THE UNIVERSITY OF DA NANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

CAPSTONE PROJECT

MAJOR: MECHATRONIC ENGINEERING

TOPIC: RESEARCH AND INTEGRATION OF AUTOMATIC COLLATING, PERFORATING AND CREASING FUNCTIONS INTO RICOH PRINTER SERIES.

Instructor : **PhD. Le Hoai Nam**

Co-Supervisor : **Mr. Nguyen Ha Thi Giang**

Reviewer : **PhD. Vo Nhu Thanh**

Student : **Nguyen Duy Dang**

Dang Ngoc Phuoc

Bui Duong Quoc

Student code : **101200345**

101200366

101200409

Class : **20CDTCLC2**

20CDTCLC2

20CDTCLC3

Da Nang, 6/2025

SUMMARY

Project name: Research and integration of automatic collating, perforating and creasing functions into Ricoh printer series.

Student	Student ID	Class
Nguyen Duy Dang	101200345	20CDTCLC2
Dang Ngoc Phuoc	101200366	20CDTCLC2
Bui Duong Quoc	101200409	20CDTCLC3

The project titled "*Research and integration of automatic collating, perforating and creasing functions into Ricoh printer series*" was carried out with the aim of utilizing second-hand printers to improve and integrate essential post-print processing features. Reusing and upgrading old printers not only helps minimize investment costs but also has high practical relevance, aligning with modern trends of cost-effective and sustainable production.

The objective of the project is to design a fully integrated system comprising three functions: perforating, perforating and creasing. The system should operate synchronously and automatically, and be installable on old Ricoh printers models.

The scope of research focuses on surveying the mechanical structure of used Ricoh printers, designing an integrated mechanical system, utilizing available motors and sensors, designing control circuits, and programming the automation of system operations.

Research methods include: studying relevant theoretical knowledge, conducting field surveys on equipment, performing experiments, and evaluating the actual system's performance.

The structure of the thesis includes four chapters:

Chapter 1: General introduction to the topic.

Chapter 2: Mechanical system Design.

Chapter 3: Design the electrical system.

Chapter 4: Building the control system.

THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

THE SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom – Happiness

GRADUATION PROJECT MISSION

No	Name	Student ID	Class	Major
1	Nguyen Duy Dang	101200345	20CDTCLC2	Mechatronic Engineering
2	Dang Ngoc Phuoc	101200366	20CDTCLC2	Mechatronic Engineering
3	Bui Duong Quoc	101200409	20CDTCLC3	Mechatronic Engineering

1. *Project name: Research and integration of automatic collating, perforating and creasing functions into Ricoh printer series.*

2. *Subjects covered by:* ☐ *Having signed an intellectual property agreement for the implementation results*

3. *Initial data and figures:*
   - Integration of 3 functions into Ricoh printers
   - Output frequency: 30 sheets per minute

4. *Supervisors's name: PhD. Le Hoai Nam*
5. *Project assignment date: 23/2/2025*
6. *Project completion date: 10/6/2025*

Da Nang, 10th June ,2025

Head of Mechatronic Division **Supervisor**

PREFACE

In today's era, the application of automation technology to save time and reduce costs has become an urgent need across many fields—especially in the printing industry. Currently, the market offers a variety of specialized machines that perform functions such as collating, perforating, and creasing. However, owning separate machines for each of these tasks is both inconvenient during operation and costly in terms of investment. On the other hand, machines that integrate multiple functions are still limited, particularly in the low-cost or repurposed equipment segment.

From that practical need, our team developed the project titled **"Research and integration of automatic collating, perforating and creasing functions into Ricoh printer series "** with the aim of utilizing old (or discarded) Ricoh printers by upgrading and integrating essential post-printing functions. This approach not only helps reduce investment in new equipment but also promotes the effective reuse of existing machines.

This thesis focuses on redesigning the mechanical structure, integrating a transmission system, selecting and applying appropriate motors and sensors from the printer, and developing a new control system to operate the collating, perforating, and creasing functions automatically and in synchronization.

We would like to express our sincere thanks to the **Faculty of Mechanical Engineering**, especially our advisor **Dr. Le Hoai Nam**, for his dedicated guidance throughout the project. We also extend our gratitude to **Mr. Nguyen Ha Thi Giang** for his support in providing equipment and financial assistance, which enabled us to complete the project. Although we have made every effort, due to limited time and experience, there may still be shortcomings in our work. We look forward to receiving valuable feedback from our teachers to further improve the project and hope that this work can eventually be applied in practical settings.

Sincerely,

Nguyen Duy Dang

Dang Ngoc Phuoc

Bui Duong Quoc

ASSURE

We, the students undertaking the project titled *"Research and integration of automatic collating, perforating and creasing functions into Ricoh printer series"*, hereby declare that all contents of this graduation thesis are the result of our own learning, research, design, and implementation efforts.

All elements presented in this report—including technical drawings, electrical circuit diagrams, programming code, and engineering solutions—have been developed based on our professional knowledge, literature research, and practical experimentation. All referenced materials used throughout the project are properly cited in accordance with academic regulations.

We take full responsibility for the accuracy, transparency, and legality of all content presented in this thesis. In the event of any violations related to academic integrity, we commit to accepting all disciplinary actions in accordance with the policies of the The University of Da Nang - University of Science and Technology.

We fully understand that honesty and sincerity are the foundation of all research activities, and we are committed to strictly adhering to ethical standards throughout every aspect of this graduation project.

Students:

Nguyen Duy Dang

Dang Ngoc Phuoc

Bui Duong Quoc

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION.

1.1. General introduction of the topic

The automatic collating, perforating, and creasing machine is a device that integrates all three functions: collating, perforating, and creasing. Currently, most printers on the market are designed to handle only one or, at best, two of these functions.

These functions typically exist as separate capabilities in different machines, depending on their intended use. As such, users often have to combine multiple machines or implement custom improvements to meet diverse production demands.

This project aims to integrate all three functions into a single machine using an old Ricoh printer, in order to meet current market needs—particularly in areas where existing machines either fall short or are not readily accessible.

1.1.1. Functions and types of automatic perforating machines

- **Main functions:**

Create perforated lines on paper, cardboard, or printed materials to facilitate easy separation or tearing along the cut. This type of cut is commonly used in products such as tickets, discount coupons, receipts, labels, or warranty cards.

- **Outstanding advantages:**

Automation: The machine operates continuously and stably, minimizing manual intervention.

High precision: Produces uniform and sharp perforated lines, suitable for various material sizes and thicknesses.

Flexible customization: Allows adjustment of the length and spacing between perforations as required

- **Applications:**

Used in printing invoices, event tickets, gift vouchers, labels, or packaging that requires easy tearing.

- **Models available on the market:**
  - ❖ **Heidelberg (Germany)**

Heidelberg is a renowned brand known for its precise and durable die-cutting machines, suitable for applications that require high quality.

- o **Flagship perforating machine models:**
  - **Heidelberg Varimatrix 105CS:**

Figure 1.1 Heidelberg Varimatrix 105CS Perforating Machine [1]

- **Heidelberg Easymatrix 106:**

❖ **YAWA (China)**

YAWA is a prominent brand in Asia, offering die-cutting machines with reasonable prices and integrated modern features.

- **Outstanding perforating machine models:**
  - **YAWA TDS-750:**

Figure 1.2 YAWA TDS-750 Perforating Machine [2]

- **YAWA ML-930:**

❖ **Horizon (Japan)**

Horizon is a leading brand from Japan, specializing in providing modern printing and finishing equipment.

- **Outstanding perforating machine models:**
  - **Horizon CRF-362:**

Figure 1.3 Horizon CRF-362 perforating machine [3]

  - **Horizon RD-4055:**

❖ **Minh Duc Mechanical Company (Vietnam)**

Minh Duc is one of the domestic manufacturers of perforating machines, meeting the needs of small and medium-sized businesses in Vietnam.

- **Highlighted perforating machine model :**
  - **Minh Duc MD-500:**
  - **Minh Duc MD-750:**

❖ **Dong Nam Equipment Company (Vietnam)**

Dong Nam specializes in providing cost-effective and high-performance perforating machines for the domestic market.

- **Highlighted perforating machine model:**
  - **Dong Nam DN-360:**
  - **Dong Nam DN-720:**

❖ **Kama (Germany)**

Kama is a renowned brand known for its high-end die-cutting machines designed for the printing and packaging industry.

- **Highlighted perforating machine model:**
  - **Kama ProCut 76:**

Figure 1.4 Perforating machine Kama ProCut 76 [4]

- **Kama FlexFold 52:**

Table 1.1 Comparison of Machine Models Based on Application Needs.

Brand	Level of Automation	Key Applications	Advantages
Bobst	Automatic	Large-scale industrial production	High durability and precision
Heidelberg	Automatic	High-end packaging, large output	Advanced technology, reputable brand
YAWA	Semi-automatic/Automatic	Ticket, voucher, label printing	Affordable price, user-friendly
Horizon	Automatic	High-end post-press processing	High accuracy and flexibility
Minh Đuc	Manual/Semi-automatic	Small workshops, small-scale orders	Low cost, easy operation
Dong Nam	Manual/Semi-automatic	Small and medium-sized enterprises	Low investment cost

1.1.2. Functions and automatic creasing machine models

- **Main functions:**

- Create creasing lines (fold lines) on paper, cardboard, or other materials to allow easy folding without tearing or deforming the material
- Commonly used in the production of paper boxes, paper bags, greeting cards, envelopes, and products that require high precision and refinement

- **Outstanding advantages:**

- Automatic operation: Equipped with an automatic feeding and collecting system, helping to reduce manual labor.
- Versatile: Capable of creasing various types of materials, from thin paper to thick cardboard.
- High quality: Produces clean and precise crease lines without tearing the paper, ensuring both aesthetic and technical quality.
- Multi-function integration: Some modern machines can combine creasing and die cutting in a single process.

- **Applications:**

- Production of product packaging boxes, premium paper bags, invitation cards, menus, and other high-end packaging products.

- **Machine models available on the market:**

❖ **Creasing machines from Bobst (Switzerland)**

Bobst is one of the world's leading manufacturers of printing and packaging processing equipment, especially known for its high-end automatic creasing machines.

o **Bobst creasing machine**

- **Key features:**

- Flexible design allows creasing on various types of materials, from thin paper to thick cardboard.
- Easy adjustments for creasing depth and spacing between crease lines.
- Electronic control system with a high level of automation.

- **Applications:**

- Used in packaging, paper printing, carton box manufacturing, and paper bag production.

o **Bobst Expertfold 110 A2**

- **Key features:**

- Combines creasing and die-cutting technology helping to save time and effort

- Equipped with an advanced control system that fully automates the creasing process

- **Applications:**

- Production of packaging products such as carton boxes, paper boxes, invitation cards, and envelopes

❖ **Creasing machines by Heidelberg (Germany)**

Heidelberg is renowned for its offset printing machines but also provides creasing solutions for industrial printing businesses

- **Heidelberg CreaseMaster Pro**

- **Key features:**

- High-speed automatic creasing machines capable of working with various types of materials.
- Precise adjustment of creasing depth and position.

- **Applications:**

- Used in companies producing carton packaging, paper products, greeting cards, and other items requiring high precision

❖ **Creasing Machines from YAWA (China)**

YAWA là một hãng sản xuất máy in và gia công bao bì với các dòng máy có giá thành hợp lý và tính năng phù hợp với các doanh nghiệp vừa và nhỏ.

- **YAWA CREASEMASTER 1060**

- **Key features:**

- Integration of the scoring and creasing system, with adjustable scoring lines and dimensions according to requirements.
- User-friendly PLC control system, capable of scoring on various materials and different thicknesses.

- **Applications:**

- Used for packaging products such as cardboard boxes, greeting cards, menus, and products with complex folds.

❖ **Creasing Machines from Kama (Germany)**

Kama is a renowned brand in the packaging industry, offering professional scoring and creasing machines for high-end products.

- **Kama ProCut 76**

- **Key features:**

- The scoring machine is integrated with cutting and creasing technology, producing finished products with high precision.
- Capable of scoring on materials such as paper, cardboard, and plastic.
- Automatic control, allowing flexible customization of scoring parameters.

- **Applications:**

- Manufacturing cardboard boxes, invitation cards, high-quality packaging, and other packaging products.

❖ **Creasing Machines from Horizon (Japan)**

Horizon is a renowned brand in the printing industry, specializing in providing industrial solutions with high-quality scoring machines.

- o **Horizon CRF-362 Creasing Machine**

- **Key features:**

- The machine is capable of scoring on various materials, from paper to cardboard.
- Uses scoring technology that prevents paper tearing, ensuring material protection throughout the processing.
- Flexible adjustment of parameters to create precise folds.

- **Applications:**

- Used in the packaging industry, for greeting cards, discount vouchers, and products that require neat and precise folds.

1.1.3. Functions and models of 3-tray collating machines

The 3-tray collating machine is a specialized device in the printing industry, designed to divide paper into multiple parts and automatically arrange them in the desired order. This is a high-end machine, especially suitable for the production of invoices, receipts, payment vouchers, or documents that need to be collated into multiple copies in precise sequence.

❖ **Outstanding features**

- **Accurate collating**

- The machine can divide paper into multiple copies based on specified sizes and quantities.
- Cutting lines are precise and sharp without tearing or damaging the paper.

- **3 Trays operating in synchronization**

- Equipped with 3 independent paper input trays, allowing simultaneous handling of 3 different paper types or sizes.
- The trays are programmed to work in coordination, ensuring the final product is arranged in the correct sequence.

- **Automatic sorting**

- The machine automatically arranges the collated sheets in a preset order, e.g., copy 1, copy 2, copy 3
- Especially useful for multi-copy invoice production, helping to save time and ensure correct order

- **Easy customization**

- The intelligent control system allows easy configuration of sheet sizes, sorting order, and operating speed
- Supports various paper sizes and thicknesses

- **Applications of the 3-Tray collating and sorting machine this machine is widely used in areas such as:**

- Multi-copy invoice printing: Sales invoices, payment/receipt slips requiring 3 copies with a clear order (copy 1: customer, copy 2: accounting, copy 3: archive)
- Ticket production: Event tickets, movie tickets, or entry/exit passes that require clear ordering
- Warranty card processing: Multi-copy warranty or product inspection slips with different content on each copy

- **Models on the market**

❖ **3-Tray collating machine from mingwei (China)**

Mingwei is a well-known brand in the production of collating machines, including 3-tray models designed for printing and packaging companies

- **Mingwei MW-350**

- **Key features:**

- The machine can accurately collate sheets and arrange them in the desired sequence.
- Automatic feeding system with three trays for sorting and arranging sheets in order.
- High-speed operation suitable for large-volume print orders.

- **Application:**

- Ideal for products such as invoices, receipts/payments, event tickets, and discount vouchers.

❖ **3-Tray collating machine from wanjie (China).**

Wanjie is one of the major manufacturers of collating machines in China, offering efficient and precise solutions.

- **Wanjie WJ-330.**

- **Key features:**

- Equipped with three automatic paper feeding trays for fast and accurate collation.
- Trays can be easily adjusted for sheet size and number of copies.
- Electronic control system allows flexible configuration of parameters.

- **Applications:**

- Commonly used in the production of multi-copy invoices, event tickets, and documents requiring ordered collation.

❖ **3-Tray Collating Machine from Shenzhen (China).**

Shenzhen is a prominent supplier of collating devices, especially in models with three trays for sequential sorting.

o **Shenzhen SH-650**

- **Key features:**

- 3-tray automatic collating machine with adjustable speed and precise sheet length control.
- Trays are designed to arrange sheets in correct order, saving time and effort for operators.

- **Applications:**

- Mainly used in printing applications such as invoices, warranty cards, and discount vouchers.

❖ **3-Tray Collating Machine from Zhejiang (China).**

Zhejiang offers a variety of high-performance collating machines, especially models with three sequential-sorting trays.

o **Zhejiang ZJ-530.**

- **Key features:**

- Capable of collating paper into three preset-length copies to manage the printing process more efficiently.
- Three independently operating trays with flexible adjustment for product requirements.
- PLC control system optimizes the cutting and sorting process.

- **Applications:**

- Used in the production of invoices, discount vouchers, event tickets, and multi-copy printed documents

❖ **3-Tray Collating Machine from Shangyin (China).**

Shangyin is a brand specializing in collating machine solutions for the printing industry.

- **Shangyin SY-600**

- **Key features:**

- 3-tray collating machine operates in synchronization, automatically collating and sorting sheets in order.
- Automatic paper feeding trays with adjustable copy size and collation speed.
- User-friendly digital control panel for quick adjustments by operators.

- **Applications:**

- Suitable for products such as tickets, receipts/payments, and sales invoices.

1.2. Purpose of Selection and Desired Outcomes

- **Reducing costs and managing industrial printer waste.**

For perforating, creasing, and collating machines from well-known manufacturers, they are always the ideal choice for large enterprises or industrial-scale production workshops thanks to their stable operation, high performance, and superior features. These machines can handle large workloads in a short amount of time, meeting strict product quality standards, thereby optimizing the production process and increasing productivity.

However, for small businesses, especially small print shops, investing in these industrial-grade machines may not be appropriate. The initial investment cost is very high, not to mention maintenance and operation expenses, making it difficult for small businesses to afford. Furthermore, they typically do not require the full range of advanced features these machines offer. Given their limited production scale, using such large industrial equipment is unnecessary and could lead to wasted resources.

On the other hand, currently, many used printers—particularly older Ricoh models—have become obsolete and are being discarded, mostly coming from countries like China and Japan. These printers are often recycled, resold, and imported into Vietnam. Instead of letting them become waste, we can repurpose and recycle these old devices to build automatic collating and creasing machines.

Modifying these used machines not only helps reduce waste but also solves some financial challenges for small businesses. By utilizing available components from old printers, we can build equipment at a significantly lower cost compared to purchasing new machines, while still meeting essential functions such as automatic collation and creasing. This approach not only contributes to environmental protection but also provides a financially viable solution, helping small businesses cut down on investment costs and enhance their competitiveness in the market.

Through this method, we not only address pollution and waste issues from old devices but also create opportunities for small enterprises to access modern technology without incurring high costs. Recycling and reusing components from old printers will be a sustainable solution that delivers both economic and environmental benefits for small businesses in the printing industry.

- **Reducing processing steps and saving time.**

The implementation of this project represents a significant advancement in the production process by integrating three functions—collating, creasing, and perforating—into a single automated system. Previously, these tasks were typically carried out entirely by hand, especially the collating process, which required workers to manually pick up each sheet, arrange them in order, and only then move on to subsequent steps such as creasing or perforating. This manual workflow was not only time-consuming but also prone to errors, which could negatively affect the quality of the final product.

Thanks to this improved system, the entire document collating process is now fully automated, eliminating the need for manual handling during the collation stage. This significantly reduces processing time, thereby increasing production output within the same time frame. Functions like creasing and perforating are also performed sequentially and accurately within the same machine, minimizing the need to transfer materials between different stages and eliminating idle time between operations.

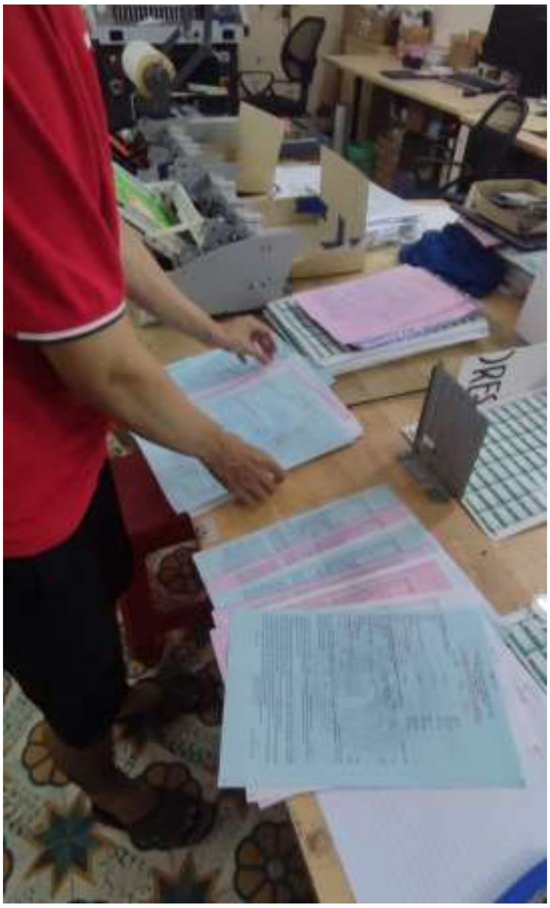


Figure 1.5 Manual collating in small print shops

Figure 1.6 Perforating and creasing with a homemade machine

1.3. Initial technical specifications set

- Customer Requirements:

Manufacturing using old printers to create a combined machine for collating, perforating, and creasing.

Flexible multi-function adjustment via control screen.

Some products: notebook binding, books, packaging, invitation cards, collating paper sets and perforating…

- **Technical specifications**: (Since the machine is made from reused old printers, technical specifications will be omitted and replaced with descriptions of the additional components integrated into the machine).

- Maximum width and length of the product (currently based on A4 size): A4 paper size 210x297mm → conveyor width > 310mm
- Paper feeding frequency, working speed: 30 sheets per minute.
- Accuracy: error-free products or ability to detect errors.
- Intermittent operation mode.
- Minimizes vibration and noise as much as possible.

CHAPTER 2: MECHANICAL SYSTEM DESIGN

2.1. General introduction to Ricoh printers

Figure 2.1 Ricoh Printer [5]

The Ricoh C5300s and Ricoh C5310s are two high-end digital printers from Ricoh's Pro series, specifically designed to meet the demands of professional printing. These are powerful machines with high-quality color printing capabilities, fast speed, and versatility, making them suitable for small and medium-sized businesses, digital printing centers, or production units requiring precision and high performance.

❖ Key deatures:

- Superior print quality:

- The Ricoh C5300s/5310s deliver a print resolution of up to 2400 x 4800 dpi, ensuring sharp images, vibrant and consistent colors. These printers are ideal for products such as flyers, brochures, business cards, books, and marketing materials that require high color accuracy.

- High print speed:

- Ricoh Pro C5300s: Print speed reaches 80 pages per minute (A4)

- Ricoh Pro C5310s: Higher speed, up to 90 pages per minute (A4). This helps optimize printing time, especially for large or urgent print jobs.
- Supports a wide range of paper sizes: The machine supports paper sizes from A6 to SRA3, and can even print on long sheets up to 1260 mm (for banners) with a maximum paper thickness of 360 gsm. This allows printing on a variety of media such as plain paper, coated paper, art paper, and labels.
- Integrated finishing capabilities: The printer comes with optional finishing features like stapling, bookbinding, folding, hole-punching, brochure creation, and even staple-free binding. All processes are automated, reducing time and labor in the finishing stage.

• Advanced technology:

- Ricoh VCSEL toner system: Uses Ricoh's latest toner technology to ensure more consistent color and reduce toner waste, helping save costs.
- Fiery controller (optional): For print centers requiring advanced color management, the printer can be integrated with the Fiery E-27B or E-37A controller, enhancing color control and optimizing print quality.
- Smart management system: 10.1-inch touch-screen interface is user-friendly and easy to operate, allowing users to customize printing parameters and monitor machine status in real time.

❖ Environmentally Friendly Features

- Lower energy consumption compared to similar printers, helping reduce electricity costs
- Certified with Energy Star and EPEAT Gold, meeting international environmental protection standards

❖ Applications

- The Ricoh C5300s/5310s are designed to meet a wide range of printing needs:
- Small and medium businesses: Suitable for internal documents, brochures, and marketing materials at a reasonable cost
- Digital print centers: Ideal for urgent print jobs requiring fast turnaround and high quality
- Education sector: Printing books, curricula, or specialized documents
- Advertising industry: Producing posters, flyers, and high-quality marketing materials

❖ **Main Technical Specifications:**

Specification	Ricoh C5300s	Ricoh C5310s
Print speed (A4)	80 pages/minute	90 pages/minute

Specification	Ricoh C5300s	Ricoh C5310s
Resolution	2400 x 4800 dpi	2400 x 4800 dpi
Supported paper sizes	A6 to SRA3, long banner	A6 to SRA3, long banner
Paper thickness	Up to 360 gsm	Up to 360 gsm
Machine dimensions	799 x 880 x 1648 mm	799 x 880 x 1648 mm
Weight	263 kg	263 kg

2.2. Mechanical system design for the collating function

2.2.1. Collating mechanism principle of the machine

Ricoh printers consist of multiple components that perform different stages of the printing process, including the paper feed unit, drum unit, fusing unit, duplexing system, output tray, and more. However, for the specific purpose of collating (multi-part form separation), several components can be removed, including non-essential parts such as the Automatic Document Feeder (ADF), scanner glass tray, and scanning or printing functionalities.

In this application, the printer will be repurposed to focus only on core components that serve the collating function: the paper feeding mechanism, drum section, and paper storage trays. Specifically, we will utilize only the lower part of the printer — the section responsible for pulling paper from the three trays to enable 3-tray sequential collating. All upper components of the printer will be dismantled.

- Principle of set collation:

- The principle of collating by set differs from collating by printing.
    + In printing-based collating, the process involves printing once on the topmost sheet of a multi-part form. The printing pressure from the drum unit or printhead transfers through the underlying sheets using carbon paper or carbonless paper, allowing the same content to appear on each layer.
    + In contrast, set-based collating refers to separating pre-printed sheets into organized sets or document groups with identical structure (e.g., copy 1, copy 2, copy 3…), for the purpose of distribution, archiving, or delivery to relevant parties.

Based on this working principle, we utilize only the essential components mentioned earlier and eliminate most unnecessary parts of the printer.

The mechanism works by sequentially pulling paper from each tray to form complete multi-part sets. Paper is picked from each tray by the feed unit and transport rollers, which guide the sheets upward for further processing steps.

2.2.2. 3D assembly of the collating system

Since the mechanical system of the printer is reused for the collating function, this section will simulate the 3D system of the components repurposed from the Ricoh printer.

The components used for collating include:

- The bottom frame of the printer, used to fix the reused components.
- Three trays of the printer.
- The paper pickup unit (identical for each tray).

Structure and principle of paper feeding from each tray:

- **Tray 1:**

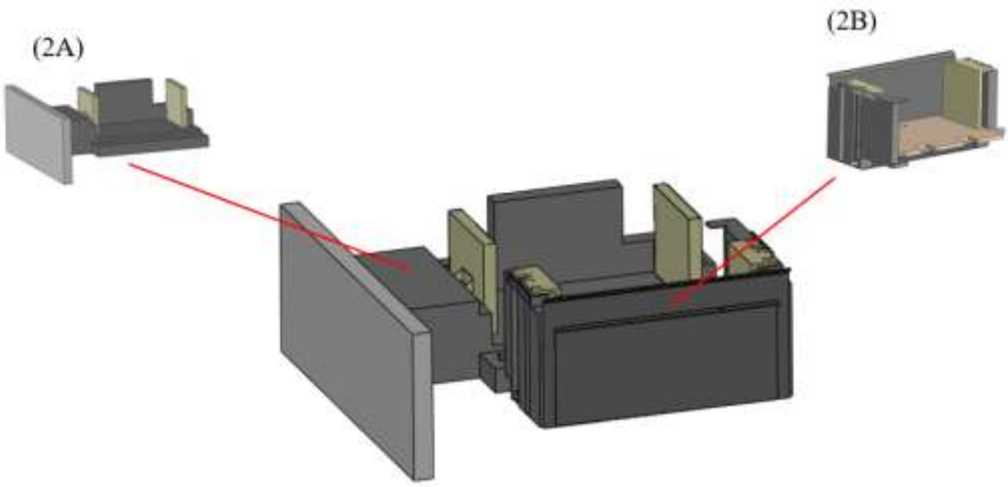


Figure 2.2 Overall structure of tray 1 components

Figure 2.2 shows the overall structure of Tray 1, consisting of two main sections with the following components and functions:

- Tray frame and paper holding component (2A) :

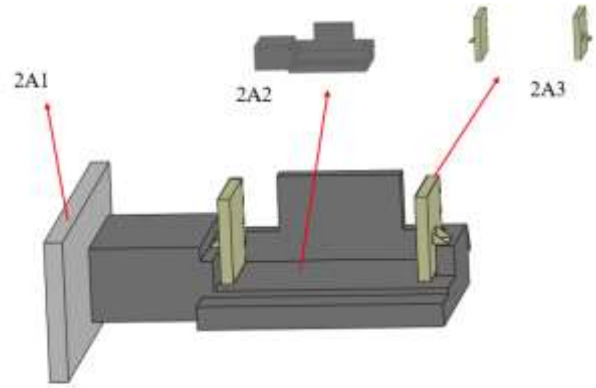


Figure 2.3 Tray frame and paper holding component

Overall structure includes:

+ Handle (2A1): facilitates convenient tray opening and closing.
+ Tray frame (2A2).
+ Paper side guides (2A3).

- 2B Paper lifting unit:

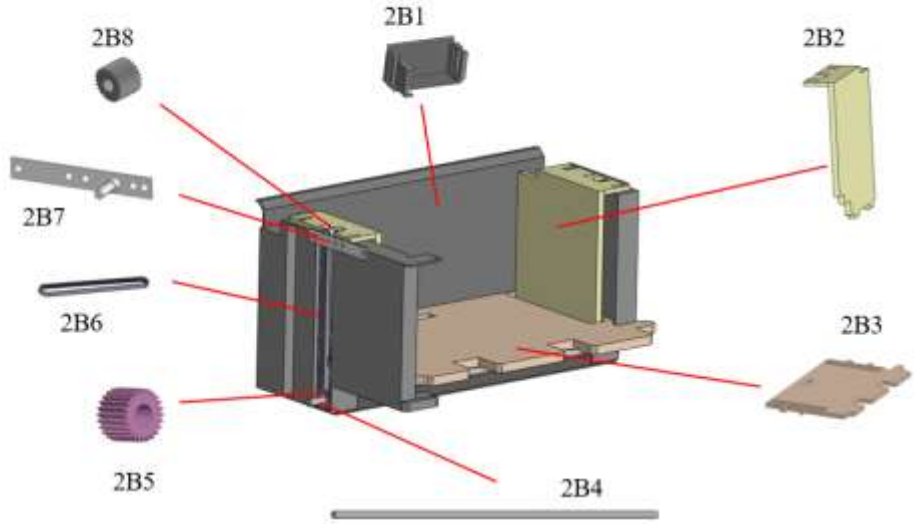


Figure 2.4 Paper lifting mechanism of tray 1

This component is designed to lift the paper up to the paper pickup assembly, enabling paper to be fed out from tray 1.

+ Consists of the following components:
    - Paper lifting frame (2B1).
    - Paper support tray (2B3).

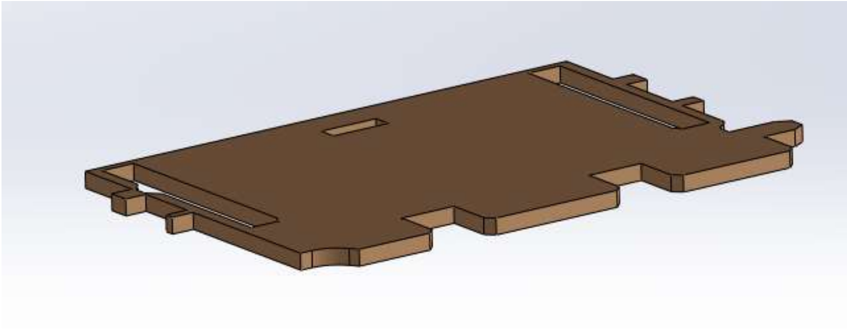


Figure 2.5 Paper support shelf of tray 1

This support shelf works together with the pulley-toothed belt transmission system to raise and lower the shelf.

+ Pulley – timing belt:
  - Lower pulley attached to the rotating shaft (2B5).
  - Upper pulley (2B8).
  - Timing belt (2B6).

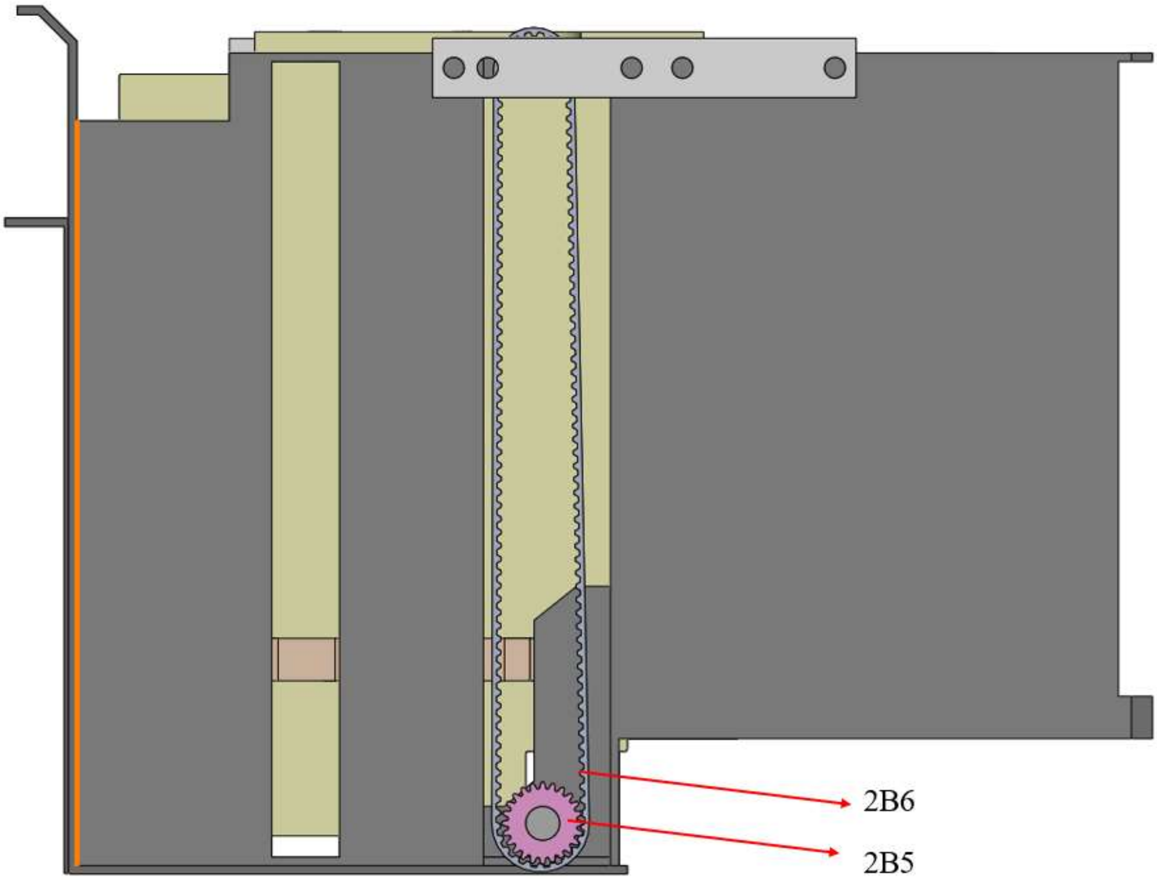


Figure 2.6 Pulley and timing belt transmission of tray 1

Based on the rotational motion of the shaft transmitted through the transmission system, the support shelf is raised.

+ Paper side fixing components (2B2):

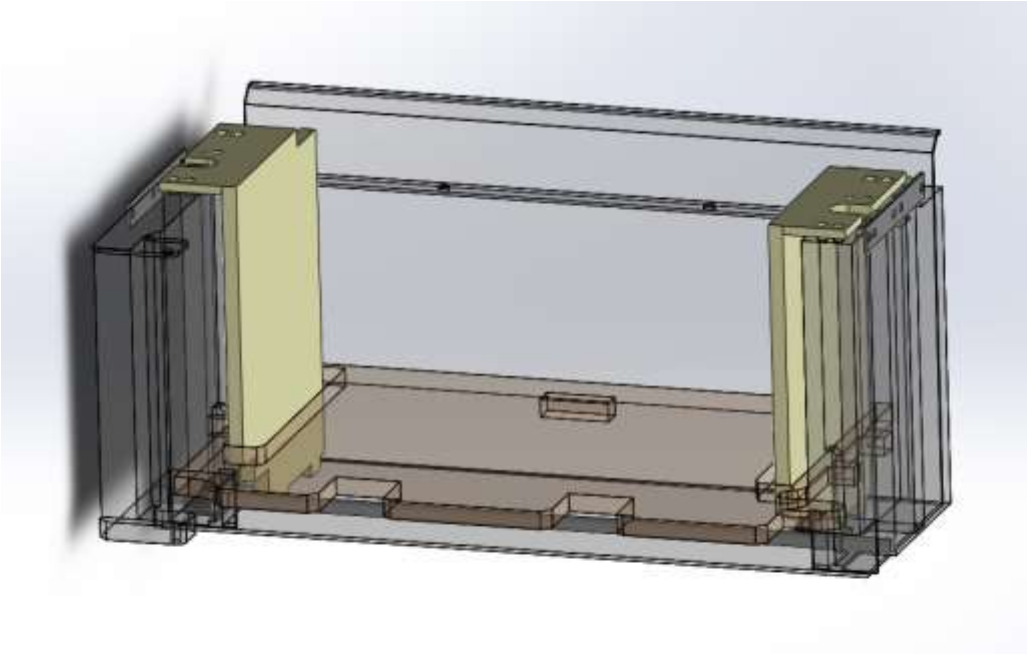


Figure 2.7 Pulley – timing belt transmission of tray 1

This component helps secure both sides of the paper to prevent misalignment..

+ Rotating shaft 2B4.

+ Upper pulley holding bar 2B7

- **Tray 2 and tray 3 :**

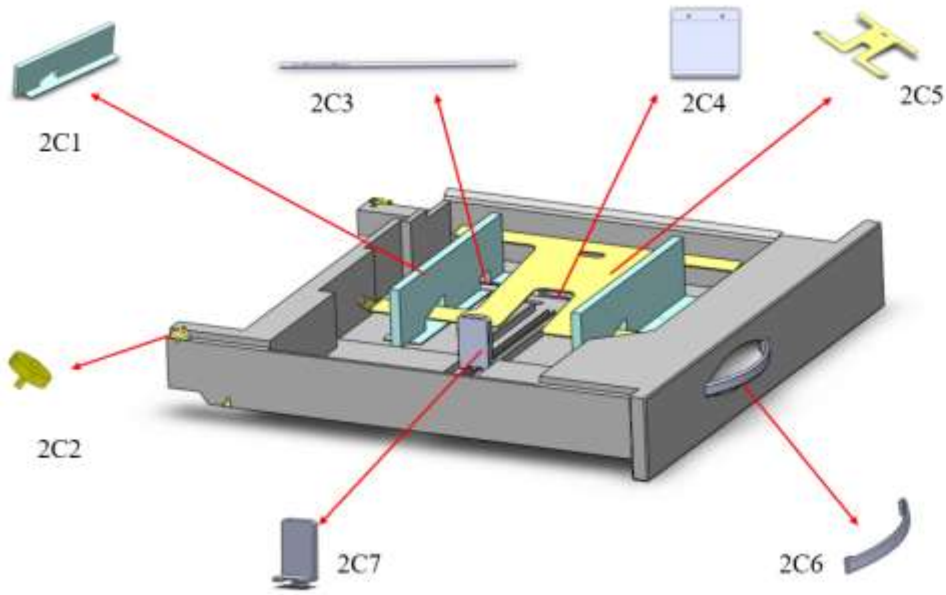


Figure 2.8 Overall structure of the top components of tray 2 and tray 3

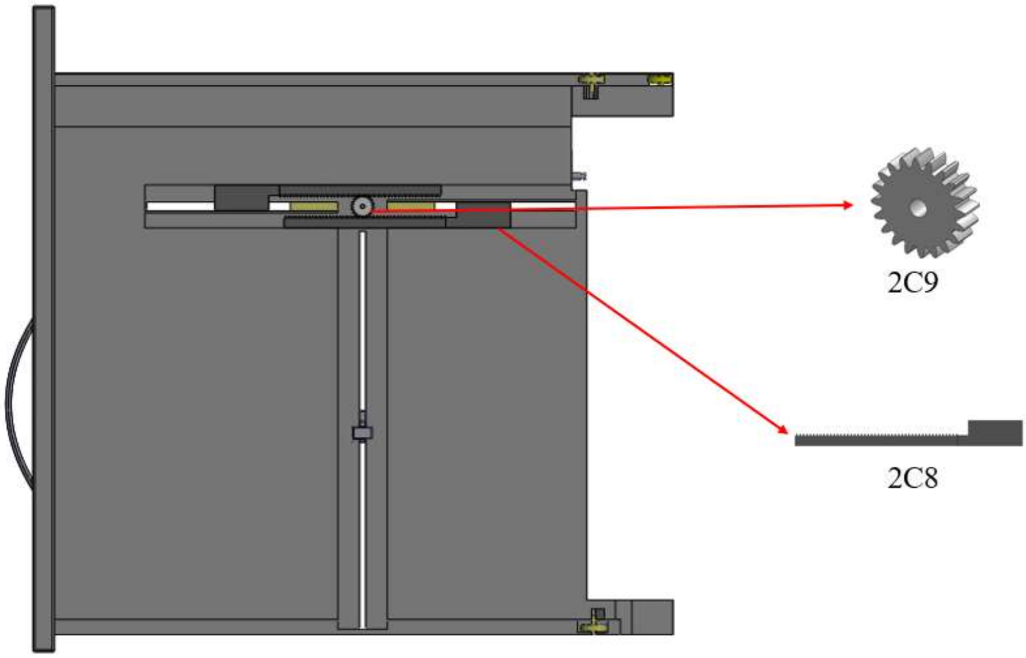


Figure 2.9 Overall structure of the upper components of tray 2 and tray 3

Figure 2.8 and figure 2.9 show the overall structure of tray 2, including its components and the functions of each component as follows:

- Side paper alignment component (2C1).
- Rack bar (2C8).
- Gear (2C9).

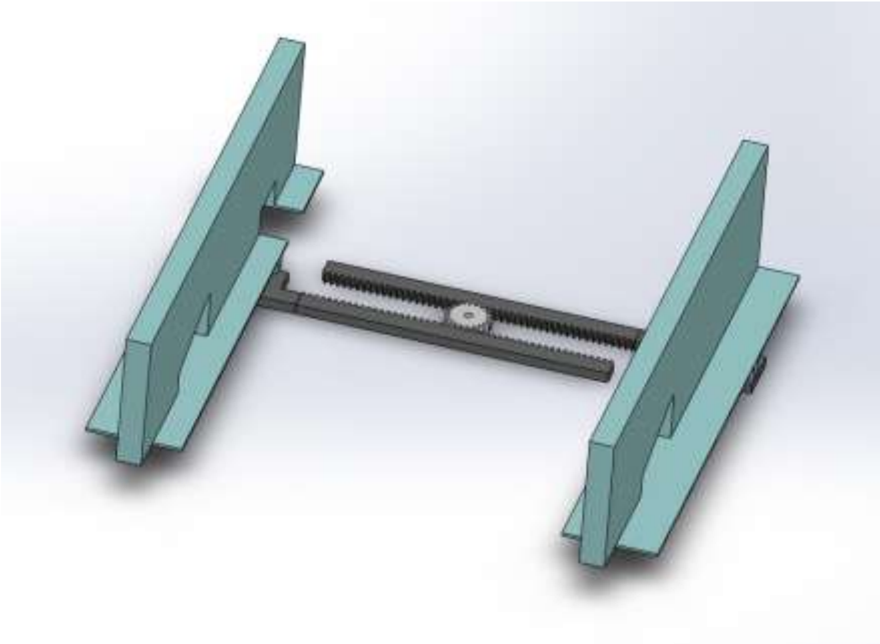


Figure 2.10 Side paper holder components of tray

This component is designed to secure both sides of the paper and operates based on the rack-and-pinion mechanism to open and close horizontally.

+ Vertical paper alignment component (2C7):

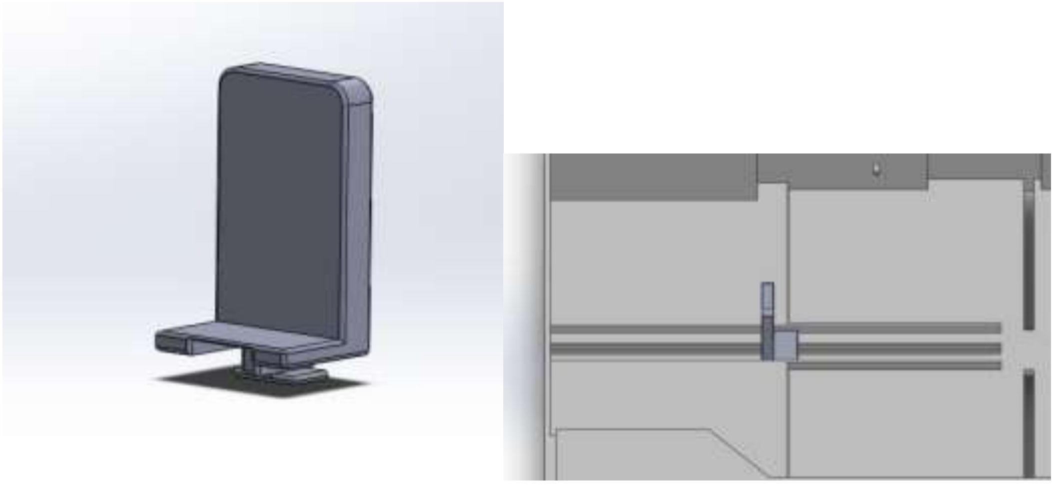


Figure 2.11 Vertical fixing component

This component secures the paper in the vertical direction. When the paper is lifted, it tends to tilt and slide backward, so this part ensures the paper remains in position and prevents it from slipping.

+ Paper lifting mechanism (2C5) :

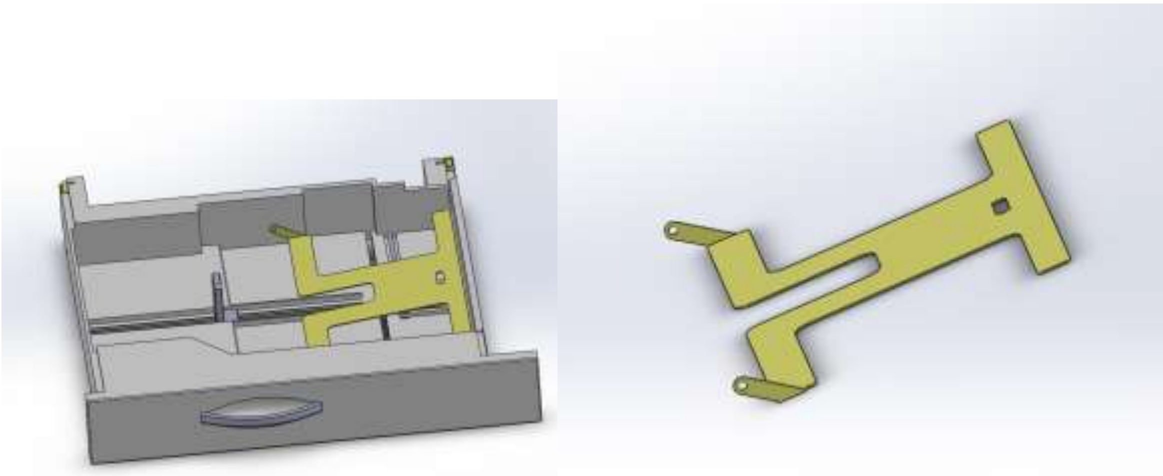


Figure 2.12 Paper lifting component of tray 2

This component lifts the paper to the upper limit, allowing the pickup assembly to feed the paper out.

+ Rotating shaft (2C3) and Support bracket attached to the rotating shaft (2C4):

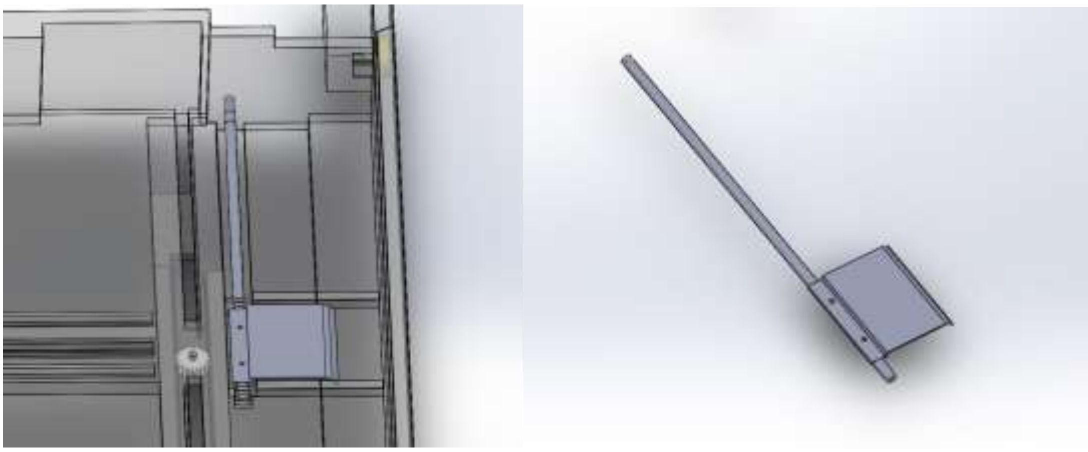


Figure 2.13 Paper lifting shaft of tray 2

This rotating shaft works in conjunction with a support plate. When the motor transmits power through the drive system to the shaft, the combined motion of the shaft and the support plate assists the paper lifting mechanism.

+ Tray frame and the remaining components:

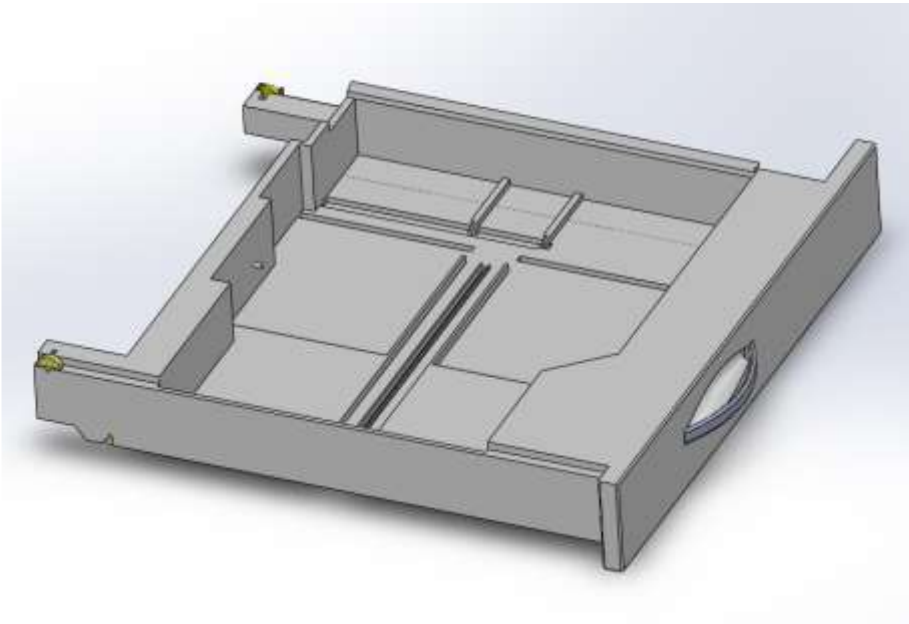


Figure 2.14 Frame of tray 2

The tray frame is a fixed component connected to the other parts and serves as the holder for the paper.

+ Roller bearing (2C2):

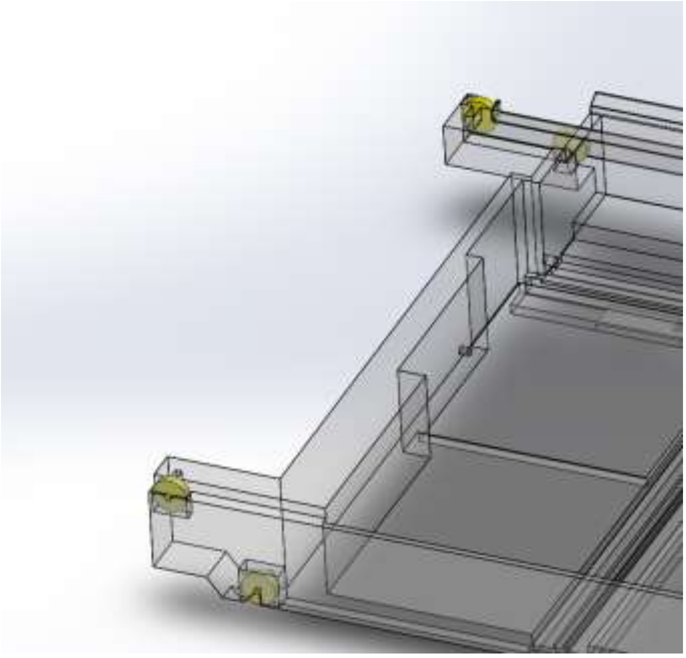


Figure 2.15 Roller bearing of tray 2

The roller bearing allows the tray to open and close by sliding along the fixed rail of the machine frame, while also helping to secure the tray in place.

+ Handle (2C6):

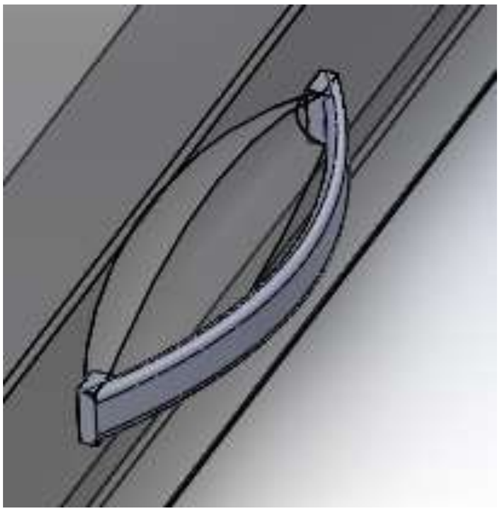


Figure 2.16 Handle of tray 2

The tray handle facilitates easy operation of opening and closing the tray.

2.2.3 Paper pickup unit

- **Paper feeding mechanism:**

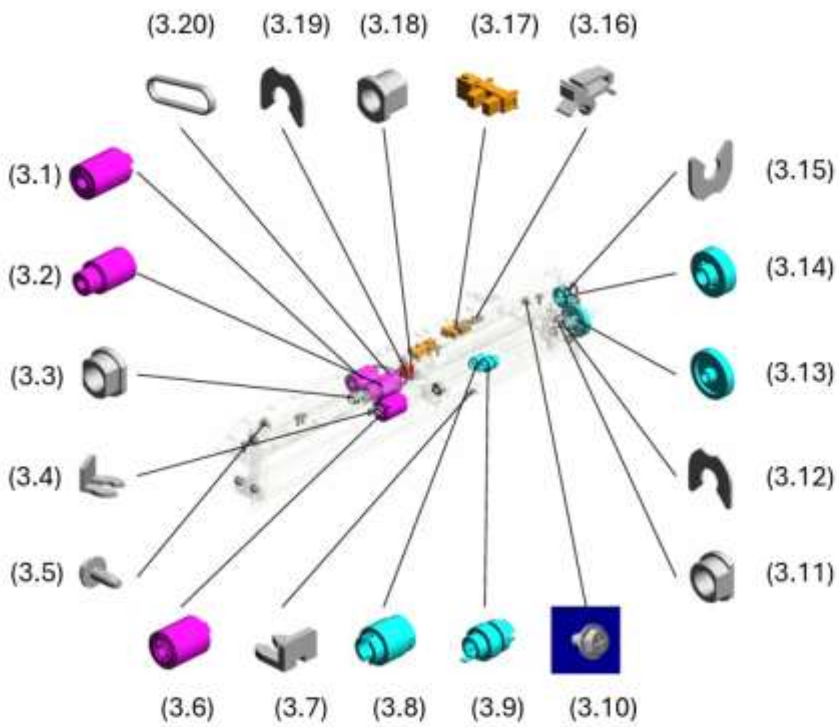


Figure 2.17 Paper picking mechanism of the trays [6]

- The components of the paper pickup assembly shown in Figure 3.5 include:

- (3.1) paper feed roller: pickup
- (3.2) paper feed rol ler:feed
- (3.3) plain shaft bearing
- (3.4) retaining ring c - separate
- (3.5) hexagon head tapping screw
- (3.6) paper feed roller: separate
- (3.7) snap ring
- (3.8) gear:separate: driven
- (3.9) gear:separate: drive
- (3.10) tapping screw - m3x6
- (3.11) plain shaft bea ring
- (3.12) retaining rings -e:silencer:dia4
- (3.13) gear:separato r:input
- (3.14) gear:feed
- (3.15) retaining rings -e:silencer:dia3

- (3.16) clamp:lwsm-030
- (3.17) photointerrup tor:gp1a173lcs
- (3.18) plain shaft bea ring
- (3.19) retaining rings -e:silencer:dia4
- (3.20) timing belt

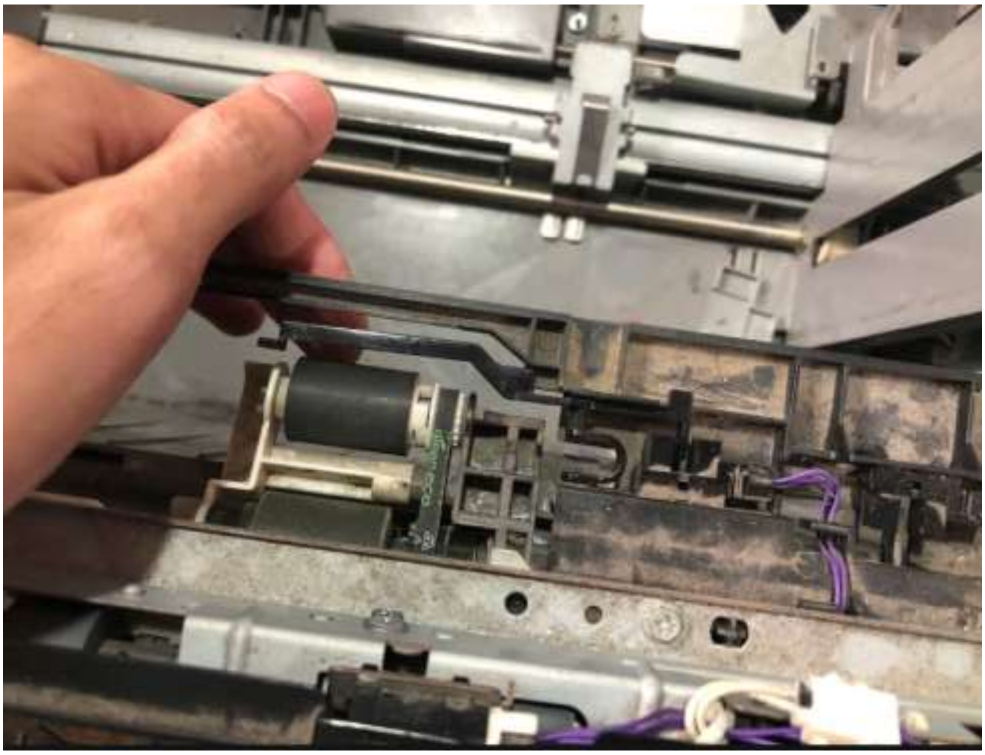


Figure 2.18 Paper picking mechanism of the trays

- **Operating principle of the paper striking and feeding mechanism:**

The paper striking mechanism using an electromagnet operates based on the interaction between electromagnetic force and mechanical components.

- **Main components:**

- Electromagnet (Solenoid): Generates magnetic pulling force when electric current passes through.
- Return spring: Returns the mechanism to its original position after the electromagnet is deactivated.
- Gear or lever: Connects to the pick-up roller and performs mechanical movement when activated by the electromagnet.
- Pick-up roller: The roller that makes direct contact with the paper to pull it from the tray.

- **Operating principle:**

- Activation signal: When an electrical signal is sent to the solenoid in the paper striking mechanism.
- Magnetic force generation:
    - Electric current flows through the solenoid coil, generating a strong magnetic field.
    - This electromagnetic force pulls a lever or mechanical linkage, changing the position of a gear or lever.
- Pick-up roller activation: As the lever is pulled, it releases the pick-up roller, allowing it to drop down and make direct contact with the paper in the tray.
- Roller rotation:
    - The roller is driven by the main motor or the printer's gear transmission system.
    - The roller pulls a sheet of paper from the tray and feeds it into the paper path.
- Return spring: When the solenoid stops receiving a signal (power is cut off), the return spring pushes the lever or gear back to its original position, ready for the next cycle.

This mechanism is used for all three trays to feed paper to the outside.

2.2.3 Actual assembly process

2.3.1. Disassembly

- Disassemble all upper components of the machine.

Figures 2.19 Disassembly of components on the printer

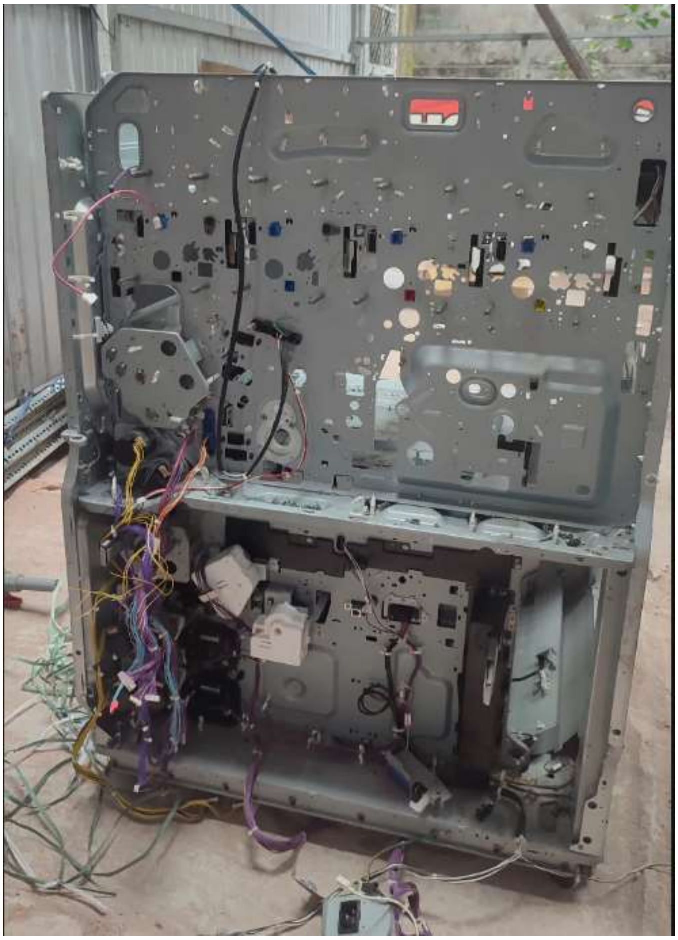


Figure 2.20 After disassembling the components on the upper part of the machine

2.3.2. Cutting off the upper part of the machine body

Figure 2.21 After cutting off the upper part of the machine

Figure 2.22 Front of the machine after cutting off

2.3. Mechanical system design for the perforating and creasing functions

2.3.1. Kinematic diagram of the perforating and creasing mechanism

The perforating and creasing processes are carried out after the paper is fed from the trays. Since the Ricoh printer does not have these functions built in, the mechanical components are designed based on the working principles, kinematic diagram, and the machine’s existing structure.

- **The creasing principle**

- Creasing is the process of creating a fold line on the surface of paper or board to support easy and accurate folding without cracking or misalignment.
- A creasing blade (concave pressing blade) presses down on the paper in combination with a counter-pressure shaft with a groove or soft pad. Under pressure, the paper deforms along the creasing blade to create a groove or ridge.

- **The perforating principle**

- Perforating is the process of creating saw-tooth cut lines (intermittent cuts) on paper to allow easy tearing at the correct position.
- A perforating die (saw-tooth blade) mounted on the perforating shaft presses onto the paper with compressive force or rotary pressure. The blade has an interrupted shape (intermittent cutting blade), creating alternating open and closed cuts.

From the working principles of the two steps above, we can see the similarity in structure and operating method. Both use blades mounted on a rotating shaft or mechanical transmission system, then press directly onto the paper with a certain force to create either a deformation or a cut. The main difference lies in the blade shape: the creasing blade is continuous, creating a groove for folding; while the perforating blade is intermittent, creating broken cuts for easy tearing.

Therefore, the kinematic diagrams of these two mechanisms are the same.

According to the principle, appropriate pressing force is applied between the saw-tooth blade (for perforating), the creasing blade (for creasing), and the soft pad onto the sheet of paper.

There are many options for designing the perforating or creasing unit by using gear movement on a flat surface to create fold lines on the paper sheet.

- Compare the difference between the two mechanisms.

Table 2.1 Comparison of differences between the mechanism with a rotating shaft and the fixed plane

Characteristics	**Saw-tooth/Creasing Blade + Rotating Shaft with Soft Pad**	**Saw-tooth/Creasing Blade + Fixed Flat Surface**
Mechanism Type	Rotary mechanism	Reciprocating or pressing mechanism
Operation	Continuous, cyclic rotation	Intermittent or requires stopping the paper to press
Contact	Cyclic point contact along rotation	Full-surface or fixed straight-line contact
Speed	Faster	Slower, usually used for manual or stationary machines

- The use of a rotating shaft with a soft pad combined with a saw-tooth blade is a suitable solution for a continuous paper feeding system. :
  + During the continuous paper feeding process, the paper moves without stopping, so the rotary mechanism allows perforating to occur simultaneously while the paper is moving, without needing to stop.
  + Reduces the number of processing steps.
  + Minimizes complex mechanisms => Compacts the working space.

= > Use the mechanism of saw-tooth/creasing blade + rotating shaft.

- **Kinematic diagram construction:**

- The diagram includes 5 key components:
  + (1) – Frame

+ (2) – Blade holder post
+ (3) – Blade
+ (4) – Rotating shaft
+ (5) – Bearing support

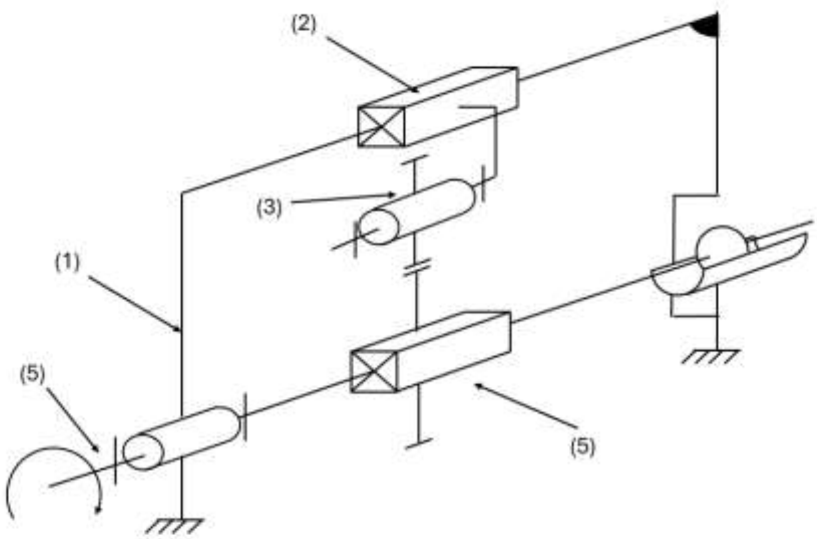


Figure 2.23 Kinematic diagram 1 of the creasing and perforating mechanism

The blade holder shaft moves horizontally along the machine frame to allow adjustment of the cutting position as desired.

The blade rotates by means of friction with the bearing support, which itself is driven by an external motor that rotates the attached shaft.

This frictional contact is intended to generate the necessary pressure on the paper sheet.

Integrating creasing and perforating functions on a single machine frame becomes simpler by replacing the blades and mounting multiple blade holder shafts onto the frame

To make it easier to install multiple blades on the machine, the structure of the blade holder shaft must be modified for quick assembly and disassembly while ensuring adjustable positioning to suit different applications.

- Modification of the kinematic diagram :

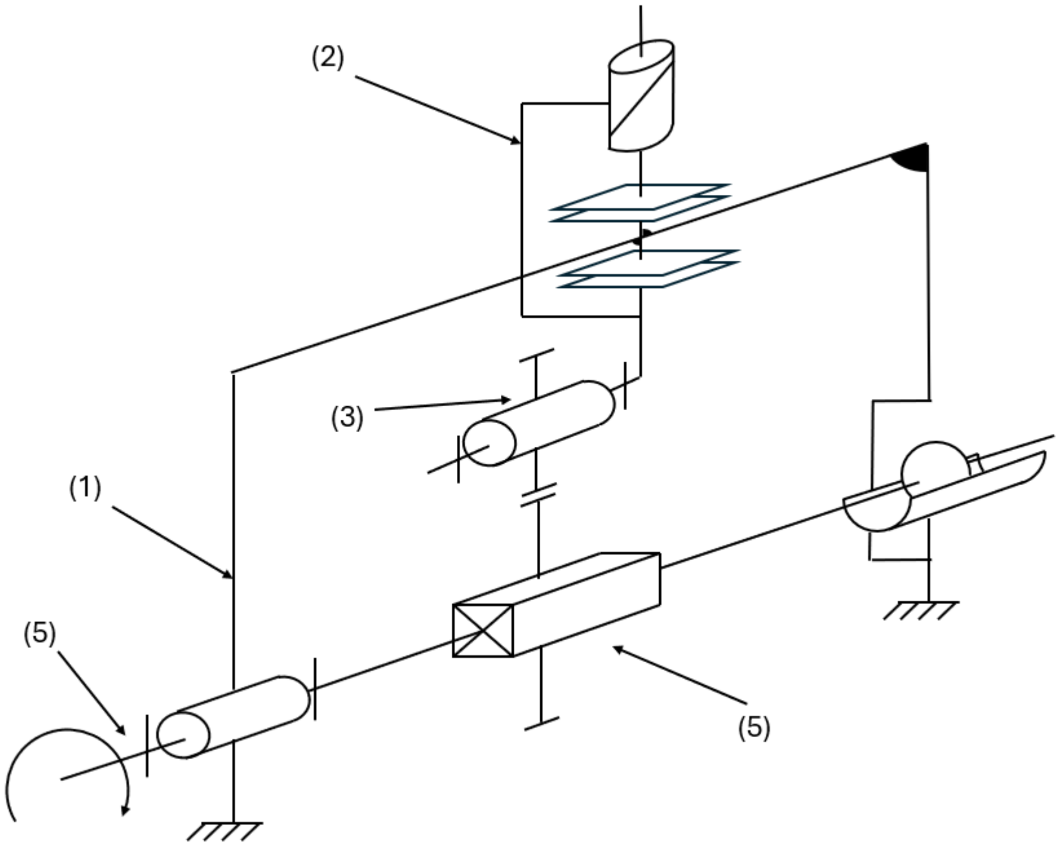


Figure 2.24 Kinematic diagram 2 of the creasing and perforating mechanism

The modification of the blade holder shaft structure into a screw joint and flat support joint is intended to allow easy disassembly and fixation. This makes replacing or adding blade holder shafts more convenient while still ensuring positional accuracy.

2.3.2. 3D design for the perforating and creasing unit

According to the technical design requirements, the conveyor width is based on the product (paper) dimensions:

- The width must be greater than 310 mm to accommodate A4 sheets.
- 155 mm × 205 mm (±2) for small notebooks, and 170 mm × 240 mm for large notebooks (books).

  → The length becomes 170 × 2 = 340 mm for a large notebook when opened. → Conveyor width > 360 mm to ensure proper accommodation.
- The increase in width also depends on the mechanical structure of the machine and available space. The purpose of increasing the width is to allow paper to be placed horizontally, enabling faster output compared to vertical placement — thereby increasing product output frequency (speed).
- Design of the spacer and the holder for the conveyor rotary shaft, as well as the perforating and creasing unit. The design is based on the specifications of the rotary shaft and the bearing block.

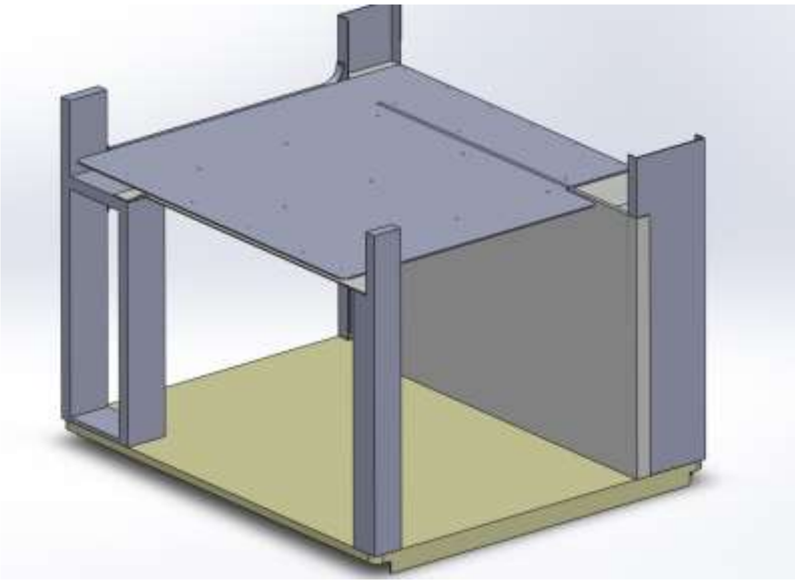


Figure 2.25 Design of the machine's liner pad

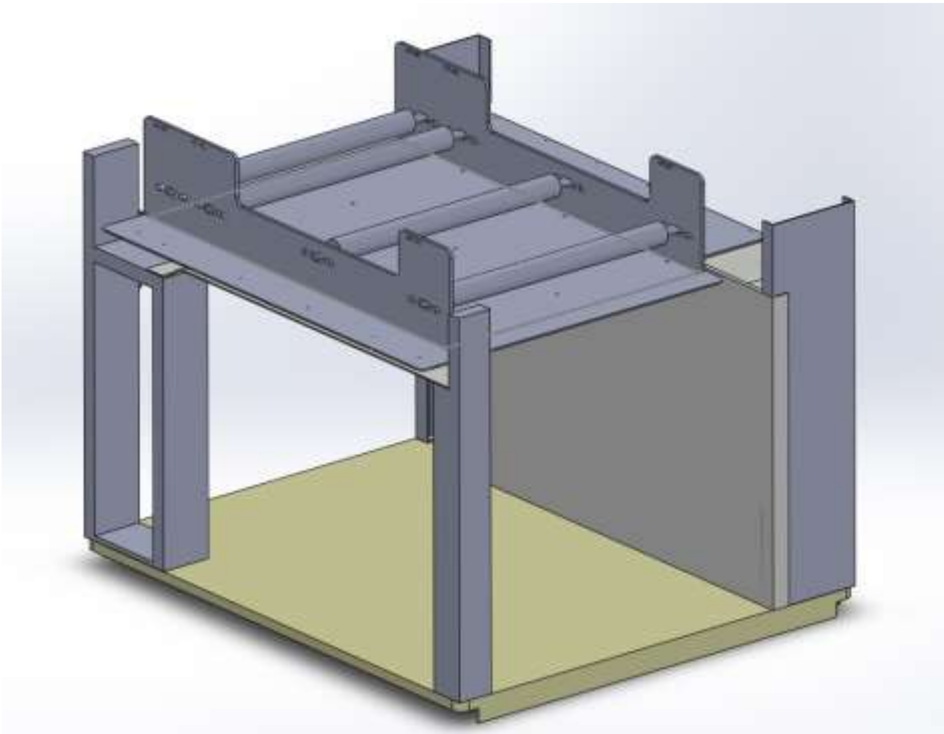


Figure 2.26 Design of the conveyor shaft support and the serrated cutting unit

2.3.3. Actual assembly

- For the conveyor drive shaft, reuse the existing shafts that were removed from the original machine.

Figure 2.27 Rotary shaft used for the conveyor belt

- Reuse the original design of the perforating and creasing shaft.

Figure 2.28 Rotary shaft bar for perforating and creasing.

- Use the available KFL000 bearing supports with a 10mm diameter for the rotating shaft.

Figure 2.29 KFL000 bearing block with 10mm diameter.

2.3.4. Technical drawing

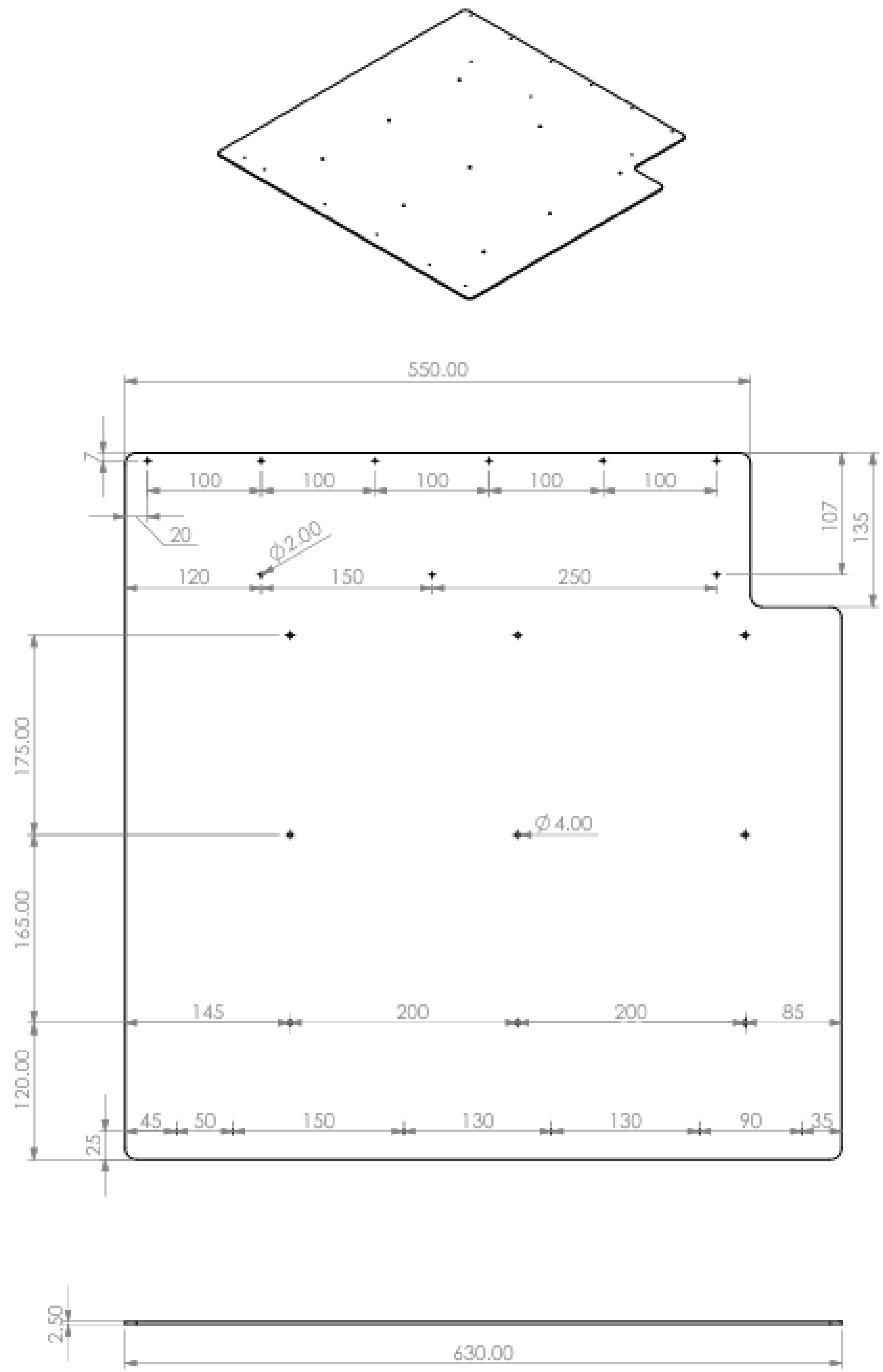


Figure 2.30 Detailed drawing of the machine shim

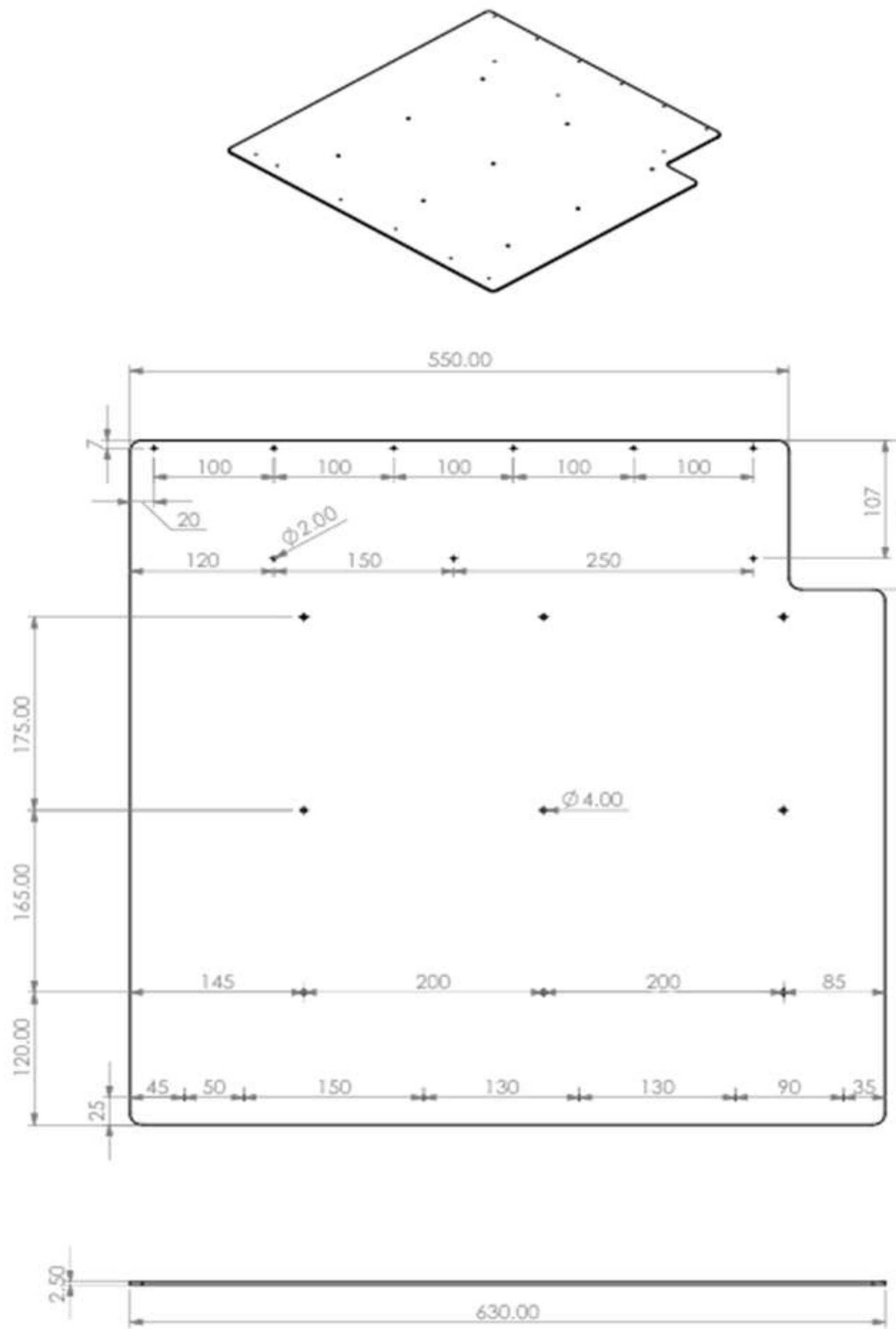


Figure 2.31 Detailed drawing of the conveyor shaft support and the serrated cutting unit

CHAPTER 3: ELECTRICAL SYSTEM DESIGN

3.1. The components of the machine's mechatronic system

3.1.1. Assembly module

The assembly module is the overall mechanical structure of the machine. It integrates other functional modules such as lifting, paper feeding, creasing, perforating, etc., into a synchronized system. At the same time, it ensures precise and durable physical connections between components, enabling the machine to operate smoothly, stably, and efficiently.

- **Components:**

- The machine frame serves as the base support structure for the entire system. It holds all components in place and protects them from dust and physical impact.
- The paper pick-up unit includes rollers that pull paper from the trays, ensuring that only one sheet is fed at a time. It also helps detect paper position and signals paper jam errors.
- The conveyor transports paper from the trays through the creasing and perforating unit.
- The creasing and perforating unit consists of feed rollers and serrated rollers for applying creases and perforations to the paper.
- The input trays consist of three paper trays storing sheets.
- The output section includes a rotating mechanism that collates printed sheets into separate sets, making it easier to organize and sort them.

- **Functions:**

- It integrates functional modules such as user interface, processing, sensing, and actuation into a unified system.
- It maintains the alignment and stability of components, ensuring that sensors, motors, and paper trays operate accurately.
- It minimizes vibrations, paper skewing, and jams, especially during high-speed operation

3.1.2. Measurement module.

This is the input section of the processing module. It consists of limit switches, push buttons inside the machine, and various sensors that send signals to the controller.

The sensors, push buttons, and limit switches used in the machine include:

- U-shaped 3-pin optical sensor (Vcc, Out, Gnd), PNP type.
- Paper detection optical sensor, 3-pin (Vcc, Out, Gnd), PNP type.

3.1.2.1. U-shaped photoelectric sensor with 3 pins (Vcc, Out, Gnd), PNP type

Figure 3.1 U-shaped photoelectric sensor

Figure 3.2 Structure and pin diagram of the U-shaped photoelectric sensor

- **Operating principle:**
- **Position and main functions of the U-shaped optical sensor used in the model:**

- Each tray uses 3 sensors (a total of 9 sensors for 3 trays, with the same configuration for each tray), with the following positions and functions:

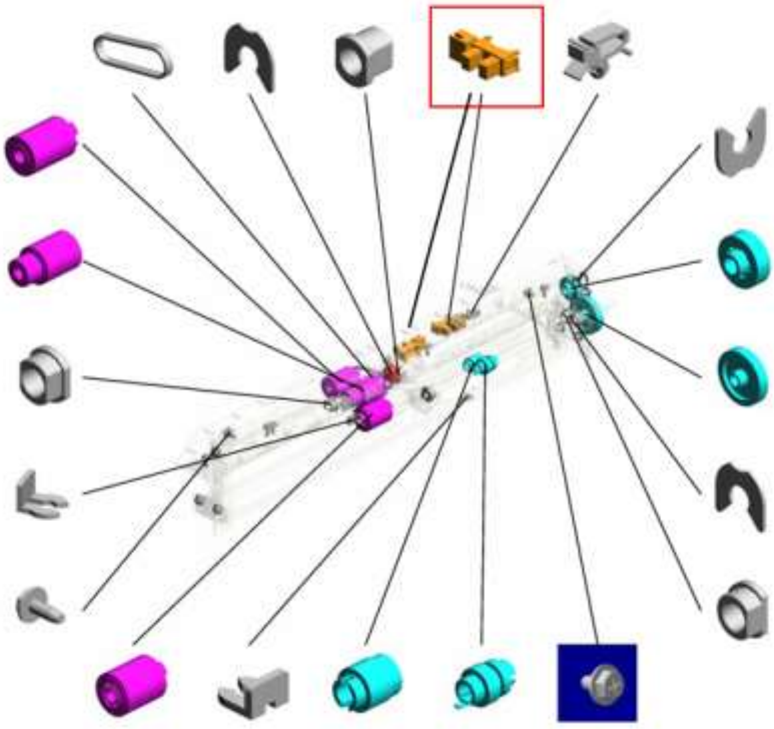


Figure 3.3 Position of the U-shaped photoelectric sensor in the paper picking assembly [6]

- The orange color represents the U-shaped optical sensor, which is positioned in the paper pick-up module with the following functions:

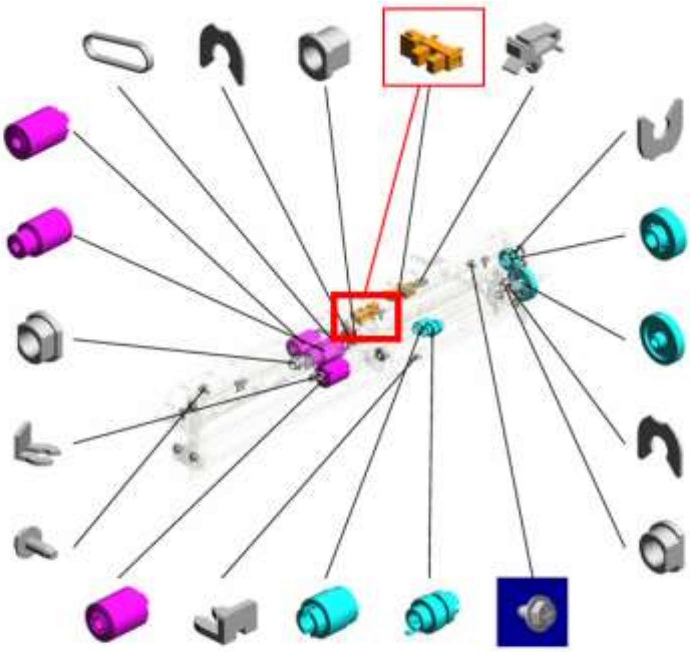


Figure 3.4 Position of the first U-shaped photoelectric sensor in the paper picking assembly [6]

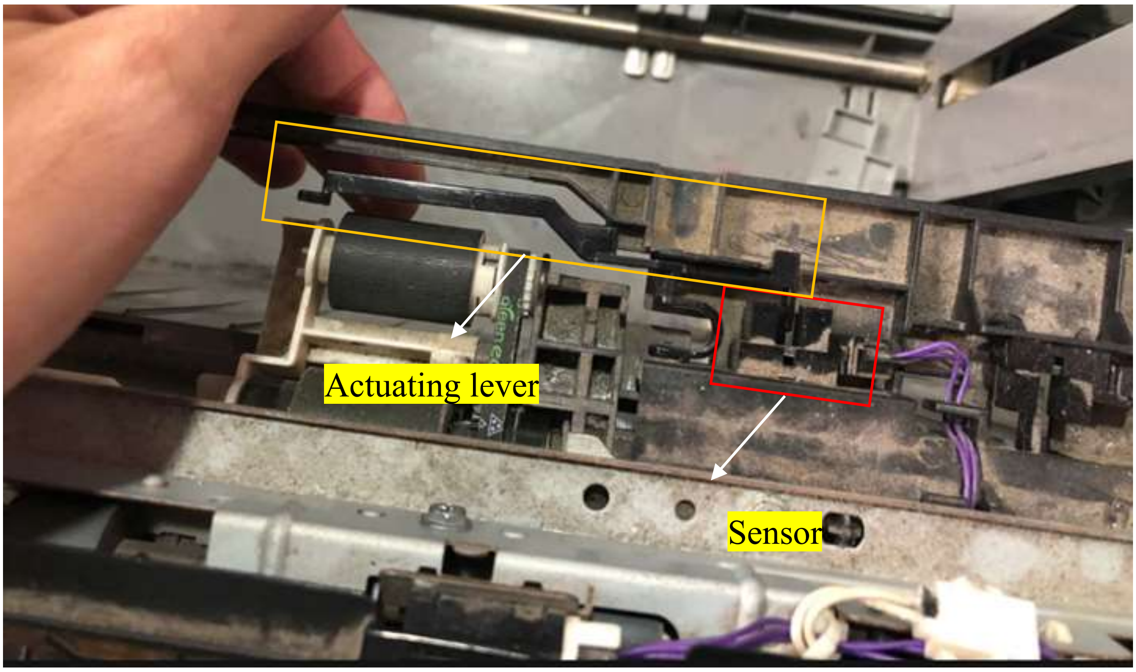


Figure 3.5 Actual position of the actuator bar and sensor

- In this position, the sensor works in conjunction with the lever mechanism to determine the uppermost limit position of the paper. At this distance, the paper is correctly aligned for the pick-up mechanism (shown in purple in the diagram).
- Operating principle: When the paper stack is lifted and contacts the lever, the rear end of the lever drops and blocks the sensor. At this point, the lifting motor stops operating. During machine operation, if the sensor no longer detects the lever, it means the paper has dropped below the pick-up range, and the lifting motor will activate until the sensor is triggered again.

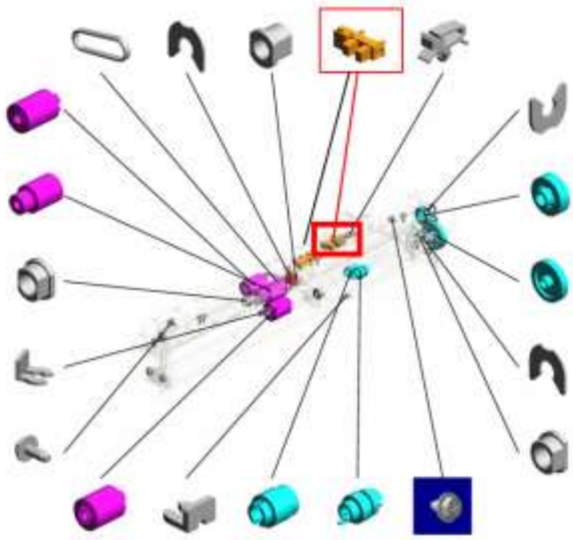


Figure 3.6 Position of the second U-shaped photoelectric sensor in the paper picking assembly [6]

- In the second position on the paper pick-up module, the sensor helps detect the upper and lower limits of the pick-up mechanism (shown in purple). It works in conjunction with Sensor 1 to determine whether the tray is out of paper.
- Operating principle: When paper is present, it will first push the lever, causing Sensor 1 to be triggered, and the system will operate normally. If there is no paper in the tray, the space where the lever normally sits will be empty. In this case, the lever will be bypassed, and the tray will continue to be lifted until it directly contacts the pick-up mechanism. When Sensor 2 detects this, the lifting motor stops, and a paper-out alert is triggered.

Table 3.1 Operating status of paper lift based on the states of two U-shaped photoelectric sensors

Sensor 1	Sensor 2	Operating Status
0	0	Continue lifting the paper.
1	0	Stop lifting. Paper present.
0	1	Stop lifting. Out of paper.
1	1	Invalid state. Error.

- The final position in the tray determines whether the tray is open or closed. The tray is attached to a mechanism consisting of a lever; when the tray is closed, this lever is positioned within the U-shaped slot of the photoelectric sensor. This indicates that the tray is closed and allows the motors to operate.

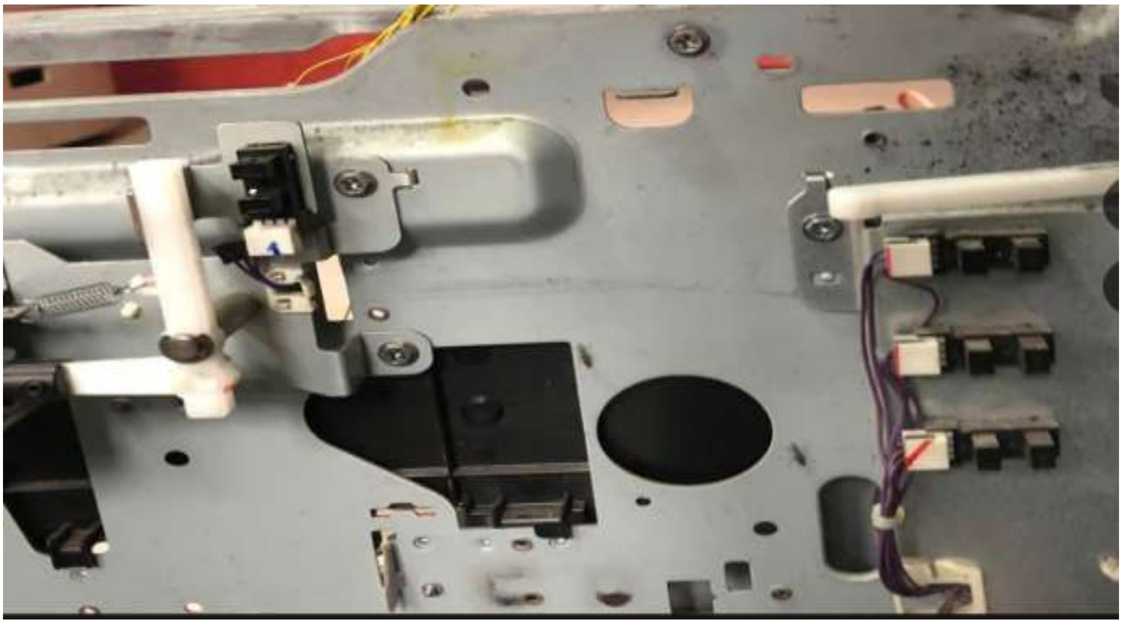


Figure 3.7 U-shaped photoelectric sensors for detecting the open/close status of the tray

3.1.2.2. PNP type 3-wire (Vcc, Out, GND) paper detection photoelectric sensor

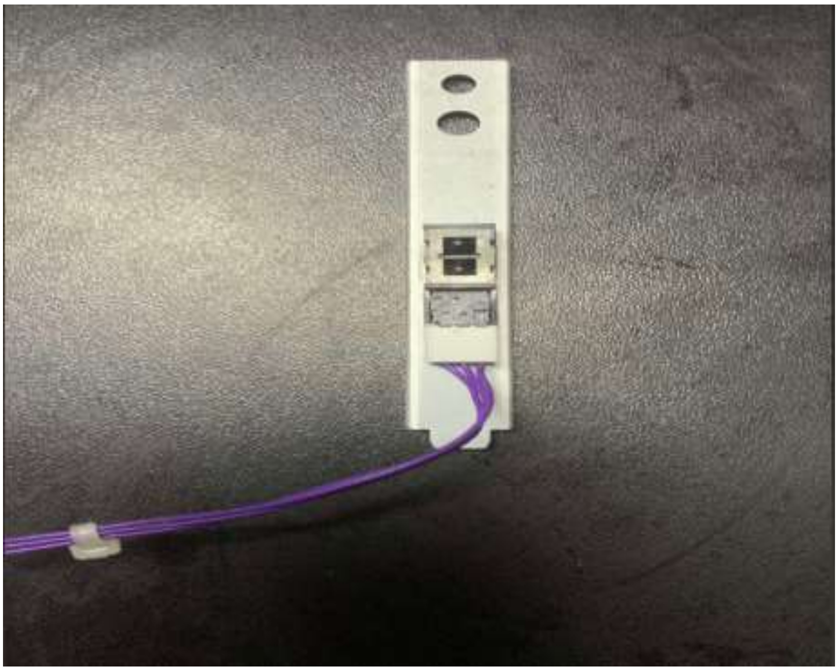


Figure 3.8 Paper reading photoelectric sensor

- **Position and main function of the photoelectric sensors used in the model:**

- In the paper pick-up module of each tray:

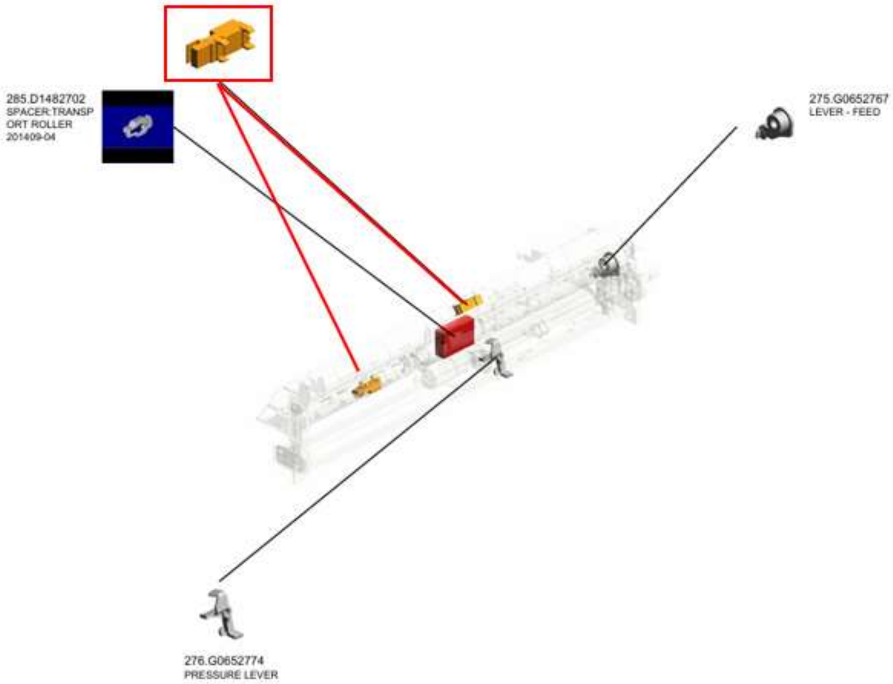


Figure 3.9 Position of the paper reading photoelectric sensor in the paper picking assembly [6]

- The orange component in the diagram represents the photoelectric sensor, which is positioned within the paper pick-up module with the following functions:

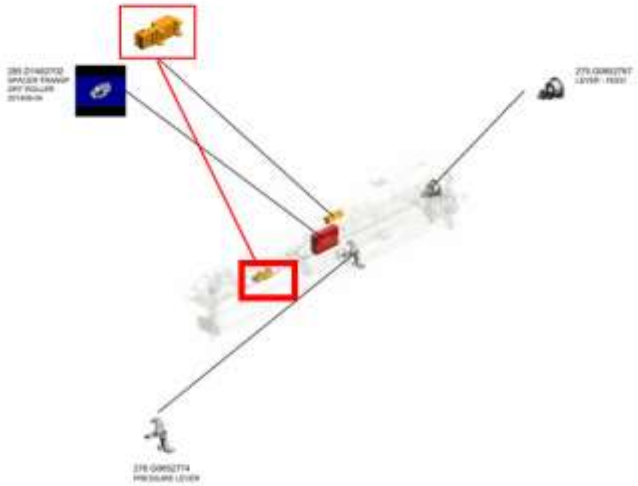


Figure 3.10 Position of the first paper reading photoelectric sensor in the paper picking assembly [6]

- At this position, the sensor helps determine whether the paper has been picked up. It works in conjunction with other mechanisms to proceed with subsequent tasks.
- Operating principle: The pick-up mechanism is positioned below to feed the paper. Once the paper is fed, the sensor detects it, triggering the lifting of the pick-up mechanism to prevent pulling in a second sheet, ensuring that only one sheet is fed at a time.
- Additionally, the system calculates the time from when the paper is detected to when it exits. If the time exceeds a defined limit, it indicates either a second sheet is being pulled or a paper jam has occurred, prompting an error notification.

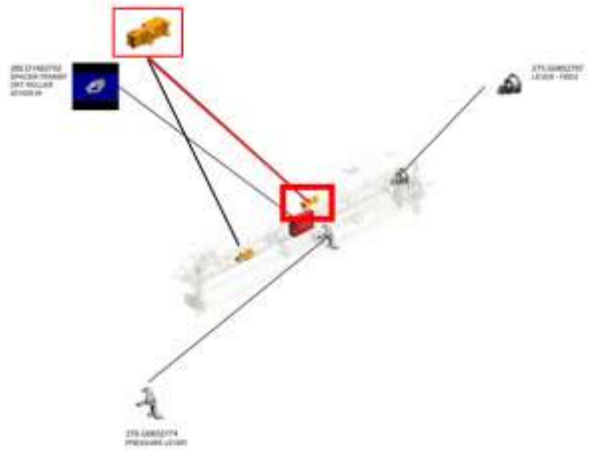


Figure 3.11 Position of the second paper reading photoelectric sensor in the paper picking assembly [6]

- At the second position on the paper pick-up module, the sensor detects whether the sheet has exited the pick-up unit and moved upward to proceed with the next operation for each tray.

3.1.3. Environment module

The environmental module is not a technical module physically located inside the machine like other hardware blocks, but rather a collection of physical and human factors that directly affect the machine's operation. Although not integrated into the electronic design, the environment is a critical factor that influences operational efficiency, system stability and lifespan, and the output quality of the creasing and perforating process.

- **Components:**

- Paper is the primary material used in all processes, and environmental factors such as temperature, humidity, and dust significantly impact the machine's performance and paper output quality. The machine operator is responsible for making adjustments, checking, and troubleshooting common errors, and also serves as the main controller of the machine.

- **Function:**

- Paper is the essential input for the system. If the input paper does not meet quality standards, the entire system will operate inefficiently. Dust can distort sensor readings, resulting in false errors such as “no paper detected” or “paper jam.” Environmental conditions also directly affect the performance, durability of machine components, and the quality of the finished paper product.

3.1.4. Actuation module

These are motors and mechanical mechanisms that control the movement of shafts through transmission systems.

The input comes from the Processing Module (ATMega 2560), which sends control signals to:

- Motor driver units
- Motors

- **Functions and positions of each motor in the system:**

3.1.4.1. Mabuchi motor ID-549XW-5031

Figure 3.12 Mabuchi motor ID-549XW-5031 [7]

❖ Mabuchi motor:

- Mabuchi motor is one of the world's leading manufacturers of DC motors, well known for its compact electric motors, especially in office equipment and consumer electronics.
- Mabuchi motors are commonly used in devices such as printers, optical drives, electronic toys, medical instruments, etc.

❖ Structure and uorking principle of mabuchi motor:

- Main components:

- Rotor: A coil of wire that generates a magnetic field when electric current flows through it.
- Stator: Permanent magnets that create a constant magnetic field.
- Brushes and Commutator: Enable current transmission from the power source to the rotor coil.
- Motor shaft and Bearings: Ensure smooth power transmission and rotation.

- Working principle:

- When current flows through the rotor coil, it generates a magnetic field that interacts with the stator's magnetic field.
- This interaction produces a **Lorentz force** acting on the coil, causing the rotor to spin.
- Mabuchi motors are brushed DC motors, where the **commutator** and **brushes** serve to switch the current direction in the rotor coils, enabling continuous rotation.

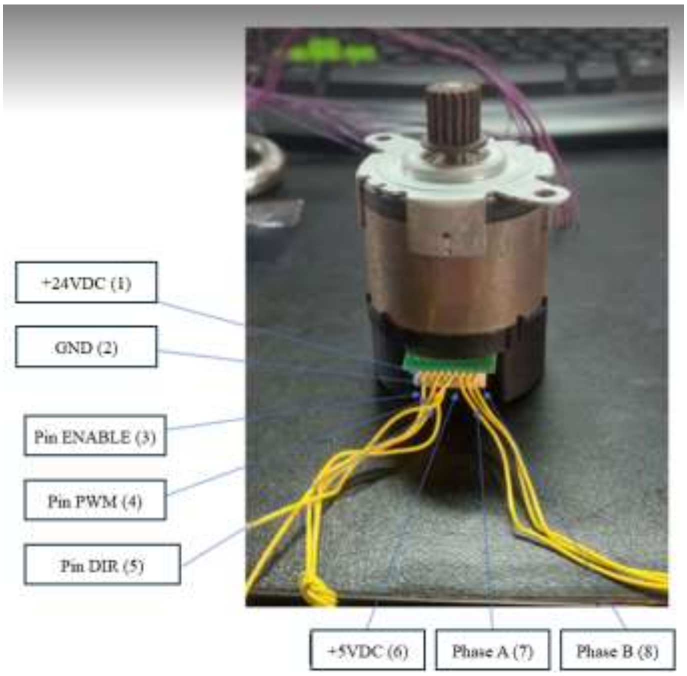


Figure 3.13 Wiring diagram of the Mabuchi motor ID-549XW-5031

❖ Mabuchi drive unit with integrated encoder – model: ID-549XW-5031

• Technical specifications:

- Compact DC motor
- Integrated encoder for position and speed monitoring
- Operating voltage: 12V – 24V
- Rotational speed: Approximately 3000 – 6000 RPM (depending on input voltage)

• Working principle of the encoder motor:

- The encoder is typically an optical or magnetic rotary sensor.
- As the motor shaft rotates, the encoder generates high-frequency pulse signals.
- These pulses are interpreted by a microcontroller to calculate the angular position and rotational speed of the motor.

• Functions in a printer system:

- Controls paper feed and paper pick-up shafts.
- Enables precise positioning of the printhead through the integrated encoder.
- Assists in the movement of the scanning head during scanning operations.

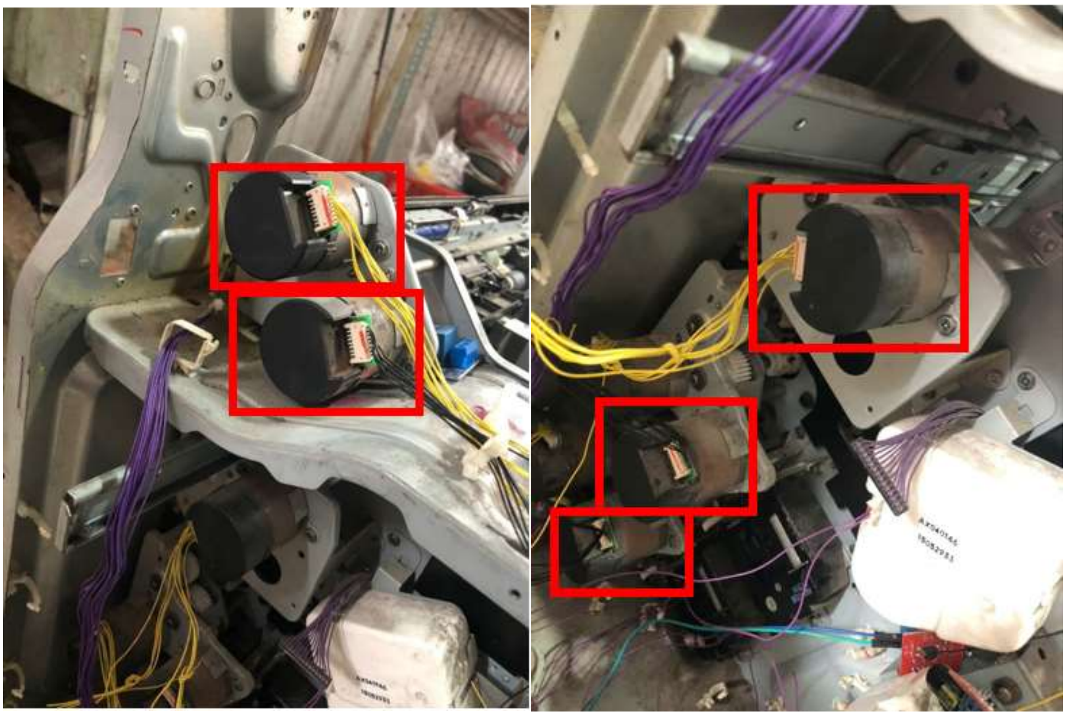


Figure 3.14 Actual installation positions of the motor

3.1.4.2. Genuine Ricoh motor D2236747 LCT lift motor

❖ General introduction

- Motor used for lifting paper tray 1:

- The Genuine Ricoh D2236747 (D223-6747) LCT Lift Motor is a specialized motor designed for use in Ricoh multifunction printers.
- It is specifically engineered to lift the Large Capacity Tray (LCT) in high-volume printing systems.
- This motor is commonly found in industrial printers and high-performance office printers requiring automated paper handling with high load capacity.

Figure 3.15 Tray lifting motor I (D2236747) [9]

❖ Main specifications

- Motor type: Brushed DC motor with integrated gear reduction.
- Power supply: 24V DC.
- High torque: Suitable for lifting large stacks of paper.
- Low rotational Speed: Optimized for high pulling force.
- Lifting capacity: Large Capacity Tray (LCT) with weight of up to hundreds of sheets of paper.

❖ Structure and working principle

- Structure:
  - Brushed DC motor with gear reducer.
  - Drive shaft directly connected to the paper lift tray.
  - Position sensor system for detecting tray position.
- Working principle:
  - When commanded by the controller, the motor rotates at low speed.
  - The drive shaft turns, lifting or lowering the paper tray.
  - Position sensors stop the motor when the tray reaches the appropriate position.

❖ Application in Ricoh multifunction printers

- The D2236747 motor is specifically designed for Ricoh printer models.
- Main functions:
  - Raise and lower the large capacity paper tray.
  - Maintain accurate paper positioning for automatic paper feeding.

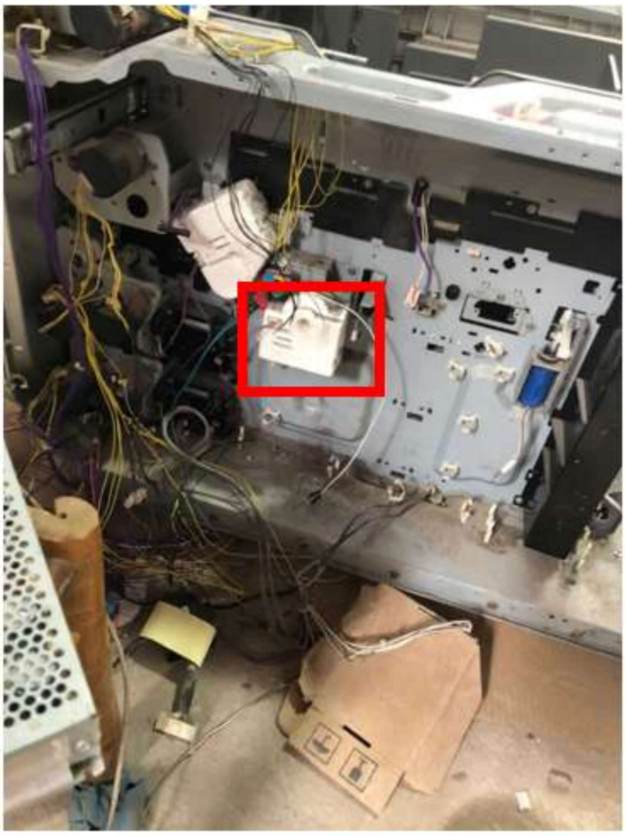


Figure 3.16 Actual installation positions of the motor

❖ Advantages and limitations

- Advantages:

- High torque, suitable for large capacity paper trays.
- Sturdy design with high durability.
- High precision thanks to integrated sensors.

• Limitations:

- High cost.
- Requires periodic maintenance due to use of brushed motor.
- Optimized for continuous operation in industrial printing environments.

3.1.4.3. Genuine Ricoh AX040119 (D331-2680) paper lift motor

❖ **General Introduction.**

- The Genuine Ricoh AX040119 (D331-2680) Paper Lift Motor is a specialized motor designed for Ricoh multifunction printers.
- Main function: Lifts and lowers the paper tray during the paper feeding process, ensuring the paper is pushed to the correct position before printing begins. It is used in Tray 2 and Tray 3

Figure 3.17 Tray lifting motors II and III (AX040119) [10]

❖ Main technical specifications.

- Motor Type: Brushed DC motor.
- Operating Voltage: 24V DC.
- Rotational Speed: Low (optimized for high torque).
- Torque Output: High, capable of lifting heavy paper loads.
- Supported Paper Weight: Suitable for both standard and photo paper.

❖ Construction and Operating Principle.

- Construction:
- Motor Body: Brushed DC motor.
- Gearbox: Integrated to reduce speed and increase torque.
- Drive Shaft: Directly connected to the paper tray lifting mechanism.
- Operating Principle:
- Upon receiving a command from the control board, the motor is supplied with 24V DC power.
- The motor rotates at low speed with high torque through the gearbox.
- The drive shaft turns and transfers motion to lift or lower the paper tray precisely.

❖ Application in Ricoh multifunction printers

- Primary functions:
- Lifts and stabilizes the paper tray during the printing process

- Ensures proper paper feeding position, helping prevent paper jams

Figure 3.18 Actual installation position of the motor

Figure 3.19 Internal structure of the motor

❖ Advantages and limitations

- Advantages:

- High torque output, suitable for lifting large volumes of paper
- Robust construction and high durability
- Optimized for continuous operation in industrial printing environments

- Limitations:

- Relatively high cost due to being a genuine Ricoh component
- Brushed motor may experience wear over time, requiring periodic maintenance

3.1.4.4. Driver module L298N

The L298N Module is a widely used integrated H-Bridge circuit for controlling DC motors and stepper motors. It is a suitable choice for embedded systems such as Arduino, Raspberry Pi, and other microcontroller platforms. With the ability to control two independent DC motors or one bipolar stepper motor, this module is commonly used in robotics, autonomous vehicles, automation equipment, and various DIY projects.

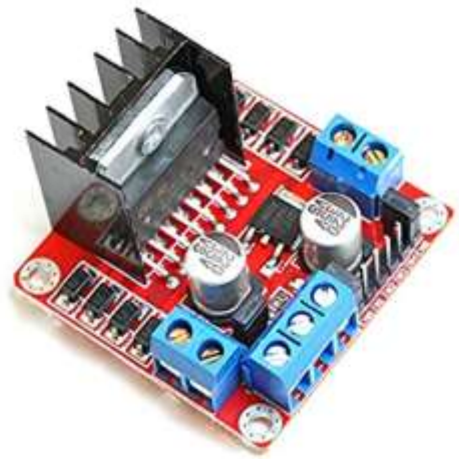


Figure 3.20 L289N module [11]

- **Structure and operating principle**

- The L298N is a power control IC capable of supplying up to 2A per channel and operating within a voltage range of 5V to 35V. The module typically includes a heat sink to dissipate heat during operation.
- Its H-bridge configuration allows for reversing the current through the motor, thereby controlling its rotation direction. Additionally, by using a PWM (Pulse Width Modulation) signal, the motor's rotation speed can be flexibly adjusted.

- Pin Functions of the Module

A standard L298N module includes the following key pins:

- IN1, IN2: Control the rotation direction of Motor A.
- IN3, IN4: Control the rotation direction of Motor B.
- ENA, ENB: Enable and control the speed of Motors A and B (can be connected to PWM pins).
- OUT1, OUT2: Output to Motor A.
- OUT3, OUT4: Output to Motor B.
- VMS (or 12V): Motor power supply input (typically 6V–12V or 24V).

- 5V: Logic voltage; can either power a microcontroller or be powered by it (if the jumper is attached).
- GND: Common ground.

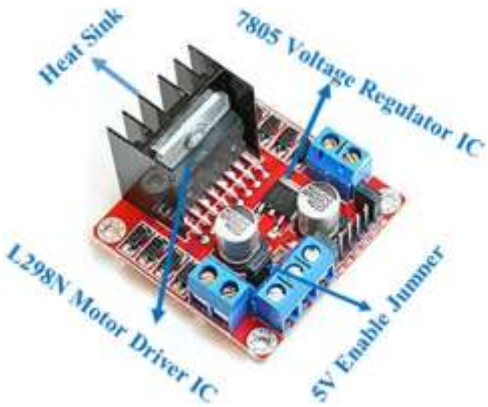


Figure 3.21 Main components of the L289N module [11]

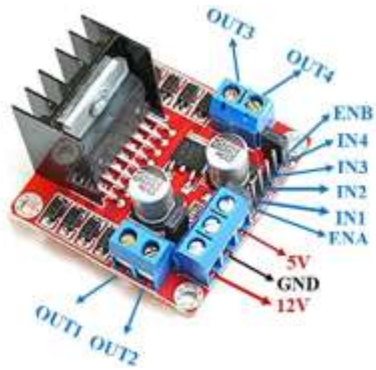


Figure 3.22 Pin diagram of the L289N module [11]

- Operating principle of dc motor control:

When a HIGH or LOW signal is applied to the IN1 and IN2 pins, it determines the direction of the current flowing through the motor, thereby controlling the rotation direction. The ENA or ENB pin, when supplied with a PWM signal, adjusts the motor's speed by varying the pulse width.

For example:

- IN1 = HIGH, IN2 = LOW → Motor rotates forward.
- IN1 = LOW, IN2 = HIGH → Motor rotates backward.
- IN1 = IN2 → Motor stops (coasts or brakes depending on the design).

3.1.4.5. Driver module BTS7960

In high-power DC motor control systems, selecting a suitable driver is crucial to ensure system performance and stability. The BTS7960 module is a high-power motor driver featuring a fully integrated H-bridge and the capability to handle large current loads. It is an ideal choice for projects requiring strong motor control such as heavy-load robots, electric model vehicles, mini conveyor belts, and more.

Figure 3.23 BTS7960 module [12]

- Overview of the BTS7960 Module

The BTS7960 module is designed based on the BTS7960 power IC from Infineon. This is a full-bridge driver capable of bidirectional control of DC motors with a maximum load current of up to 43A and an operating voltage range from 6V to 27V. Additionally, the module supports speed control via PWM (Pulse Width Modulation) signals and includes essential protection features such as over-temperature and over-current protection.

- Pin Configuration and Functions

The BTS7960 module includes the following interface pins, as shown in the table below:

Table 3.2 Functions of the Pins on the BTS7960 Module.

Pin Name	**Function Description**
RPWM	Controls speed/forward rotation via PWM signal
LPWM	Controls speed/reverse rotation via PWM signal
R_EN	Enables operation of the right half of the H-bridge
L_EN	Enables operation of the left half of the H-bridge
VCC	Supplies 5V logic voltage (from Arduino or other microcontroller)
GND	Common ground between logic power and motor power
B+	Positive terminal of motor power supply (6–27V)
B-	Ground (negative terminal) of the motor power supply
Motor+ / -	Connect directly to the two terminals of the DC motor

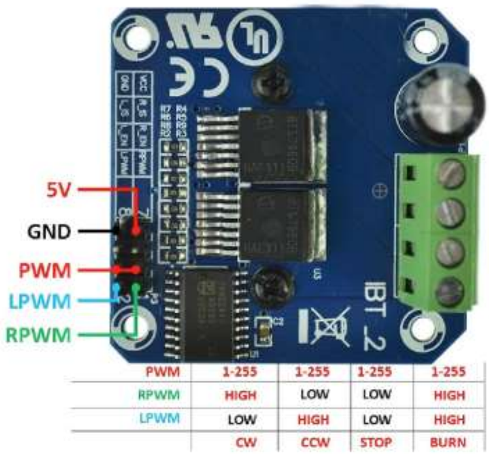


Figure 3.24 Pin diagram of the BTS7960 module [12]

- Operating principle:

- The BTS7960 operates based on the H-bridge principle, allowing the direction of current through a DC motor to be reversed. Motor control is carried out using two PWM signals, RPWM and LPWM, as follows:
- When RPWM = PWM, LPWM = 0 → the motor rotates forward
- When RPWM = 0, LPWM = PWM → the motor rotates in reverse
- When both = 0 → the motor stops (idle or non-driving state) Additionally, both R_EN and L_EN pins must be set to HIGH to activate the H-bridge circuit.

- Technical specifications:

Table 3.3 Technical Specifications of the BTS7960 Module

Specification	Value
Operating Voltage	6V – 27V
Continuous Drive Current	Up to 43A (recommended under 30A)
Peak Pulse Current	Up to 55A for short durations
Direction Control	Yes (bidirectional)
Speed Control	Yes (via PWM)
Protection Features	Over-temperature, short-circuit, over-current protection

3.1.4.6. Electromagnet

An electromagnet is a device capable of generating a magnetic field when an electric current flows through a coil of wire. Unlike a permanent magnet, the magnetism of an electromagnet can be turned on or off and adjusted according to the current, making it widely used in control systems, automation, and electromechanical applications.

Figure 3.25 Electromagnet [13]

- Structure

An electromagnet typically consists of the following main components:

- **Conductor coil (usually copper):** wound in multiple turns around a ferromagnetic core.
- **Ferromagnetic core (usually soft iron or low-carbon steel):** enhances the magnetic flux generated by the coil.
- **Power supply:** provides either direct current (DC) or alternating current (AC), commonly at 5V, 12V, or 24V depending on the application.
- When voltage is applied, current flows through the coil, generating a magnetic field that magnetizes the core, turning it into a temporary magnet.

- Operating Principle

The electromagnet operates based on Ampère’s law and the magnetic field induced by current flow:

- When electric current flows through the coil, it generates a magnetic field around the coil.
- This magnetic field magnetizes the ferromagnetic core, resulting in a much stronger overall magnetic effect compared to the coil alone. The strength of the magnetic field depends on:
- The current intensity (amperage).
- The number of coil turns (windings).
- The magnetic permeability of the core material.

When the power is cut off, the magnetic field collapses, and the core returns to a non-magnetized (or weakly magnetized) state.

- Key technical parameters

Table 3.4 Technical specifications of the electromagnet.

Specification	Description
Operating Voltage	Typically 5V, 12V, or 24V DC
Current Consumption	Depends on size, ranges from a few hundred mA to several A
Pull Force	Holding force for metal objects, measured in Newtons or kilograms
Dimensions	Varies by application (small, medium, industrial sizes)
Response Time	Usually from a few milliseconds to several tens of milliseconds

• Practical applications:

Electromagnets are used in many systems:

- Electronic locks.
- Relays and contactors.
- Electromagnetic lifters for metal handling.
- Ferrous metal sorting systems and recycling machines.
- Robotic pick-and-place systems.
- Actuators in printers and photocopiers.
- Magnetic latches in automatic doors.

Figure 3.26 Electromagnet in the paper picking assembly

- Advantages and disadvantages:

Advantages:

- Can be turned on and off quickly.
- Strong magnetic field when needed.
- Easy to adjust the pull force by changing the current.

Disadvantages:

- Requires continuous power supply → electrical consumption.
- Generates heat during prolonged operation.
- Magnetism is lost immediately when power is cut (unlike permanent magnets).

- Application in the System:

- The electromagnet is placed in the paper pickup module, connected to the paper pickup mechanism, helping to raise and lower the part that directly contacts the paper to extract it.

3.1.5. Communication module

3.1.5.1. UART TTL communication between MEGA 2560 and UNO

- Introduction to UART:

UART (Universal Asynchronous Receiver/Transmitter) is a widely used asynchronous serial communication standard in microcontrollers and embedded systems for transmitting and receiving data between two devices.

- Asynchronous: It does not use a separate clock signal to synchronize data.
- Data is transmitted bit by bit in sequence, beginning with a start bit, followed by the data bits, optionally a parity bit (for error checking), and ending with a stop bit.

UART TTL refers to UART communication using standard TTL logic voltage levels (typically 0V for logic 0 and 3.3V or 5V for logic 1). It is suitable for short-distance communication, usually within a few meters.

- Transmission mode used:

- Full-duplex, meaning data can be transmitted in both directions simultaneously:
- Device A transmits (TX) data to Device B’s receiver (RX).
- At the same time, Device B can transmit (TX) data to Device A’s receiver (RX).

Therefore, a UART full-duplex system uses two separate signal lines: one TX and one RX.

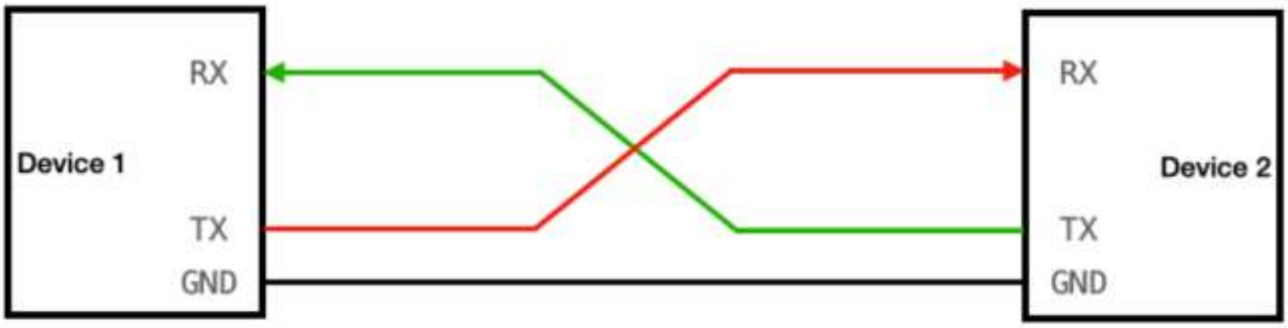


Figure 3.27 Hardware connection model of the system's UART TTL

Figure 3.28 UART data transmission frame

- **Start Bit**

The UART data transmission line is typically held at a high voltage level when idle (no data being transmitted). To initiate a data transmission, the transmitting UART pulls the line low for one (1) clock cycle. When the receiving UART detects this voltage transition from high to low, it begins reading the bits of the data frame at the baud rate frequency.

- **Data Frame**

The data frame contains the actual data being transmitted. It can be 5 to 8 bits long if a parity bit is used. Without a parity bit, the data frame can be up to 9 bits. In most cases, data is transmitted with the **least significant bit (LSB)** first.

• **Parity Bits**

Parity indicates the evenness or oddness of a number. The parity bit allows the UART to detect whether any data was altered during transmission. Bit errors can occur due to electromagnetic interference, mismatched baud rates, or long-distance transmission.
After the receiving UART reads the data frame, it counts the number of 1s and determines whether the total is even or odd:

- If the parity bit is 0 (even parity), the number of 1s in the data frame should be even.
- If the parity bit is 1 (odd parity), the number of 1s in the data frame should be odd.

If the parity bit matches the expected parity, the UART assumes the data is error-free. If not—i.e., the parity bit is 0 but the count is odd, or the parity bit is 1 but the count is even—then the UART detects a transmission error.

• **Stop Bits**

To signal the end of the data packet, the transmitting UART pulls the data line from low voltage back to high voltage for a duration of one (1) to two (2) bits.

3.1.5.2. I2C communication between MEGA 2560 (Master) and PCF8575 (Slave).

- **Overview of I2C**

- I2C is a serial communication interface that supports multi-master and multi-slave configurations, used to connect microcontrollers with sensors, memory chips, displays, and other peripheral ICs.
- Developed by Philips (now NXP) in the 1980s.
- Purpose: to transfer data between chips on the same circuit board using very few signal lines.

- **Basic characteristics**

- Two main communication lines:
  - **SDA (Serial Data Line):** bidirectional data line.
  - **SCL (Serial Clock Line):** clock signal line generated by the master device.
- Both lines require pull-up resistors, typically to 3.3V or 5V.
- Common communication speeds:
  - Standard mode: 100 kbps
  - Fast mode: 400 kbps
  - Fast mode plus: 1 Mbps
  - High-speed mode: 3.4 Mbps (less commonly used)

- **Communication structure**

- I2C uses a master-slave model:
  - Master: controls communication, generates the SCL clock, and initiates data transfer.
  - Slave: responds to master commands.
- A single I2C bus can support multiple masters and multiple slaves.

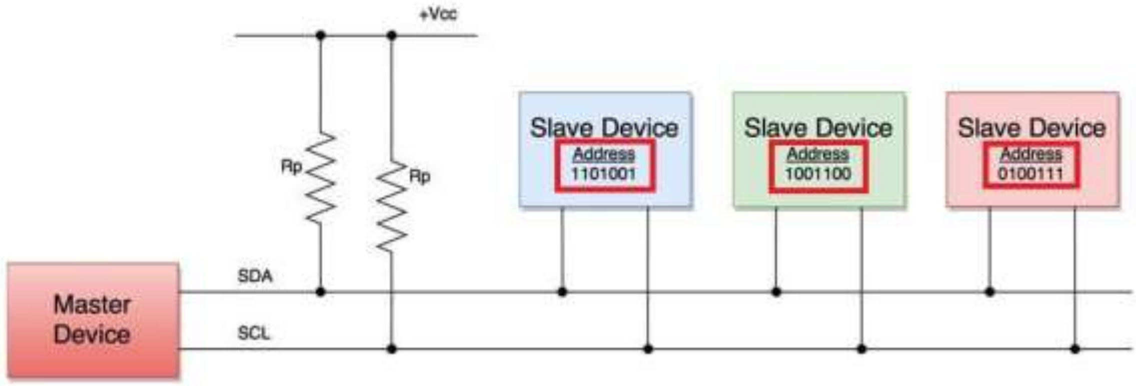


Figure 3.29 Physical connection of I2C

- The resistor R_P is used as a pull-up resistor to keep the voltage on the SDA and SCL lines at a high level when the bus is idle, preventing signal interference, read/write errors, and device recognition failures.

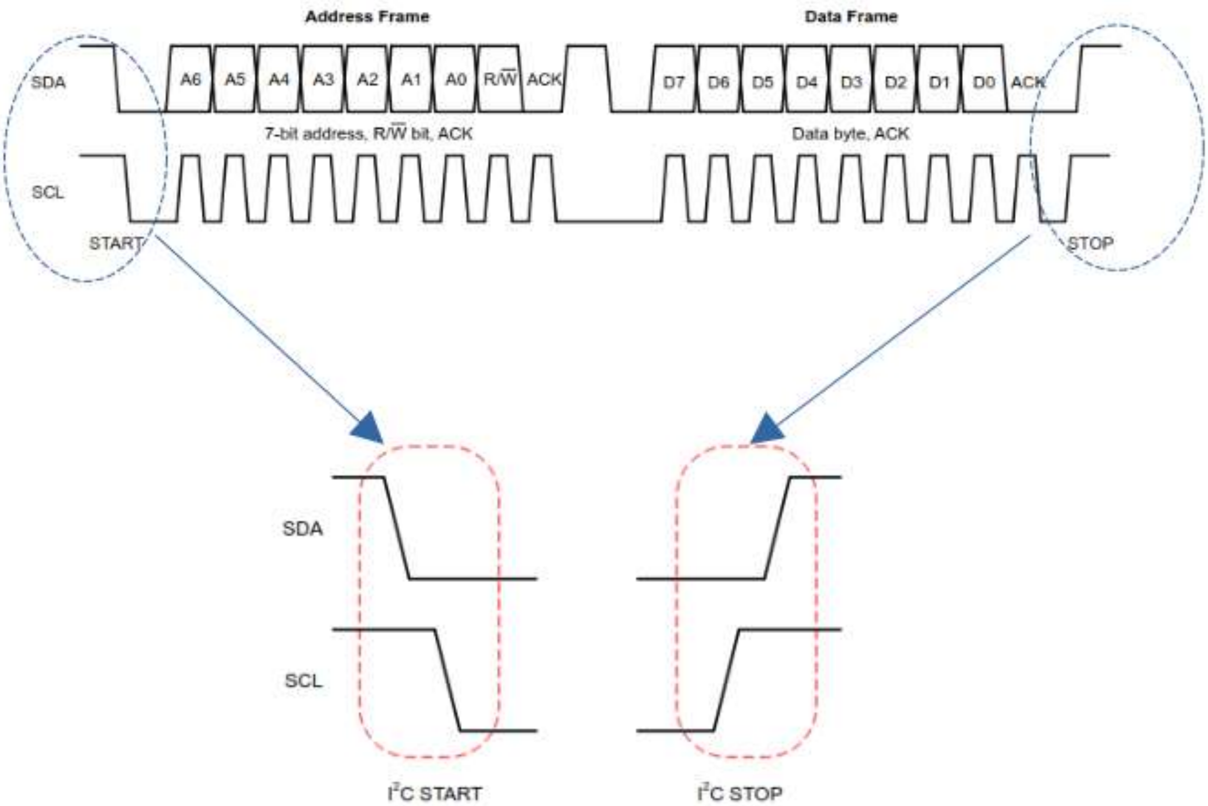


Figure 3.30 Transmission frame and data transfer process of the I2C protocol

When a device wants to communicate with another device on the bus, the process typically proceeds as follows:

- The I2C bus must be in a "free" state, meaning both SDA and SCL lines are "High".

- A "START signal" is generated to take control of the bus. All other devices on the bus will "LISTEN" to determine whether they are the target of the communication.
- Clock pulses are provided on the SCL line. This clock is shared by all devices on the bus. Data on the SDA line is "latched" when the SCL line transitions from low to high.
- The address of the target device is sent onto the bus.
- A bit is sent to specify whether data will be SENT to or RECEIVED from the target device.
- An ACK bit is requested from the target device to confirm it is ready to communicate.
- After receiving the ACK, data exchange begins.
- After every 8-bit data transfer, the transmitter requests an ACK bit from the receiver to confirm successful communication.
- After data exchange is complete, the bus is released and a "STOP signal" is generated to end the communication process.

- **Overview diagram of communication methods used in the system.**

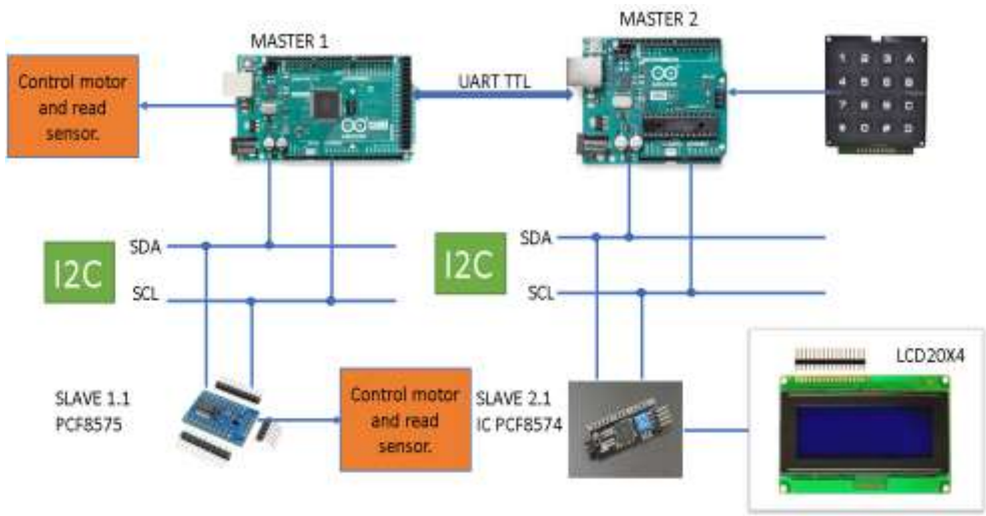


Figure 3.31 Overview of the communication methods used

3.1.6. Software module

This is the section that handles algorithms and logic. The software contains programs to process sensor signals and perform computations for controlling actuators.

- The software used in this system includes:

- Arduino IDE software: This IDE is used to communicate with the Arduino Atmega2560 via the USB port and upload algorithm code to the Arduino.

Figure 3.32 Arduino IDE software [14]

- SolidWorks software is used to design and draw mechanical components, as well as to analyze forces within the system.

Figure 3.33 SolidWorks software [15]

- Proteus software is used to design control circuits and printed circuit boards (PCBs).

Figure 3.34 Proteus software [16]

3.1.7. Processor module:

The processor is responsible for calculating information and processing data from the software module and the measurement module.

The Arduino ATmega 2560 receives signals from the interface module or control screen, limit switches, and sensors to control other actuators and motors.

❖ **Arduino atmega 2560**

Figure 3.35 Arduino Atmega 2560 [17]

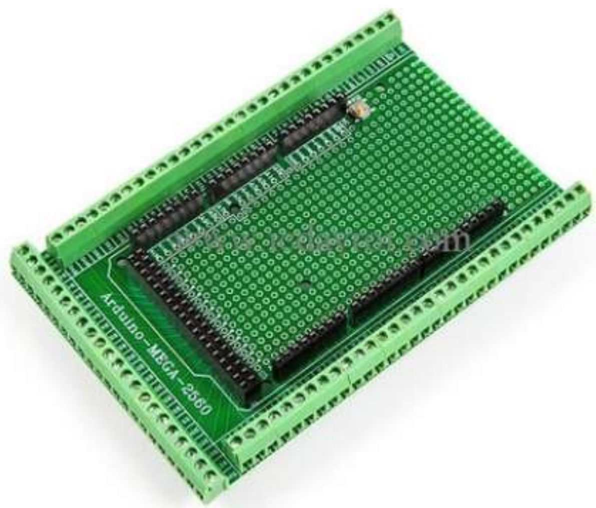


Figure 3.36 Arduino Atmega 2560 expansion board [18]

- Technical Specifications

- Processing Speed: Up to 16 MIPS (Million Instructions Per Second) at 16 MHz
- Flash Memory: 256 KB, with 8 KB reserved for the bootloader
- Peripheral Interfaces:
- USART: 4 UART/USART modules for serial communication
- I2C (TWI): I²C/TWI (Two Wire Interface) communication interface
- USB: No native USB support integrated
- Control and Timing Features:
- PWM: 4 channels of 8-bit PWM and 12 channels of 16-bit PWM

❖ **Introduction to the PCF8575 Module**

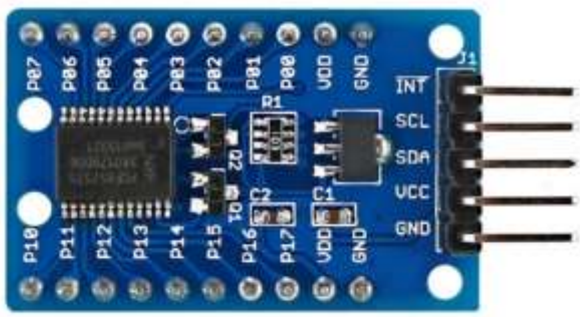


Figure 3.37 Pin diagram of the PCF8575 [19]

- **Main Functions of the PCF8575 Module:**

Module PCF8575 is an I/O expander IC using I²C communication, which helps increase the number of input/output pins for microcontrollers (such as Arduino, ESP32, STM32...) through I²C interface (2 wires: SDA, SCL).

- Main Specifications:

- Main IC: PCF8575 (by NXP or Texas Instruments).
- Number of I/O pins: 16 pins (P0 → P15).
- Communication interface: I²C (SDA, SCL).
- Operating voltage: 2.5V ~ 5.5V.
- Maximum current per I/O pin: ~25mA.
- I²C address range: 0x20 to 0x27 (configurable via A0, A1, A2 jumpers).

- Pin Configuration (on typical module):

- VCC, GND: Power supply for the module.
- SDA, SCL: I²C communication lines to the microcontroller.
- P0 – P15: 16 extended I/O pins.
- A0, A1, A2: I²C address selection (via jumper or connected to GND/VCC).

3.1.8. Interface module

The Interface Module serves as the bridge between the user and the machine system. It is where the operator inputs control commands (such as selecting mode, number of sets, etc.) and also receives feedback from the machine (operational status, error alerts...). This module plays a key role in visualizing the machine's operation and helps users interact with the machine easily and accurately**.**

- Components:

LCD Display:

- Provides direct communication between the user and the machine. Detailed functions include:

- Displaying control menus (mode selection, tray selection, number of sets, etc.)
- Displaying device status (Ready, Operating, Initializing, Error)

Physical Buttons or Keypad Matrix:

- Enables basic operations even if the screen is unresponsive or faulty. Common buttons include:
- Start / Stop: Start or stop the current operation
- Reset: Reboot the system
- Numeric Keys: For selecting mode, number of sets, and tray

Status LEDs and Buzzer:

- Indicate system status through color changes, blinking patterns, or sound alarms. For example:
- Paper out
- Paper jam

Functions:

- Receive Control Commands:

Takes user inputs such as mode, paper size, number of sets, tray selection, etc., and sends them to the Arduino for execution.

- Display System Status:

Real-time visualization of the entire system's status (e.g., sheet count, set count).

- Error Alerts and Troubleshooting Guidance:

When an error is detected by sensors, the system displays error codes or specific icons (e.g., paper jam in Tray I, out of paper).

3.2. Wiring diagram and electrical circuit design for fabrication

3.2.1. Power supply selection

3.2.1.1. Power supply selection for the motor

- **Brushless DC Motor-BLDC:**

Figure 3.38 Brushless DC motor specifications from the manufacturer

Accordingly, this motor has a power output of 28.1 W.

- **2-phase stepper motor:**

Figure 3.39 Stepper motor specifications from the manufacturer

Accordingly:

- Rated voltage: 7.7 V
- Current per phase: 1.1 A

$$P = U * I = 2 * 7.7 * 1.1 = 16.94(W)$$

- **Brushless DC Motor with Integrated Driver and Encoder - Model: ID-549XW-5031:**

Figure 3.40 Brushless DC Motor with Integrated Driver and Encoder [7]

According to the manufacturer, this motor has a power output of 20 W.

When the system is operating, the maximum number of these motors running simultaneously is 4.

⇒ The total power supplied for this type of motor is:

$$P_t = n * P = 4 * 20 = 80(W)$$

- **Tray lifting motor:**

Figure 3.41 DC motor for tray lifting [9]

Since the current specifications for this motor are not available, a test was conducted by lifting 100 sheets of paper, and the measured current consumption was 0.7 A.

⇒ The power supplied for this type of motor is:

$$P = U * I = 0.2 * 24 = 4.8 \ (W)$$

- **Paper suction fan on the conveyor:**

Figure 3.42 Paper suction fan

According to the manufacturer, this fan has a current rating of 0.17 A, and the conveyor system uses two of them to hold the paper in place.

⇒ The power supplied for this type of fan is:

$$P = n * U * I = 2 * 0.17 * 24 = 8.16 \ (W)$$

- **Electromagnet:**

Figure 3.43 Electromagnet

According to the manufacturer, this electromagnet has a voltage rating of 24 V and a coil resistance of 96 Ω.

⇒ The operating current through the electromagnet is:

$$I = \frac{U}{R} = \frac{24}{96} = 0.25(A)$$

⇒ The power supplied for this electromagnet is:

$$P = n * U * I = 0.25 * 24 = 6 \ (W)$$

- **Power supply selection:**

When the system operates under the condition where multiple motors run simultaneously, the following motors are active:

- Brushless DC Motor (BLDC): 28.1 W
- 2-phase stepper motor: 16.94 W
- 4 Brushless DC motors with integrated driver and encoder: 80 W
- Tray lifting motor: 4.8 W
- 2 suction fans: 8.16 W

⇒ The maximum power that the power supply circuit needs to deliver is:

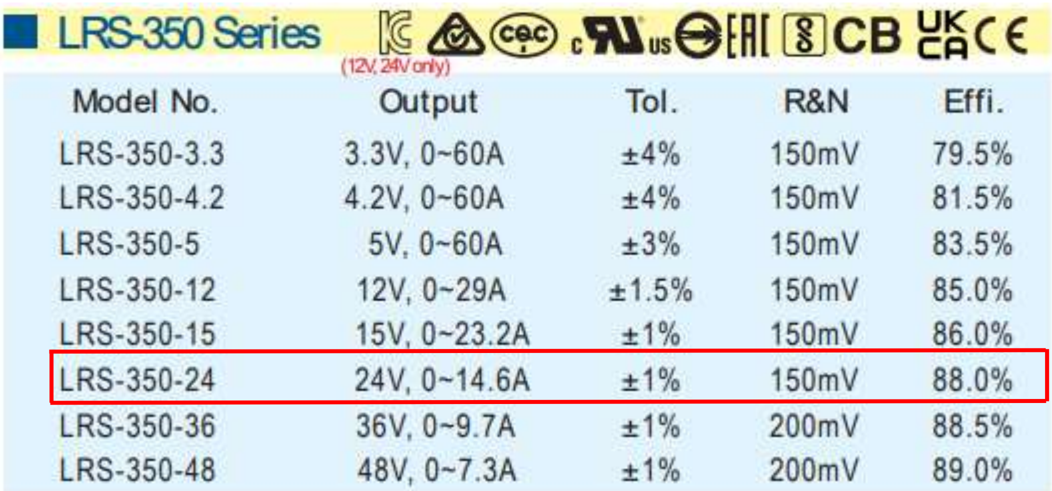
$$P_{total} = 28.1 + 16.94 + 80 + 4.8 + 9.16 = 139\ (W)$$

To prevent overload and voltage drop, the power supply is selected with a power rating of:

$$P_{chose} > \eta * P_{total}$$

We select a safety factor $\eta = 2$

$$\Rightarrow P_{chose} > 2 * 139 = 278\ (W)$$


LRS-350 Series
(12V, 24V only)

Model No.	Output	Tol.	R&N	Effi.
LRS-350-3.3	3.3V, 0~60A	±4%	150mV	79.5%
LRS-350-4.2	4.2V, 0~60A	±4%	150mV	81.5%
LRS-350-5	5V, 0~60A	±3%	150mV	83.5%
LRS-350-12	12V, 0~29A	±1.5%	150mV	85.0%
LRS-350-15	15V, 0~23.2A	±1%	150mV	86.0%
LRS-350-24	24V, 0~14.6A	±1%	150mV	88.0%
LRS-350-36	36V, 0~9.7A	±1%	200mV	88.5%
LRS-350-48	48V, 0~7.3A	±1%	200mV	89.0%

Figure 3.44 LRS-350 Series Power Supply Selection Specifications [8]

From that, we can select the power supply with model **LRS-350-24**, which has a maximum power output of **350 W**.

3.2.1.2. Selection of power supply for sensors

The system uses 18 photoelectric sensors, each consuming approximately 20 mA during operation.

Thus, the total current consumption of the entire sensor system is:

$I = 20 \text{ mA} \times 18 \approx 360 \text{ mA}$

We select the LM2596 power module, which provides an output voltage of 5 V and a maximum output current of 3 A, which is sufficient to supply the sensor system.

3.2.2. Overview of the electrical wiring diagram

- General Diagram of Components Used and Pin Configuration:

Table 3.5: Sensors used

Sensor	Qty	Wires	Pull-up Resistor	Power Connections	Output to Arduino	Input from Arduino
Optical sensor (AW010151)	2	3	Yes	VCC → 5 V, GND → GND	Connected	No
Optical sensor (AW020234)	2	3	No	VCC → 5 V, GND → GND	Connected	No
Push-button (SW1AG-550)	1	2	Yes	GND → GND	Connected	No

Table 3.6: Actuating mechanisms used

Actuator	Quantity	Wires	Power Connections	Output to Arduino	Input from Arduino
Lifting motor	1	2	VCC → 24V, GND → GND	No	Connected
Paper feeding motor	2	5	VCC → 24V, GND → GND	No	Via L298N
Electromagnet	1	2	VCC → 24V, GND → GND	No	Via BTS7960

Table 3.7: Control modules used

Module	Quantity	Wires	Power Connections	Output	Input from Arduino
BTS7960	3	10	VCC → 24V, GND → GND	Electromagnet	No
L298N	2	12	VCC → 24V, GND → GND	Lifting Motor	No
Arduino	1	4	VCC → 24V, GND → GND	To Arduino itself	No

- **Circuit Wiring Diagram:**

- Tray I :

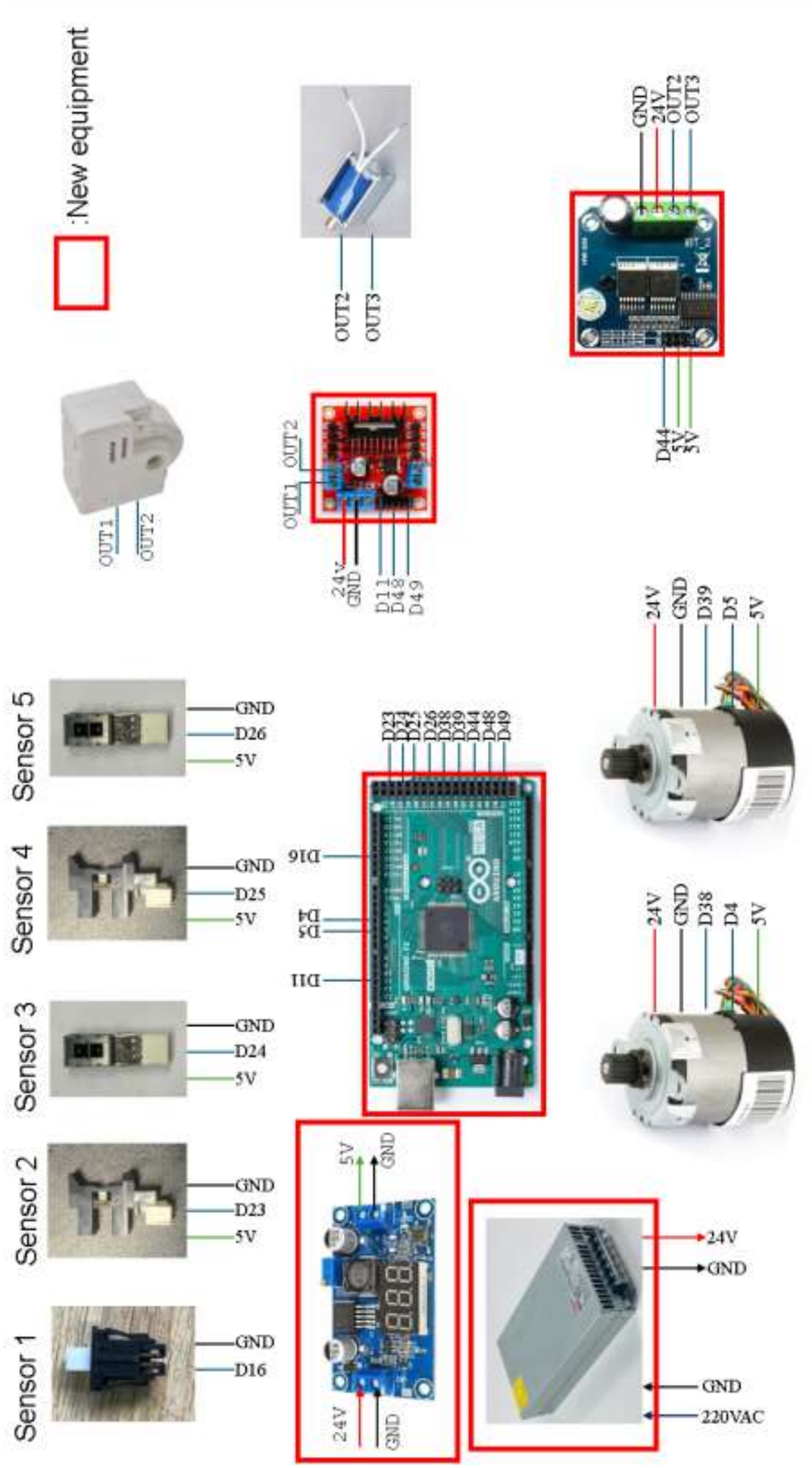


Figure 3.45 Wiring diagram of tray I

- Tray II :

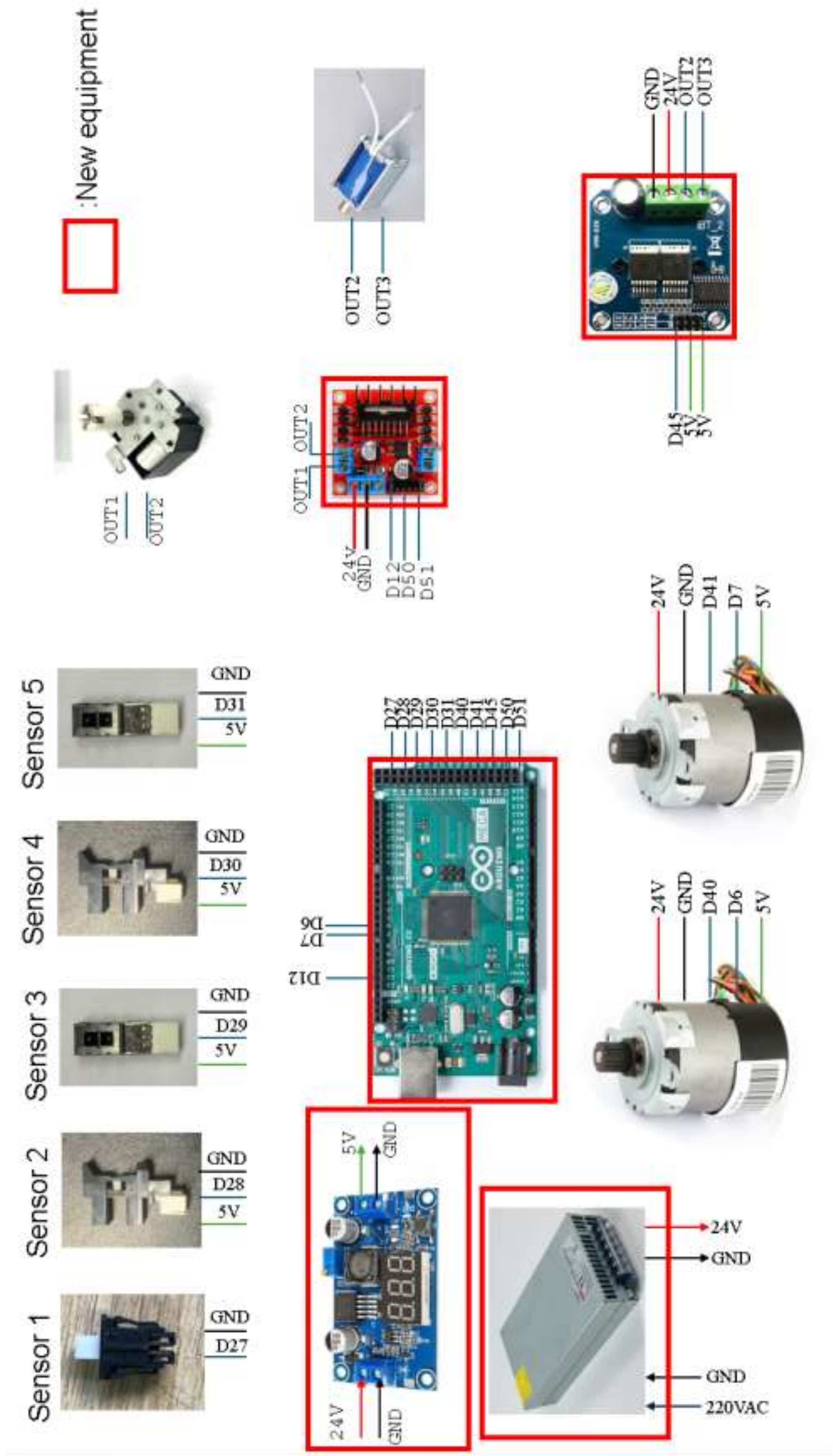


Figure 3.46 Wiring diagram of tray II

- Tray III :

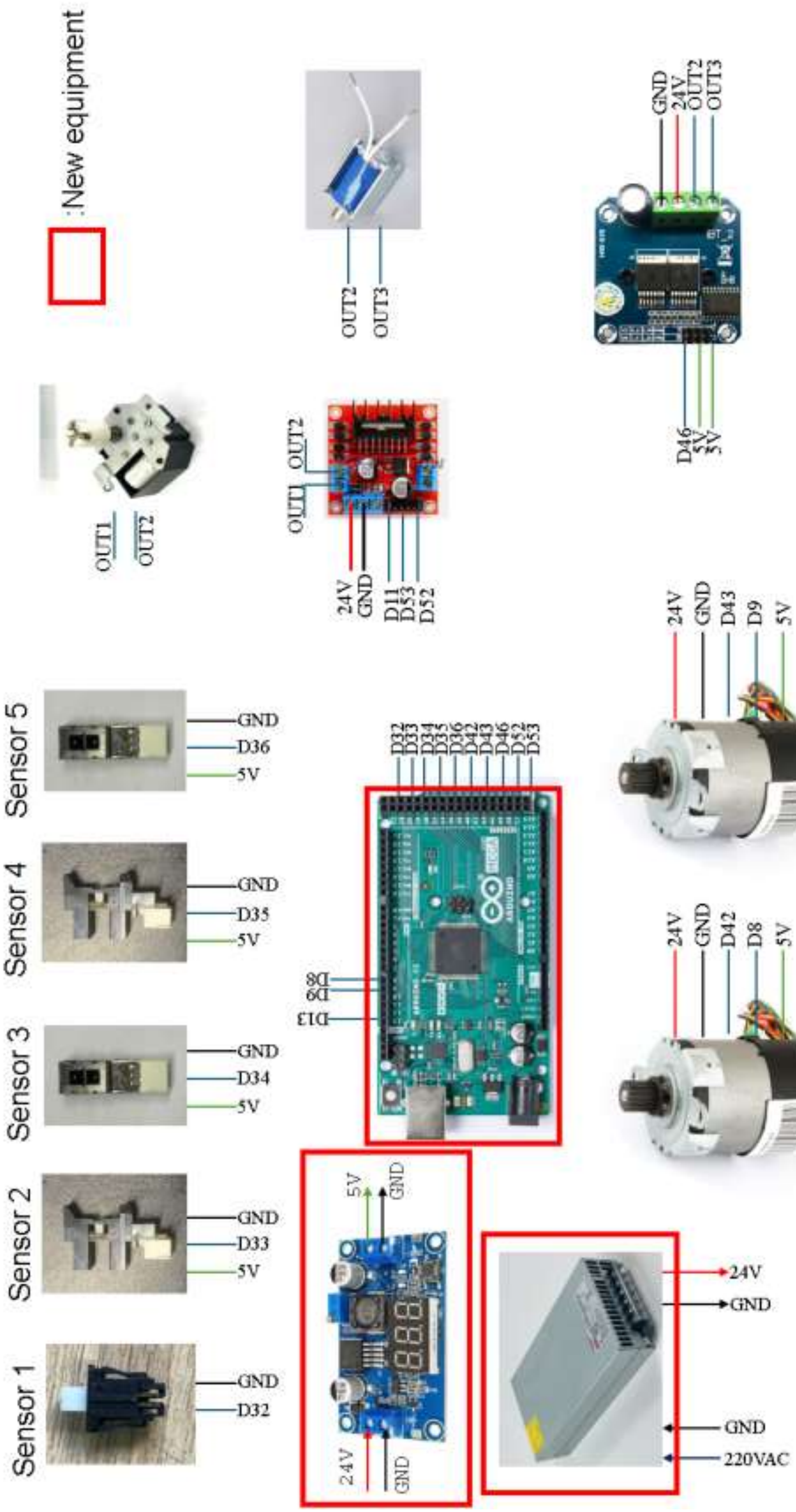


Figure 3.47 Wiring diagram of tray III

- Assembly B.

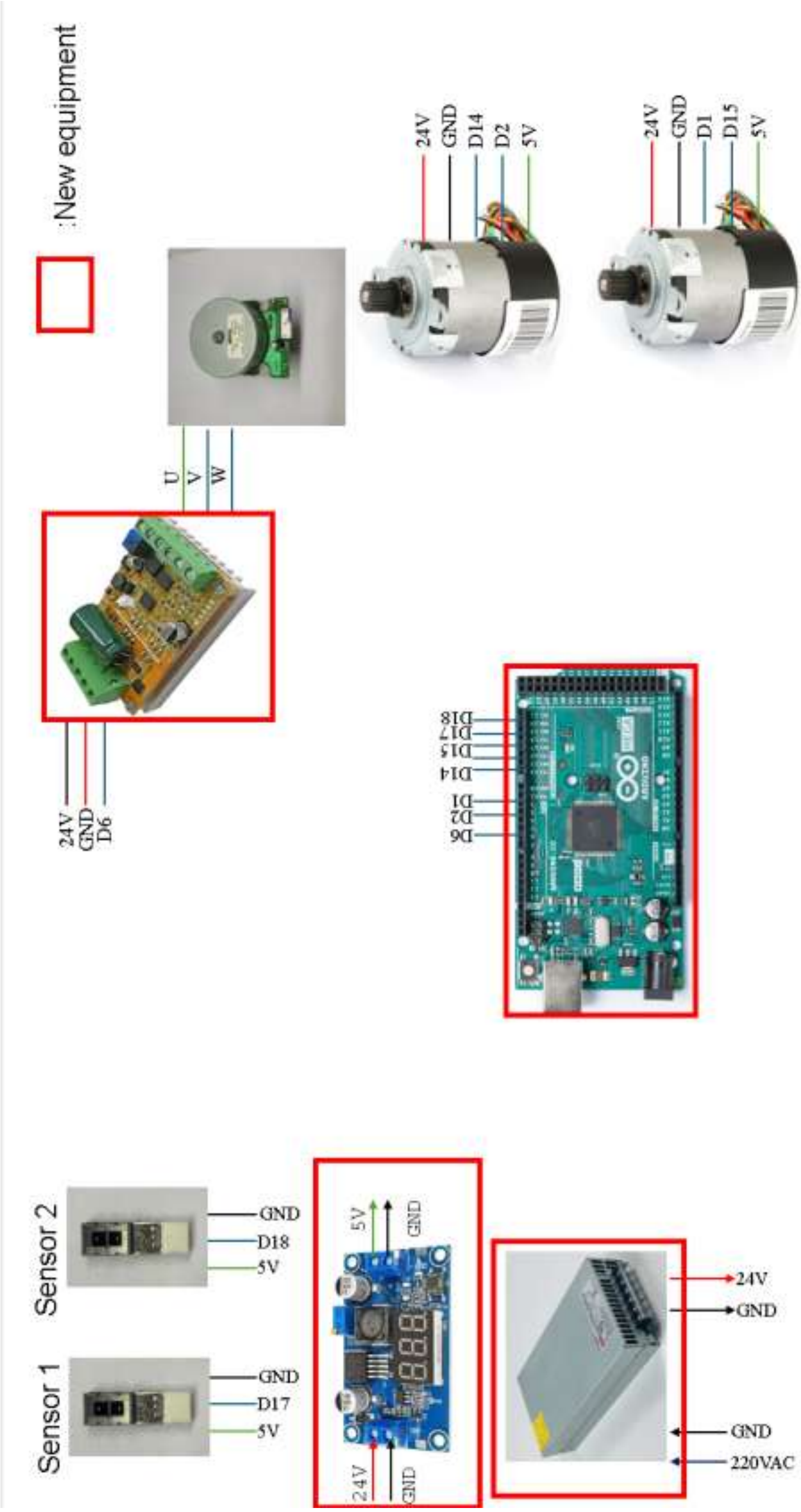


Figure 3.48 Wiring diagram of assembly B

- Power **:**

In any electrical or electronic system—especially in highly automated devices—the power supply plays a crucial role in ensuring that all modules operate reliably, synchronously, and safely. Designing and managing an efficient power supply system not only enhances overall system reliability but also minimizes the risk of component damage, short circuits, or failures during operation.

- Characteristics of Input Power Supply:

- The machine operates using alternating current (AC) from the household power grid, with the following typical specifications:
- Voltage: 220V–240V AC, frequency 50/60 Hz (compatible with the power standards in Vietnam and many other countries), then converted to 110V/120V AC.
- Current consumption: ranges from 3A to 15A, depending on the power and configuration of the machine (e.g., small office machines vs. high-speed industrial models).
- This AC power is then processed through multiple conversion stages to supply suitable voltage levels for each functional module within the device.

- Internal Power Processing Components:

- Main power supply: receives AC power and delivers DC power to main circuits and motors.
- Switching power supply: converts AC power into stable DC voltages for control circuits (such as 5V, 12V, 24V...).
- Protection circuits: disconnect the power supply in case of faults such as overload, overheating, or short circuits.

- Common Internal DC Voltage Levels:

- 5VDC: supplied to microcontrollers (e.g., Arduino) and sensors.
- 24VDC: supplied to the paper feed motors, tray lifting mechanism, electromagnets, and conveyor belts.

3.2.3. Electrical circuit design and fabrication

- **Circuit design on proteus:**

The schematic diagram of the main circuit is shown in Figure 3.42

The main circuit consists of 7 blocks:

- Power block: Provides power for the entire circuit.
- Tray B assembly
- Tray I assembly

- Tray II assembly
- Tray III assembly
- Push-button block
- High-load block
- Communication (interface) block

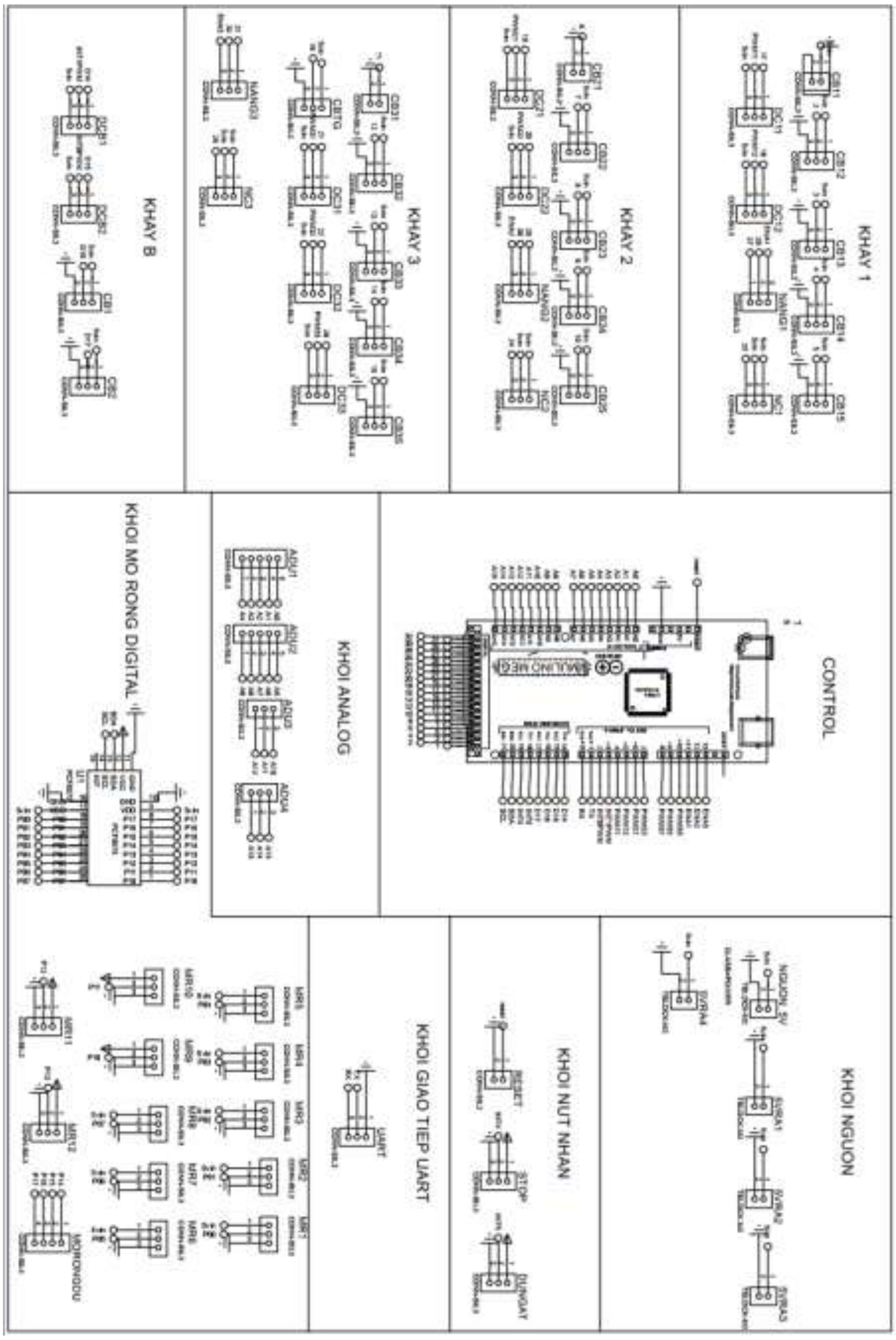


Figure 3.49 Schematic diagram of the main circuit

- **PCB layout:**

- The main circuit layout is shown in Figure 3.43

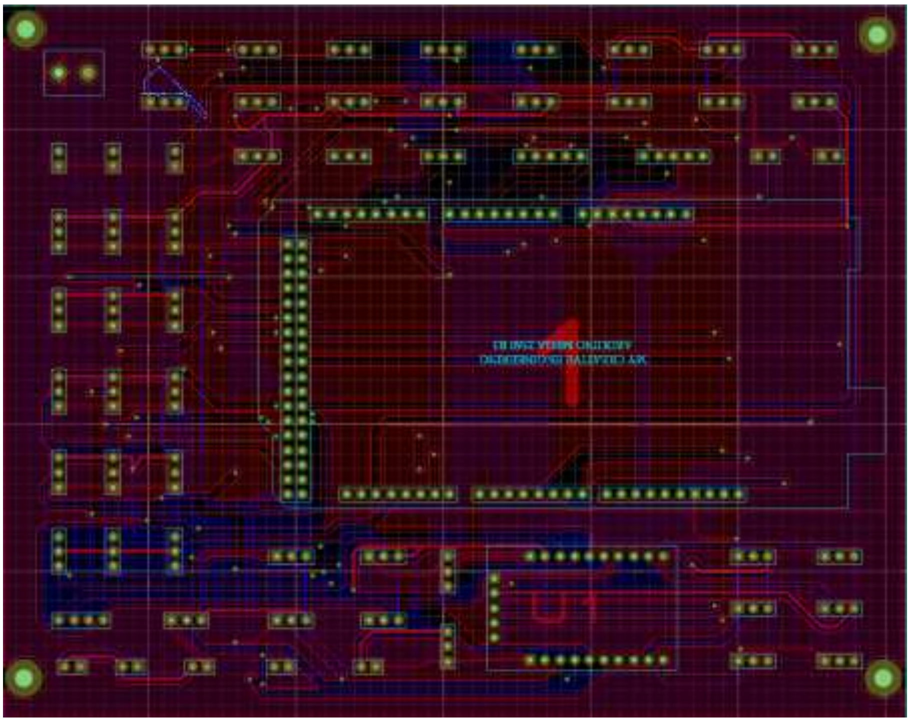


Figure 3.50 Layout of the main circuit

- **Practical circuit design.**

- The actual peripheral circuit is shown in Figure 3.44

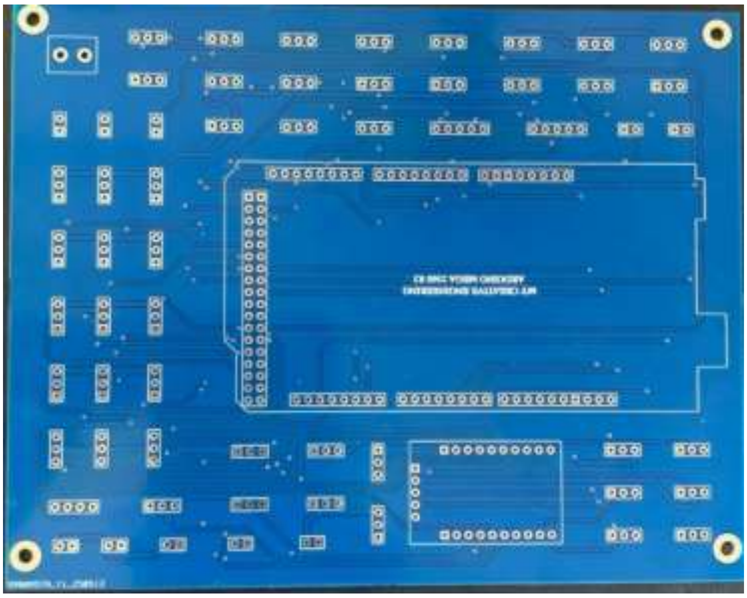


Figure 3.51 Actual peripheral circuit

- The circuit after soldering and assembling the components is shown in Figure 3.45 and figure 3.46.

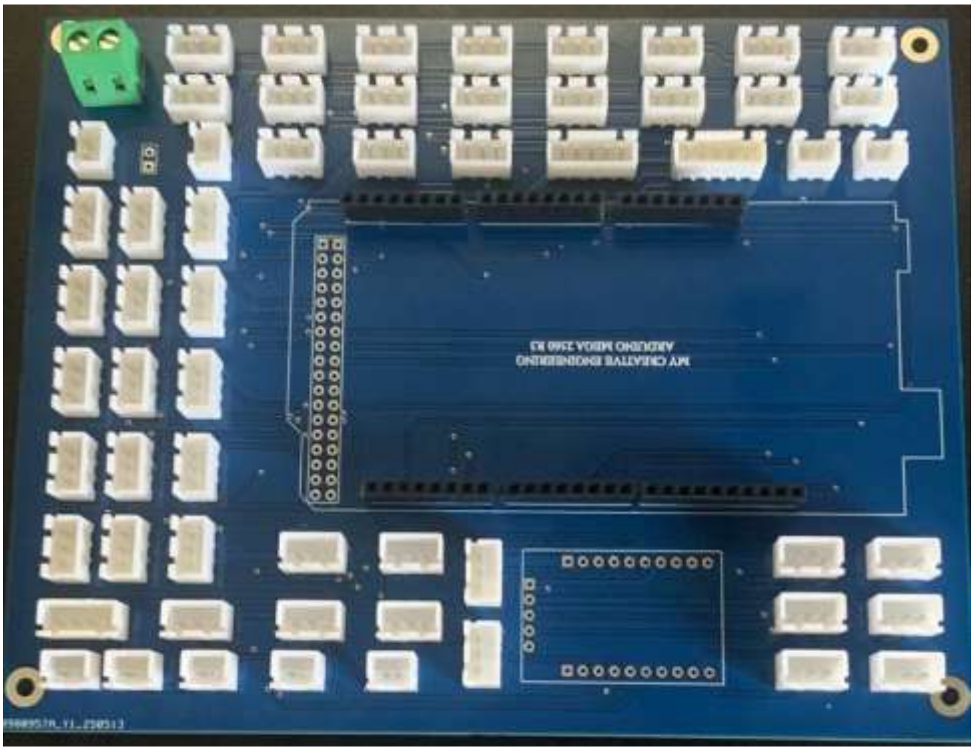


Figure 3.52 Front side of the circuit board

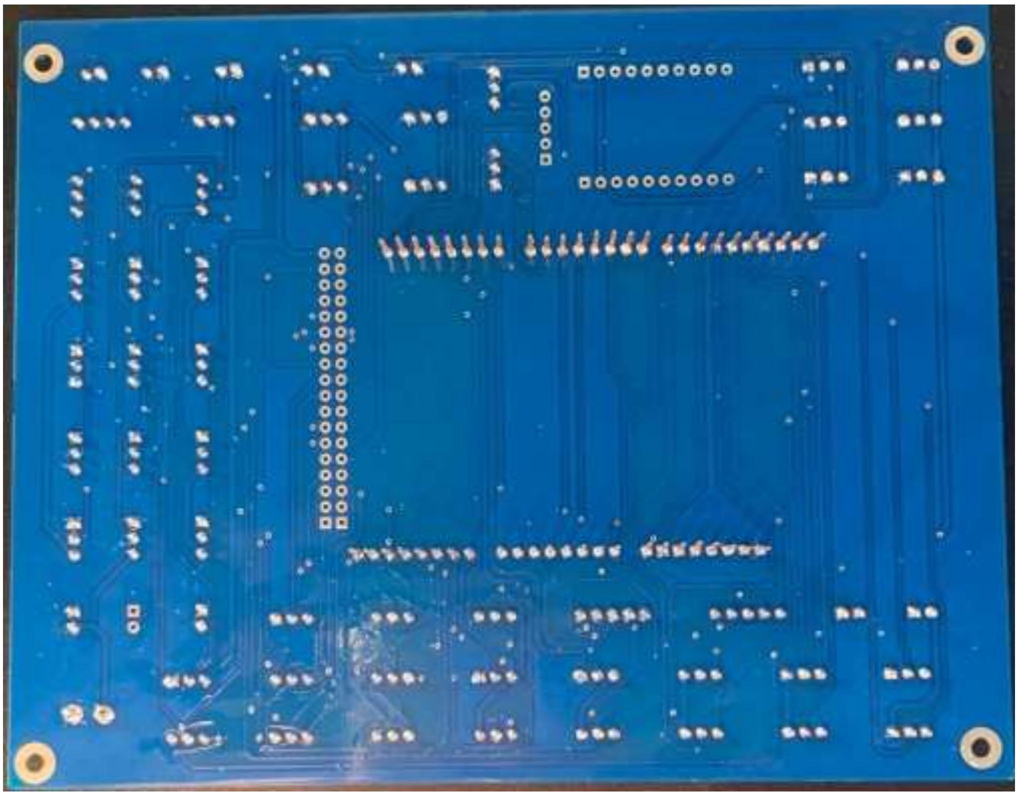


Figure 3.53 Back side of the circuit board

CHAPTER 4: CONTROL SYSTEM

4.1. Input requirements

The system will operate in two modes depending on user requirements.

Mode 1: Select the order of tray operation => Set the number of sheets to 1 for each tray => Set the number of sets to be processed => Execute the creasing or perforating mode.

Mode 2: Select the active tray => Select the order of tray operation => Set the number of sheets for each tray => Set the number of sets to be processed => Execute the creasing or perforating mode.

his mode is used for creating booklets, notebooks, books, etc.

- Based on these requirements, automatic and custom modes can be created depending on the desired operation to support both modes.
- From these requirements, we build an algorithm describing the operating principle for both modes.

4.2. Algorithm and control principle

- The main program is designed to select operating modes and input information such as tray order, number of sets, etc.

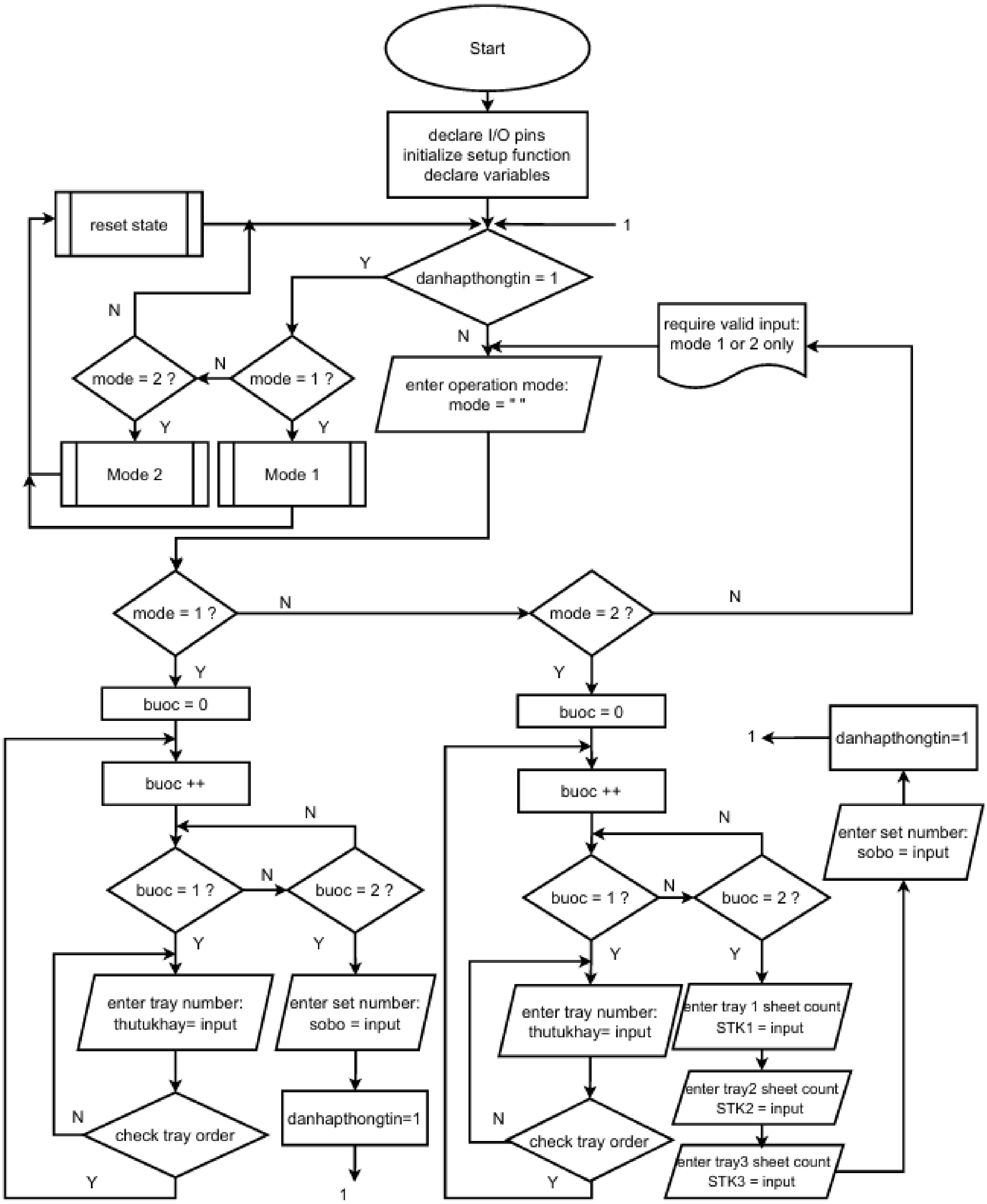


Figure 4.1 Main program algorithm

- After entering the data in the main program, the process will proceed to subprogram mode 1 or mode 2.

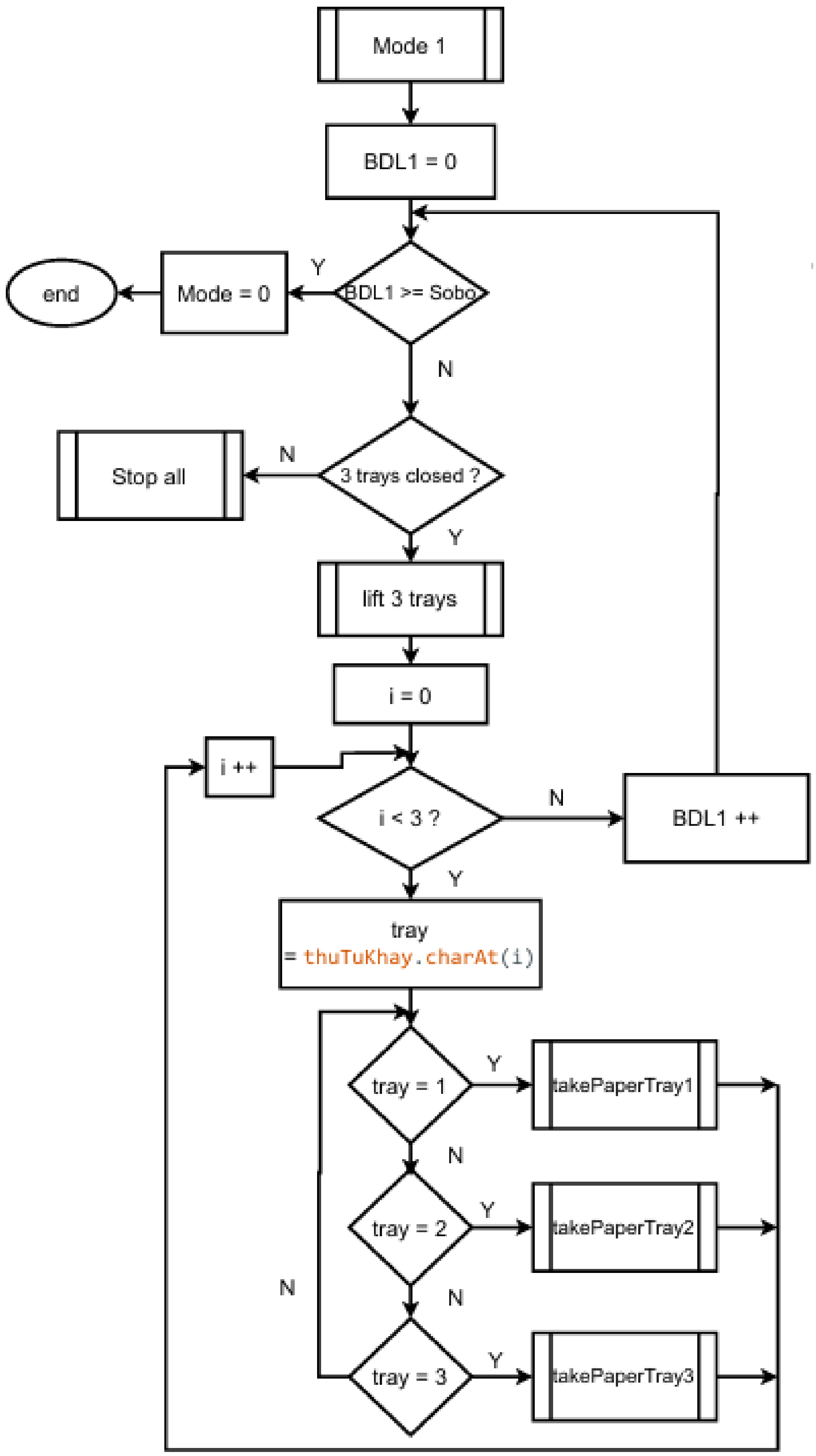


Figure 4.2 Subroutine algorithm for mode 1

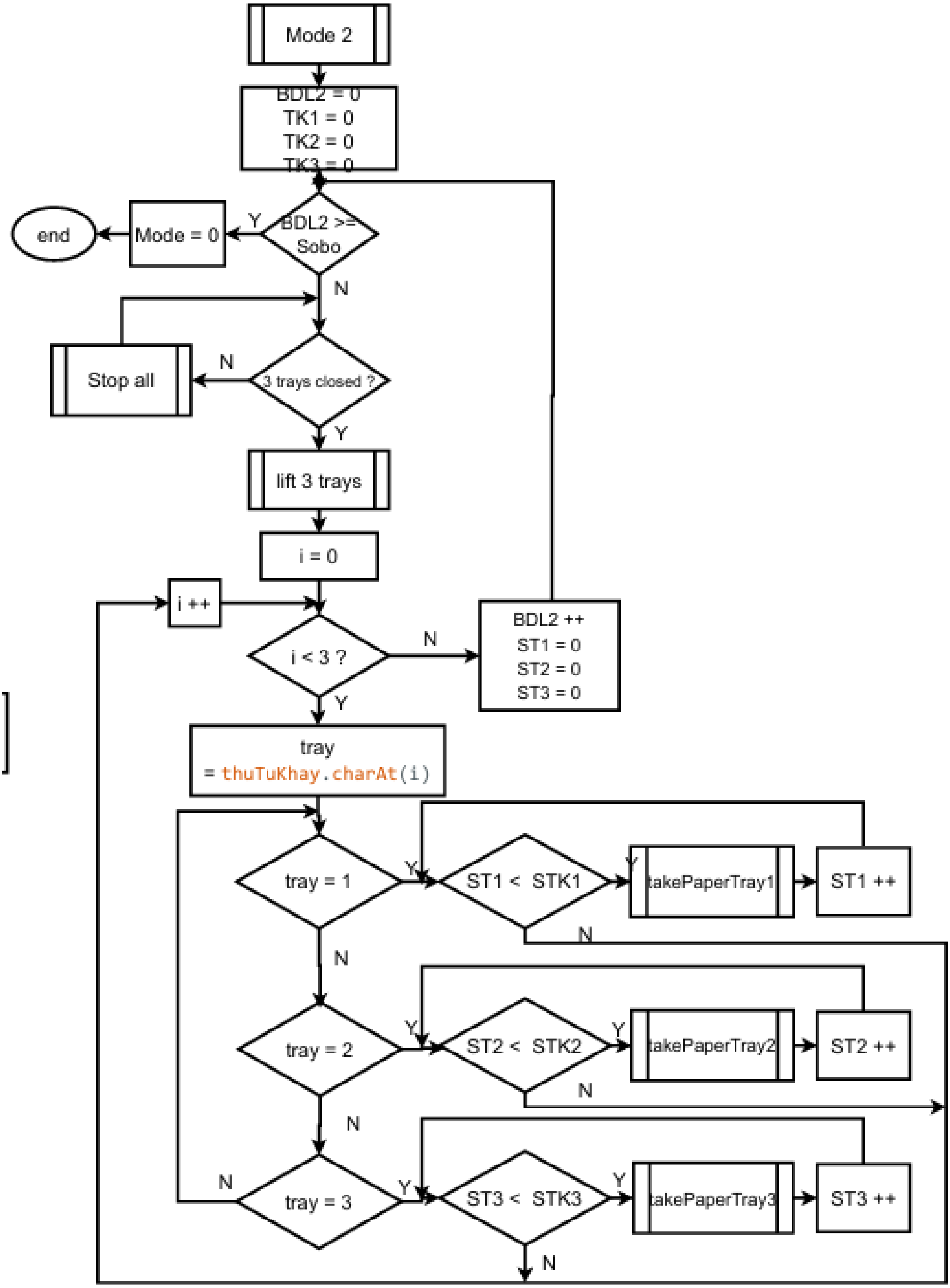


Figure 4.3 Subroutine algorithm for mode 2

- The "Stop All" routine is designed to halt all motors in the event of an error or when the paper runs out.

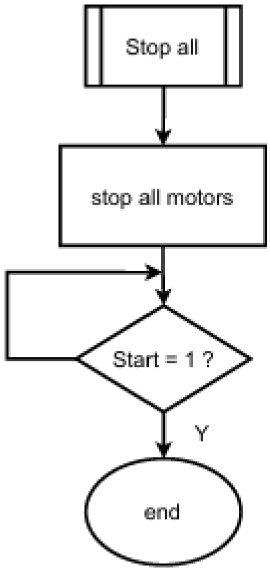


Figure 4.4 Stop ALL program algorithm

- The "Lift 3 Trays" routine is designed to simultaneously lift all three trays at the start of operation in order to reduce time, instead of lifting each tray individually.

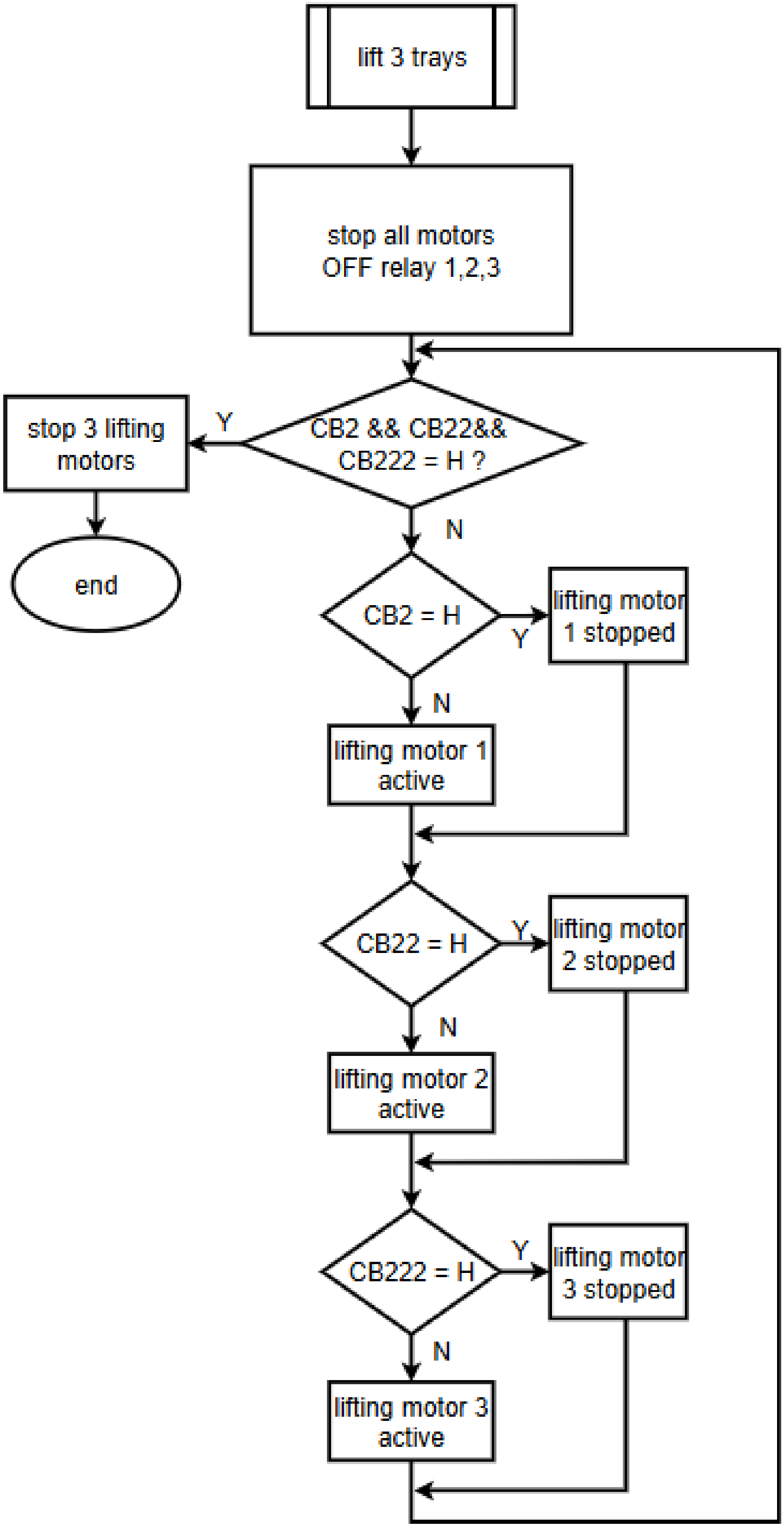


Figure 4.5 Tray lifting program algorithm for three trays

- The paper pickup algorithm for the three trays is designed to pick one sheet of paper from each tray. The operating principle is as follows: lower the paper pickup mechanism → check if paper is within operational range, if not, lift the paper → pick the paper from the tray → move the paper upward → delay for a short period before proceeding to the next tray's pickup sequence.

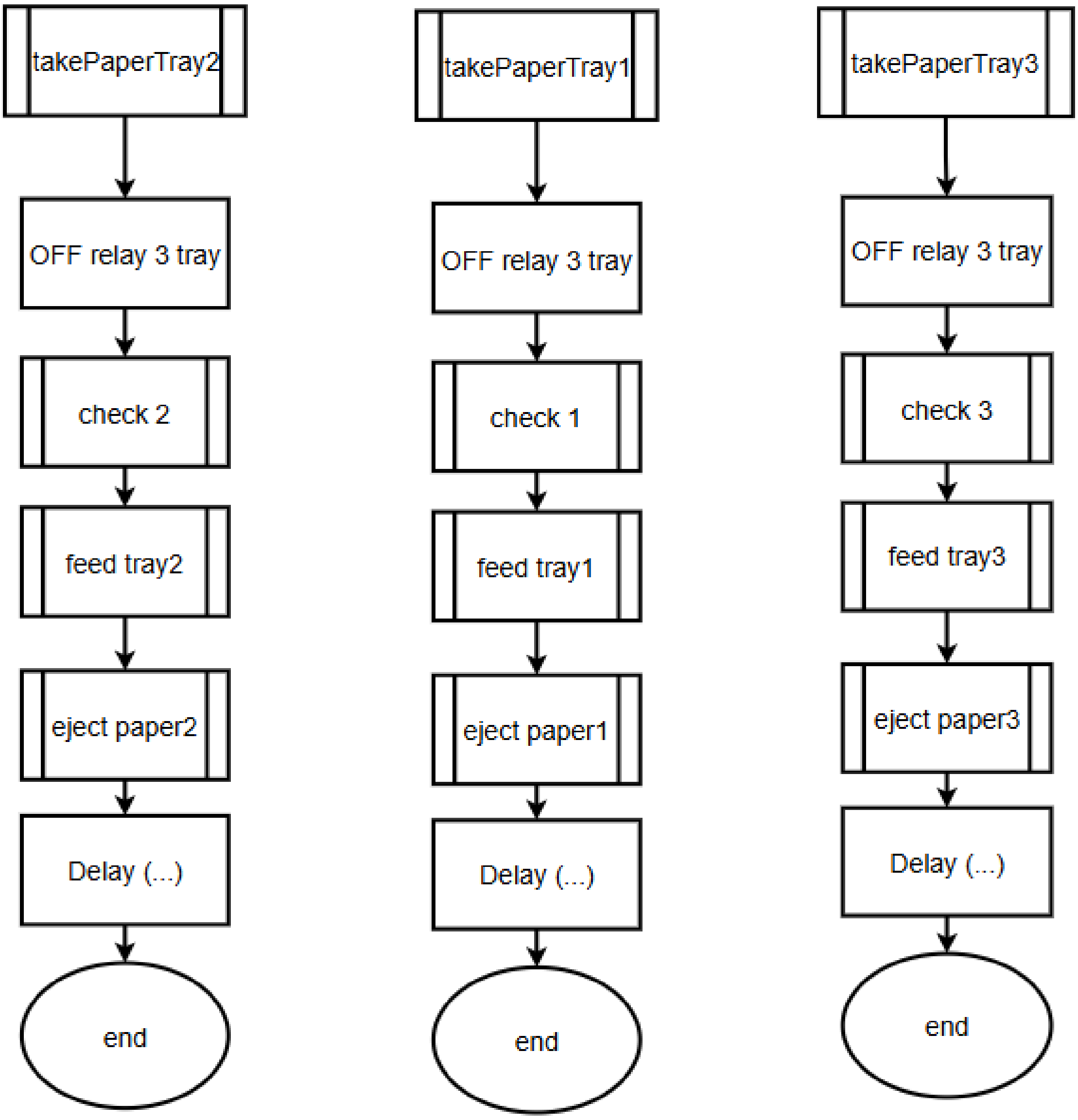


Figure 4.6 Subroutine algorithm for paper picking from three trays

The checking algorithm is designed to verify whether the paper is within the reachable range for pickup. If the paper is not within range, the system will lift the paper so that the pickup mechanism can successfully retrieve it.

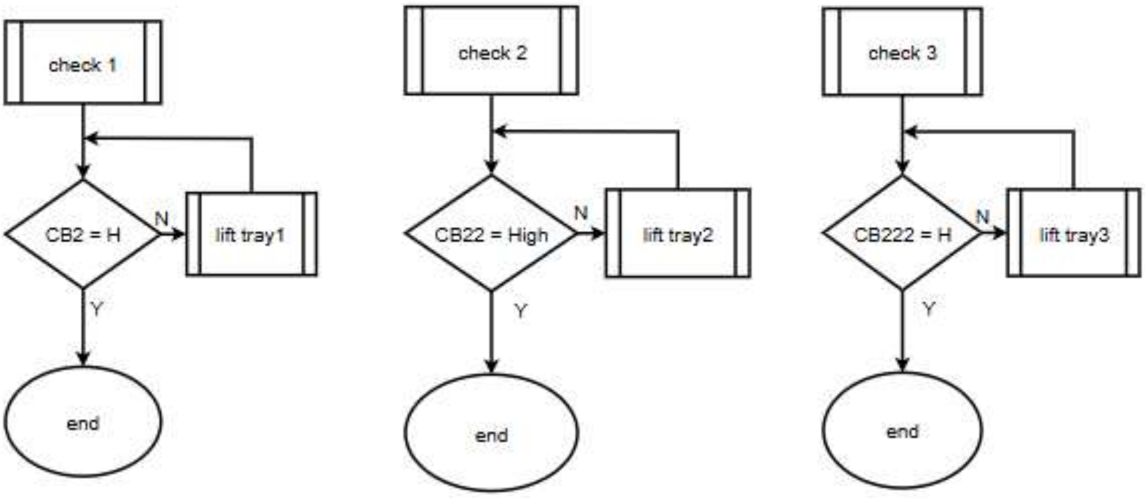


Figure 4.7 Checking program algorithm for three trays

- The paper-picking algorithm is designed to lower the pickup mechanism and feed a single sheet of paper to the paper feeding unit, with the process monitored by an optical sensor that detects the presence of paper inside.

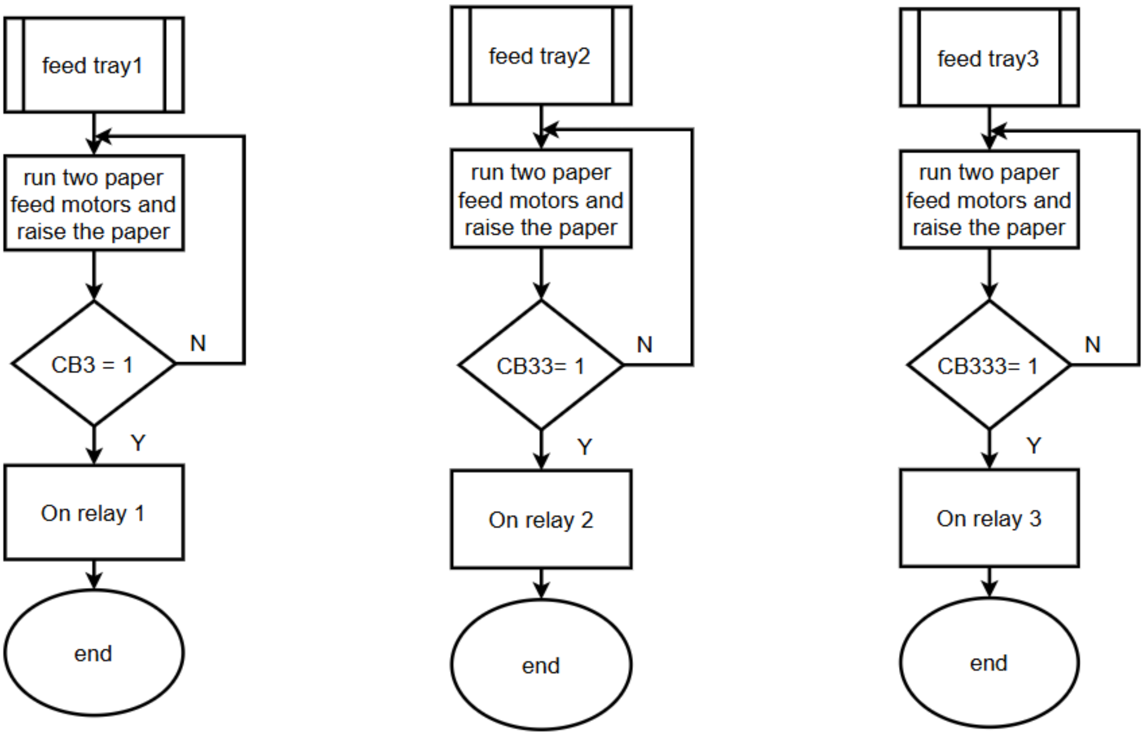


Figure 4.8 Paper picking algorithm up to the paper feeding section for three trays

- The algorithm for the paper-feeding program to Unit B is designed to transfer the paper onto the conveyor.

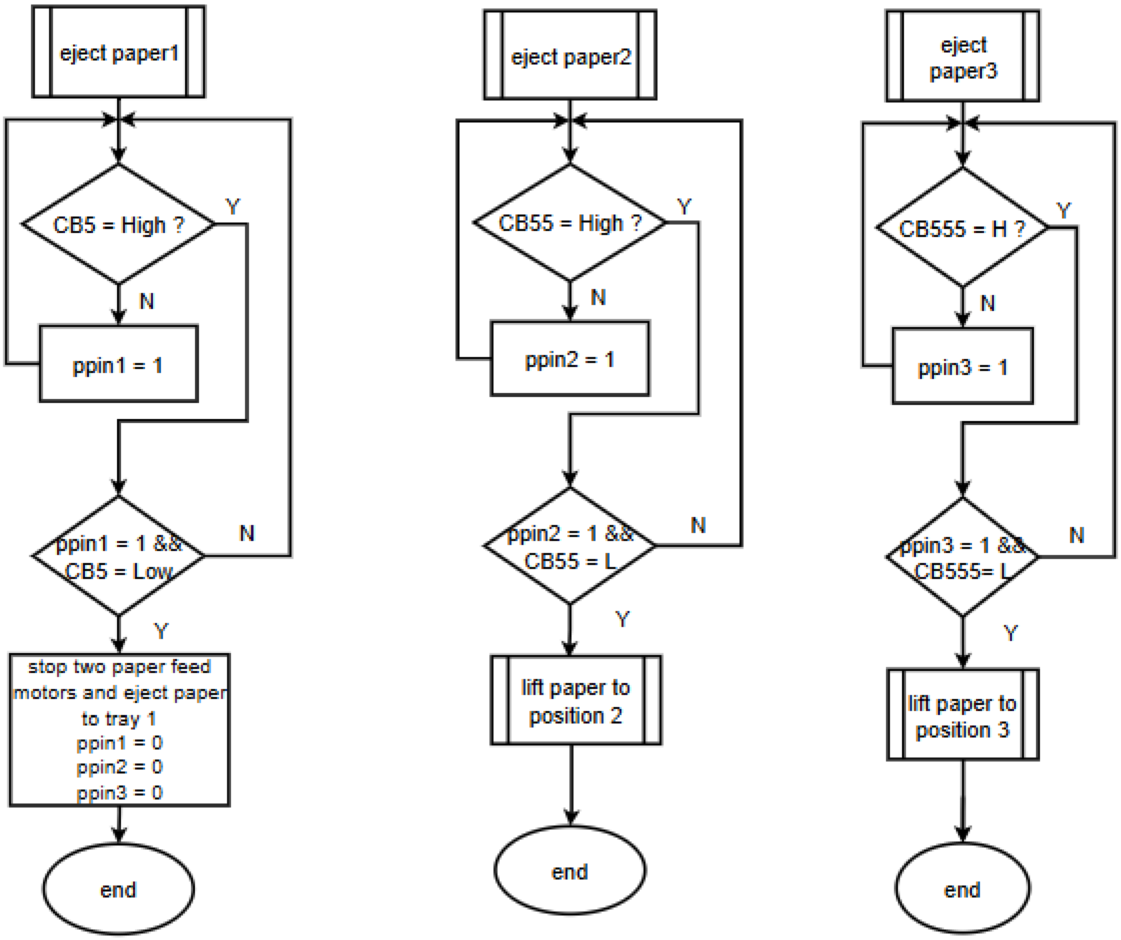


Figure 4.9 Algorithm for loading paper onto cluster B of 3 trays

- Since Tray 1 is located at the top, transferring paper to Unit B is faster, while Trays 2 and 3 require passing through an intermediate motor. Therefore, additional routines are needed to transfer paper from these trays to Unit B.

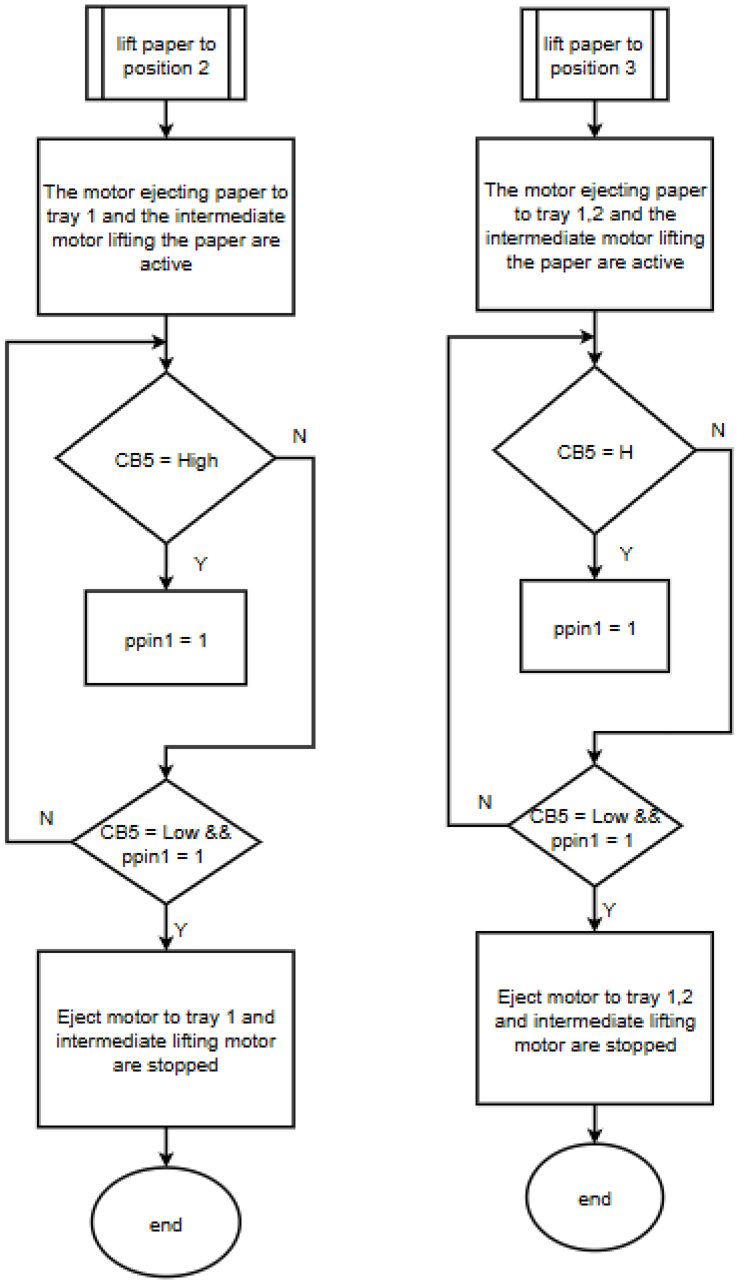


Figure 4.10 Algorithm for transferring paper from trays 2 and 3 via intermediary to cluster B

- The lift algorithm for each tray is executed when the paper falls below or rises above the operating range of the pickup mechanism.

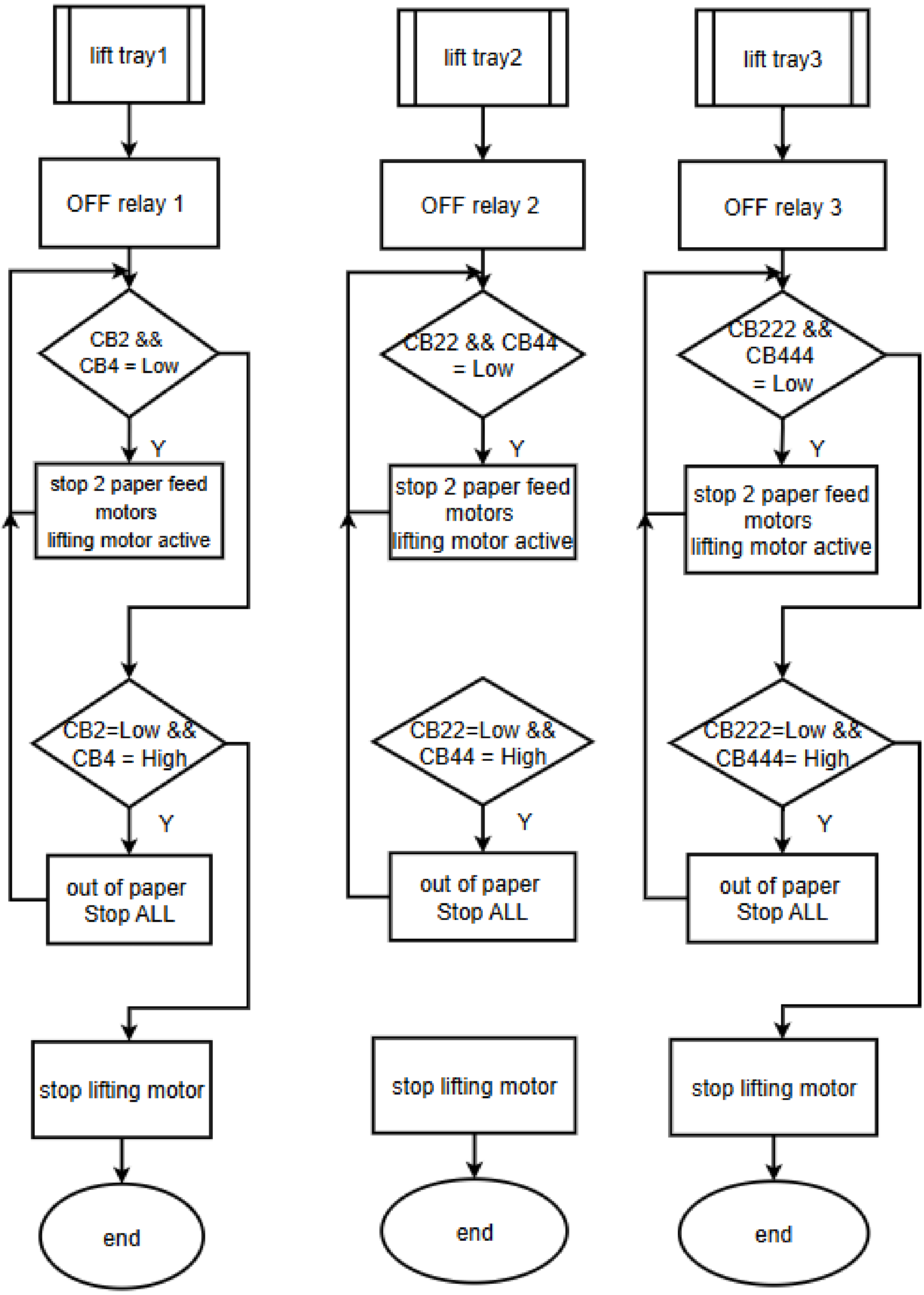


Figure 4.11 Algorithm for lifting paper from each tray

CHAPTER 5: RESULTS AND CONCLUSIONS.

Upon completion, the system operated stably and performed well in the collating function, achieving the required output frequency. The paper jam rate was relatively low, mainly due to the quality of the paper and the aged condition of the machine components. The system successfully supported the selection of operating modes that integrate all three functions: collating, perforating, and creasing on a single device.

However, the project still has some limitations, particularly in terms of aesthetics, due to time constraints and the team's limited experience. Additionally, the perforating and creasing functions still require manual adjustments and are not yet fully automated. Further improvements are necessary to enhance the system's flexibility and operational convenience.

- **Machine Operation and Experimental Results**

- **Mode 1**: with 20 sets.
- **Mode 2**: with 20 sets, each output consisting of 2 sheets taken from separate trays.

Figure 5.1 Loading paper into tray 2 and tray 3

Figure 5.2 Loading paper into tray 1

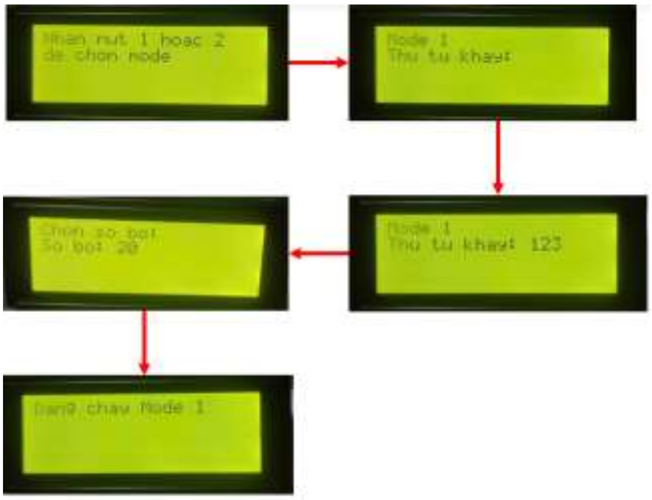


Figure 5.3 Parameter input steps for Mode 1

Figure 5.4 Parameter input steps for Mode 2

Figure 5.5. Actual output results of paper from the trays

- **Evaluation of Experimental Run Results:**

The following are the experimental results of the machine, based on the criteria: output frequency, errors when two sheets are output consecutively, paper jams, and sheet misalignment.

Table 5.1 Experimental run results (5 trials) in Mode 1

Trial	Frequency	Errors	Paper Jams	Misalignment
1	30 sheets/min	0	0	1
2	30 sheets/min	1	0	0
3	30 sheets/min	0	0	0
4	30 sheets/min	0	0	0
5	30 sheets/min	1	0	1

Table 5.2 Experimental run results (5 trials) in Mode 2

Trial	Frequency	Errors	Paper Jams	Misalignment
1	30 sheets/min	2	0	3
2	30 sheets/min	1	0	0
3	30 sheets/min	0	0	0
4	30 sheets/min	2	0	4
5	30 sheets/min	0	0	2

- **General Evaluation Based on Experimental Results:**

Table 5.3 System Operation Results

Evaluation Criteria	**Result**	**Remarks**
Operating Frequency	25–35 sheets/min	Satisfactorily meets the required operating frequency
Vibration Level	Moderate	Within acceptable limits; affected by the aging machine and suboptimal motor control
Paper Jam Rate	1–2%	Mainly caused by curled or low-quality paper
Defective Paper Rate	5–8%	Includes uneven collating, slight perforation misalignment, and improper output

To overcome the current limitations and improve the system's operational efficiency in the future, several specific improvements should be implemented:

- Fully automating the perforating and creasing functions is essential to reduce reliance on manual operations, thereby increasing accuracy and saving time.
- Replacing or reinforcing worn mechanical components such as rollers, gears, and bearings.
- Using anti-vibration bases or shock-absorbing pads at motor mounting positions.
- Redesigning the protective casing and control panel for clearer navigation, and developing a touch interface (HMI) or computer interface to facilitate configuration and monitoring of the system.
- Designing the output mechanism after the collating stage and the input mechanism for the perforating and creasing units to prevent paper misalignment and reduce defective products.

APPENDIX

❖ Machine control code

```
#include <Wire.h>
#define PCF8575_ADDR 0x25  // Địa chỉ PCF8575
uint16_t pcfState = 0xFFFF; // Mặc định tất cả HIGH (inactive)

// Khai báo chân cảm biến và điều khiển động cơ L298N
const int interruptPin1 = 16;
const int interruptPin2 = 23;
const int interruptPin3 = 24;   // Ngắt ngoài
const int interruptPin4 = 25;
const int interruptPin5 = 26;
///////////////////////////////////////////////////////////
const int interruptPin6 = 37;
////////////////////////////////////////////////////////////
const int interruptPin11 = 27;
const int interruptPin22 = 28;
const int interruptPin33 = 29;   // Ngắt ngoài
const int interruptPin44 = 30;
const int interruptPin55 = 31;
/////////////////////////////////////////////////////////////
const int interruptPin111 = 32;
const int interruptPin222 = 33;
const int interruptPin333 = 34;   // Ngắt ngoài
const int interruptPin444 = 35;
const int interruptPin555 = 36;
////////////////////////const int ENA22 = 12;
const int IN122 = 51;
const int IN222 = 50;
//////////////////////const int ENA = 11;
const int IN1 = 48;
```

- const int IN2 = 49;
- /////////////////động cơ nâng khay 3//////////////////////
- const int ENA33 = 13;
- const int IN133 = 53;
- const int IN233 = 52;
- ///////////////////động cơ 1 khay 1//////////////////////
- const int PWM_Khay1_1 = 4;
- const int Enable_Khay1_1 = 38;
- //////////////////////động cơ 2 khay 1//////////////////////
- const int PWM_Khay1_2 = 5;
- const int Enable_Khay1_2 = 39;
- ////////////////////////động cơ 1 khay 2//////////////////////
- const int PWM_Khay2_1 = 6;
- const int Enable_Khay2_1 = 40;
- ////////////////////động cơ 2 khay 2//////////////////////
- const int PWM_Khay2_2 = 7;
- const int Enable_Khay2_2 = 41;
- ///////////////////////động cơ 1 khay 3//////////////////////
- const int PWM_Khay3_1 = 8;
- const int Enable_Khay3_1 = 42;
- /////////////////////động cơ 2 khay 3//////////////////////
- const int PWM_Khay3_2 = 9;
- const int Enable_Khay3_2 = 43;
- ////////////////////động cơ TG//////////////////////
- const int PWM_Trung_Gian = 10;
- const int Enable_Trung_Gian = 47;
- //
- const int PWM_B_1 = 3;
- const int Enable_B_1 = 14;
- //

- const int PWM_B_2 = 2;
- const int Enable_B_2 = 15;
- ////////////////nam châm khay 2 //////////////////////
- const int RPWM22 = 45;
- ///////////////////nam châm khay 1 //////////////////////
- const int RPWM = 44;
- /////////////////nam châm khay 3 //////////////////////
- const int RPWM33 = 46;
- /////////////////các biến //////////////////////
- int mode = 0;
- int soBo = 0;
- String thuTuKhay = "";
- int Khay1To = 0;
- int Khay2To = 0;
- int Khay3To = 0;
- int sotokhay1 = 0;
- int sotokhay2 = 0;
- int sotokhay3 = 0;
- int bodalam1=0;
- int Bodalam2=0;
- int i = 0;
- int a = 0;
- int b = 0;
- int c = 0;
- int ppin1=0;
- int ppin2=0;
- int ppin3=0;
-
- bool daChonMode = false;
- bool daNhapThongTin = false;

```
-
- void setup() {
-   Wire.begin();
-   Serial.begin(9600);
-     //Serial1.begin(9600);
-
-   writePCF8575(pcfState);
-   bitClear(pcfState, 9);
-   bitClear(pcfState, 10);
-   writePCF8575(pcfState);
- ///////////////////////khay_1 và trung_giang//////
-   pinMode(interruptPin1, INPUT_PULLUP);
-   pinMode(interruptPin2, INPUT_PULLUP);
-   pinMode(interruptPin3, INPUT_PULLUP);
-   pinMode(interruptPin4, INPUT_PULLUP);
-   pinMode(interruptPin5, INPUT_PULLUP);
-   pinMode(interruptPin6, INPUT_PULLUP);
-
-   pinMode(ENA, OUTPUT);
-   pinMode(IN1, OUTPUT);
-   pinMode(IN2, OUTPUT);
-
-   pinMode(RPWM, OUTPUT);
-
-   pinMode(Enable_Trung_Gian, OUTPUT);
-   pinMode(PWM_Trung_Gian, OUTPUT);
-
-   pinMode(Enable_B_1, OUTPUT);
-   pinMode(PWM_B_1, OUTPUT);
-
```

```
- pinMode(Enable_B_2, OUTPUT);
- pinMode(PWM_B_2, OUTPUT);
-
- pinMode(Enable_Khay1_1, OUTPUT);
- pinMode(PWM_Khay1_1, OUTPUT);
-
- pinMode(Enable_Khay1_2, OUTPUT);
- pinMode(PWM_Khay1_2, OUTPUT);
-
- ///////////////////////////////////////////////// khay_2/////////////////////////////////
- pinMode(interruptPin11, INPUT_PULLUP);
- pinMode(interruptPin22, INPUT_PULLUP);
- pinMode(interruptPin33, INPUT_PULLUP);
- pinMode(interruptPin44, INPUT_PULLUP);
- pinMode(interruptPin55,INPUT_PULLUP);
-
- pinMode(ENA22, OUTPUT);
- pinMode(IN122, OUTPUT);
- pinMode(IN222, OUTPUT);
-
- pinMode(RPWM22, OUTPUT);
-
- pinMode(Enable_Khay2_1, OUTPUT);
- pinMode(PWM_Khay2_1, OUTPUT);
-
- pinMode(Enable_Khay2_2, OUTPUT);
- pinMode(PWM_Khay2_2, OUTPUT);
- /////////////////////////////////////khay_3////////////////////////////////////
- pinMode(interruptPin111, INPUT_PULLUP);
- pinMode(interruptPin222, INPUT_PULLUP);
```

```
-     pinMode(interruptPin333, INPUT_PULLUP);
-     pinMode(interruptPin444, INPUT_PULLUP);
-     pinMode(interruptPin555,INPUT_PULLUP);
-
-     pinMode(ENA33, OUTPUT);
-     pinMode(IN133, OUTPUT);
-     pinMode(IN233, OUTPUT);
-
-     pinMode(RPWM33, OUTPUT);
-
-     pinMode(Enable_Khay3_1, OUTPUT);
-     pinMode(PWM_Khay3_1, OUTPUT);
-
-     pinMode(Enable_Khay3_2, OUTPUT);
-     pinMode(PWM_Khay3_2, OUTPUT);
-
-     stopMotor1();
-     OFFrelay1();
-
-     //attachInterrupt(digitalPinToInterrupt(interruptPin3), sensor3ISR, CHANGE);
-   }
-   //Điều khiển động cơ TG
-   void runMotor_TG(int pwmVal) {
-     pwmVal = constrain(pwmVal, 0, 255);
-     digitalWrite(Enable_Trung_Gian, HIGH);     // Enable động cơ 1 khay 1
-     analogWrite(PWM_Trung_Gian, pwmVal);       // Điều chỉnh tốc độ
-   }
-
-   void stopMotor_TG() {
-     digitalWrite(Enable_Trung_Gian, LOW);      // Tắt động cơ 1 khay 1
```

```
-  analogWrite(PWM_Trung_Gian, 255);        // Dừng động cơ (giữ pwm ở mức cao)
- }
-
- // Điều khiển động cơ 1 khay 1
- void runMotor1_Khay1(int pwmVal) {
-  pwmVal = constrain(pwmVal, 0, 255);
-  digitalWrite(Enable_Khay1_1, HIGH);     // Enable động cơ 1 khay 1
-  analogWrite(PWM_Khay1_1, pwmVal);       // Điều chỉnh tốc độ
- }
-
- void stopMotor1_Khay1() {
-  digitalWrite(Enable_Khay1_1, LOW);      // Tắt động cơ 1 khay 1
-  analogWrite(PWM_Khay1_1, 255);          // Dừng động cơ (giữ pwm ở mức cao)
- }
-
- // Điều khiển động cơ 2 khay 1
- void runMotor2_Khay1(int pwmVal) {
-  pwmVal = constrain(pwmVal, 0, 255);
-  digitalWrite(Enable_Khay1_2, HIGH);     // Enable động cơ 2 khay 1
-  analogWrite(PWM_Khay1_2, pwmVal);       // Điều chỉnh tốc độ
- }
-
- void stopMotor2_Khay1() {
-  digitalWrite(Enable_Khay1_2, LOW);      // Tắt động cơ 2 khay 1
-  analogWrite(PWM_Khay1_2, 255);          // Dừng động cơ (giữ pwm ở mức cao)
- }
-
- // Điều khiển động cơ 1 khay 2
- void runMotor1_Khay2(int pwmVal) {
-  pwmVal = constrain(pwmVal, 0, 255);
```

- digitalWrite(Enable_Khay2_1, HIGH); // Enable động cơ 1 khay
- analogWrite(PWM_Khay2_1, pwmVal); // Điều chỉnh tốc độ
- }
-
- void stopMotor1_Khay2() {
- digitalWrite(Enable_Khay2_1, LOW); // Tắt động cơ 1 khay 2
- analogWrite(PWM_Khay2_1, 255); // Dừng động cơ (giữ pwm ở mức cao)
- }
-
- // Điều khiển động cơ 2 khay 2
- void runMotor2_Khay2(int pwmVal) {
- pwmVal = constrain(pwmVal, 0, 255);
- digitalWrite(Enable_Khay2_2, HIGH); // Enable động cơ 2 khay 2 // Điều khiển chiều quay
- analogWrite(PWM_Khay2_2, pwmVal); // Điều chỉnh tốc độ
- }
-
- void stopMotor2_Khay2() {
- digitalWrite(Enable_Khay2_2, LOW); // Tắt động cơ 2 khay 2
- analogWrite(PWM_Khay2_2, 255); // Dừng động cơ (giữ pwm ở mức cao)
- }
- void runMotor1_Khay3(int pwmVal) {
- pwmVal = constrain(pwmVal, 0, 255);
- digitalWrite(Enable_Khay3_1, HIGH); // Enable động cơ 1 khay
- analogWrite(PWM_Khay3_1, pwmVal); // Điều chỉnh tốc độ
- }
-
- void stopMotor1_Khay3() {
- digitalWrite(Enable_Khay3_1, LOW); // Tắt động cơ 1 khay 2
- analogWrite(PWM_Khay3_1, 255); // Dừng động cơ (giữ pwm ở mức cao)
- }

```
-
- // Điều khiển động cơ 2 khay 2
- void runMotor2_Khay3(int pwmVal) {
-   pwmVal = constrain(pwmVal, 0, 255);
-   digitalWrite(Enable_Khay3_2, HIGH);     // Enable động cơ 2 khay 2
-   analogWrite(PWM_Khay3_2, pwmVal);       // Điều chỉnh tốc độ
- }
-
- void stopMotor2_Khay3() {
-   digitalWrite(Enable_Khay3_2, LOW);      // Tắt động cơ 2 khay 2
-   analogWrite(PWM_Khay3_2, 255);          // Dừng động cơ (giữ pwm ở mức cao)
- }
- void runMotor1_B(int pwmVal) {
-   pwmVal = constrain(pwmVal, 0, 255);
-   digitalWrite(Enable_B_1, HIGH);     // Enable động cơ 2 khay 2
-   analogWrite(PWM_B_1, pwmVal);       // Điều chỉnh tốc độ
- }
-
- void stopMotor1_B() {
-   digitalWrite(Enable_B_1, LOW);      // Tắt động cơ 2 khay 2
-   analogWrite(PWM_B_1, 255);          // Dừng động cơ (giữ pwm ở mức cao)
- }
- void runMotor2_B(int pwmVal) {
-   pwmVal = constrain(pwmVal, 0, 255);
-   digitalWrite(Enable_B_2, HIGH);     // Enable động cơ 2 khay 2
-   analogWrite(PWM_B_2, pwmVal);       // Điều chỉnh tốc độ
- }
-
- void stopMotor2_B() {
-   digitalWrite(Enable_B_2, LOW);      // Tắt động cơ 2 khay 2
```

- analogWrite(PWM_B_2, 255); // Dừng động cơ (giữ pwm ở mức cao)
- }
- // Hàm chạy động cơ nâng giấy
- void runMotor1(int speed) {
- stopMotor1_Khay1();
- stopMotor2_Khay1();
- OFFrelay1();
- digitalWrite(IN1, LOW);
- digitalWrite(IN2, HIGH);
- analogWrite(ENA, speed);
- }
-
- void runMotor2(int speed) {
- stopMotor1_Khay2();
- stopMotor2_Khay2();
- OFFrelay2();
- digitalWrite(IN122, LOW);
- digitalWrite(IN222, HIGH);
- analogWrite(ENA22, speed);
- }
- void runMotor3(int speed) {
- stopMotor1_Khay3();
- stopMotor2_Khay3();
- OFFrelay3();
- digitalWrite(IN133, LOW);
- digitalWrite(IN233, HIGH);
- analogWrite(ENA33, speed);
- }
- // Hàm dừng động cơ
- void stopMotor1() {

```
-   digitalWrite(IN1, LOW);
-   digitalWrite(IN2, LOW);
-   analogWrite(ENA, 0);
- }
-
- void stopMotor2() {
-   digitalWrite(IN122, LOW);
-   digitalWrite(IN222, LOW);
-   analogWrite(ENA22, 0);
- }
-
- void stopMotor3() {
-   digitalWrite(IN133, LOW);
-   digitalWrite(IN233, LOW);
-   analogWrite(ENA33, 0);
- }
- // Hàm bật relay khay 1
- void ONrelay1() {
-   analogWrite(RPWM, 235);
-
- }
-
- // Hàm tắt relay khay 1
- void OFFrelay1() {
-   analogWrite(RPWM, 0);
-
- }
-
- // Hàm bật relay khay 2
- void ONrelay2() {
```

```
-     analogWrite(RPWM22, 235);
-
-
-
-   }
-
-   // Hàm tắt relay khay 2
-   void OFFrelay2() {
-     analogWrite(RPWM22, 0);
-
-   }
-   void ONrelay3() {
-     analogWrite(RPWM33, 235);
-
-   }
-
-   // Hàm tắt relay khay 3
-   void OFFrelay3() {
-     analogWrite(RPWM33, 0);
-
-   }
-   void StopALL(){
-     stopMotor1();
-     stopMotor2();
-     stopMotor3();
-     stopMotor1_B();
-     stopMotor2_B();
-     stopMotor_TG();
-     stopMotor1_Khay1();
-     stopMotor2_Khay1();
```

- stopMotor1_Khay2();
- stopMotor2_Khay2();
- }
- ////....................KHAY 1....................
- ////Nâng khay 1..........
- void nang1(){
- while (true) {
- OFFrelay1();
- if (digitalRead(interruptPin2) == LOW && digitalRead(interruptPin4) == LOW) {
- //Serial.println("1");
- runMotor1(120);
- stopMotor1_Khay1();
- stopMotor2_Khay1();
- }
- else if (digitalRead(interruptPin2) == LOW && digitalRead(interruptPin4) == HIGH){
-
- delay(45);
- StopALL();
- }
- else {
- //Serial.println("2");
- delay(45);
- stopMotor1();
- delay(300);
- break;
- }
- }
- }
-

```
- //////Lấy giấy khay 1.........
- void laygiay1(){
-  if(a<=20) {
-   OFFrelay1();
-   OFFrelay2();
-   OFFrelay3();
-   kiemtra1();
-   dem1();
-   motor1();
-   }
-  else{
-   a=0;
-   NGUOC1(50);
-   delay(1000);}
-  }
-
- void kiemtra1(){
-  if (digitalRead(interruptPin2) == LOW ){
-   nang1();
-   }
-  else {
-   return;
-  }}
-
- ///Đếm giấy khay 1........
- void dem1(){
- while (true){
- bool hientai1 =digitalRead(interruptPin3);
- runMotor1_Khay1(220);
- runMotor2_Khay1(220);
```

```
-  OFFrelay1();
-  if  (hientai1 == LOW) {
-     ONrelay1();
-     a++;
-     break;
-    }
-  }
-  }
-  void motor1(){
-  while (true){
-   bool pin1 =digitalRead(interruptPin5);
-   if (pin1==LOW){
-     ppin1=1;
-    }
-   else if (ppin1==1 && pin1==HIGH)
-    {
-     stopMotor1_Khay1();
-     stopMotor2_Khay1();
-     ppin3=0;
-     ppin2=0;
-     ppin1=0;
-     break;
-    }}
-  }
-  void NGUOC1(int speed) {
-   stopMotor1_Khay1();
-   runMotor2_Khay1(220);
-   OFFrelay1();
-   digitalWrite(IN1, HIGH);
-   digitalWrite(IN2, LOW);
```

```
-    analogWrite(ENA, speed);
-  }
-   ////....................KHAY 2....................
-
-  void nang2(){
-   while (true) {
-     OFFrelay2();
-     if (digitalRead(interruptPin22) == LOW && digitalRead(interruptPin44) ==
   LOW) {
-        runMotor2(120);
-        stopMotor1_Khay2();
-        stopMotor2_Khay2();
-     }
-     else if (digitalRead(interruptPin22) == LOW && digitalRead(interruptPin44)
   == HIGH){
-
-        StopALL();
-     }
-     else {
-        //delay(40);
-        stopMotor2();
-        delay(200);
-        break;
-     }
-     }
-  }
-  void laygiay2(){
-    if(b<=20) {
-      OFFrelay1();
-      OFFrelay2();
-      OFFrelay3();
```

```
-     kiemtra2();
-     dem2();
-     motor2();
-     }
-   else{
-     b=0;
-     NGUOC2(50);
-     delay(1000);}
-   }
- void kiemtra2(){
-   if (digitalRead(interruptPin22) == LOW ){
-     nang2();
-     }
-   else {
-     return;
-   }}
-
- void dem2(){
- while (true) {
-   bool hientai2 =digitalRead(interruptPin33);
-   runMotor1_Khay2(220);
-   runMotor2_Khay2(220);
-   OFFrelay2();
- if  (hientai2 == LOW) {
-     ONrelay2();
-     b++;
-     break;
-   }
- }
- }
```

```
- void motor2(){
- while (true) {
-  bool pin2 =digitalRead(interruptPin55);
-  if (pin2==LOW){
-   ppin2=1;
-   runMotor_TG(220);
-   runMotor2_Khay1(220);
-  }
-  else if (ppin2==1 && pin2==HIGH)
-  {
-   stopMotor1_Khay2();
-   //stopMotor2_Khay2();
-   motor22();
-   ppin3=0;
-   ppin2=0;
-   ppin1=0;
-   break;
-  }
- }}
-
- void motor22(){
-  while (true) {
-  bool pin1 =digitalRead(interruptPin5);
-  if (pin1==LOW){
-   ppin1=1;
-  }
-  else if (ppin1==1 && pin1==HIGH)
-  {
-   stopMotor_TG();
-   stopMotor2_Khay1();
```

```
-      stopMotor2_Khay2();
-      break;
-     }
-   }
- }
-
- void NGUOC2(int speed) {
-   stopMotor1_Khay2();
-   runMotor2_Khay2(220);
-   OFFrelay2();
-   digitalWrite(IN122, HIGH);
-   digitalWrite(IN222, LOW);
-   analogWrite(ENA22, speed);
- }
-  ////....................KHAY 3....................
- void nang3(){
-  while (true) {
-    OFFrelay3();
-    if (digitalRead(interruptPin222) == LOW && digitalRead(interruptPin444) ==
  LOW) {
-       runMotor3(120);
-       stopMotor1_Khay3();
-       stopMotor2_Khay3();
-     }
-    else if (digitalRead(interruptPin222) == LOW && digitalRead(interruptPin444)
  == HIGH){
-
-       StopALL();
-     }
-    else {
-       delay(40);
```

```
-         stopMotor3();
-         delay(500);
-         break;
-       }
-       }
-   }
-   void laygiay3(){
-     if(c<=20) {
-       OFFrelay1();
-       OFFrelay2();
-       OFFrelay3();
-       kiemtra3();
-       dem3();
-       motor3();
-       }
-     else{
-       c=0;
-       NGUOC3(50);
-       delay(1000);}
-     }
-
-   void kiemtra3(){
-     if (digitalRead(interruptPin222) == LOW ){
-       nang3();
-       }
-     else {
-       return;
-     }}
-
-   void dem3(){
```

```
- while (true) {
-   bool hientai3 =digitalRead(interruptPin333);
-   OFFrelay3();
-   runMotor1_Khay3(220);
-   runMotor2_Khay3(220);
- if  (hientai3 == LOW) {
-     ONrelay3();
-     c++;
-     break;
-   }
- }
- }
- void motor3(){
- while (true) {
-   bool pin3 =digitalRead(interruptPin555);
-   if (pin3==LOW){
-     ppin3=1;
-     runMotor_TG(220);
-     runMotor2_Khay1(220);
-     runMotor2_Khay2(220);
-   }
-   else if (ppin3==1 && pin3==HIGH)
-   {
-     stopMotor1_Khay3();
-     //stopMotor2_Khay3();
-     motor33();
-     ppin3=0;
-     ppin2=0;
-     ppin1=0;
-     break;
```

```
-     }
-   }}
-
-   void motor33(){
-     while (true) {
-     bool pin1 =digitalRead(interruptPin5);
-     if (pin1==LOW){
-       ppin1=1;
-     }
-     else if (ppin1==1 && pin1==HIGH)
-     {
-       stopMotor_TG();
-       stopMotor2_Khay1();
-       stopMotor2_Khay2();
-       stopMotor2_Khay3();
-       break;
-       }
-     }
-   }
-   void NGUOC3(int speed) {
-     stopMotor1_Khay2();
-     runMotor2_Khay2(220);
-     OFFrelay1();
-     digitalWrite(IN122, HIGH);
-     digitalWrite(IN222, LOW);
-     analogWrite(ENA22, speed);
-   }
-   void nang3khay(){
-     while (true) {
-       OFFrelay1();
```

```
-     OFFrelay2();
-     OFFrelay3();
-     stopMotor1_Khay1();
-     stopMotor2_Khay1();
-     stopMotor1_Khay2();
-     stopMotor2_Khay2();
-     stopMotor1_Khay3();
-     stopMotor2_Khay3();
-     bool done1 = digitalRead(interruptPin2) == HIGH;
-     bool done2 = digitalRead(interruptPin22) == HIGH;
-     bool done3 = digitalRead(interruptPin222) == HIGH;
-     if (done1 && done2 && done3) {
-       stopMotor1();
-       stopMotor2();
-       stopMotor3();
-       break;
-     }
-     if (!done1) runMotor1(140); else stopMotor1();
-     if (!done2) runMotor2(140); else stopMotor2();
-     if (!done3) runMotor3(140); else stopMotor3();
-   }
- }
- ////////////////Mode1.................
- void mode1(){
-   Serial.println("BatDauMode1");
-   runMotor1_B(220);
-  runMotor2_B(220);
-
- int  bodalam1 = 0;
- while (true) {
```

```
-   bool sensorState1 = digitalRead(interruptPin1);
-   bool sensorState2 = digitalRead(interruptPin11);
-   bool sensorState3 = digitalRead(interruptPin111);
-    if (bodalam1 >= soBo){
-       mode = 0;
-       Serial.println("HoanTat");
-       break;
-   }
-    else {
-        if (sensorState2 == LOW && sensorState1 == LOW && sensorState3 == LOW ) {
-       nang3khay();
-
-       for (int i = 0; i < 3; i++) {
-          char khay = thuTuKhay.charAt(i);
-          if (khay == '1') {
-            laygiay1();
-          } else if (khay == '2') {
-            laygiay2();
-          } else if (khay == '3') {
-            laygiay3();
-          }
-        }
-
-        bodalam1++;
-        Serial.println("DaLam:" + String(bodalam1));
-      }
-       else{
-         StopALL();
-      }
-   }
```

```
- }
- }
- /////////////////Mode2....................
- void mode2(){
-  Serial.println("BatDauMode2");
-  runMotor1_B(220);
- runMotor2_B(220);
- int Bodalam2 = 0;
- int sotokhay2=0;
- int sotokhay1=0;
- int sotokhay3=0;
- while (true) {
-
- bool sensorState1 = digitalRead(interruptPin1);
- bool sensorState2 = digitalRead(interruptPin11);
- bool sensorState3 = digitalRead(interruptPin111);
- if (Bodalam2 >= soBo ){
-    Serial.println("HoanTat");
-    break;
-   }
- else {
-      if (sensorState1 == LOW && sensorState2 == LOW &&sensorState3 == LOW  ) {
-         nang3khay();
-          if (sotokhay1 >= Khay1To && sotokhay2 >= Khay2To && sotokhay3 >= Khay3To) {
-             Bodalam2++;
-             Serial.println("DaLam:" + String(Bodalam2));
-             sotokhay1 = 0;
-             sotokhay2 = 0;
-             sotokhay3 = 0;
```

```
-         }
-           else {
-           for (int i = 0; i < 3; i++) {
-     char khay = thuTuKhay.charAt(i);
-
-     if (khay == '1') {
-       while (sotokhay1 < Khay1To) {
-         laygiay1();
-         sotokhay1++;
-         delay(10);
-         Serial.println("sotokhay1:" + String(sotokhay1));
-       }
-     }
-     else if (khay == '2') {
-       while (sotokhay2 < Khay2To) {
-         laygiay2();
-         sotokhay2++;
-         delay(10);
-         Serial.println("sotokhay2:" + String(sotokhay2));
-       }
-     }
-     else if (khay == '3') {
-       while (sotokhay3 < Khay3To) {
-         laygiay3();
-         sotokhay3++;
-         delay(10);
-         Serial.println("sotokhay3:" + String(sotokhay3));
-       }
-     }
-   }
```

```
-           }
-         }
-       else{
-           stopMotor1();
-           stopMotor1_Khay1();
-           stopMotor2_Khay1();
-           stopMotor2_Khay2();
-           stopMotor1_Khay2();
-           }
-       }
-   }
-   }
-   void resetTrangThai() {
-     daChonMode = false;
-     daNhapThongTin = false;
-     mode = 0;
-     soBo = 0;
-     bodalam1 = 0;
-     Bodalam2 = 0;
-     Khay1To = 0;
-     Khay2To = 0;
-     Khay3To = 0;
-
-   }
-   //////////////Main...........
-   void loop() {
-     runMotor1_B(220);
-     runMotor2_B(220);
-    if (Serial.available() > 0) {
-       String input = Serial.readStringUntil('\n');
```

```
-     input.trim();
-     // Nếu chưa chọn mode
-     if (!daChonMode) {
-       if (input == "1") {
-         mode = 1;
-         daChonMode = true;
-
-       } else if (input == "2") {
-         mode = 2;
-         daChonMode = true;
-
-       } else {
-
-       }
-     }
-     else if (!daNhapThongTin) {
-       static int buoc = 0;
-       buoc++;
-       if (mode == 1) {
-         if (buoc == 1) {
-           thuTuKhay = input;
-           if (thuTuKhay.length() != 3 || !thuTuKhayIsValid(thuTuKhay)) {
-
-             buoc--;
-             return;
-           }
-
-         }
-          else if (buoc == 2) {
-           soBo = input.toInt();
```

```
-           daNhapThongTin = true;}}
-         else if (mode == 2) {
-           if (buoc == 1) {
-             thuTuKhay = input;
-             if (thuTuKhay.length() != 3 || !thuTuKhayIsValid(thuTuKhay)) {
-                     buoc--;
-               return;
-             }}
-           if (buoc == 2) {
-             Khay1To = input.toInt();
-           }
-           else if (buoc == 3) {
-             Khay2To = input.toInt();
-           }
-           else if (buoc == 4) {
-             Khay3To = input.toInt();
-           }
-           else if (buoc == 5) {
-             soBo = input.toInt();
-             daNhapThongTin = true;
-           }
-         }
-       }
-     }
-     if (daChonMode && daNhapThongTin) {
-       if (mode == 1) {
-         mode1();
-       } else if (mode == 2) {
-         mode2();
-
```

```
-     }
-   resetTrangThai();
-   }
-   void writePCF8575(uint16_t value) {
-   Wire.beginTransmission(PCF8575_ADDR);
-   Wire.write(value & 0xFF);       // Byte thấp
-   Wire.write((value >> 8) & 0xFF); // Byte cao
-   Wire.endTransmission();
-   }
- 
-   bool thuTuKhayIsValid(String s) {
-   if (s.length() != 3) return false;
-   for (int h = 0; h < 3; h++) {
-     char d = s.charAt(i);
-     if (d != '1' && d != '2' && d != '3') return false;
-   }
-   // Kiểm tra không trùng ký tự
-   if (s.indexOf('1') == -1 || s.indexOf('2') == -1 || s.indexOf('3') == -1) return false;
-   return true;
-   }
```

❖ **Button and LCD code**

```
- #include <Wire.h>
- #include <LiquidCrystal_I2C.h>
-
- const int buttonPins[5] = {2, 3, 4, 5, 6};
- LiquidCrystal_I2C lcd(0x27, 20, 4);
-
- int mode = 0, state = 0, soBo = 0;
- String thuTuKhay = "";
- int soToKhay[3] = {0, 0, 0};
- int khayDangChon = 0;
-
- void setup() {
-   Serial1.begin(9600);
-   Serial.begin(9600);
-   lcd.init(); lcd.backlight();
-   lcd.setCursor(0, 0);
-   lcd.print("Nhan nut 1 hoac 2");
-   lcd.setCursor(0, 1);
-   lcd.print("de chon mode");
-   for (int i = 0; i < 5; i++) pinMode(buttonPins[i], INPUT_PULLUP);
- }
-
- void loop() {
-   int nut = docNutNhan();
-
-   if (mode == 0) {
-     if (nut == 1) { mode = 1; state = 1; lcd.clear(); lcd.print("Mode 1"); nut = 0; delay(500); }
-     if (nut == 2) { mode = 2; state = 1; lcd.clear(); lcd.print("Mode 2"); nut = 0 ; delay(500); }
```

```
-   }
-
-   if (mode == 1) xuLyMode1(nut);
-   else if (mode == 2) xuLyMode2(nut);
-
-   // Nhận dữ liệu từ Arduino 1
-   if (Serial1.available() > 0) {
-     String nhan = Serial1.readStringUntil('\n');
-     nhan.trim();
-
-     if (nhan.startsWith("DaLam:")) {
-       lcd.setCursor(0, 3);
-       lcd.print("Da lam: ");
-       lcd.print(nhan.substring(6));
-       lcd.print("     ");
-     }
-     else if (nhan == "HoanTat") {
-       lcd.clear(); lcd.print("Da hoan tat!");
-       delay(2000);
-       lcd.clear(); lcd.print("Nhan nut 1 hoac 2 ");
-       lcd.setCursor(0, 1);
-       lcd.print("de chon mode");
-       mode = 0; state = 0;
-       thuTuKhay = ""; soBo = 0;
-       soToKhay[0] = soToKhay[1] = soToKhay[2] = 0;
-     }
-     else if (nhan.startsWith("BatDauMode")) {
-     }
-   }
- }
```

```
-
- int docNutNhan() {
-   for (int i = 0; i < 5; i++) {
-     if (digitalRead(buttonPins[i]) == LOW) {
-       delay(200);
-       return i + 1;
-     }
-   }
-   return 0;
- }
- void guiSerial(String data) {
-   Serial1.println(data);
- }
-
- void xuLyMode1(int nut) {
-   switch (state) {
-     case 1:
-       lcd.setCursor(0, 1);
-       lcd.print("Thu tu khay: " + thuTuKhay + "   ");
-       if (nut >= 1 && nut <= 3 && thuTuKhay.length() < 3) {
-         char c = '0' + nut;
-         if (thuTuKhay.indexOf(c) == -1) thuTuKhay += c;
-         else { lcd.setCursor(0, 2); lcd.print("Trung khay!      "); }
-       }
-       if (nut == 4 && thuTuKhay.length() == 3) {
-         guiSerial("1");
-         guiSerial(thuTuKhay);
-         state = 2;
-         nut = 0;
-         lcd.clear(); lcd.print("Chon so bo:");
```

```
-           lcd.setCursor(0, 1); lcd.print("So bo: " + String(soBo));
-         }
-         break;
-       case 2:
-         if (nut == 1) soBo++;
-         if (nut == 2) soBo += 10;
-         if (nut == 3 && soBo > 0) soBo--;
-         if (nut == 4 && soBo >= 10) soBo -= 10;
-         lcd.setCursor(0, 1);
-         lcd.print("So bo: " + String(soBo) + "   ");
-         if (nut == 5 && soBo > 0) {
-           guiSerial(String(soBo));
-           lcd.clear(); lcd.print("Dang chay Mode 1");
-           mode = 0; state = 0; nut = 0;
-         }
-         break;
-     }
-   }
-
-   void xuLyMode2(int nut) {
-     switch (state) {
-       case 1:
-         lcd.setCursor(0, 1);
-         lcd.print("Thu tu khay: " + thuTuKhay + "   ");
-         if (nut >= 1 && nut <= 3 && thuTuKhay.length() < 3) {
-           char c = '0' + nut;
-           if (thuTuKhay.indexOf(c) == -1) thuTuKhay += c;
-           else { lcd.setCursor(0, 2); lcd.print("Trung khay!      "); }
-         }
-         if (nut == 4 && thuTuKhay.length() == 3) {
```

```
-           guiSerial("2");
-           guiSerial(thuTuKhay);
-           state = 2;
-           khayDangChon = 0;
-           nut = 0;
-           lcd.clear(); lcd.print("So to khay 1:");
-           lcd.setCursor(0, 1); lcd.print("So to: " + String(soToKhay[0]));
-         }
-         break;
-
-       case 2: case 3: case 4: {
-         int k = khayDangChon;
-         if (nut == 1) soToKhay[k]++;
-         if (nut == 2) soToKhay[k] += 10;
-         if (nut == 3 && soToKhay[k] > 0) soToKhay[k]--;
-         if (nut == 4 && soToKhay[k] >= 10) soToKhay[k] -= 10;
-         lcd.setCursor(0, 1);
-         lcd.print("So to: " + String(soToKhay[k]) + "   ");
-         if (nut == 5) {
-           guiSerial(String(soToKhay[k]));
-           khayDangChon++;
-
-           if (khayDangChon < 3) {
-             state++;
-             lcd.clear(); lcd.print("So to khay " + String(khayDangChon + 1) + ":");
-             lcd.setCursor(0, 1); lcd.print("So to: " + String(soToKhay[khayDangChon]));
-           } else {
-             state = 5;
-             lcd.clear(); lcd.print("Chon so bo:");
-             lcd.setCursor(0, 1); lcd.print("So bo: " + String(soBo));
```

```
-          }
-        }
-        break;
-      }
-      case 5:
-        if (nut == 1) soBo++;
-        if (nut == 2) soBo += 10;
-        if (nut == 3 && soBo > 0) soBo--;
-        if (nut == 4 && soBo >= 10) soBo -= 10;
-        lcd.setCursor(0, 1);
-        lcd.print("So bo: " + String(soBo) + "   ");
-        if (nut == 5 && soBo > 0) {
-          guiSerial(String(soBo));
-          lcd.clear(); lcd.print("Dang chay Mode 2");
-          mode = 0; state = 0; nut = 0;
-        }
-        break;
-    }
-  }
```

REFERENCES

[1] PressXchange, “Heidelberg Varimatrix 105CS Perforating Machine,” 2007. [Online]. Available: http://pressxchange.com/en/heidelberg-varimatrix-105-c-year-2007/machine-id/401254/. [Accessed: Mar. 10, 2025].

[2] SF Services, “YAWA TDS-750L 560x750mm Platen Foiling Cutting Creasing Embossing,”[Online].Available:https://www.sfservices.com/yawa-tds750l-560x750mm-platen-foiling-cutting-creasing-embossing/ . [Accessed: Mar. 10, 2025].

[3] Northern Machine, “Horizon CRF-362,” [Online]. Available: https://northernmachine.com/product/crf-362/ . [Accessed: Mar. 10, 2025].

[4] KAMA GmbH, “ProCut 76 – Die Cutting and Embossing Machine,” [Online]. Available:https://www.kama.info/en/products/die-cutting-and-embossing-machines/procut76/ . [Accessed: Mar. 10, 2025].

[5] Ricoh Vietnam, “IM 6000 Black and White Multifunction Printer,” [Online]. Available:https://www.ricoh.com.vn/products/office-printing/multifunction-printers/mfp-black-and-white/im-6000/ . [Accessed: Mar. 10, 2025].

[6] Scribd, “Part Catalog 6502-1,” [Online]. Available: https://www.scribd.com/document/781273497/ Part-Catalog-6502-1. [Accessed: Apr. 10, 2025].

[7] Motorbank, “Mabuchi Motor ID-549XW-5031,” [Online]. Available: https://www.motorbank.kr/goods/goods_view.php?goodsNo=1000009303 . [Accessed: Mar. 15, 2025].

[8] MEAN WELL Enterprises Co., Ltd., “LRS-350 Series Power Supply Selection Specifications,” [Online]. Available: https://www.meanwell.com/Upload/PDF/catalog_s.pdf. [Accessed: Jun. 15, 2025].

[9] Máy Văn Phòng Hà Anh, “Mô tơ nâng khay photocopy Ricoh 1060/2060/7500/6001/9001/6002/7503/503 (D2236747),” [Online]. Available: http://mayvanphonghaanh.com/mo-to-nang-khay-photocopy-ricoh-1060-2060-7500-6001-9001-6002-7503-503.html . [Accessed: Mar. 15, 2025].

[10] Precision Roller, “Genuine Ricoh AX040119 (D331-2680) Paper Lift Motor,” [Online]. ,Available: ,https://www.precisionroller.com/ax040119-for-ricoh/details_pn_29832.html . [Accessed: Mar. 15, 2025].

[11] MakerZone, “L298N Stepper Motor Driver Dual H-Bridge for Arduino,” [Online]. Available: https://makerzone.com.au/product/l298n-stepper-motor-driver-dual-h-bridge-for-arduino/ . [Accessed: Apr. 5, 2025].

[12] Điện Tử Nhật Tùng, “Module điều khiển BTS7960 43A – Driver động cơ DC,” [Online]. Available: https://dientunhattung.com/san-pham/module-dieu-khien-bts7960-43a-driver-dong-co-dc/ . [Accessed: Apr. 7, 2025].

[13] Điện Tử Nguyên Hiền, “Nam châm điện DC 12V – 0.8kg,” [Online]. Available: https://dientunguyenhien.vn/show/1106 . [Accessed: Mar. 15, 2025].

[14] Wikipedia, “Arduino Logo,” [Online]. Available: https://en.m.wikipedia.org/wiki/File:Arduino_Logo.svg . [Accessed: Apr. 20, 2025].

[15] Drexel University College of Engineering, “SolidWorks,” [Online]. Available: https://tech.coe.drexel.edu/software/solidworks/ . [Accessed: May 25, 2025].

[16] Điện Tử Tương Lai, “Phần mềm Proteus,” [Online]. Available: https://dientutuonglai.com/phan-mem-proteus.html . [Accessed: May 25, 2025].

[17] Chợ Trời Hà Nội, “Kit Arduino Mega2560 R3 tại Linh kiện điện tử 3M,” [Online]. Available: https://chotroihn.vn/kit-ardiuno-mega2560-r3-tai-linh-kien-dien-tu-3m-1 . [Accessed: May 25, 2025].

[18] IndiaMART, “Arduino Mega 2560 R3 Terminal Board,” [Online]. Available: https://www.indiamart.com/proddetail/arduino-mega-2560-r3-terminal-board-21650633612.html . [Accessed: May 25, 2025].

[19] L. Llamas, “Control up to 16 GPIO via I2C with PCF8575 and Arduino Expander,” [Online]. Available: https://www.luisllamas.es/en/control-up-to-16-gpio-i2c-pcf8575-arduino-expander/ . [Accessed: May 25, 2025].