Hochschule Schmalkalden

Mechatronics & Robotics Engineering

Pen plotter model

Submitted by:

STUDENT NAME	MATRICULATION NUMBER
Arjun Dineshbhai Savaliya	315446
Jay Manishkumar Gajjar	315621
Kashyap Mahendrabhai Khunt	315154

In

Workshop of Mechatronics II.

for an effective completion

towards a Master of Engineering in Mechatronics and Robotics

Supervisor: Prof. Dr. Stefan Roth

(Department of Mechanical Engineering)

Supervisor: Prof. Dr. Ing. Silvio Bachmann

(Department of Electrical Engineering)

DECLARATION

The report for the semester project "Pen plotter model" that is being presented, in partial fulfillment of the requirements for an M.Eng in Mechatronics & Robotics is officially declared. Nothing in this project report has been taken verbatim from other sources or directly lifted from reports written by other students without proper citation. Under the guidance of Professors Dr. Stefan Roth and Ing. Silvio Bachmann, the study was conducted.

Student name	Signature
Arjun Dineshbhai Savaliya	315446
Jay Manishkumar Gajjar	315621
Kashyap Mahendrabhai Khunt	315154

OVERVIEW

A pen plotter is a precision graphics output device that creates pictures on paper with a motorised plotter. Typically, a control system, motors for positioning, and a moving pen is included in its design. Software instructions control the pen plotter's operation, enabling it to accurately replicate digital patterns on a physical medium. Pen plotters are useful because of their accuracy and adaptability, and they are frequently employed in industries including engineering, architecture, and the arts.

We developed the product using approaches we were previously familiar with this semester, starting with the V-model notion. A graphical depiction of a systems development lifecycle is called the V-model. Rigid project management and development lifecycle models are created with it. [1] For the mechanical and coding portions of our project, we also employed a variety of software tools, such as Solid Works, Circuito.io, and the Arduino IDE. The assignment description and lecture notes contained the majority of the information and specifications

They do have some drawbacks, though, such a slower pace and fewer color possibilities. Pen plotters are still used in situations where their special qualities are appreciated, even if they are competing with alternative printing technologies. Pen plotters of today are very adaptive in the digital era, as seen by their integration with sophisticated design software.

We encountered difficulties and made some design modifications while working, which provided us with practical prototype development expertise and a grasp of the trouble spots for next iterations of the pen plotter design.

During the work, we faced challenges and made some changes in the design, which gave us the real experience of prototyping development and understanding of problematic points for future versions of the design of the pen plotter.

TASK SUMMARY

The following salient features of a pen plotter model's task summary were included:

Goal: Create a pen plotter model to produce accurate images on paper.

Structure and Elements:

Build a physical framework that includes necessary parts such as paper feed mechanisms, motors, and pens.

Examine several pen plotter kinds (such as drum and flatbed) and choose the best design.

Usability:

Describe the whole operating process, including data intake and output.

Put software control systems in place for precise and effective plotting.

Systems of Control:

Provide controls that let users provide plotting directions

Verify that it works with a range of software interfaces and communication channels.

Uses:

Determine and rank the pen plotter's target industries and applications.

Provide characteristics that address particular requirements in engineering, architecture, art, etc.

Benefits and Drawbacks:

Add elements that will increase accuracy and adaptability

Take steps to improve performance by addressing constraints like speed.

In contrast to Other Technologies:

Analyze and compare with alternative printing and plotting technologies.

Emphasize the pen plotter's special qualities and applications.

Patterns and Advancements:

Observe the most recent developments in pen plotter technology.

Put into practice elements that work with CAD tools and contemporary design software.

Testing and Approval:
To guarantee the pen plotter's precision and dependability, implement a thorough testing procedure.
Verify how well it performs in actual situations and fix any problems found.

User Interface and Documentation:
Provide users with thorough documentation.
Provide an intuitive user interface to enable smooth communication with the pen plotter.

Future Improvements:
Describe prospective next enhancements and upgrades.
Think about your capacity to scale and adjust to new technologies.

This work brief offers a development plan for a pen plotter model, encompassing a range of topics from applications and future upgrades to design and functionality.

TABLE OF CONTENT

1 INTRODUCTION

Within the field of accurate graphic replication, the pen plotter is a reliable representation of painstaking workmanship and digital creativity. The pen plotter is a technological marvel that goes beyond conventional printing techniques. It uses a mechanical arm to precisely and painstakingly create complex drawings on paper. This combination of digital and analog skills has found use in a variety of domains, including engineering, architecture, and the arts.

As we dive into the world of pen plotters, we take you on a tour through the development of this technology, examining its basic architecture, intricate workings, and the numerous uses that have elevated it to a key position in both the technical and artistic sectors.

We include the V-Model, a methodical technique that mimics the complexities of the pen plotter's own design, into the story as we begin our investigation into pen plotters. The V-Model develops symmetrically, connecting each level of creation to its matching testing phase, much like the plotting process, which begins with a digital design and transforms it into a concrete masterpiece.

The requirements analysis and system design phases are represented by the left arm of the "V" in the pen plotter representation. Here, we explore the basic architecture of the pen plotter, including its physical layout, key parts, and the complex interaction between the software and mechanical components.

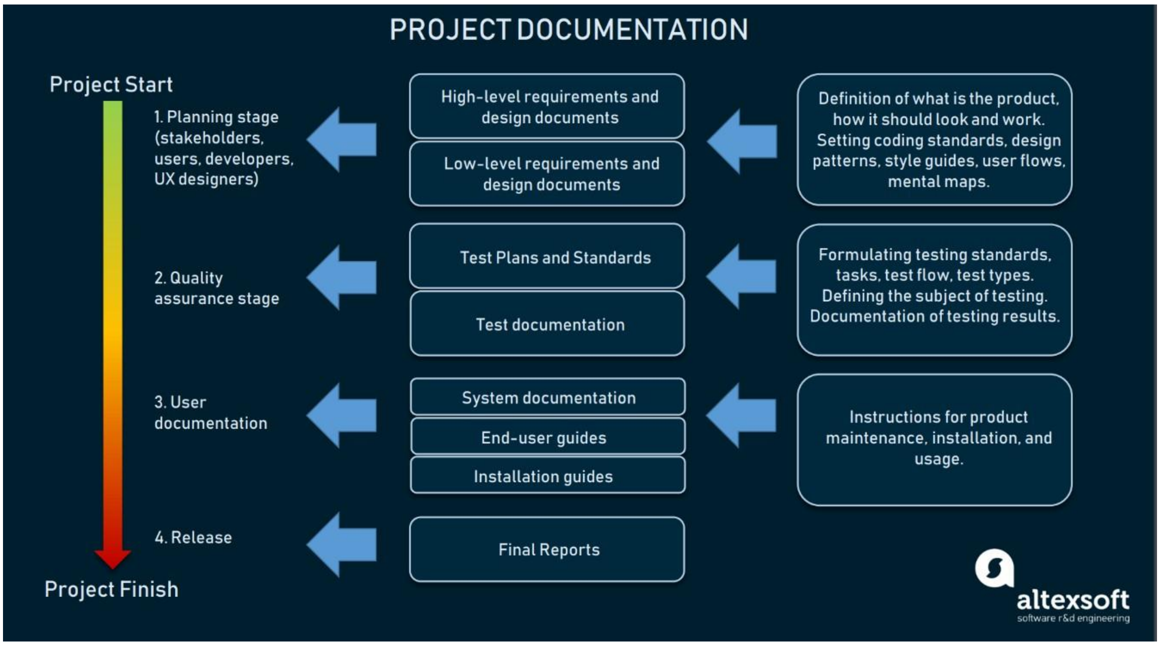
The coding and implementation step begins as we move down the "V," simulating the pen plotter's arm's downward trajectory as it transfers digital designs to paper. The control mechanisms are specified here, guaranteeing smooth communication between the pen plotter and the user. We make comparisons between the painstaking programming that controls the plotter's exact motions and the coding process.

Join us as we explore the relationship between pen plotters and the V-Model in this symbiotic inquiry, learning about the similarities between the disciplined approach to development and the dexterity of pen-and-paper accuracy in the field of graphic reproduction.

2 PLANNING OF THE PROJECT

Creating a work breakdown structure, evaluating requirements, identifying stakeholders, establishing objectives, and assessing risks are all part of project planning for a pen plotter model. Dependencies are used to arrange tasks, and a timetable with milestones is created. A communication strategy is developed together with the allocation of resources and funding. The strategy includes quality assurance, testing, and documentation with an emphasis on stakeholder reviews and change management. A post-implementation strategy takes into account potential future improvements, continuing support, and user training. This all-encompassing strategy guarantees a successful and methodical growth process.

2.1.1 Establishing the documentation for engineering needs



2.1.2 The scheduling and division of work.

Along with allocating tasks to the team members, we also made the project schedule.

Table 2.1: division of work

Job Title	Description of work	Name
Mechanical part	Design of the plotter we Use Solidworks to construct the design of the pen holder, plot area (plot board, link, base, and so on).	Arjun, Kashyap
Electrical setup and circuit	Connect all of the cables and electrical components to construct the electrical setup and circuit.	Jay, Arjun
Software Part	Development of the control pro-gramme and To write and test the code on C lan-guage.	Kashyap
Project management	To create the Engineering requirements, report mak-ing, planning, task defining.	Jay

Table 2.2: Timetable of the project

Stage	Description	Deadline	Status
Team formation	Assemble the team for the work-shop, considering members for me-chanical, hardware, and software aspects	03.05.2023	Completed
User Requirements	Establish user requirements for the workshop using Padlet	25.05.2023	Completed
Technical Specifications	Outline technical specifications on Padlet for the workshop	29.05.2023	Completed
Verification Plan	Develop a verification plan for the workshop	10.06.2023	Completed
Mechanical Parts develop-ment	Conceptualize and design mechani-cal parts, send final design for print-ing.	Start: 16.06.2023/ End: 30.06.2023 (2 weeks)	Completed
Hardware development	Develop hardware concept, create a working model	Start: 01.07.2023/ End: 30.07.2023 (4 weeks)	Completed
Software development	Code the logic of the plotter, devise operational algorithms based on	Start: 3.08.2023/ End: 23.08.2023(3 weeks)	Completed

	user requirements and technical specifications		
Plotter Assembly	Assemble all components (mechanical, hardware, software) to create a functional plotter prototype	Start: 02.09.2023/22.09.2023 (3 week)	Completed
Verification and Validation	Execute plotter tests according to the verification plan	Start: 25.09.2023/14.10.2023 (3 week)	Completed
Project Dossier	Compile the project dossier for the workshop	Start:20.10.2023/End:26.11.2023	Completed
Oral presentation	Prepare and deliver a presentation and report for the workshop	12.12.2023	Completed

3 V-MODEL APPROACH

We developed our project using the V-model technique of system lifecycle development. Software development methodologies such as the V-Model show the link between development phases and matching testing stages as a "V." Early testing is encouraged throughout the development process to make sure that errors are found and fixed at every turn. While validation activities make sure the entire system complies with requirements, verification activities verify the results of development phases. The concept facilitates early fault discovery by offering a clear link between design and testing. However, once the process starts, it could have trouble adjusting to changes.

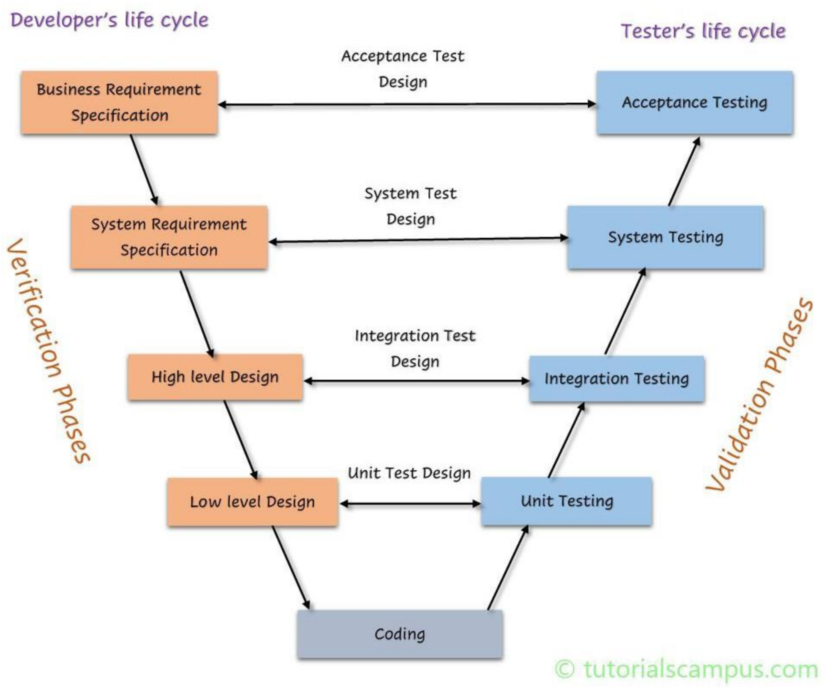


Fig. 3.1: V-model reference

4 DESIGN PHASE

4.1 Design of mechanical part of the project

Throughout the pen plotter model's design process, we concentrated on developing a thorough blueprint that describes the device's functional and structural features. The following are some essential components:

System Architecture:

Describe the general layout of the pen plotter, including the parts that make it up and how they work together.

Indicate the connections between the hardware, software, and mechanical components.

Mechanical Design:

Describe the general layout of the pen plotter, including the parts that make it up and how they work together.

Indicate the connections between the hardware, software, and mechanical components.

Hardware Design:

Describe the various parts of the hardware, such as the actuators, sensors, and electronic control units.

Indicate the interfaces and connections between the various hardware components.

Software Architecture:

Describe the software architecture, including the logic for the movement of the pen and the control algorithms.

Indicate how drawing commands should be interpreted and carried out by the software.

User Interface Design:

Create the UI that will communicate with the pen plotter.

Think about features that are easy for users to utilize, such feedback systems and input methods.

Communication Protocols:

Indicate the method by which the pen plotter will speak with other hardware or software

If applicable, define the communication protocols (e.g., USB, serial).

Safety Considerations:
If the pen plotter has moving parts, pay extra attention to safety features and concerns in the design.
Describe the safety precautions and emergency shutdown protocols.

Scalability and Modularity:
Think about how scalable the design is in case it has to be improved upon or modified in the future.
The plotter was designed with flexibility in mind, making it simple to modify or replace individual parts.

Power Requirements:
Specify how much electricity the pen plotter needs.
Indicate the current, voltage, and any power management functions.

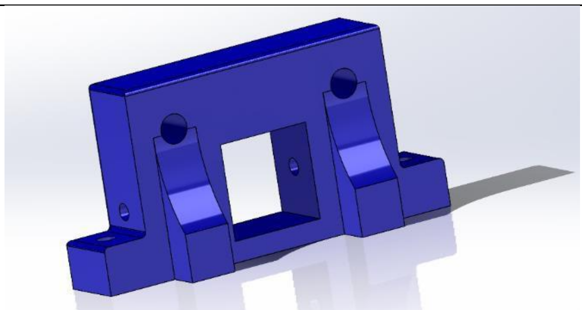
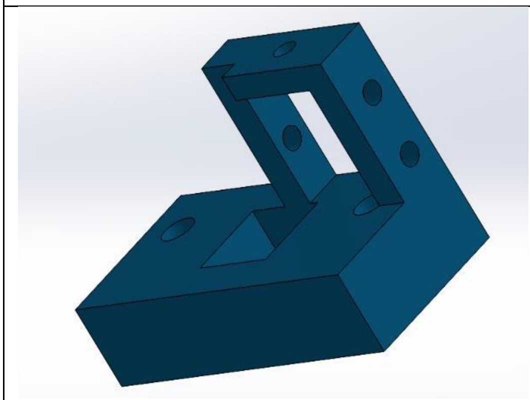
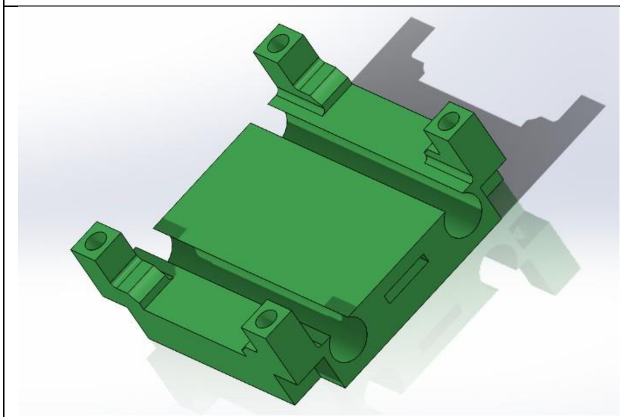
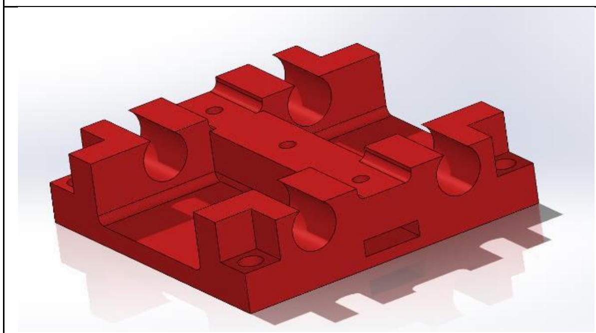
Cost Considerations:
Give an approximate idea of the design's expected costs, including those for production and materials.

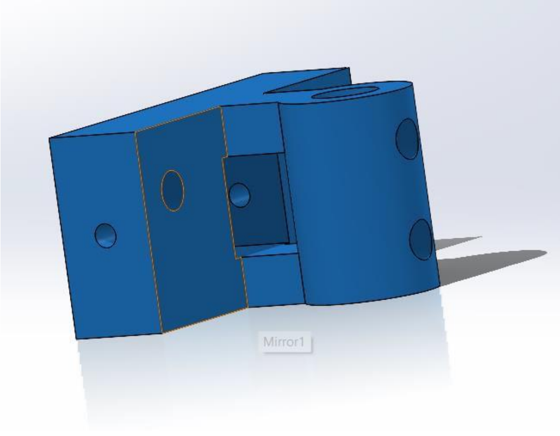
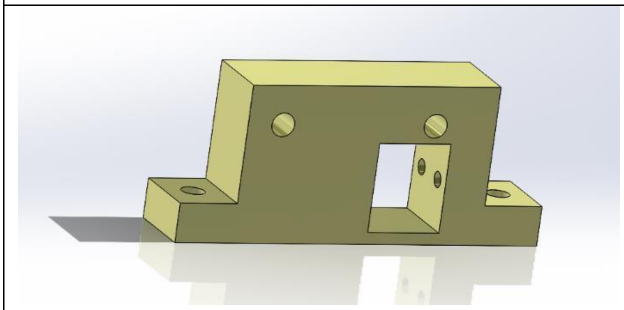
Feasibility Analysis.
To make that the suggested design is both technically feasible and compliant with project limitations, perform a feasibility study.

Design Review:
Arrange for a procedure of design review with pertinent parties involved to get input and make required modifications.

Prototyping Plan:
Plan how to test and validate the design with prototypes before going into full-scale manufacturing.
Make sure your design documentation is clear and comprehensive so that it can serve as a strong basis for the pen plotter model's further development stages.

Table 4.1: Table of main components of the plotter frame

View of the component	Name of the component	Functional description of component
	Fix axis idle end	accountable for traveling in a Y direction. The center of the plate is home to the pulley stand. includes a screw hole for adjusting and fixing the shafts.
	Moving axis motor end	Responsible for moving axis and to hold the motor for moving axis.
	Centre top part	It holds the bearing and shaft assembly for moving axis. Also for pivoting belt drive for moving axis.
	Centre bottom part	It holds the bearing and shaft assembly for fixed axis. Also for pivoting belt drive for fixed axis.

	Pen holder	carries a pen with it. The design allows the use of many pen variations by adjusting the ad-justable screw. .Also supports the belt drive with use of idle pulley.
	Fix axis motor end	For fix axis it holds the motor and supports the shafts.

4.1.1 How the belt mechanism functions

We chose a specific type of belt, that is a plus belt, for a pen plotter model. we decided to go with this model after considering various factors related to the application requirements, precision, durability, and overall performance. Here we will show why one might choose a plus belt for a pen plotter:

Why a Pen Plotter Model Should Use a Plus Belt:

Plotting with accuracy:
Because plus belts have a toothed construction, precise placement is made possible, enabling the pen plotter to accurately replicate fine features. In applications like architectural blue-prints or intricate artwork, this is very crucial.

Strength and Lifespan:
Plus belts' long lifespan is a result of its premium construction and materials. As a result, the pen plotter will have a longer operating lifespan, require less maintenance, and be more re-liable overall.

Decreased Uncertainty:

Slippage is reduced by the plus belt's teeth engaging positively with the pulley. This is essential to make sure the pen stays on the planned course, especially when making quick or complex motions.

Maintaining Performance Consistency:

Minimal flexibility and minimal blowback are factors.
Without deviations from belt deformations or system play, the pen plotter can accurately reproduce patterns.

High-speed operation compatibility:

Furthermore, belts can be used in high-speed applications. A plus belt may manage the demands without sacrificing precision if the pen plotter needs quick movements for efficiency.

Considerations for Maintenance:

Plus belts' long-lasting qualities frequently translate into less maintenance needs. For consumers looking for a pen plotter model that requires less maintenance and downtime, this might be helpful.

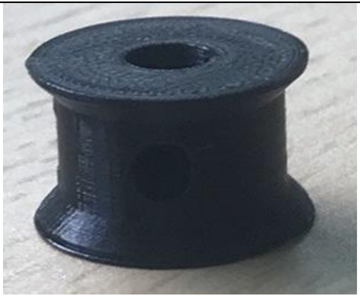
In conclusion:

The particular needs of the application determine which additional belt is best for a given pen plotter model. Key factors include things like accuracy, robustness, reduced backlash, and suitability for high-speed operation. Plus belts are a good option for pen plotters where precision and dependability are critical since they have these benefits.

4.2 Supplies and given data

A table with the supplied components and details is provided below.

Table 4.2: Given components

View of component	Specification and name of component
	Belt Pulley Length : 10.5 mm Inner Ø: 14 mm Outer Ø: 17 mm Shaft Ø: 5.2 mm Fix Ø : 3.5 mm
	Gear Pulley Length : 15.5 mm Gear Ø : 10 mm Big Ø : 14 mm Shaft Ø: 3.2 mm Fix Ø : 3.5 mm
	Ground plate 305 x 305 x 12 mm
	Timing belt Width: 6 mm Length: as per requirement

	Guiding Shaft Diameter : 6 mm Length : 300 mm
	Linear ball bearing inside diameter 6 mm outer diameter 12 mm length 19 mm
	Storage Box (to store the model) Width: 56 cm3 Length: 39 cm3 Height : 28 cm3 Capacity: 45 L
	Screws, Bolts and Nuts Standard size M4 x (8, 10, 12, 16, 20, 25) M5 x (10, 12, 16, 20, 25, 30) M6 x (16, 20, 25, 30, 35)

Figures following show how the plotter's main body is laid out.

Fig. 4.7: Motor holder

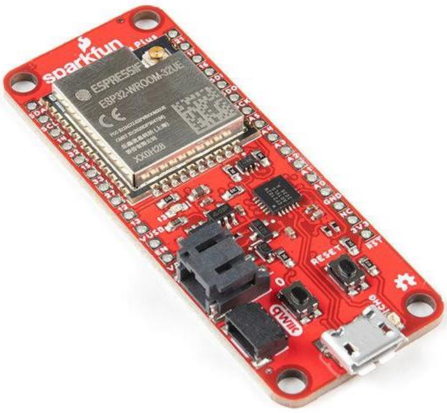
5 DEVELOPMENT OF ELECTRICAL PART

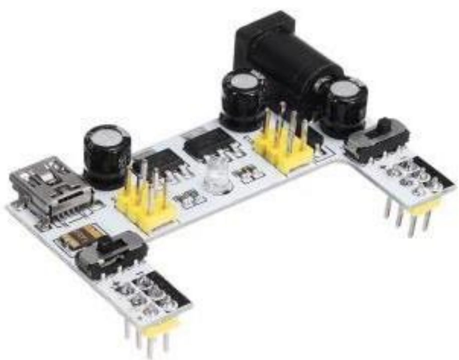
5.1 Development of the hardware part of the project

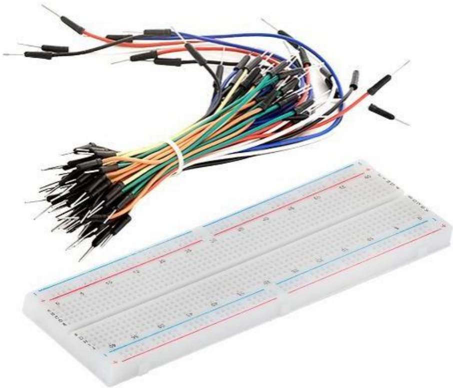
The actual components that make up the plotter are designed and made as part of the pen plotter project's hardware development. This include integrating necessary hardware components, choosing suitable materials, and outlining the mechanical structure. The emphasis is on dependability, accuracy, and the smooth operation of parts including pulleys, motors, and pen holders. In order to guarantee the pen plotter's precise and dependable performance in converting digital designs into physical drawings, the hardware development stage is essential.

All of the necessary parts are available for constructing the circuit. A table with information and component specifications may be found below.

Table 5.1: Provided electrical components

View of component	**Information and specification of component**
	**ESP 32 Thing Plus** Used for control the process. – Dimensions: 2.30 x 0.90 Inches. – Xtensa® dual-core 32-bit LX6 microprocessor. – Up to 240MHz clock frequency – 16MB of flash storage – 520kB internal SRAM – Integrated 802.11 BGN WiFi transceiver – Integrated dual-mode Bluetooth (classic and BLE) – 2.3 to 3.6V operating range – 21 GPIO – 8-electrode capacitive touch support – Hardware accelerated encryption (AES, SHA2, ECC, RSA-4096) – 2.5 µA deep sleep current [8]

	**Qwiic Motor Driver** Operates small robot drive motors without a heat sink. – 3.3V default VCC and logic – 127 levels of DC drive strength. – Controllable by I2C or TTL UART signals – Direction inversion on a per motor basis [8]
	**Geared DC Motor** Generates the movements – Dimensions: L: 35 mm B: 12 mm W: 10 mm – Voltage: 6V – Max. current: 230mA, – RPM: idle: 77 rpm, norm: 46.7 U/min – Torque: norm: 98 Nm Stall: >245mNm
	**XD 42 Power Supply Module** – Recessed design, save more space for – breadboard – 2 channel design, can choose 5 V or 3.3 V – Can be powered via USB power and DC – power.[9]

	**Limit Switch** A limit switch is an electromechanical device operated by a physical force applied to it by an object. Limit switches are used to detect the presence or absence of an object. [10]
	**Push Button** A push-button (also spelled pushbutton) or simply button is a simple switch mechanism to control some aspect of a machine or a process.[11]
	**Additional materials** Cables Breadboard Pin header Soldering material

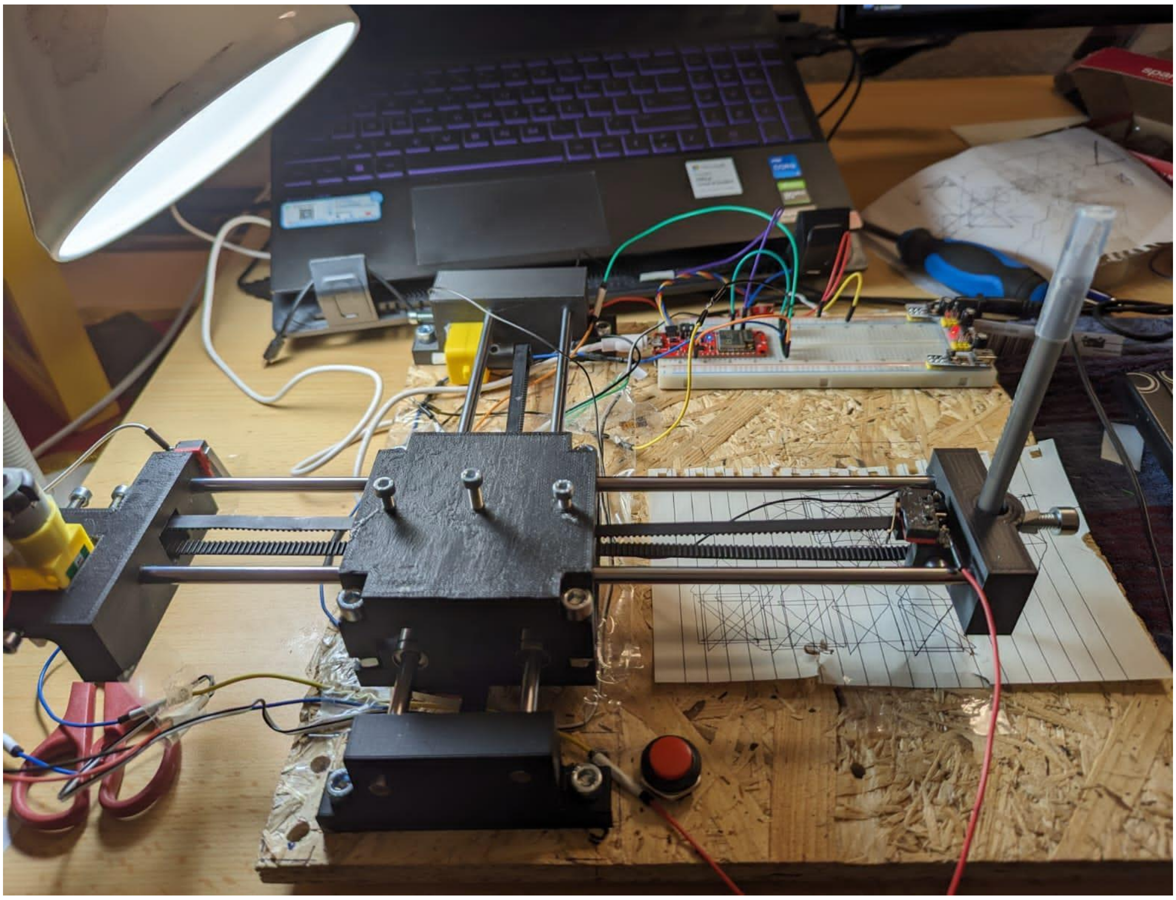


Fig. 5.2: Final design layout (Laptop connection; ESP32 microcontroller; Qwiic Motor Driver; breadboard; push button; XD 42 Power Supply Module; grid connection)

5.2 Control architecture development

The code was written using the Arduino IDE. The code was created with the needs of the user in mind. There are comments for every step of the code. The Arduino IDE is a freely available application with a sizable independent programming community.

```
#include <Arduino.h>
#include <stdint.h>
#include "SCMD.h"
#include "SCMD_config.h"
#include "Wire.h"
#include <BluetoothSerial.h>

#define LED_PIN 13
#define LIM_F_M 14
#define LIM_F_I 15

int limf;
int limm;
```

```
SCMD myMotorDriver;
BluetoothSerial SerialBT;

void toggleLed() {
    digitalWrite(LED_PIN, HIGH);
    delay(50);
    digitalWrite(LED_PIN, LOW);
    delay(50);
}

void driveMotors(int drive1, int drive2, int speed, int duration) {
    myMotorDriver.setDrive(0, drive1, speed);
    myMotorDriver.setDrive(1, drive2, speed);
    delay(duration);
    myMotorDriver.setDrive(0, 0, 0);
    myMotorDriver.setDrive(1, 0, 0);
}

void setup() {
    Serial.begin(9600);
    SerialBT.begin("ESP32_BT");
    Serial.println("Bluetooth activated");
    pinMode(LED_PIN, OUTPUT);

    Serial.println("Starting sketch.");

    myMotorDriver.settings.commInterface = I2C_MODE;
    myMotorDriver.settings.I2CAddress = 0x5D;
    myMotorDriver.settings.chipSelectPin = 10;

    while (myMotorDriver.begin() != 0xA9) {
        Serial.println("ID mismatch, trying again");
        delay(500);
    }
    Serial.println("ID matches 0xA9");

    Serial.print("Waiting for enumeration...");
    while (!myMotorDriver.ready());
    Serial.println("Done.");
    Serial.println();

    while (myMotorDriver.busy());
    myMotorDriver.enable();
}

void loop() {
    if (SerialBT.available()) {
        String command = SerialBT.readString();
        int CMD = command.toInt();
        Serial.println(CMD);

        limf = digitalRead(LIM_F_I);
        limm = digitalRead(LIM_F_M);
```

```
        Serial.println(limf);
        Serial.println(limm);

        if (CMD == 1) {
            if (limf == 0) {
                myMotorDriver.reset();
                toggleLed();
                delay(500);
                myMotorDriver.setDrive(0, 1, 150);
                delay(500);
            } else {
                myMotorDriver.reset();
                myMotorDriver.setDrive(0, 0, 160);
                Serial.println("Stop!");
                delay(500);
                myMotorDriver.setDrive(0, 0, 0);
                myMotorDriver.reset();
            }

            if (limm == 0) {
                myMotorDriver.reset();
                toggleLed();
                delay(1200);
                myMotorDriver.setDrive(1, 1, 150);
            } else {
                myMotorDriver.reset();
                myMotorDriver.setDrive(1, 0, 160);
                Serial.println("Stop!");
                delay(500);
                myMotorDriver.setDrive(0, 0, 0);
                myMotorDriver.setDrive(1, 0, 0);
                delay(1000);
                myMotorDriver.reset();
            }
        } else if (CMD == 2) {
            if (limf == 0 && limm == 0) {
                myMotorDriver.reset();
                driveMotors(1, 1, 0, 500);
                // ... other drive sequences ...
            } else {
                myMotorDriver.reset();
                toggleLed();
                // ... alternative behavior ...
            }
        } else if (CMD == 3) {
            myMotorDriver.setDrive(0, 0, 150);
            delay(400);
            myMotorDriver.setDrive(0, 0, 0);
            myMotorDriver.reset();
        } else if (CMD == 4) {
            myMotorDriver.setDrive(0, 1, 150);
            delay(400);
            myMotorDriver.setDrive(0, 0, 0);
            myMotorDriver.reset();
```

```
        } else if (CMD == 5) {
            myMotorDriver.setDrive(1, 0, 150);
            delay(400);
            myMotorDriver.setDrive(1, 0, 0);
            myMotorDriver.reset();
        } else if (CMD == 6) {
            myMotorDriver.setDrive(1, 1, 150);
            delay(400);
            myMotorDriver.setDrive(1, 0, 0);
            myMotorDriver.reset();
        }
    }
}
```

6 SUMMARY OF PROTOTYPE.

6.1 Working principle of prototype

Conceiving the function and features of a pen plotter model, developing the hardware and software, and prototyping for testing and improvement are all necessary steps in the creation process. After the finished product is assembled, it is put through rigorous testing to ensure accuracy and dependability. The development cycle is finished with documentation, project management, deployment, and continuing maintenance. The main components of success include meticulous testing, respect to design criteria, and effective teamwork.

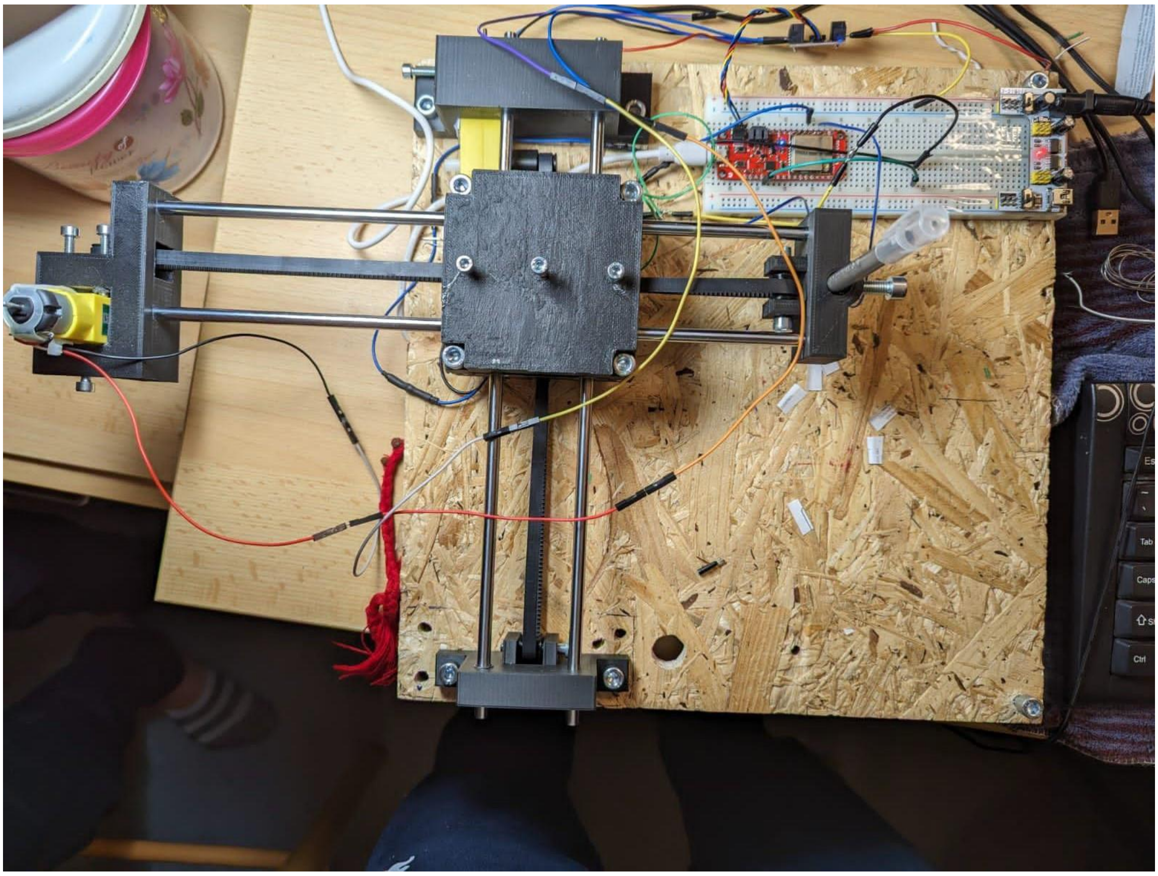


Fig. 6.1: Final mechanical and electrical layout

6.2 Problems and their solutions

While making this pen plotter we face couple of problems and we made solutions accordingly. Few of them are mentioned below:

Problems:

1. Problems with Precision:

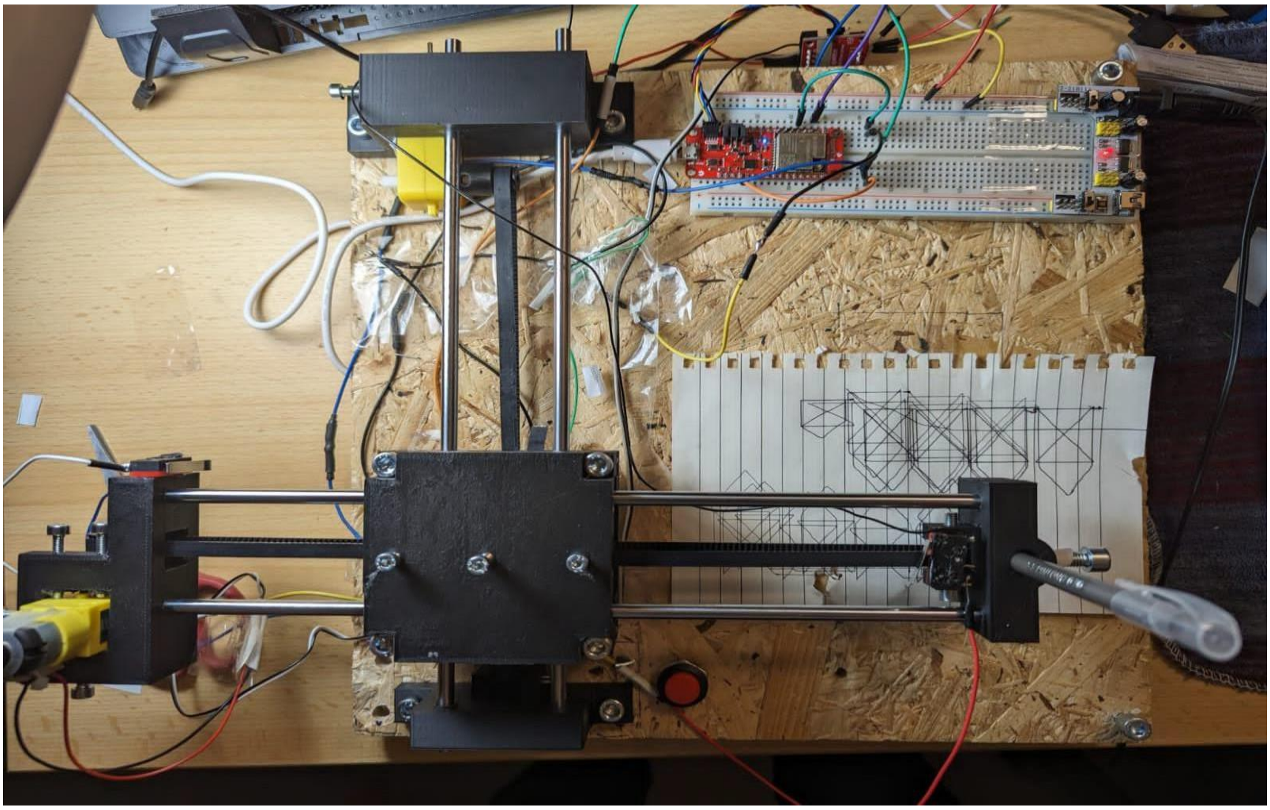
Problem: Wear and tear, misalignments, and mechanical tolerances made it difficult to achieve constant and exact motions of the pen holder.

Solution: To fix this discrepancies, we did regular calibration. The accuracy did improved by using feedback systems such as encoders and high-quality, precision components.

2. Integration of Software:

Problem: Software defects, unexpected behaviors, and compatibility problems arose when integrating software with mechanical components.

Solution: During software development, we put in place a comprehensive testing procedure aids in the early detection and resolution of integration problems. We guaranteed smooth compatibility, the software and hardware teams must work together continuously.



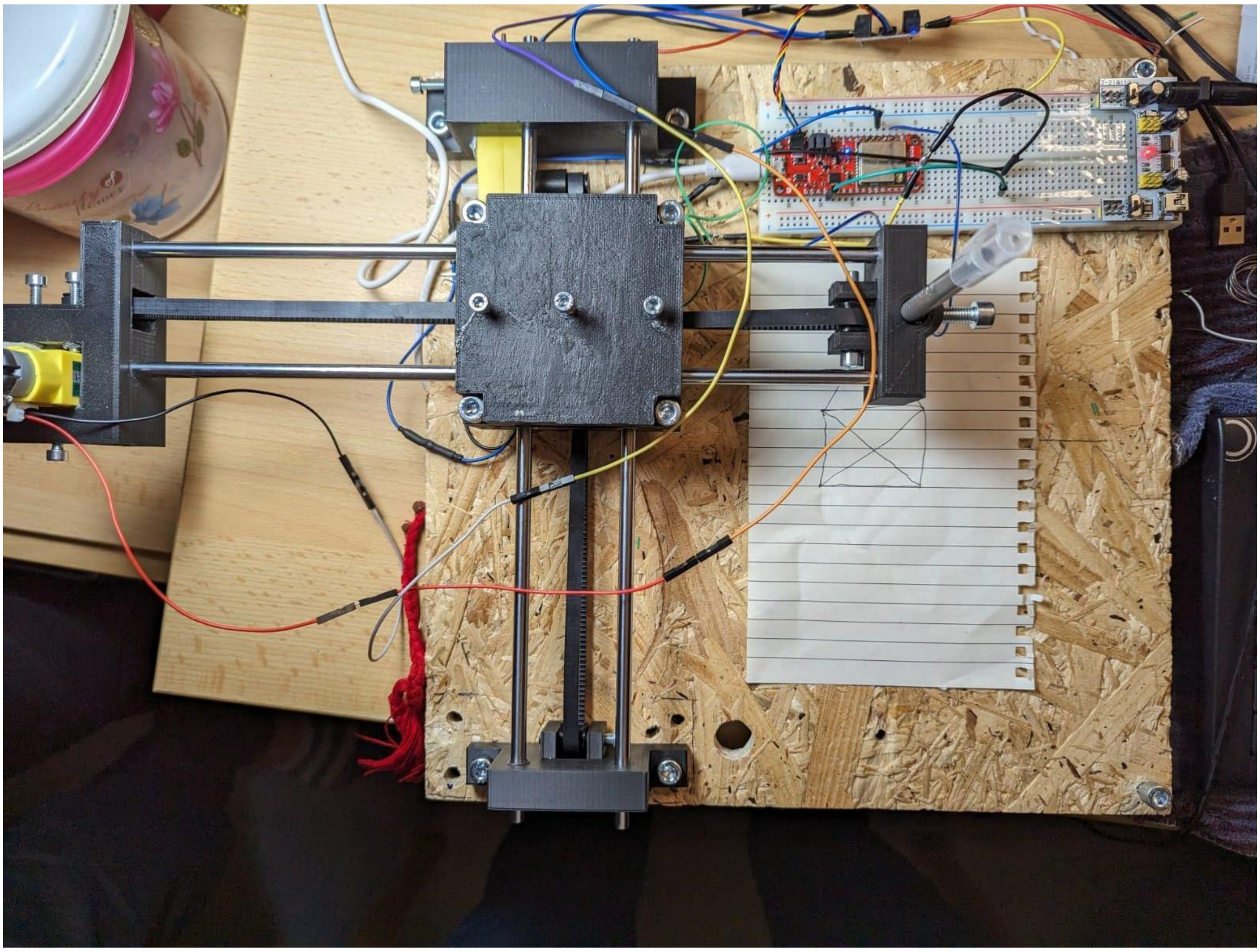
These difficulties we successfully handled by tackling accuracy problems with calibration and component selection, as well as by putting strong testing and cooperation for software integration into practice.

6.3 Potential ways to use

The integration of AI and AR, support for 3D printing, improved wireless connectivity, cloud-based collaboration, and increased material compatibility are some of the potential future applications for pen plotter models. Possible improvements include haptic feedback, energy-efficient designs, and enhanced user interfaces. Pen plotters are anticipated to become more adaptable and integrated with contemporary design workflows through integration with IoT for data transmission and real-time changes.

6.4 Conclusion

In conclusion, creating a pen plotter model with fixed proportions based on Nickolaus' home offers a fascinating chance to combine design and technology. Pen plotters' focus on accuracy fits very nicely with the fine detail needed to create accurate architectural depictions. The pen plotter can accurately replicate complex designs by following the given measurements, making it an invaluable tool for illustrating and presenting architectural concepts. Plotting findings will be precise and dependable only if careful consideration is given to mechanical accuracy, material selection, and software integration. This pen plotter's promise is found not only in its technical skills but also in its capacity to improve architectural concepts' visualization and communication within Nickolaus' home.



REFERENCES

1. URL: https://medium.com/nightingale/pen-plotters-are-the-perfect-tool-for-dat -sto-rytelling-b05c71ceadd5

2. URL:https://www.researchgate.net/publication/344547887_Review_on_Build-ing_A_Cost_Efficient_Pen_Plotter

3. URL: https://hackaday.com/2023/10/04/lessons-learned-while-building-a-diy -pen-plotter/

4. URL: https://szymonkaliski.com/writing/2023-10-02-building-a-diy-pen-plotter/

5. URL: https://learn.sparkfun.com/tutorials/esp32-thing-plus-hookup-guide/all

6. URL: https://howtomechatronics.com/projects/diy-pen-plotter-with-automatic -tool-changer-cnc-drawing-machine/

7. URL: https://hackaday.com/2021/08/06/pen-plotter-is-about-as-simple-as-it-ca -get/

8. URLhttps://davidbliss.com/2021/09/13/penelope/

9. URL: https://andrewsleigh.github.io/plotter/