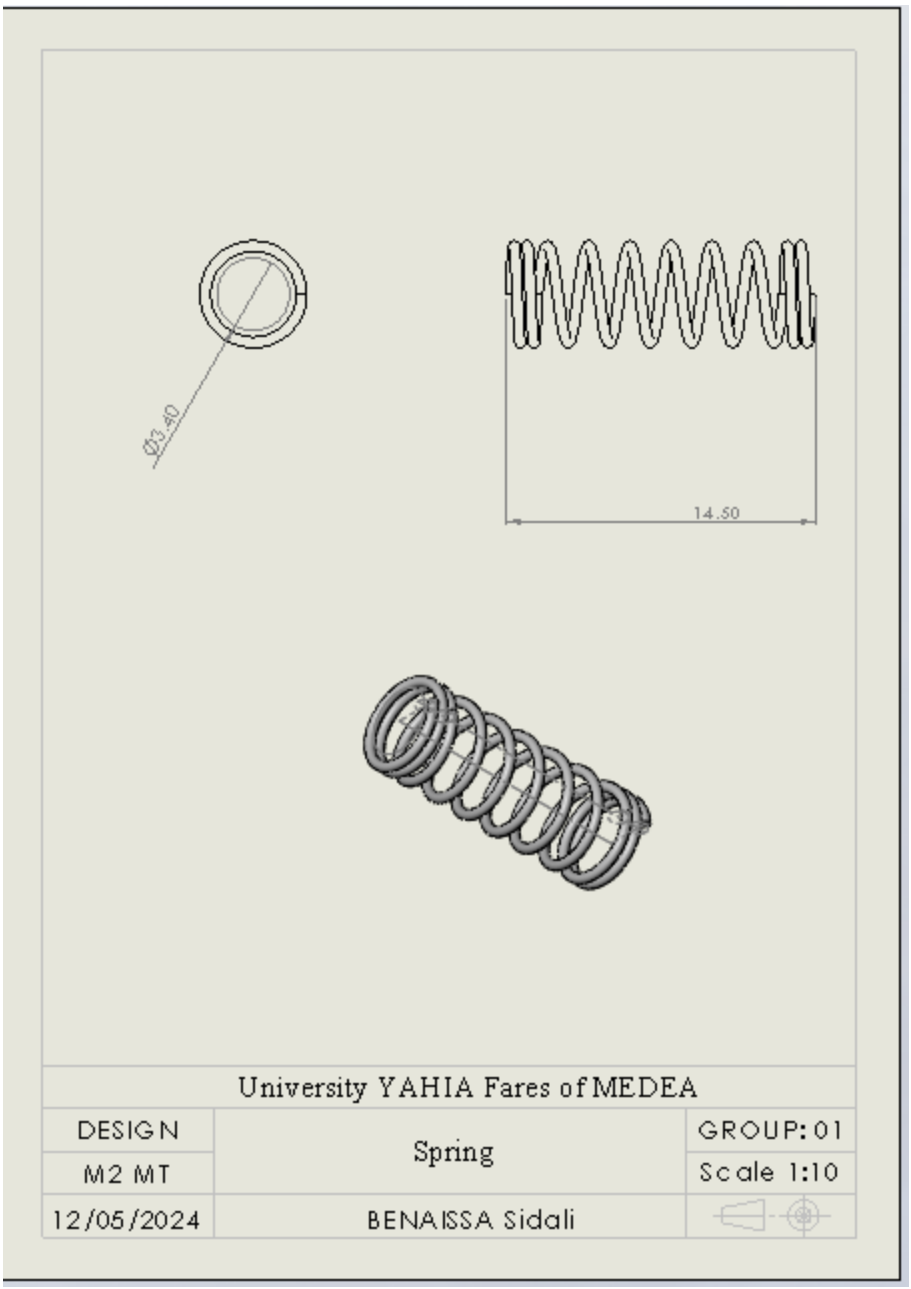


Figure 82: Spring.

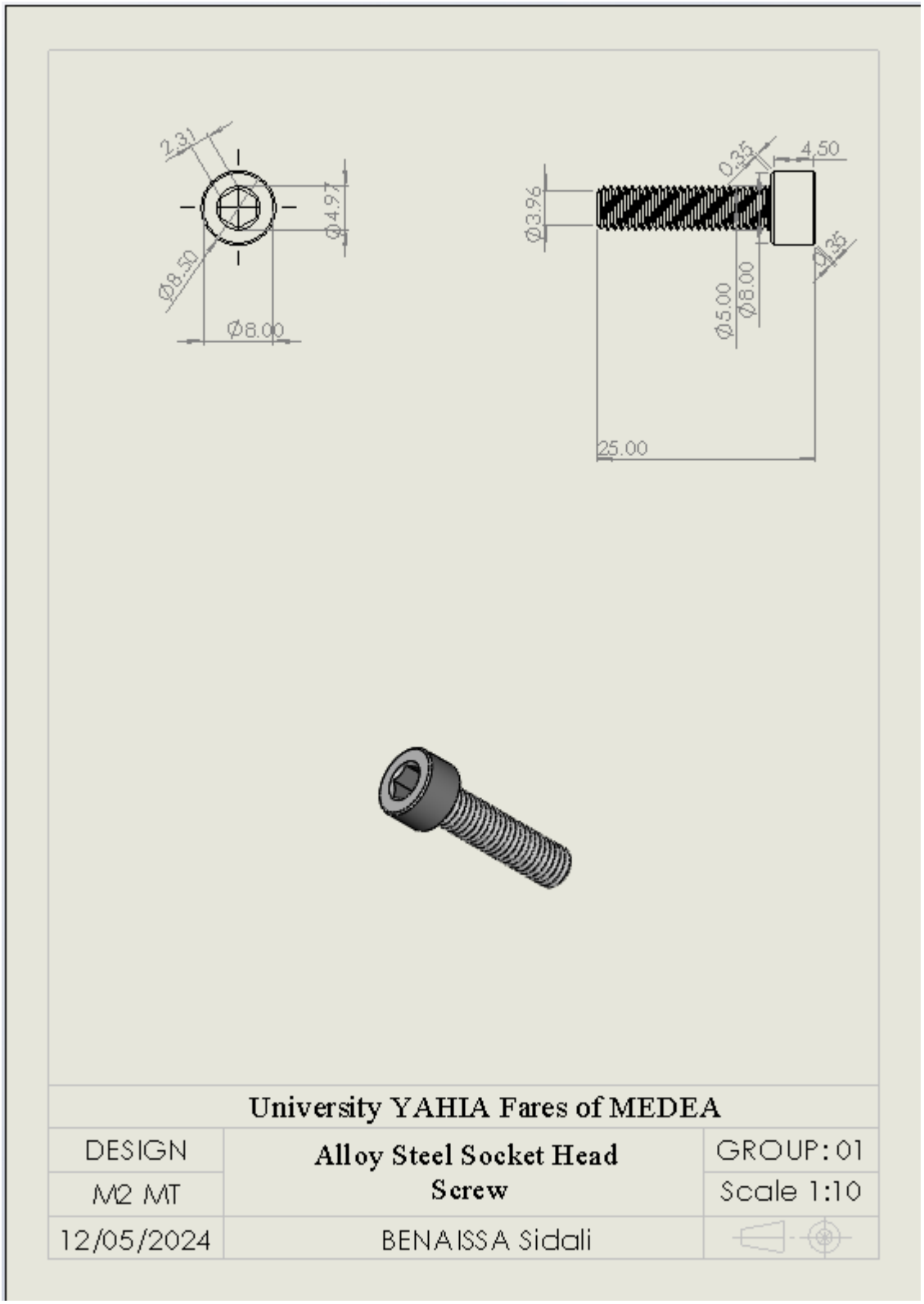


Figure 83:Alloy Steel Socket Head Screw.

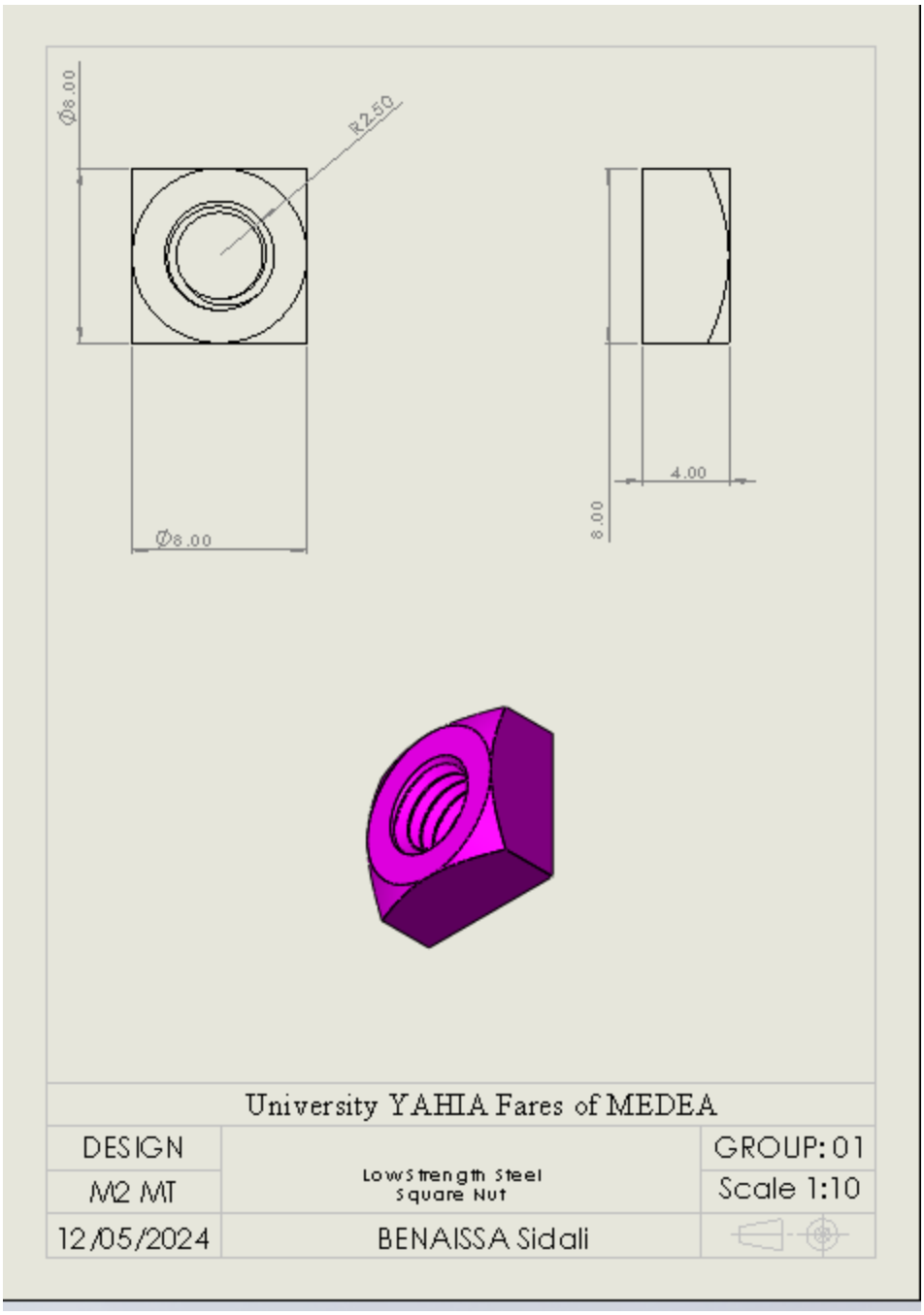


Figure 84:Square Nut.

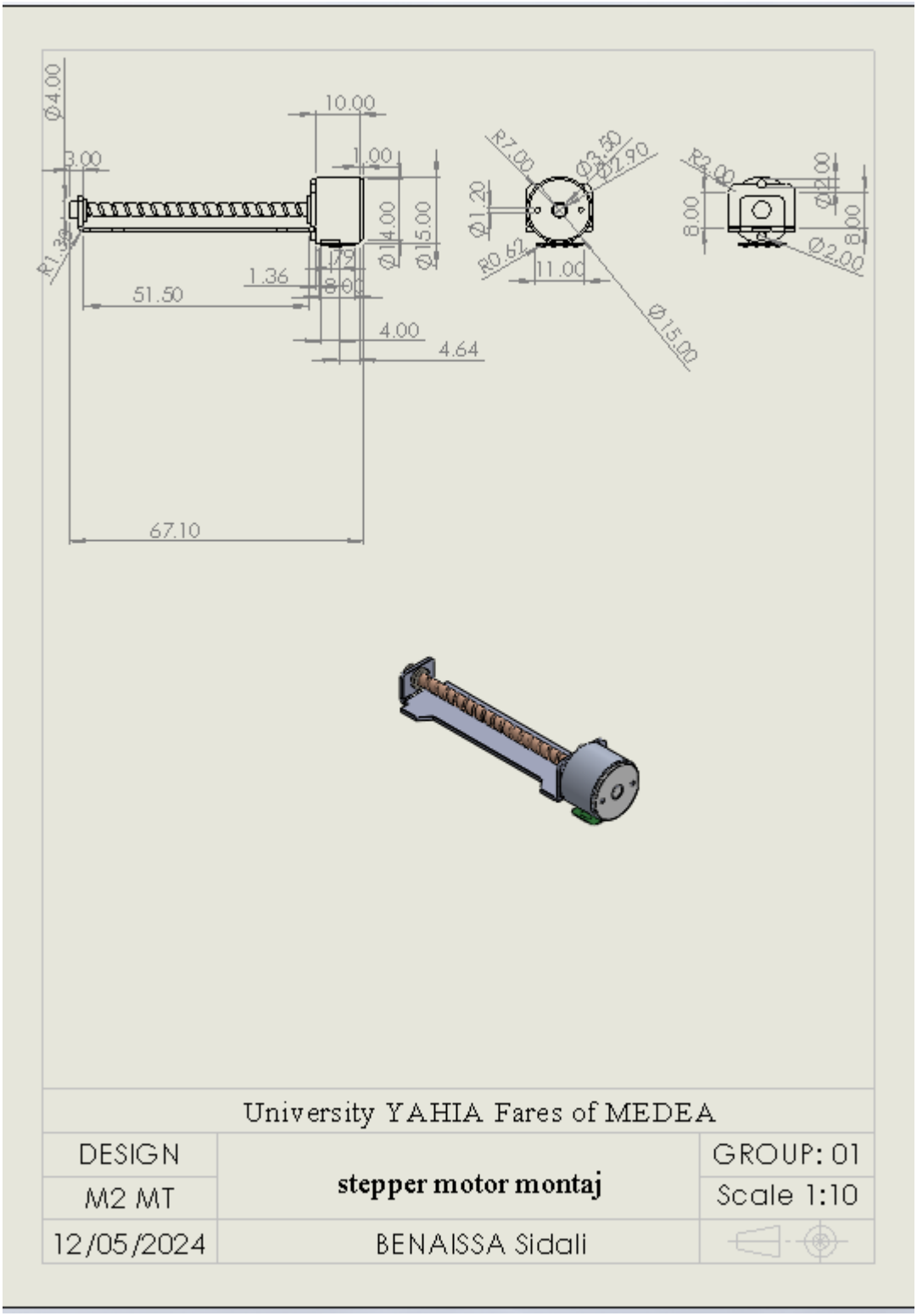


Figure 85: Stepper Motor.

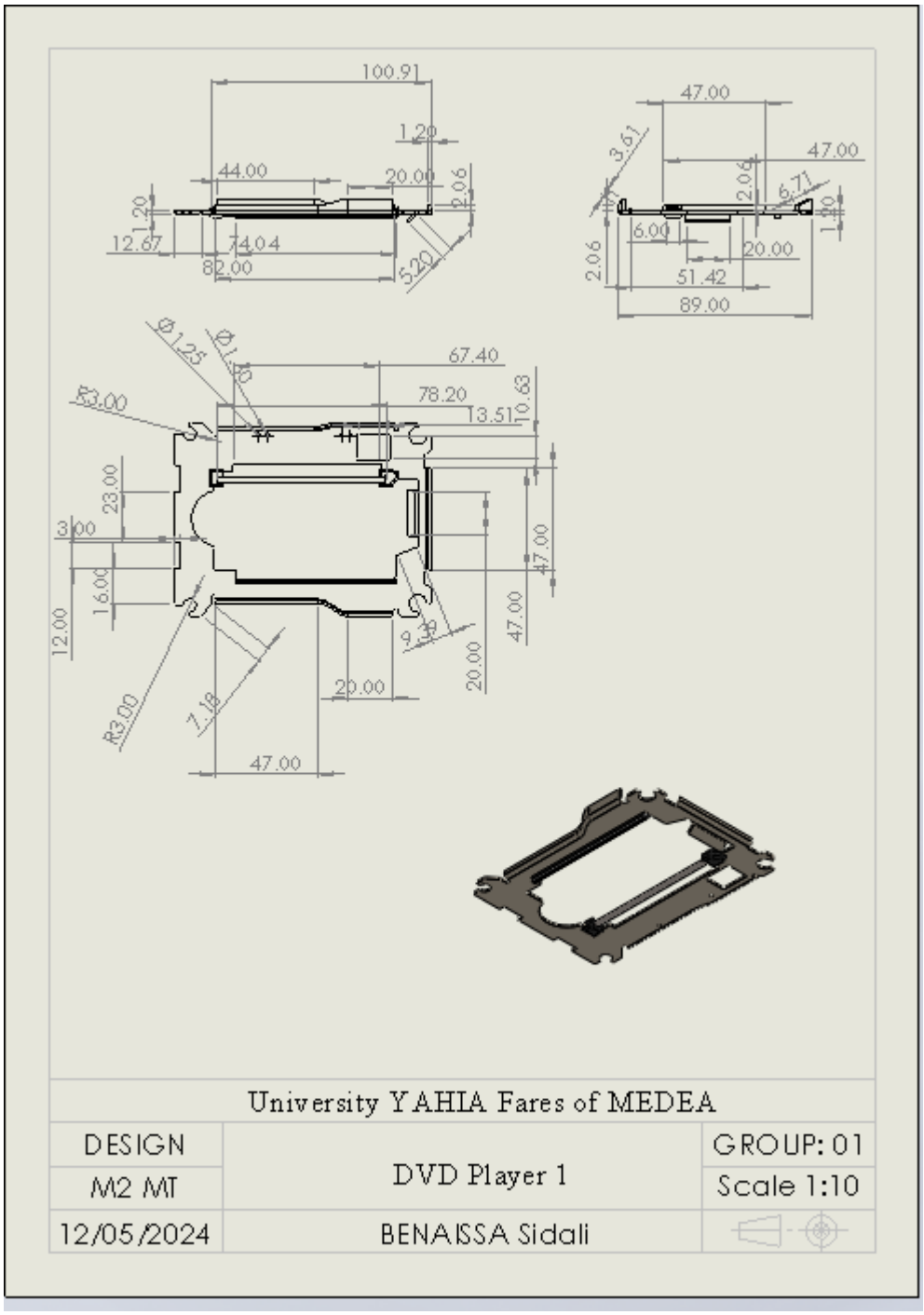


Figure 86: DVD Player.

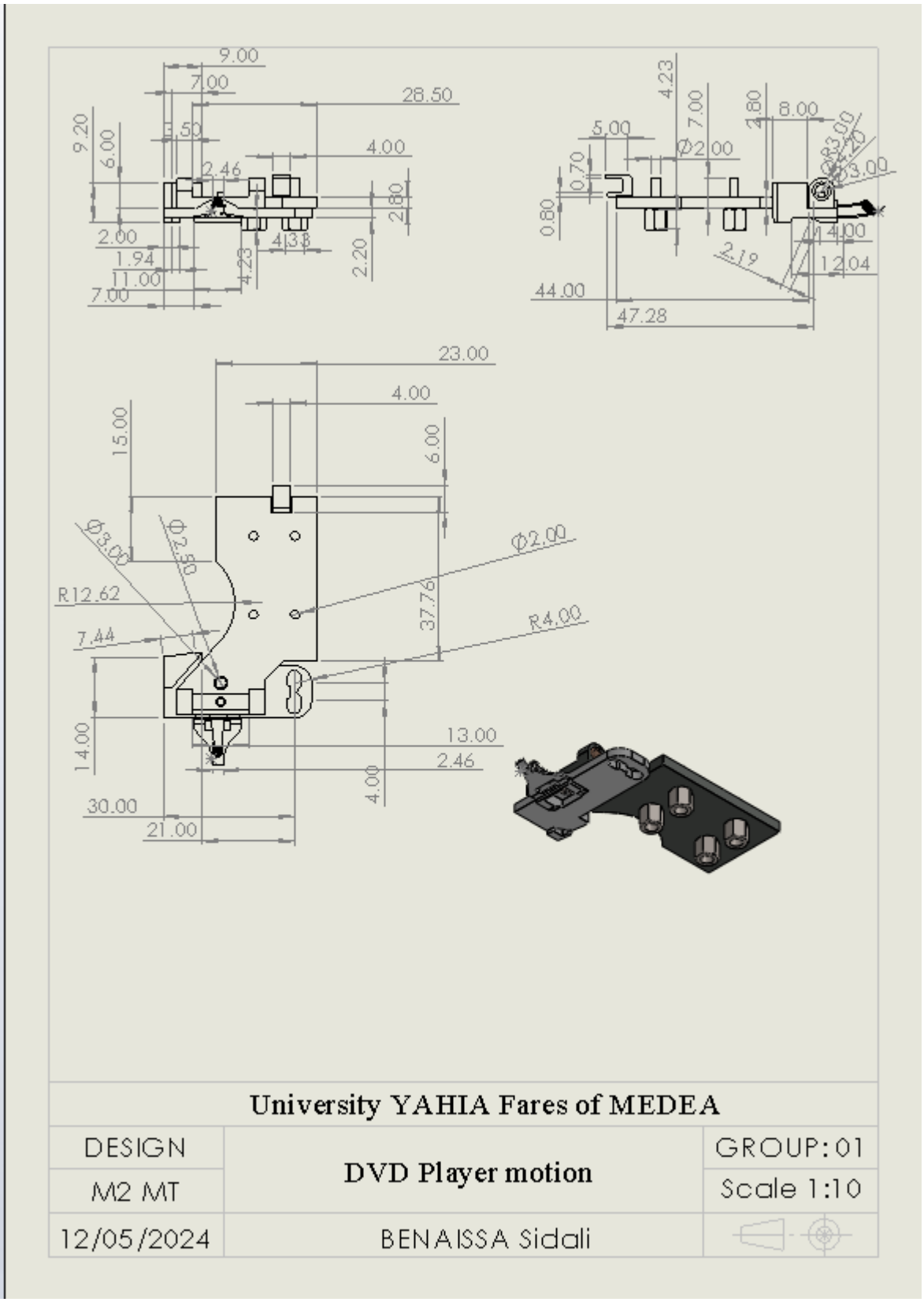


Figure 87: DVD Player motion.

Appendix B

Used codes.

Arduino code

```
/*How to make a mini CNC Plotter machine using Arduino*/

#include <Servo.h>
#include <AFMotor.h>

AF_Stepper motory(48, 1);
AF_Stepper motorx(48, 2);
Servo servo;

#define penup 115
#define pendown 160
#define Spin 9

//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;

//Steps per width and height of the drawing area
float StepsPerMillimeterX = 164.0;
float StepsPerMillimeterY = 250.0;

//Max and min values of the drawing area(mm)
float Xmin = 0;
float Xmax = 38;
float Ymin = 0;
float Ymax = 38;
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;



void setup() {
  Serial.begin( 9600 );
  servo.attach(Spin);
  servo.write(penup);
  delay(100);
```

```
  motorx.setSpeed(600);
  motory.setSpeed(600);
  Serial.println("Your CNC machine is ready...");
}

void loop() {
  delay(100);
  char line[512];
  char c;
  int lineIndex;
  bool lineIsComment, lineSemiColon;

  lineIndex = 0;
  lineSemiColon = false;
  lineIsComment = false;

  while (1) {


    while ( Serial.available() > 0 ) {
      c = Serial.read();
      if (( c == '\n') || (c == '\r') ) {
        if ( lineIndex > 0 ) {
          line[ lineIndex ] = '\0';
          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) ) {
    if ( c == ')' )  lineIsComment = false;
  }
  else {
    if ( c <= ' ' ) {
    }
    else if ( c == '/' ) {
    }
    else if ( c == '(' ) {
      lineIsComment = true;
    }
    else if ( c == ';' ) {
      lineSemiColon = true;
    }
    else if ( lineIndex >= 512 - 1 ) {
      Serial.println( "ERROR - lineBuffer overflow" );
      lineIsComment = false;
```

```
          lineSemiColon = false;
        }
        else if ( c >= 'a' && c <= 'z' ) {
          line[ lineIndex++ ] = c - 'a' + 'A';
        }
        else {
          line[ lineIndex++ ] = c;
        }
      }
    }
  }
}

void processIncomingLine( char* line, int charNB ) {
  int currentIndex = 0;
  char buffer[ 64 ];
  struct point newPos;

  newPos.x = 0.0;
  newPos.y = 0.0;

  while ( currentIndex < charNB ) {
    switch ( line[ currentIndex++ ] ) {
      case 'U':
        penUp();
        break;
      case 'D':
```

```
    penDown();
    break;
  case 'G':
    buffer[0] = line[ currentIndex++ ];
    //      buffer[1] = line[ currentIndex++ ];
    //      buffer[2] = '\0';
    buffer[1] = '\0';

    switch ( atoi( buffer ) ) {
      case 0:
      case 1:
        // /!\ Dirty - Suppose that X is before Y
        char* indexX = strchr( line + currentIndex, 'X' );
        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++ ];
    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:
        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 );
      }
    }
  }
}

void drawLine(float x1, float y1) {

  if (verbose) {
    Serial.print("fx1, fy1: ");
    Serial.print(x1);
    Serial.print(",");
    Serial.print(y1);
    Serial.println("");
  }

  y1 = y1 * -1;


  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("");
}
```

```
x1 = (int)(x1 * StepsPerMillimeterX / 2);
y1 = (int)(y1 * StepsPerMillimeterY / 2);
float x0 = Xpos;
float y0 = Ypos;



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) {
    motorx.onestep(sx, MICROSTEP);
    over += dy;
    if (over >= dx) {
      over -= dx;
      motory.onestep(sy, MICROSTEP);
    }
    delay(StepDelay);
  }
} else {
  for (i = 0; i < dy; ++i) {
    motory.onestep(sy, MICROSTEP);
    over += dx;
```

```
    if (over >= dy) {
      over -= dy;
      motorx.onestep(sx, MICROSTEP);
    }
    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(LineDelay);
Xpos = x1;
Ypos = y1;
```

```
}


void penUp() {
  servo.write(penup);
  delay(50);
  Zpos = Zmax;
}

void penDown() {
  servo.write(pendown);
  delay(50);
  Zpos = Zmin;
}
```

Java code used in Processing IDE.

```
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;
import processing.serial.*;

Serial port = null;

String portname = null;
boolean streaming = false;
float speed = 0.001;
String[] gcode;
int i = 0;

void openSerialPort(){
  if (portname == null) return;
  if (port != null) port.stop();

  port = new Serial(this, portname, 9600);

  port.bufferUntil('\n');
}

void selectSerialPort(){
  String result = (String) JOptionPane.showInputDialog(frame,
    "Select the COM port on the Arduino board",
    "Select serial port",
```

```
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup(){
  size(500, 120);
  openSerialPort();
}

void draw(){
  background(0);
  fill(255);
  int y = 24, dy = 12;
  text("Run your CNC machine..", 12, y); y += dy;
  text("p: select serial port", 12, y); y += dy;
  text("g: select G-code file", 12, y); y += dy;
  text("x: stop your CNC machine", 12, y); y += dy;
  y = height - dy;
  text("Powered by SriTu Hobby", 12, y); y -= dy;
  text("current serial port: " + portname, 12, y); y -= dy;
}
```

```
void keyPressed(){

  if (!streaming) {
    if (key == 'p') selectSerialPort();
  }

  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) {
      streaming = false;
      return;
    }

    if (gcode[i].trim().length() == 0) i++;
    else break;
  }

  println(gcode[i]);
  port.write(gcode[i] + '\n');
  i++;
}

void serialEvent(Serial p){
  String s = p.readStringUntil('\n');
  println(s.trim());

  if (s.trim().startsWith("ok")) stream();
  if (s.trim().startsWith("error")) stream(); // XXX: really?
}
```

The End.