Design and implementation of an affordable Braille Reader for Converting PDF Text into Refreshable Braille Output

By
Md. Hafizur Rahman
21321059
Mubtasik Nafe Mostofa
21321067
Nafis Ahmed Alvee
21121027
Mohammad Asifur Rahman Sheikh
21121026
Shabab Scimitar Chowdhury
21221024

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronics Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering
BRAC University
January 2026

Design and implementation of an affordable Braille Reader for Converting PDF Text into Refreshable Braille Output

By
Md. Hafizur Rahman
21321059
Mubtasik Nafe Mostofa
21321067
Nafis Ahmed Alvee
21121027
Mohammad Asifur Rahman Sheikh
21121026
Shabab Scimitar Chowdhury
21221024

Academic Technical Committee (ATC) Panel Member:

Dr. Touhidur Rahman
Professor
BRAC University
Department of Electrical and Electronic Engineering

Shahed Alam
Senior Lecturer
BRAC University
Department of Electrical and Electronic Engineering

Atib Mohammad Oni
Lecturer
BRAC University
Department of Electrical and Electronic Engineering

Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronics Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

Declaration

It is hereby declared that

1. The Final Year Design Project (FYDP) submitted is our own original work while completing a degree at Brac University.
2. The Final Year Design Project (FYDP) does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The Final Year Design Project (FYDP) does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Hafizur Rahman
21321059

Nafis Ahmed Alvee
21121027

Mubtasik Nafe Mostofa
21321067

Mohammad Asifur Rahman Sheikh
21121026

Shabab Scimitar Chowdhury
21221024

Approval

The Final Year Design Project (FYDP) titled "Affordable Braille Reader for Converting PDF Text into Refreshable Braille Output" submitted by

1. Md. Hafizur Rahman (21321059)
2. Nafis Ahmed Alvee (21121027)
3. Mubtasik Nafe Mostofa (21321067)
4. Mohammad Asifur Rahman Sheikh (21121026)
5. Shababab Scimitar Chowdhury (21221024)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 4th March, 2026.

Examining Committee:

Academic Technical Committee (ATC): (Chair)	______________________ **Dr. Touhidur Rahman** Professor BRAC University Department of Electrical and Electronic Engineering
Final Year Design Project Coordination Committee: (Chair)	______________________ **Dr. Mirza Rasheduzzaman** Associate Professor, Department of EEE, BRAC University
Department Chair:	______________________ **Md. Mossaddequr Rahman, PhD** Professor and Chairperson BRAC University Department of Electrical and Electronic Engineering

Ethics Statement:

This project titled 'Affordable Braille Reader for Converting PDF Text into Refreshable Braille Output' was completed by our group following the ethics standards. All of the mentioned work are our own except for the sections that have been cited and been given proper credit. All references have been added at the end of the report. Moreover, progress to the project was contributed by every member of the group. We acknowledge the authors of the works cited and maintain the proper academic standards in crediting them.

The similarity report provides an estimated 8% of our work is similar to previous texts which ensures and follows the rules for FYDP submission.

Acknowledgement

We concluded our work for ‘Affordable Braille Reader for Converting PDF Text into Refreshable Braille Output’ through perseverance and hard work. It was a great learning experience for us in many ways including technical engineering knowledge and stakeholder management.

We would firstly like to thank our knowledgeable ATC members for guiding us throughout this academic journey. The constant support from our ATC chair Dr. Touhidur Rahman helped us immensely in the completion of this project. Our respected ATC member Shahed Alam also provided valuable advice and resources which was instrumental for our success. Moreover, we received regular motivation and guidance from our ATC member Atib Mohammad Oni and we are extremely grateful to all of our ATC members.

Lastly, we would like to thank our friends, respected teachers of the EEE department and LTOs for their insights and overall help in completing our project.

Abstract:

This project demonstrates the development and the design process of an Electronic Braille Module Display which is affordable and user friendly. Additionally, it also broadens the accessibility for the visually impaired people. The processing unit, which is Raspberry Pi in our case, controls solenoid for Braille output. It works real time by translating digital texts (PDFs) into braille output. The process of hardware design emphasises on low-cost ensuring reliability and suitable for the education for the visually impaired people. The validation of functionality confirms the fast refreshing, tactile feedback. This project not only addressed the high priced barrier of existing braille modules but also promotes the braille literacy among the visually impaired people across the globe

Keywords—- Braille Reader, Visually Impaired, Refreshable Braille, Processing Unit, Digital Accessibility, Low-Cost Design

Table of Contents Page no.

List of Tables

List of Figure

Chapter 1: Introduction

1.1.1 Introduction

This paper includes the whole implementation of the electromechanical refreshable braille display module along with the design, development and full implementation of hardware. This project addressed the much needed accessibility gap faced by the people who are unable to see globally. Due to the high cost of braille displays that are available in the market, most people can not have access to them. By keeping this thought in mind, this Final Year Design Project aims to develop a low cost braille module which will do the same job as that of those available in the market. From the very start, this project progresses through several careful operating methods of the braille dots, and finally going with choice of Solenoid base driven and actuation ensuring low cost and effectiveness.

1.1.2 Problem statement and Background Research

According to the World Health Organization, recent data indicate that around 43 million people globally are visually unable to see.[6] In Bangladesh, the total population of unable to see individuals is approximately 750,000, which is about 1.5% of the current adult population.[7]

A vital tool empowering blind people to live independently, learn, and fully engage in society is Braille literacy. Still, especially in low- and middle-income nations, the lack of reasonably priced and effective Braille technology is a major obstacle. The World Health Organization (WHO) estimates that under 10% of visually impaired people in such areas have access to Braille-based learning resources [1]. Refreshable Braille displays that are currently available in the market are pricey and fragile mostly because of their dependence on piezoelectric actuators, which are very costly to produce and in need of very high operating voltages to operate. The high prices therefore prevent millions from accessing dynamic text material including e-books, navigation systems, and real-time communications. Therefore, making a low-cost, and robust refreshable Braille module would significantly increase the access to information for the visually impaired people. It would also help the visually impaired people to increase their literacy rates, help those people find work, which also means improve their life, and they will be able to be more inclusive on digital platforms. This solutions are not just for people but also various institutions including libraries, schools, and companies all over the world. Creating inexpensive assistive technology is more and more in line with world development goals as digital accessibility criteria (such as WCAG 2.1) and the UN Sustainable Development Goals (Goal 4: Quality Education) get more attention.Around the world, many efforts have been made to meet the demand for refreshable Braille devices:

Traditional commercial products like those from HumanWare and Freedom Scientific employ piezoelectric actuators to lift Braille dots [2]. Although consistent, these systems are quite pricey, usually running more than $3,000 for 40-cell screens.

Al-Faruque et al. and other researchers looked at SMA-based actuation techniques to cut power use and size [3]. Besides SMAs, refresh rates are poor and also have overheating issues, which makes it less efficient for it to work in real time.

Recent research has looked at soft actuators and micro-electromagnets to refresh Braille cells. Although Kim et al., proposed Braille display controlled by electromagnetic actuation which is less costly also has enhanced refresh rates and affordability [4]. But the long term issues were still there like strength as well as scalability.

Projects such as OpenBraille [5] have looked at 3D-printed parts to help lower expenses. Even if it has some creativity, it lacks tactile quality, also for force sensitivity and dimension it does not fulfill the ISO braille standard.

Most of the recent designs still can not make sure a combination of affordability, durability, standard compliance in one system although having advancement significantly. Many experimental designs also suffer from low refresh rates, constrained cell scalability, or mechanical fragility. A reasonably priced, portable, energy-efficient refreshable Braille module using electromagnetic actuation combined with microcontroller-based smart control for real-time content adaptation is still very much needed. The project is focused on developing an affordable solution to close the accessibility gap for the people unable to see by using advanced low-power electronics, driven by electromagnetic actuation, and low cost production methods (e.g., 3D printing).

1.1.3 Literature Gap

Countries like Bangladesh where the majority of the people are in poverty, can not afford to get braille displays that are available in the market, even though there is a lot of research going on over the past decades, advancement in technology still could not fill the void of affordability. Few research and methods come into light every now and then, but it never goes beyond the lab due to high costs again and not being able to reach the people who actually need them. The problems start to rise when the quality, cost and accessibility does not have the same direction. Even making parts locally is a hurdle because of the supply chain not working properly. Because of that often most projects ignore the actual feedback of the user which ends up with missing basic safety rules and function quality.

First of all, we can not have any real solution while having cost, performance and accessibility in mind. Most of the braille displays that are available in the market use piezoelectric tech to drive their devices, which are highly costly priced above 3000 US Dollars [2]. This price alone takes away the willingness of getting a braille display. Due to this high price, research labs and universities are trying to make and design a cheaper model to actuate the dots. Like Al-Faruque and teams developed a model where they used thin alloy wire to acute the dots to cut down the cost and bulkiness [3]. But later on reviews pointed out various problems such as sluggish reaction form the heating cycles, overheating, for live display needs constant update [10]. Also take Kim et al., research, his magnetic approach with speed looks good on prices as well [4]. But these magnetic parts depend on laser cut metal pieces and hand spun tiny coils which is a nightmare for production which also raises the cost. So it is rare to have a low priced dot actuating methods while getting the high output until now.

Another problem is there in those studies which is the local production gap. Works like OpenBraille, heavily relies on the parts networks of worldwide, also cutting edge technology like fine 3d printer and circuit board cutter [5]. Also for places like Bangladesh, often it takes months to get a specific product. Also the scarcity of the right technology has been a real problem as well. So one major thing we kept in mind while designing this project was to get the required product as easily as possible. We picked up small electronics components like diodes, mosfets and registers in the local market. Also as for the 3D printing of the required parts also available in different workshops throughout the city. Which makes sure that if any problem happens while using it, it would be fixed locally.

On the other hand, recent studies do not pay much attention to how to assemble everything together and use it in real time, especially while scaling up the project. If we look at the recent papers or past papers, it only focused on one specific problem to actuate one dot at a time [3], [4], [10]. Also there is paper or project that shows the whole project procedure like electronics, coding structure and physical layouts. Also to raise a single dot, a signal has to go through all cells in the circuit, taking care of sudden power drops and so on which they skipped entirely or showed things purely conceptual. But our work explains the project from the grounding up. From which electronics we are using, how it is actually working, all the 3D designs where all the components fit properly, coding structure on how to how the braille letters are shown in each cell.

Therefore, a clear gap exists and we aim to cover that in this project by focusing on affordability.

1.1.4 Relevance to Current and Future Industry

We can see a rapid growth over the last decade in the assistive tech markets because of the number of senior citizens increasing by day and the directives from the UN Sustainable

Development Goals and the Web content accessibility Guidelines, especially ensuring quality education and inequality reduction (SDG10) [8]. Also there is growing talk within the industry towards using more low-cost and user friendly devices. The importance of this work is that it demonstrates a way to disrupt the existing market for high-cost refreshable Braille Modules. The use of low priced and available components such as solenoids, Raspberry Pi, and integrated circuits (ICs) and production and design processes such as PCB and 3D printing makes this product well-suited for future industry.

1.2 Objectives, Requirements, Specification and constant

1.2.1 Objectives

- To design a low cost effective refreshable Braille cell.
- To ensure real-time tactile feedback.
- Able to take PDF as input and convert it to the braille output
- To ensure portability and compatibility with standard braille dimensions (ISO/IEC 24751-1).
- Also using machine learning to process text in an image into braille output

1.2.2 Requirements, Constraints and Specifications

Requirements are identified from the user's perspective and are categorized into two types: functional and non-functional. The necessary criteria for completing a project objective are called functional requirements, where non functional requirements are mainly based on user experience and aesthetic of the product

Functional Requirements

- The module has to show 6-dot Braille cells following ISO/IEC 24751-1 criteria.
- Real-time refreshing is required of the system; update time per cell is under 1 second.
- A tactile device has to give enough force for dot perception without unintentional flattening (1.8 kg·cm of torque).

- Convert the text (from pdf and images) to braille format accurately.

Non-functional requirement:

- Small design: Braille cell size ≤10 cm^3.
- Use of lightweight, long-lasting materials.

Constraints

- **Miniature Module:** The module needs to fit the existing standards of the braille tablets.
- **Budget constraint:** Must be very less per module (for scalability).
- **Manufacturing constraints:** Compatibility with standard 3D printing methods and electronic components.
- **Power Consumption**: Increasing the number of braille cells will also increase the power consumption.
- **Text Conversion Limitation**: The module can only translate plain text, it is unable to convert tables or mathematical equations.

Conflicting Requirements:

1. Producing a lightweight, portable device while also ensuring it is durable is a conflicting requirement.
2. Having more features can increase the price of the device.
3. Having a rechargeable battery with long battery backup without making the device too heavy.

1.2.3 Specifications:

Table-1: Product Specifications

Feature	Specification
Primary Function	Real-time PDF & Image-to-Braille conversion (OCR).
Output	Standard 6-dot tactile Braille cells.
Refresh Speed	< 200ms per cell (Instantaneous reading flow).
Power Input	Direct Supply (Optimized for low energy consumption).
Tactile Force	≥ 1.8 kg·cm (Pins resist retraction when touched).
Durability	500,000 actuation cycles per pin.
Noise Level	< 40 dB (Whisper-quiet operation).
Efficiency	Low-heat design with energy-saving actuation logic.

1.2.4 Technical and Non-Technical Constraints

Technical Constraints

- Raspberry Pi has limited GPIO pins, which demanded to use series to parallel shift resistor
- MOSFET switching and separate power rails needed
- Required careful design to ensure smooth and aligned dot movement.

Non-Technical Constraints:

- Budget Constraint: Need to make sure low budget to maintain affordability
- Manufacturability: locally or online the parts must be sourceable, and specialized tools must not used during assembly

- Time: All the implementation, testing and solving the various problems must completed with the given timeline

1.2.5 Applicable Compliance, Standards, and Codes

The standards that we followed to ensure safety, interoperability, and usability while doing the designing and implementation:

1. ISO/IEC 24751-1 (Individual accessibility and adaptability in training, e-learning, education): give the Braille dots physical dimensions and sensitivity of the force to have proper tactile recognition [7].

2. IEEE 12207 (Systems and software engineering – life cycle processes of the softwares: structured approach must be informed to development and integration of the system of the software.

3. WCAG 2.1 (Web Content Accessibility Guidelines) Principles: While primarily for digital content, the principles of Perceivable and Operable directly informed the need for reliable, accurate, and timely tactile output.

1.3 Systematic overview of the project

There are 43 million people all over the world who are visually impaired people, among 0.75 million people are in Bangladesh. Sadly, most of the people can afford to have the braille technology by which they get access to digital text such as pdfs. They have to rely on others to get access to those materials when they can do it by themselves by having a braille device. But they can not afford these gadgets due to the high prices. Now we tried to solve this problem by taking initiative to rethink the whole design process and making sure it works efficiently but costs only a little. Our goal is not just make a device but making sure to design something that is trustworthy, ready for bulk production as well as within the financial reach.

This project's main aim is to design a prototype of five braille cells where each cell will represent one letter at a time. Keeping in mind about the cost, we used solenoid based electro magnetic actuation to raise the dots. We also sourced the electronic component in the local market such Diodes, Mosfets, Shift resistors. We designed a novel circuit for the whole system and printed PCB layouts for each of the cells. We designed the physical layout in the 3D softwares which are mentioned in detail later on this paper. This whole design made sure the low cost also ensured the efficiency while following ISO standards for braille cells and force sensitivity.

Finally, All in all, this project and our effort of building a working model form a fresh idea which shows how actually the refreshable braille module worked effectively and also within the financial reach of the people. This project opens a door to the visually impaired people to get access to education, access to knowledge and fairness, which was beyond their reach due to the high cost.

1.4 Conclusion:

This project is being developed while focusing on the identified gap that we mentioned before. Our design and development tackles the lack of the low cost component for making a refreshable braille module as well as high prices for making an efficient device. Among all these obstacles, we were able to custom made our circuitry, able to get electronic components which are not costly and made the physical layout with local 3d printers which ultimately made sure an affordable low-cost braille display ensuring reliability, following the set standards and also efficient, yet each cell cost approximately 10 dollar. The cost for our product is far below the market price. Overall, this project is not just about the cost, it is also about how an engineer's design impacts society in establishing fairness. Especially in Bangladesh where less privileged people can not get access to this technology just because of not being able to afford it. Therefore, this solution will help those visually impaired and unprivileged people to get access to education.

.

Chapter 2: Project Design Approach

2.1 Introduction

Based on the actuation method we came up with two different approaches for our refreshable braille display. One is based on the principle of electromagnetism, using electromagnetic fields to actuate the dots. Another is using controlled servo motor rotation to linearly move the dots. Both designs are refined based on the feasibility, electronic safety, power efficiency and cost to evaluate the suitable approach for our project.

2.2 Identify Multiple Design Approaches

2.2.1 Design Approach 1: Electromagnetic Braille Dot Actuation Using Mini Solenoids

- **Working Principle:** In default state the solenoid plunger keeps retracted, thus the dot connected with the angular rod keeps lowered, representing the off state. When voltage is applied to the solenoid due to the magnetic field the plunger goes outwards in a controlled manner, pushing the angular rod upwards for a tactile sensory experience, representing the designated braille alphabet.

- **Electronic Control:** A Raspberry Pi serves as the central controller,reads the pdf file and extracts the texts. Then converts the alphabets into a braille signal with predefined braille alphabet mapping. Transmitting the data serially to the shift register (74HC595), which then drives the parallelly connected logic level mosfet (IRL540N). Which acts as a switch to actuate the solenoid.

2.2.2 Design Approach 2: Micro-Servo Motor Actuation This design employs standard micro-servo motors for precise angular control of the tactile pins.

Working Principle: Each Braille dot is mechanically coupled to a cam or lever mechanism attached to the servo's horn. The default position, rotation of 0°, is the lowered position of dots.

- Raising the dot: Since servo motors have 3 pins, the actuation process is slightly more complicated. The processing unit will send a signal to the PWM driver which will convert it to a PWM signal. This PWM signal will be sent to the servo’s control pin, which instructs the servo to rotate to a certain degree, in our case it is 60°. At the same time, a signal to the PMOS controls the power flow to the servo.

- Lowering the dot: Lowering dot works the same way as raising the dot. Once the PWM signal arrives from the servo driver, the servo rotates to the default position and the power is disconnected through PMOS.

2.3 Analysis of Multiple Design Approaches

Table-2: Comparison of Design Approaches

Criteria	**Approach 1**	**Approach 2**
Cost	Lower because solenoids are cheaper and driven by MOSFETs	Higher due to servo and expensive servo drivers
Efficiency	Consumes power during actuation only and it is lower than servo power consumption overall	Servos consume higher power during rotation and holding position
Manufacturability	Highly scalable thanks to the compact design	Less scalable due to bulkiness
Usability	Easier to setup and maintain, making it more usable	Complex design makes it slightly less usable and harder to maintain
Sustainability	More sustainable with lower components and power consumption	Less sustainable due to higher hardware needs and higher power consumption

2.4 Conclusion

From the comparison, a conclusion can be drawn that both designs have their merits, however the solenoid approach is a better fit for us. The important parameters were evaluated by us and we realized that for low cost and scalable operations, the solenoid approach is suitable. However, that is not to say the servo approach does not have any merit or the solenoid approach is perfect. As a result, further evaluation is done in the next sections.

Chapter 3: Use of Modern Engineering and IT Tool.

3.1 Introduction

Modern engineering tools are required to understand the different parameters and to compare them to arrive to an optimal solution. Furthermore, the tools are used for implementation of the project in both hardware and software sections. Understanding performance from the relevant values are extremely important. Therefore, This section explains the tools we chose and the reasoning behind them.

3.2 Selected Appropriate Engineering AND IT Tools

3.2.1 Software tool selection

- Python
- Proteus
- MATLAB
- Fusion 360
- Raspberry Pi Thonny

3.2.2 Hardware tool selection

- Raspberry Pi 5
- 74HC595 Shift Register
- IRL540
- Push-pull linear mini solenoid
- Buck converter
- Flyback diode
- **3.3.1 Software Tools**

Table-3:Software Tools

Tool	Application in Project	Key Features Utilized	Justification for Use
Python	Text-to-Braille conversion algorithm development; automation of PDF parsing	Python was the main coding language used. It converted given text to Braille mapping and signalled the solenoids to actuate through shift registers.	We used Python because it is very versatile. It integrates well with Raspberry Pi. This enables rapid development, reliable text-to-Braille processing, and precise low-level hardware control within a single, efficient software environment..
Proteus	Electronic circuit simulation (NMOS, servo control)	Virtual testing of control circuits Visualization of current/voltage response Debugging before hardware implementation	Allows validation of circuits (shift registers, drivers, PSU) without physical components, reducing prototyping costs and risks.
MATLAB	Performance analysis	Graph plotting such as Temp vs Time, Angular rotations vs Time and so on.	Allows for evaluations and parameter comparison for different approaches.
Fusion 360	3D modeling of Braille cell modules and enclosure	CAD-based modular design - Stress and durability analysis - Export to 3D printing	Enables creation of lightweight, durable, and ergonomic casing; supports rapid iteration and manufacturability checks.

Raspberry Pi 5 Thonny (Hardware + Development Platform)	Central processing unit for device (Braille conversion + actuation control)	2.4 GHz quad-core ARM Cortex-A76 CPU – Enhanced GPIO control with full support for Python and Linux OS.	Cost-effective, high-performance microcomputer; integrates software and hardware, enabling real-time processing, precise GPIO control, and scalable operation for multiple Braille cells.

3.3.2 Hardware Tools

Table-4:Hardware tools

Tool	Application in Project	Key Features Utilized	Justification for Use
Raspberry Pi 5	Functioning as the main processing unit	PDF extraction, OCR processing, GPIO activation	High processing power, cost effectiveness, scalable
74HC595 Shift Register	8-bit serial-to-parallel output	Used to reduce the number of GPIO pins needed for the project. First shift register needs 3 pins to operate and every subsequent shift register only needs one additional pin to function	Low power consumption, makes the project easily scalable
IRL540N	Used to control the power supply to the actuation system	Utilized for logic level activation and very low latency	Fast response, low power consumption, compatible with Raspberry Pi's 3.3V GPIO output
Push-pull linear mini solenoid	Main actuation device	5.5mm shaft actuation used for raising the 3D printed Braille dots	Compact size, relatively low power consumption, fast time response, scalable
Buck Converter	Supply the voltage to the solenoid	To efficiently step down the input	Output: 1.2V-36V High frequency low

		voltage while maintaining the high current drive necessary for solenoid actuation.	resistance capacitors Voltage surge protection 10A rated 95% efficiency
Flyback Diode	Protect from back EMF	Suppress inductive voltage spike during magnetic field collapse	1000V rated Forward current upto 1A Low leakage current Fast switching and rectification
AC-DC power adapter	Convert 120V AC into 24V DC	Convert the AC voltage into DC to operate the DC solenoids	Input: AC100-240V, 50/60Hz Output Voltage: DC 3V-24V Output Current: Up to 5A Efficiency: 95%

1. Raspberry Pi: Functioning as the main processing unit. Used for PDF extraction, OCR processing, GPIO activation.

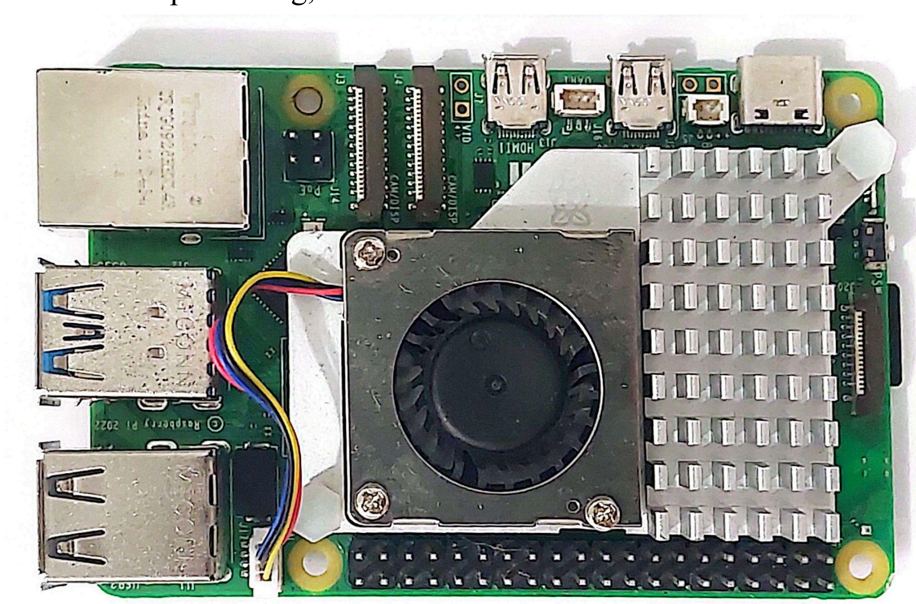

Figure 1: Raspberry Pi 5

2. 74HC595 Shift Register: Used to reduce the number of GPIO pins needed for the project. The first shift register needs 3 pins to operate and every subsequent shift register only needs one additional pin to function.

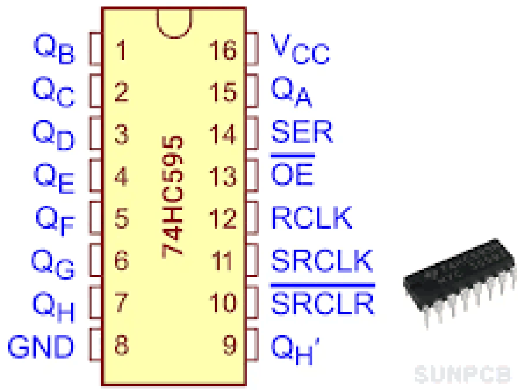

Figure 2: 74HC595 with its pin diagram

3. IRL540N: Used to control the power supply to the actuation system. Utilized for logic level activation and very low latency.

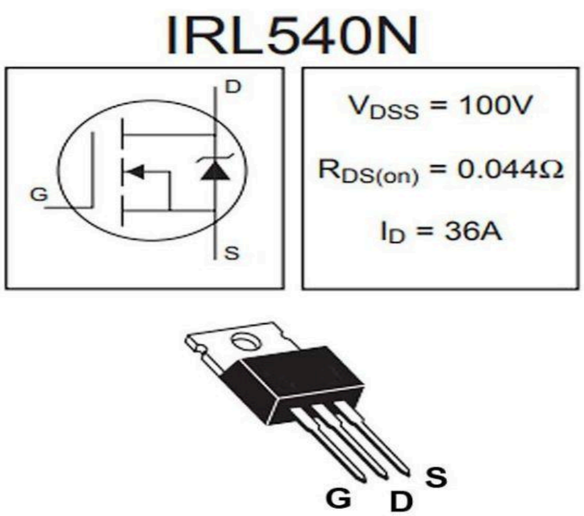

Figure 3: IRL540 N-channel MOSFET

4. Push-pull linear mini solenoid: Main actuation device. 5.5mm shaft actuation used for raising the 3D printed Braille dots.

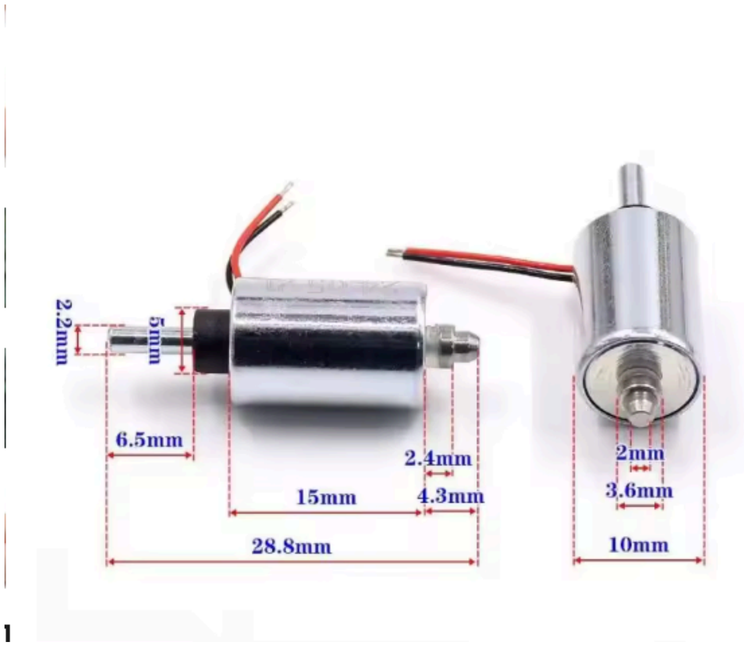

Figure 4: Mini solenoid with dimension

5. **Buck Converter:** To efficiently step down the input voltage while maintaining the high current drive necessary for solenoid actuation.

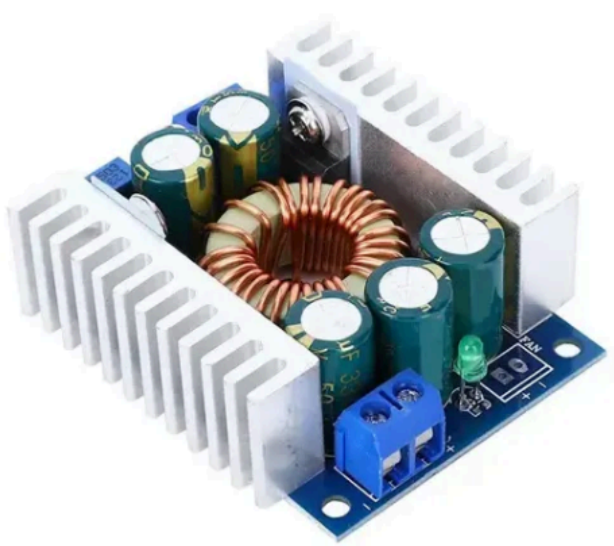

Figure 5: DC-DC Buck converter

6. **Flyback Diode:** Suppress inductive voltage spike during magnetic field collapse.

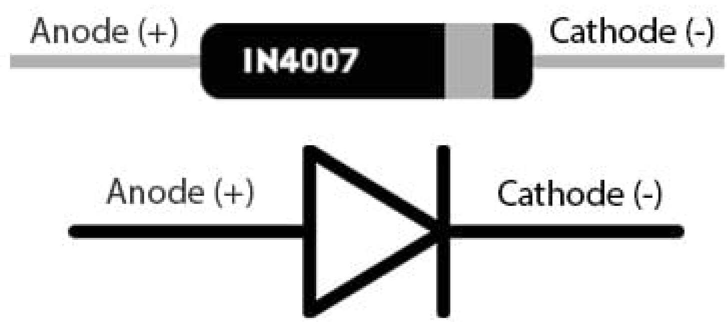

Figure 6: Flyback diode (1N4007)

7. **AC-DC Power adapter:** To convert household AC voltage into 24V DC voltage to match our solenoid requirements.

Figure 7: Variable AC-DC power adapter

3.4 Conclusion:

We identified and used the proper tools for our purposes. Platforms like Proteus and MATLAB allowed us to simulate the approaches and compare the results to arrive at the better design choice. In conclusion, knowing which tools to use is part of the project and it needs to be completed within the timeframe.

Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution.

4.1 Introduction

The existing refreshable braille modules that are available in the market are very expensive, for being able to compete in this assistive technology market, our project has developed two design approaches which will ensure efficiency and cost effectiveness. With approach 1 which is Solenoid based actuation and approach-2 which is Servo Motor actuation, our goal is to finalize the optimal solution ensuring the high refresh and response time, low cost and a reliable system for the users.

4.1.1 Design Approach-1:

Electromagnetic Braille Dot Actuation Using Mini Solenoids

This design employs the basic principles of electromagnetism by using miniature solenoids in a controlled order to move each braille dot to represent the respective alphabets to enable the reader to identify the alphabet effectively.

Working Principle

Each braille dot is actuated by a miniature solenoid (10mm diameter) with 5.5mm stroke length, Positioned beneath the braille dots with an angular rod. In its default state, the solenoid shaft remains retracted, keeping the Braille dot in the lowered position.

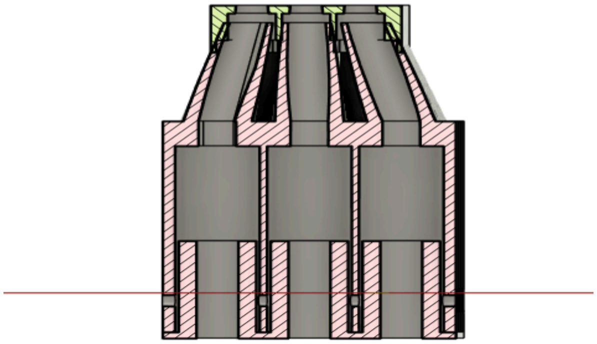

Figure 8: Cross section of solenoid based actuation mechanism

Raising a Dot: When a voltage is applied to the solenoid coil, it becomes energized and generates a magnetic field that pushes the solenoid plunger outward. As angular rods connected to the braille dots are placed just above the solenoid coils, the upward linear motion of the solenoid plunger is mechanically transferred and consequently pushes the braille upward (ON state). The right pattern of upward position of braille dots represents a significant alphabet. The tactile formation of braille dots allow the user to recognize the alphabets and read the text effectively. To represent G (G=0b00001111) solenoid 1,2,3,4 are energized.

Lowering a Dot:

As a part of braille alphabets some dots need to be in OFF state(lowered). When the voltage is removed, the magnetic field collapses. The built in stabilizing spring mechanism pulls the plunger in default position (lowered). As the plunger is pulled, due to gravitational force the angular rod connected to the braille dot is also lowered. To represent H (H=0b00001101) after G(G=0b00001111) voltage is removed from solenoid 2, which will lower the braille dot 2.

In order to provide protection to the circuit, a flyback diode is connected parallel with each of the solenoid actuators. It suppresses the high voltage inductive kickback generated by the coil when the magnetic field collapses and protects the processing unit and electronic components.

System Architecture and Block Diagram

The electronic control system for this approach is designed for scalability:

1. **Central Controller (Raspberry Pi 5):**
   The raspberry pi takes the input (pdf or jpg) file and reads the file to extract the text from the file. Then using predefined binary braille mapping (e.g., 0b00000001 for **A**, 0b00001111 for **G**). converts the alphabets in braille format. Each bit represents the ON/OFF state of a designated solenoid corresponding to a braille dot. After conversion sends the signal to each gpio pin dedicated to the braille reader.

2. **Shift Register (74HC595):**
   The binary Braille pattern is transmitted serially from the Raspberry Pi to the data input pin(DS) of the 74HC595 shift register through gpio pin. The 8 bit SIPO(Serial Input Parallel Output) sends the data parallel to six MOSFET connected to the output pins(Q1-Q6) of the shift register. Each bit of the binary braille signal is shifted on the rising edge of the shift clock(SHCP). After all the bits are shifted to the internal shift register the latch pulse (STCP) is applied, which moves the signal from the internal shift register to the output latches. The clock and latching ensures that the signal is moved to the parallel output at the same time, avoiding flickering which will disrupt the reading process of the user. Additionally, as the clock and latch are shared between all the shift registers, it minimizes the number of required gpio pins.

3. **MOSFET Switching Stage (IRL540N):**
   Each of the output pins of the shift register is connected to the gate of an IRL540N, N channel logic level MOSFET. Here the MOSFET is used as an electronic switch as the gpio pins can supply upto 20mA current at max and the solenoid draws 190mA. So, the solenoid is supported with an external power supply and the MOSFET used as an electronic switch.
   A pull down resistor is connected to the gate of the MOSFET to ensure LOW state to avoid floating during latch transition. Which helps to avoid flickering. The solenoid is placed in series with the drain and the DC supply, with mosfet source grounded. When the shift register sends a HIGH signal, the gate is driven HIGH and the MOSFET turns on, allowing the current to flow through the solenoid from drain to source. WHen the signal goes to LOW, the pull down resistor discharges the gate capacitance, turning the MOSFET OFF in a controlled manner to avoid noise.

4. **Flyback Diode:**

A flyback diode is connected parallel to each solenoid. As the solenoids are inductive load, while the voltage turns off, the magnetic field collapses generating a high voltage transient also known as back emf or inductive kickback. The sudden voltage spike can exceed the breakdown ratings of the circuitry and also the processing unit (Raspberry Pi in our case).
Thus the flyback diodes work as a low impedance route for the voltage spike to slowly decay while protecting the IRL540N MOSFET, 74HC595 shift register, Raspberry Pi from any potential damage. Additionally, it reduces electromagnetic interference and system stability during fast switching of the solenoids.

5. **Power Supply Unit:**

A dedicated DC power supply provides the operating voltage and current for the solenoids, as pi gpio pins can support neither the voltage nor the required current. The use of dedicated power supply ensures Pi safety. A buck converter is used as the supply unit as it can step down the voltage.
Mains the AC power is converted to DC using a AC-DC adapter, then a dedicated buck converter is used to further step down to 8V, matching the solenoid's operating voltage. This multi-stage power architecture is used to allow the user to connect directly to the household supply lines, isolate the high current drawing solenoids from the PI ensuring system stability and protecting electronics from load transient.

6. Switch:

The switches act as an user navigation interface. In this formation two switches are connected to the gpio pins configured with a 10k pulldown resistor to ensure constant LOW state. Which will prevent flickering, noise and accidental triggers. The other side of the switches are connected to the VCC pins. When the switch is pressed the circuit is bridged with the VCC, sending HIGH at the gpio pins. The rising edge triggered pi considers the HIGH trigger as an interrupt to operate certain tasks. This "one-press" logic mechanism is refined by the use of a debouncing algorithm, which filters out the rapid, mechanical switching of the contacts, so that one physical push of the switch corresponds to one command without missing characters.

These switches are basically part of the user interface, as it allows the user to read at his own pace. When the first switch is pressed the system increments the index of the extracted character string, and sends the defined binary signal to the designated pins. Eventually moving to the next word or the sentence. On the other hand, the second switch decreases the index of the extracted string and prints the previous word or sentence.

Rather than a fixed scrolling speed, this switch mechanism allows the user to pause at complex words, backtrack to further clarify, mimicking the process of a physical braille page.

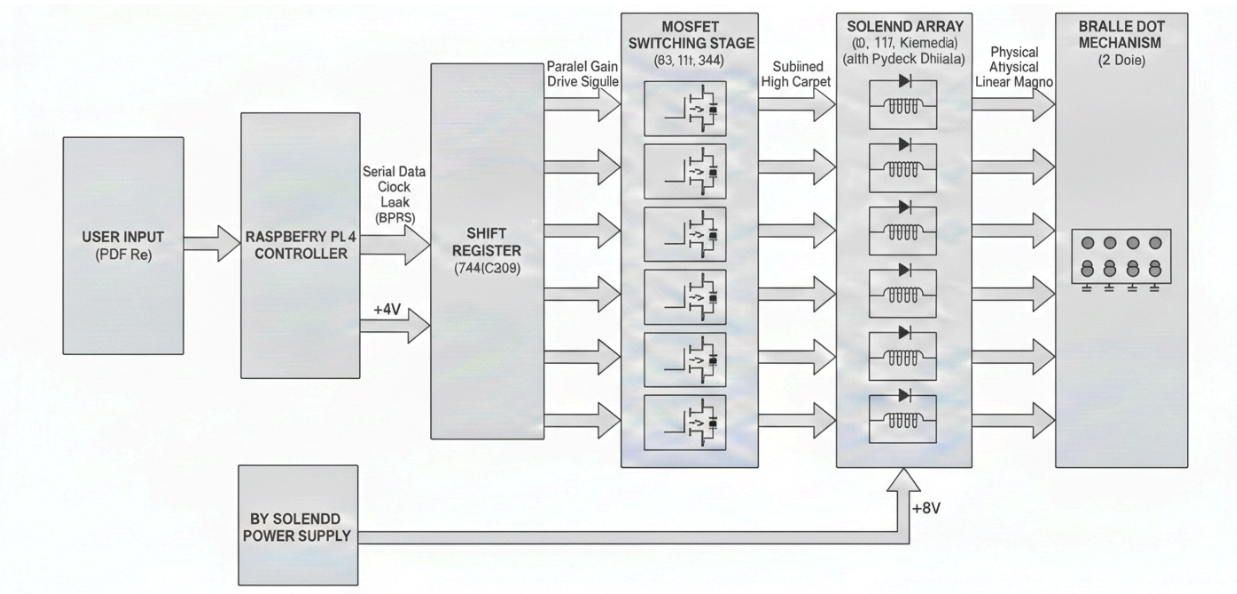

Figure 9: Block diagram of Solenoid actuation

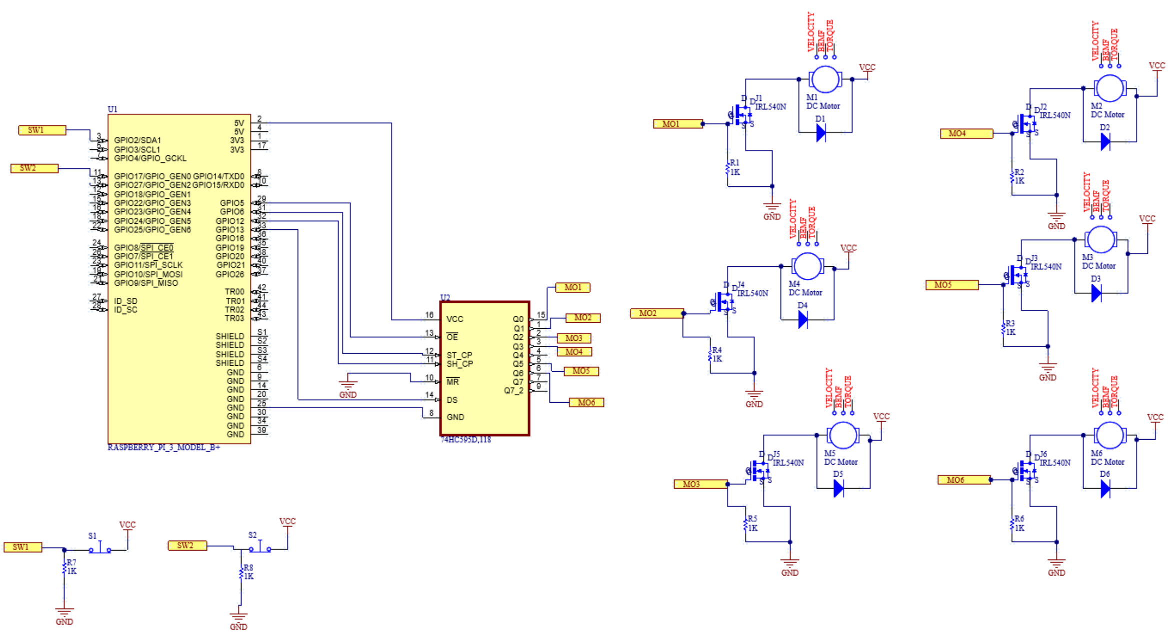

Figure 10: Schematic of approach-1 circuit diagram for single cell. (solenoids represented as DC motor)

Components Modeled: Raspberry Pi, 74HC595 shift register, IRL540n, 1N4007 diode, and DC motors to visually represent the actuation (through the electromagnet).

Simulation Outcome: The simulation verified correct serial data transfer from the Raspberry Pi to the shift registers and proper MOSFET switching based on the Braille bit patterns. The observed output behavior confirmed accurate control logic and signal timing for reliable solenoid actuation.

Design Approach-2:

4.2 Design Approach 2: Micro-Servo Motor Actuation

Compact servo motors are used for Braille dot movement. Micro servos provide precise movement using feedback systems.

4.2.1 Working Principle

The servo motor actuation process converts the angular rotation of the servo motor into vertically upward/downward motion of the braille dots connected with the servo shaft using a lever mechanism.

- **Closed-Loop Control:**The servo motors act like a closed loop feedback device to maintain the predefined position(angle). The closed loop feedback system allows precise control of the braille dots position. Avoiding any physical error and unintentional actuation of the servo.[1]
- **Tactile "Raised" State:** To raise the braille dots into ON state the servo motor is fed a PWM signal to move 60 degrees anticlockwise, which pushes the shaft attached to the braille dot, eventually raising the dot vertically allowing the user having a tactile feel to for efficient reading.
- **Tactile "Lowered" State:** For the lowered state the servo motors needs to be back at its default position (0 degree). The PWM signal moves the servo motor to 0 degrees or 60 degrees clockwise from the ON state position. Consequently, lowering the connected braille dot shaft.

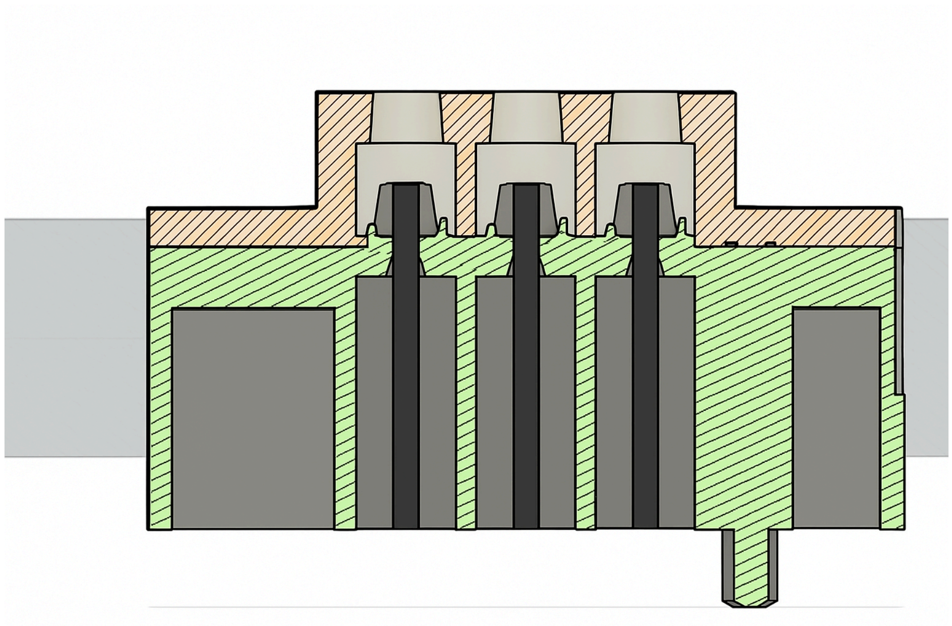

Figure 11: Cross section of servo controlled braille cell

While servos are mechanically simpler to implement, they require continuous power to maintain their position, which is a significant drawback for energy efficiency.

4.2.2 System Architecture and Electronic Control

We designed an optimal control architecture for our purpose.

1. **Central Processing Unit (Raspberry Pi):** The processing unit takes the text input and coverts them to Braille signal following the Braille map. It uses I2C protocol.
2. **PWM Expansion (PCA9685 Driver):** A PWM Driver is needed to control a large number of servos since GPIO pins are limited. With a driver, only 2 GPIO pins are needed for 16 servos.
3. **Active Power Management (PMOS Switching Stage):** Since servos can draw power even when inactive, a PMOS provides power to servos only when they need to be activated.
4. **Decoupled Power Supply:** A separate power supply is used for Raspberry Pi to ensure proper operation. All grounds are connected together.

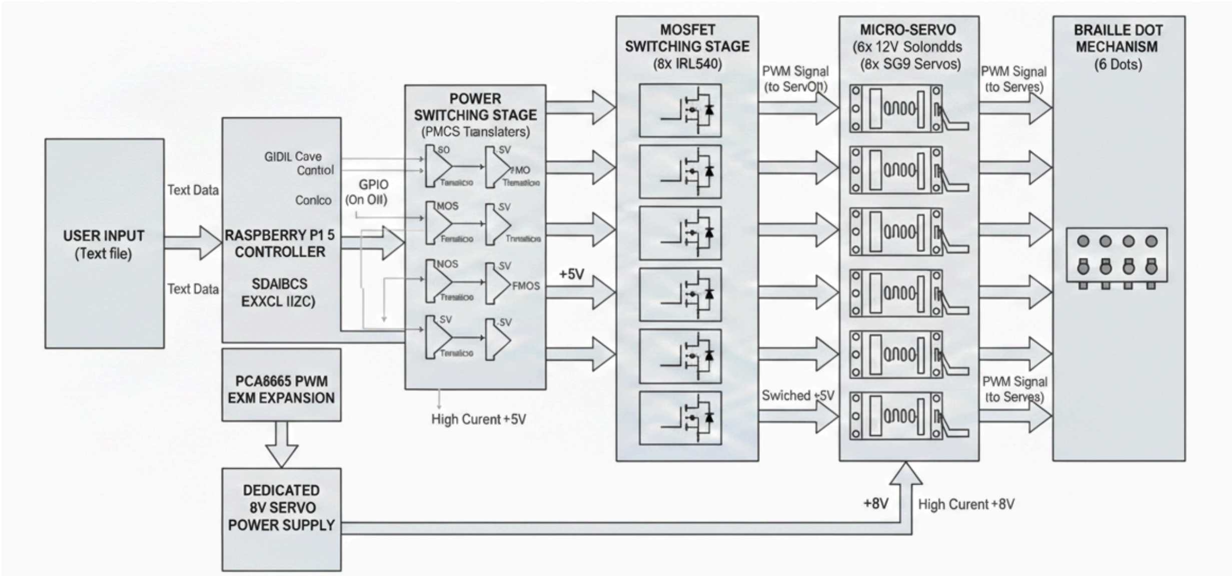

Figure 12: Servo based Braille Dot Actuation system block diagram.

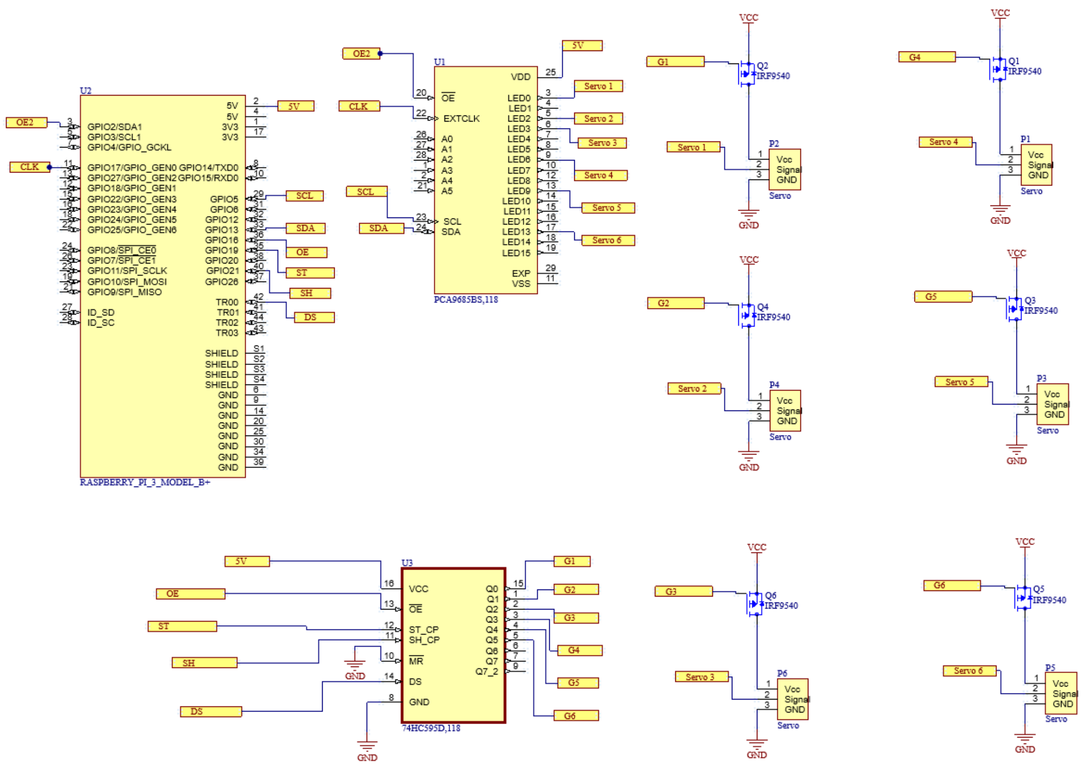

Figure 13: schematic of the servo actuation approach (for one cell)

4.2.3 Analyze the Multiple Design Solutions:

The finalized architecture is based on using a cylindrical solenoid with outward actuation system to move each braille dots vertically. Braille dots are connected to the plunger of the compact solenoid with an angular shaft. A dedicated PCB with integration of IRL540n logic level MOSFET, 74HC595 shift register takes the input signal and sends it to the solenoids while also supplying the solenoid actuating voltage at the drain of the MOSFET. The processing unit processes the input file (JPG/JPEG/PDF) converts into a binary array and sends the signal while maintaining clock and latch. The built in spring mechanism in the solenoid pulls the solenoid plunger into OFF state when there is no HIGH signal.

Considering all the advantages of the approach we moved to the solenoid actuation due to its fast response time, less electrical and mechanical complexity, scalability and reliability for a physical prototype.

4.3 Identify optimal design approach

4.3.1 Time Response:

Approach-1:

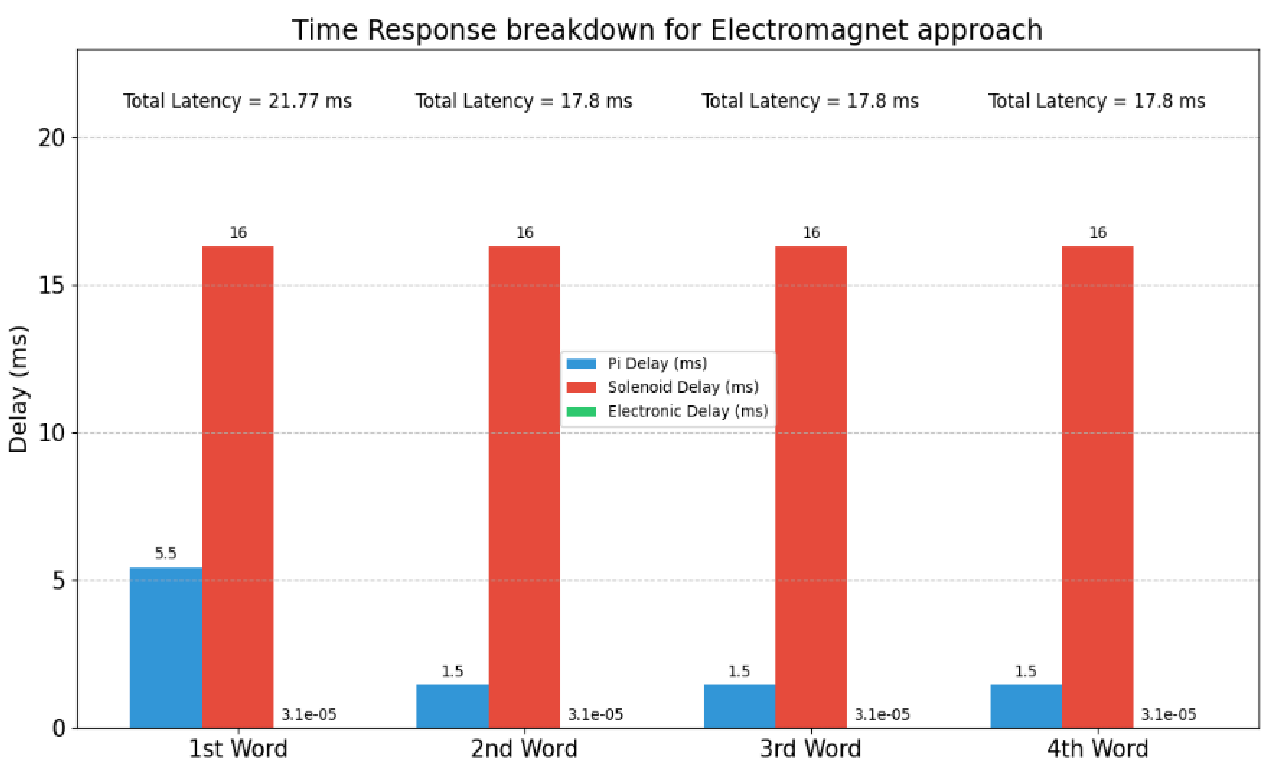

Figure 14: Time Response breakdown for Electromagnet approach

Technical Justification of System Latency Analysis

We can see that the entire system is fast as it works within a fraction of a second. Moving on, to further understand the process, we have divided the total time in various sections and explained the time delays.

1. Processing Latency

The processing delay, which is experienced by the processing unit, was determined by us through code that tracks the delay. The phases of these delays are:

- 1st Word Overhead (5.45): The time it takes to process the first word in the PDF is higher than the subsequent words. The table below provides further information.
  Table 5: PDF parsing latency

PDF extraction time	4.78ms
Text parsing time	0.16ms
Gpio output time	0.45ms
Total latency	5.40ms

- Sequential Optimization (1.48ms): Any words encountered after the first word consumed less time to process because the PDF has been parsed by the Raspberry Pi by then.

Table 6: Latency of switching to next word

Button pressed	**Word Displayed**	**Latency**
1	HELLO	1.48ms
2	WORLD	1.45ms
3	THIS	1.46ms

4	IS	1.46ms
5	BRAIL	1.46ms

2. Solenoid Actuation Latency (16.32ms)

Direct hardware approach was taken by us to understand the solenoid actuation delay.

- Measurement Method: There is a voltage dip when voltage is applied. As a result, this timing can be used to understand the amount of time it takes for actuation.

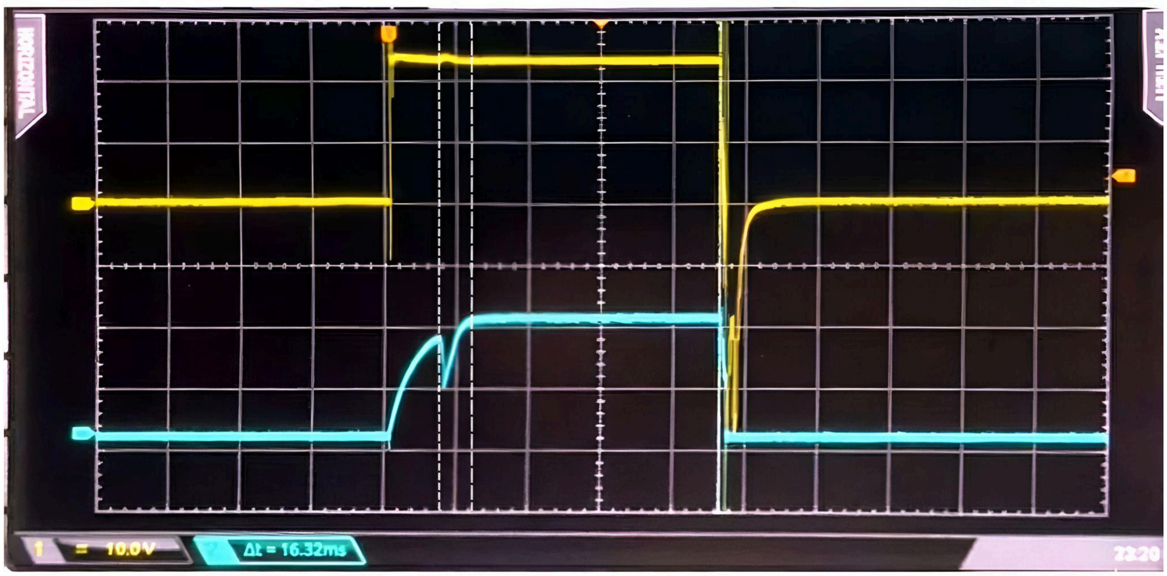

Figure 17: Measurement of solenoid latency

- Mechanical Constraint: As we know, the plunger needs to travel 5.5 mm for actuation. Our measurements show that this time is 16.32ms and it is the largest delay in the system.

3. Electronic Switching Latency (0.000031 ms)

- Electronic Component: The shift register 74HC595 and the MOSFETs have an extremely small delay for activation.

  Total electronic delay= Shift Register delay+ MOSFET turn on delay

Total electronic delay=20ns+11ns
Total electronic delay=31ns

- Nanosecond Level Activation: Obtained from the respective datasheets, we can observe that the electronic components operate on nanosecond scales.

Approach-2

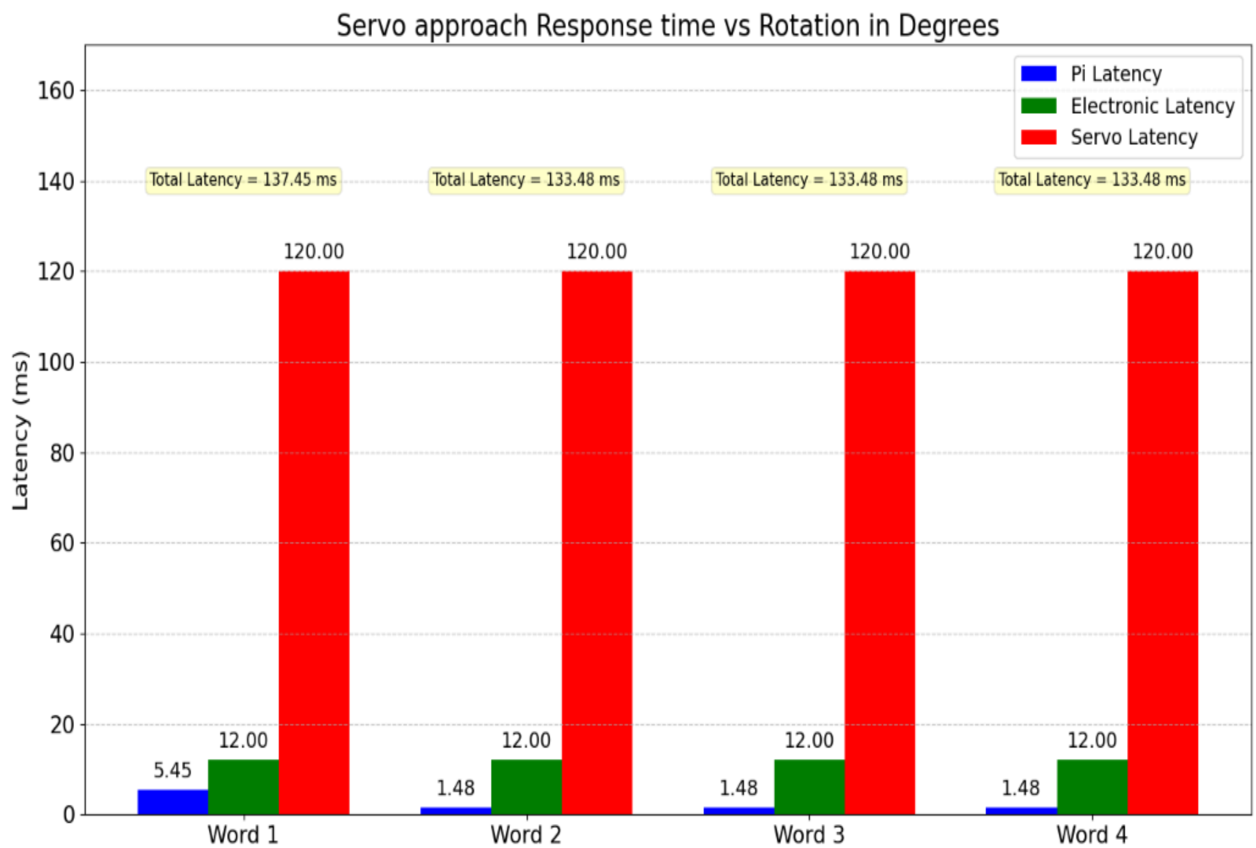

Figure 18: Servo Approach Response Time vs Rotation in Degrees

Latency Breakdown and Calculation

The total timing breakdown can be observed from this graph. The explanation for this timing is given below:

1. Pi Latency (Processing)

The Pi latency represents the time taken for the software to process the command and send it over the I^2C bus.

- Word 1 (5.45ms): this includes extracting the text from pdf converting into binary braille format and sending the signal to gpio pins.
- Words 2-4 (1.48ms): Here the pdf is already extracted, the system is only sensing the switch and moving to the next word.

2. Electronic Latency (12ms)

This is the "Physical Communication" delay. Our chosen SG90 servo operates at 50Hz frequency, meaning every 20ms a PWM pulse is being sent.

- Justification: On average, the command from the Pi will arrive in the middle of a PWM cycle. The PCA9685 must wait for the next cycle to update the pulse width.
- Calculation:
  Average wait=20ms x0.6
  Average wait=12ms

3. **Servo Latency** (120ms)

This is the dominant bottleneck, representing the mechanical movement of the motor.

- Justification: A typical hobby servo has a speed rating of approximately 0.2s per 60 degrees.
- Calculation for 60 degree movement
  The servo moves at a rate of 2 ms per degree.
  T = 60 degree x 2ms/degree=120ms

As seen in the red bars, the physical movement takes up **over 85%** of the total response time.

Total System Latency

The total latency for the first word is:

5.45(pi)+12ms(electronic)+120ms(servo 0-90degree)=137.45ms

4.3.2 Power Consumption:

This graph represents the total power consumption of the approach-1, solenoid actuation method across five distinct operational scenarios. Two primary parts are the variable load of solenoids(according to alphabet) and the static base load from the control unit.

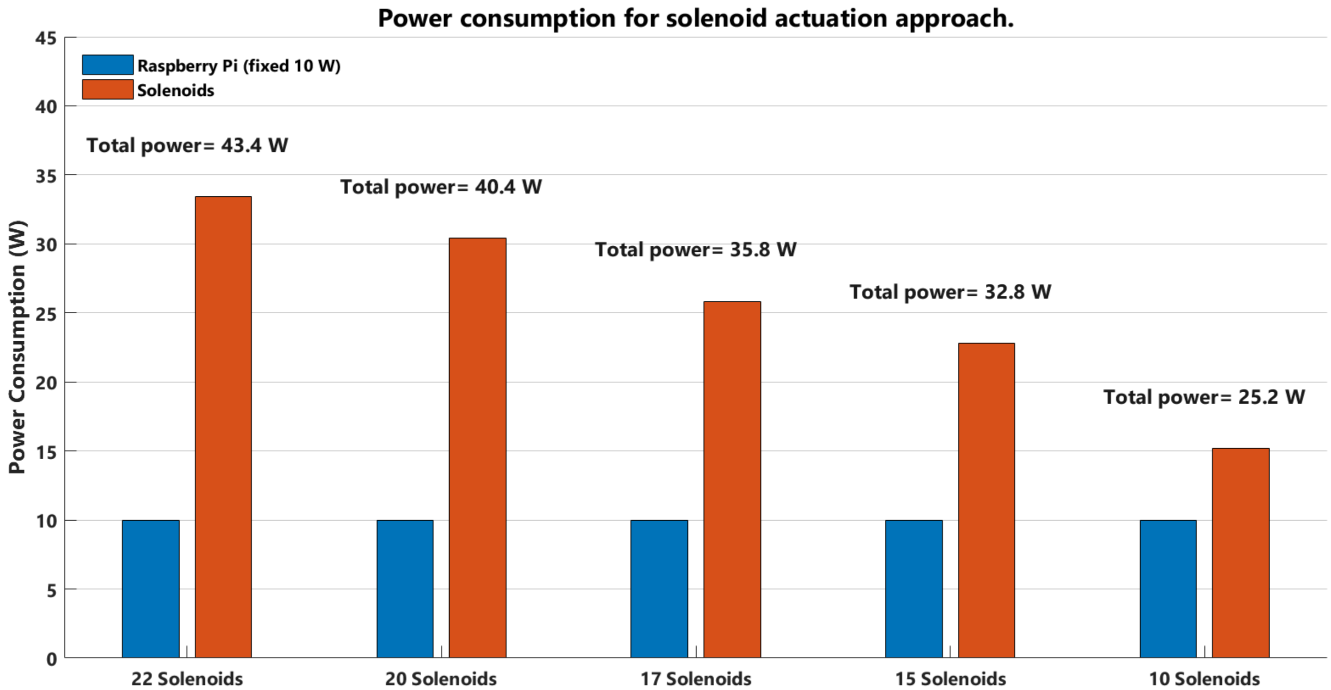

Figure 19: Power consumption for solenoid actuation approach.

1. Solenoid Power Justification

The operational power consumption is directly proportional to the numbers of solenoids actuating at a time.

- Individual Load: Each solenoids draw 190mA(0.19A) at 8V.

$P_{solenoid} = V \times I$

$P_{solenoid} = 8V \times 0.19A$

$P_{solenoid} = 1.52W$

- Operational Context: The 5 letter prototype will be able to print 5 Braille characters at once. A typical Braille cell contains 6 dots, so the numbers of solenoids active will vary greatly with the letters forming.
  - Light Load: Simple letters ('a', 'b') may use only 1–2 dots(1-2 solenoids).
  - Heavy Load: Complex letters ('q', 'y') or contractions may use 4–5 dots(4-5 solenoids).
- Scenario Modeling: The graph represents power consumption for 5 letter batches from low density scenario(10 dots) to high density scenario(22 dots).

2. **Control Unit (Raspberry Pi) Justification**

Here we have used raspberry pi as the controlling unit, which does the pdf parsing, extracting text, converting each alphabet of the string to a binary representation, sending the signal to gpio and controlling the clock-latch mechanism.

- Assumed Load (10W): Though the average idle power consumption is 3-5W we have allocated 10W to consider the peak cpu usage translation and OCR. Additionally for attached peripherals(cooling fan) and gpio pins(15mA, 3,3V).

Table-5 : Solenoid power consumption

Scenario (Active Dots)	**Solenoid Power Calculation**	**Solenoid Total**	**Pi Power**	**Total System Power**
22 Solenoids	22 x 1.52W	33.44W	10W	**43.44W**
20 Solenoids	20 x 1.52W	30.40W	10W	**40.40W**
17 Solenoids	17 x 1.52W	25.84W	10W	**35.84W**
15 Solenoids	15 x 1.52W	22.80W	10W	**32.80W**
10 Solenoids	10 x 1.52W	15.20W	10W	**25.20W**

Approach-2:

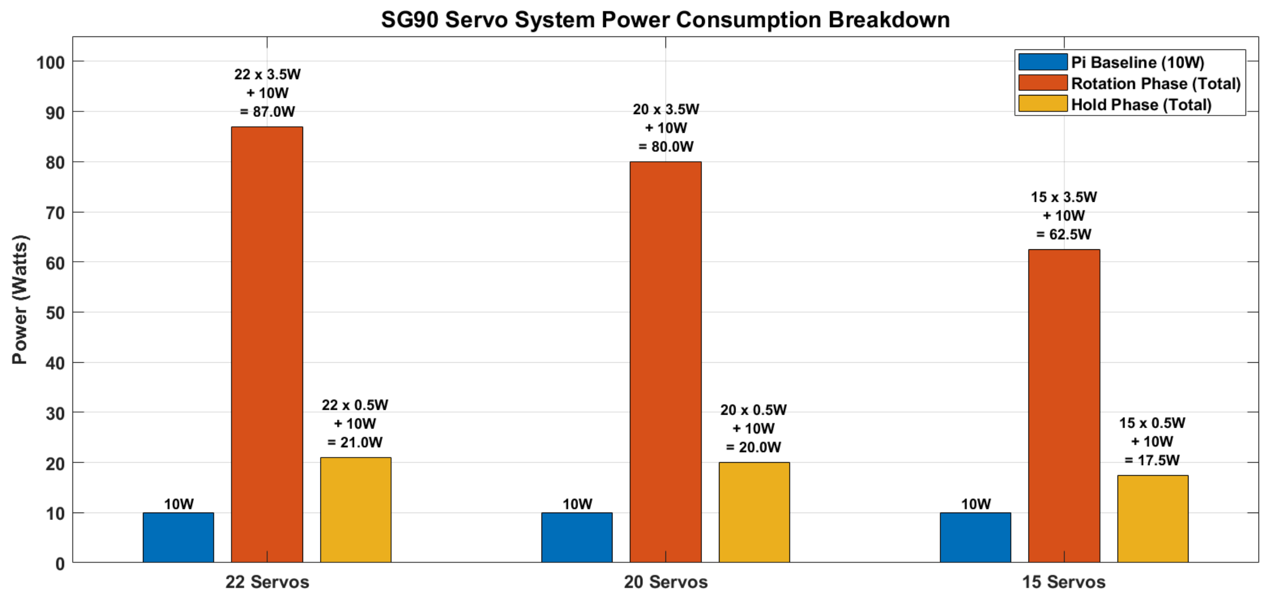

Figure 20: Servo Power Consumption Breakdown

1. Individual Servo Load Justification

Each SG90 servo motor operates in two electrical phases during the Braille dot actuation process:

- **Rotation Phase (0 to 60 degree):** When the system sends signal to raise a dot, the servo motor actuates and moves from 0 degree to 60 degree to fully elevate Braille dot, the servo operating at 5V draws approximately 700mA (0.7A).
  - Power: 5V x 0.7A = **3.5W per servo**.
- **Hold Phase (maintain ON state):** After elevating the dot to maintain the position it draws approximately 100mA.

  power: 5V x 0.1A = **0.5W per servo**.

2. Operational Context

Our device prints 5 characters at the same time. Each cell has 6 dots and the number of solenoid activated depends on the letter.

- **Light Load:** Letters such as 'a' or 'b' may only require 1–2 active servos per cell.
- **Heavy Load:** Complex letters like 'q' or 'y' or specific contractions can require 4–5 active

servos per cell.

- **Scenario Modeling:** The graph analyzes 3 likely real life scenario where different number of solenoids are active. 22 for high load, 20 for medium load and 15 for low load.

Table-6 : **Servo calculation**

Scenario	Rotation Phase Calculation	Hold Phase Calculation
22 Servos	(22 x 3.5W) + 10W = **87.0W**	(22 x 0.5W) + 10W = **21.0W**
20 Servos	(20 x 3.5W) + 10W = **80.0W**	(20 x 0.5W) + 10W = **20.0W**
15 Servos	(15 x 3.5W) + 10W = **62.5W**	(15 x 0.5W) + 10W = **17.5W**

Comparison parameters verification Results:

Table-7: Result Comparison

Module Type	Response Time (Seconds)	Power (W)	Cost(TK)
Servo Motor	**136ms (60° rotation)**	**80 (During rotation for 20 servos)**	**37800**
Solenoid	**21.77ms (First Word)**	**40.40 (20 solenoids)**	**33600**

Weighted Decision Analysis for Design Alternatives

Table-8:Weighted Decision

Criteria	Weight	Solenoid Actuation (Score 1–5)	Weighted Score	Servo Motor Actuation (Score 1–5)	Weighted Score
Response Time	5	**5** (21.77ms)	25	**2** (136ms)	10
Power Consumption	4	**4** (40.40W, 20 solenoids)	16	**2** (80W,During rotation for 20 servos)	8
Cost	5	**4** (33600tk)	20	**3** (37800tk)	15
Scalability	4	**5** (Highly scalable due to compact size and MOSFET control)	20	**3** (servo count limits scaling)	12
Ease of Maintenance	2	**4** (simple replacement of solenoids)	8	**3** (servo gears harder to repair)	6
Total Score	—	**89**		**51**	

The multi-criteria evaluation confirms that **Solenoid Actuation** is the best design approach for achieving the project's goals of **affordability, reliability, and energy efficiency**. The servo approach, while functional, is less suited for long-term, scalable deployment.

Justifications For Weighted Scores:

1. **Response time:** Comparing to the standard Piezoelectric Braille Tablet time response, which is 100ms for single lines [14], we can conclude that solenoid design is deserving of the high score. Meanwhile, the 136ms servo motor response time is decent considering the standard for Piezoelectric modules, it is significantly less suitable than solenoid design. This is the reason for the score it was assigned.
2. **Power Consumption:** Solenoids in general consume less power than servo motors because they only draw current when actuated. The actuation power consumption is also lower servo rotation or servo hold power consumption. Both designs have higher power consumption than a typical Piezoelectric tablet which is 12W for full activation [14]. This is one of the constraints of our designs. We aim to address the issue in our future work.
3. **Cost:** The price of the module is one of the strongest aspects of our project. Typical Piezoelectric modules have very high price ranges. We have provided detailed economic analysis in chapter 8. Our designs are multiple times cheaper than standard Piezoelectric modules. Both of our designs are close in cost but since solenoid design is slightly more inexpensive, we have assigned '4' to the cost of solenoid design.
4. **Scalability:** Solenoids are compact devices capable of easy scalability. The dimensions are 28.88mm in length and 10mm in diameter. The MOSFET control system is also scalable since the components can be arranged in a compact Braille cell. Meanwhile the servo design is bulkier and less compact due to motor, gearbox, wiring and relatively large servo drivers. Scaling servo design will also significantly increase power consumption, reducing scalability. Therefore the score assigned to the designs is justified.
5. **Ease of maintenance:** Solenoids are mainly composed of coils and plungers. Solenoids are easier to repair because issues are usually caused by coil burnout or wear and tear. It is easy to replace because of the simple circuitry. MOSFETs and other components are easily replaced as well. Servos are mechanically more complex. Complex linkages increase maintenance time and cost. Both of our designs are easier to maintain compared to Piezoelectric Braille displays because they consist of fragile and very small components[14]. Moreover, they use high voltage circuitry which is expensive and complicated to repair. In conclusion, solenoids are a superior choice in this category.

4.4 Performance Evaluation of Developed Solution

We considered several parameters to understand the performance of this design approach. They are listed below:

Processing and Extraction Accuracy (OCR)

In order to check the effectiveness of the system in processing the visual data, a test image in JPG format with 50 letters was fed into the system. The Optical Character Recognition (OCR) and text extraction process, which is a part of the computational logic implemented in the Raspberry Pi, successfully recognized 42 out of 50 letters, thereby showing an accuracy of 84%.

- **Character Misinterpretation**: The system misread 8 letters during this test.
- **Common Confusions**: The errors were primarily due to visual similarities between characters; the system read 'B' instead of 'R', and struggled to distinguish between 'U' and 'V', as well as 'D' and 'O'.
- **Impact on Performance**: While these errors are software-based rather than mechanical, they represent the current threshold of the system's parsing logic for non-digital (image-based) inputs.

Mechanical Reliability

The mechanical consistency was evaluated by running the program 30 times and documenting the output.

- **Reliability Rate:** Out of 30 documented runs, it produced perfect results 26 times, which means an 86.67% mechanical success rate.
- **Mechanical Errors:** In 4 instances, one or a few dots could not actuate according to the defined letter. Some got stuck in the frame and could not retract. This was traced to physical constraints, friction between the Braille cell, misalignment, and accuracy of the 3D printer.
- **Electronic Stability:** The electronic signal chain, which employs the 74HC595 shift register and the IRL540N MOSFET, registered 100% success rate with an "instantaneous" electronic switching latency of 31 ns.

Response Time and Latency Analysis

The system is highly optimized for speed, ensuring a fluid reading experience:

- **Initial Response**: **5.45 ms** is the delay of this section as it converts text to Braille map.
- **Optimization**: **1.48 ms** is needed to process subsequent words
- **Mechanical Speed**: The physical travel distance is covered in **16.32ms**

Scalability and Power Management

The architecture of Approach-1 is designed for affordable mass production and expansion:

- **Scalability**: Since we are using shift registers, we can scale up the device easily since we require very low number of GPIO pins.
- **Power Efficiency**: We observe that the system has predictable power consumption. Each solenoid draws 1.52W during activation.

4.5 Conclusion

In conclusion, the experimental results and performance metrics clearly indicate that Approach-1 (Solenoid Actuation) is the best configuration for a maximum efficiency, cost-effective Braille reader. As the response time of the proposed system is only 21.77 ms, which is almost six times faster than the alternative solution using a servo motor, the proposed system provides a near-instantaneous experience to the user with a peak power consumption of only 40.40 W.This technically superior is paired with a more affordable price range of 33,600TK which is 4,200Tk less than the approach-2. Though there is more room for improvement as the performance evaluation portrays that the image to text conversion using OCR is still having some character identification problem (U/V, O/D etc) and the mechanical reliability is still not at its best due to friction and inaccuracy of 3D fabrication. In spite of our prototype still can't match the market products, the 84% accuracy of OCR and 86.67% mechanical accuracy can be enhanced. In conclusion, considering overall performance evaluation and scalability the approach-1 is the most suitable prototype to empower the visually impaired with a smart and affordable literacy tool.

Chapter 5: Completion of Final Design and Validation.

5.1 Introduction:

This chapter is the documentation of the process and validation of the 5 cell Braille module using solenoids. FYDP-C started out with trying to find various ways of solenoid actuation methods and choosing the best option for us. This was integral for the prototype development which will be discussed further in this chapter. The full hardware setup, trial and error and the challenges we faced will also be included in this chapter.

5.2 Completion of final design

Prototype Development:

Since we were using solenoids as the main actuation method, there are several ways to power solenoids for our purpose. We attempted 3 different methods for solenoid actuation. The three designs that were used are:

1. Motor Driver
2. Relays
3. MOSFETs

Out of these three, the MOSFET option was the best approach for us due to the low power consumption and almost no voltage drops. Moreover, MOSFETs are compact and they scale well. Overall, for unidirectional applications such as ours, MOSFETs are the superior choice. While the small and compact solenoids were being shipped from abroad, we used locally available higher power and larger solenoids to demonstrate that our design functions properly. We were able to demonstrate solenoid switching aligned with Braille mapping to our ATC members. Therefore this step was effective in finding the correct method of actuation which will be utilized in the form of PCB in the final product.

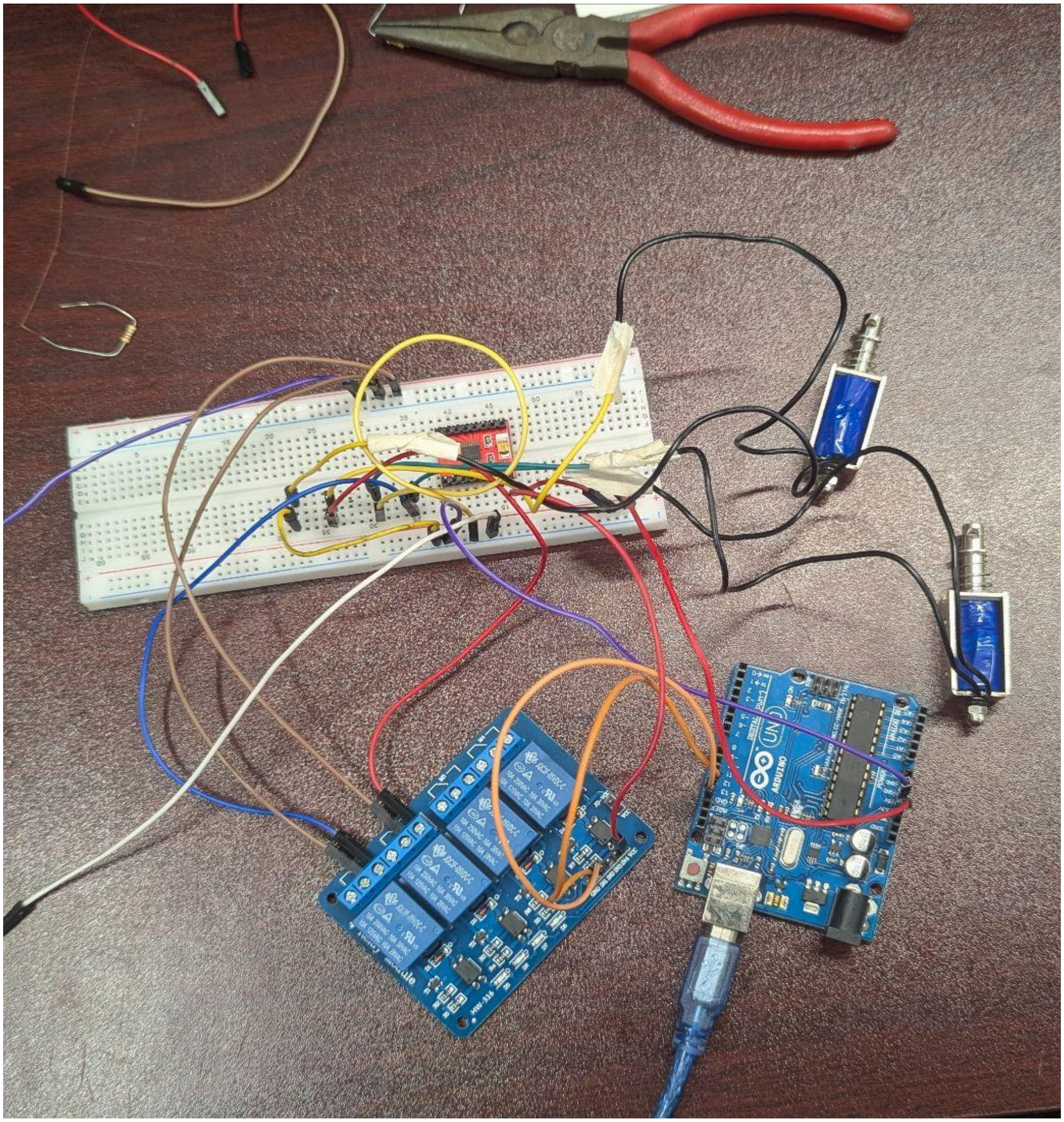

Figure 21: Relay Approach

<u>Relay Approach:</u>

At the very beginning we explore using electromagnetic relays as the switching mechanism for the solenoids. Electromagnetic relays function by energizing a small coil to create a magnetic field which attracts the metal contact to complete the circuit and do the switching.

The Relay Implementation Process

To drive the solenoids with relays, we connected the relays with the six output pins of the shift register. The small logic signal from the shift register energizes the coil to pull the metal contact , completing the circuit which allows 8V to flow through that designated solenoid. Initially the relay was an attractive option due to its ‘dry contacts’ referring to it having almost zero resistance at closed circuit and handling high current surge without overheating.

Why Relays Were Not Ideal

- **Physical Bulk:** Relay modules are quite bulky in size, as our prototype needs to be as small as possible to compete with the products in the market, scaling the prototype will create a bulkier and heavier product which doesn't match the industry standard.
- **Inductive kickback:** As relays are actuated by coils, when voltage is turned off, the magnetic field collapses, sending a high voltage spike towards the upstream circuitry.
- **Arc Effect:** When magnetic field collapses and metal contact moves to its default position the inductive load tries to maintain the current flow. Causing a small spark between the metal contacts.
- **Electromagnetic Interference:** The small sparks generate Electromagnetic interference(EMI). Which can move upstreams to the raspberry pi, causing interference with the gpio pins signals.
- **Durability and Power Consumption:** As the relays are mechanical moving parts, the arcing carbonizes the metal contact, eventually slowly increasing the resistance and failing to work. Additionally, to maintain the ON state it requires more current than our other options(MOSFET) increasing the power consumption.

The MOSFET Implementation

We opted for MOSFETs as the switching element because they are ideal for solenoids. By interfacing the shift register outputs directly with the gates of these Logic-Level N-channel MOSFETs, we can switch the 8V load with nanosecond precision(11ns). Because the IRL540N is designed with a low gate-source threshold voltage (VGH_{th}) it can be fully saturated by the 3.3V or 5V logic signals from the shift register without needing additional driver stages.

Why MOSFETs are Optimal

- **Logic-Level Compatibility:** The "L" in IRL540N signifies its logic-level operation. Unlike more available MOSFETs that require 10V at the gate to fully open, the IRL540N achieves a very low drain-to-source resistance (RDS(on)) at logic-level voltages suitable for operating with Raspberry Pi due to Pi's 3.3V logic level It also minimizes heat dissipation and ensures the solenoids receive maximum power.
- **Compact Form Factor:** As the PCB and the product needs to be lightweight and compact, MOSFET is a suitable solution.
- **Solid-State Reliability:** Since there are no moving parts, MOSFETs are more reliable and are less likely to get damaged due to switching. It also has a very long lifespan.
- **EMI and Signal Integrity:** MOSFETs are immune to sparks and it is ideal for our application as it avoids EMI issues and ensures GPIO signals are not corrupted.
- **Minimal Power Overhead:** MOSFETs consume very negligible amounts of power even when ON. This makes them better than relays because they consume considerable amounts of more power.

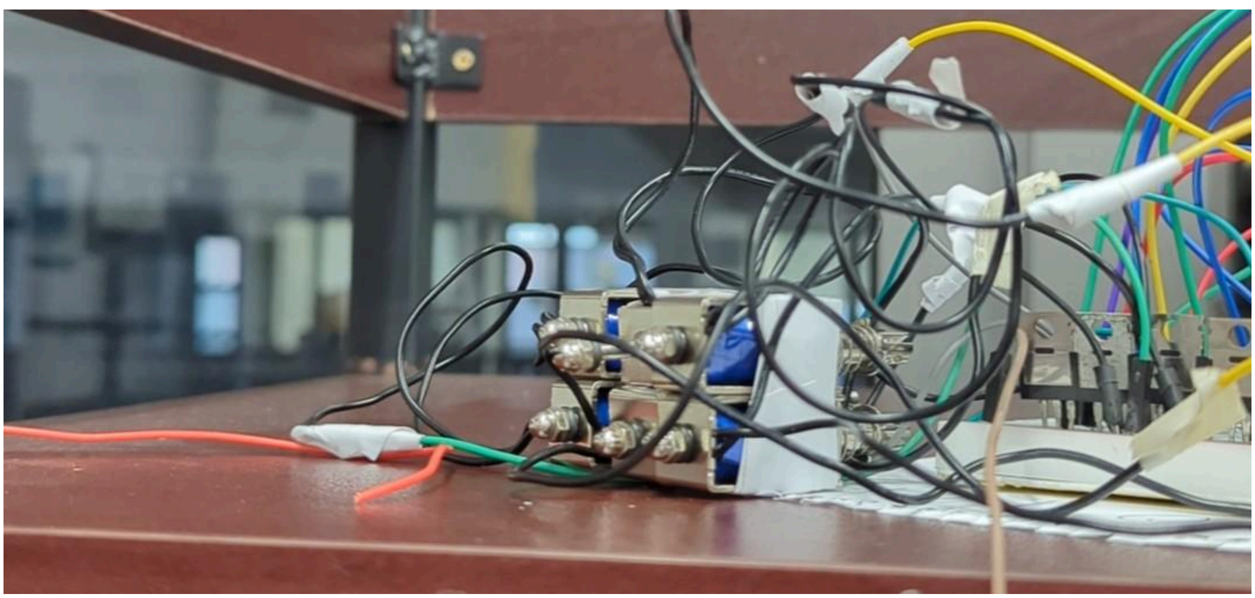

Figure 22: 6 solenoid MOSFET prototype

PCB Design

Optimized Routing and Signal Integrity

Our project is a combination of electronics and mechanical aspects. We decided to use a single layer PCB because it fits our purpose and it is more economical. One of the key obstacles in single layer PCB is to avoid ‘bridge’ wire while maintaining high current routes. To handle the high current of 190-200mA the traces wire widened enough to avoid overheating, on the contrary the logic wires were kept thinner to save space as it carries only 15-20mA. To solve the routing mess, we placed the IRL540N MOSFET in a single array. Which creates a pattern where the logic voltage is entering from one side and high voltage current flowing from the other side, representing a clean circuit.

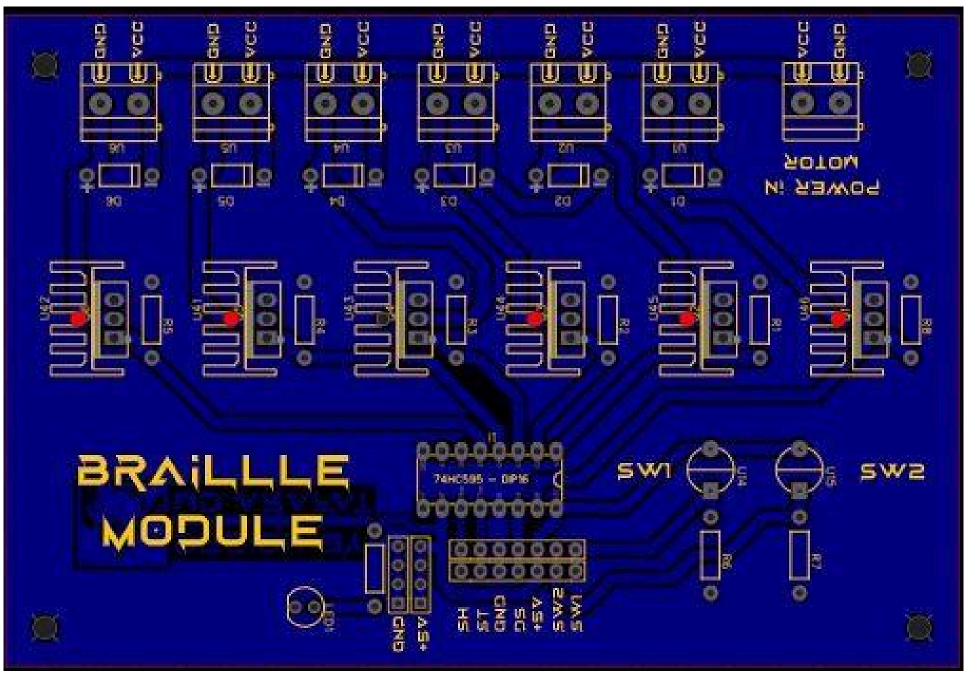

Figure 23: Gerber file view of PCB

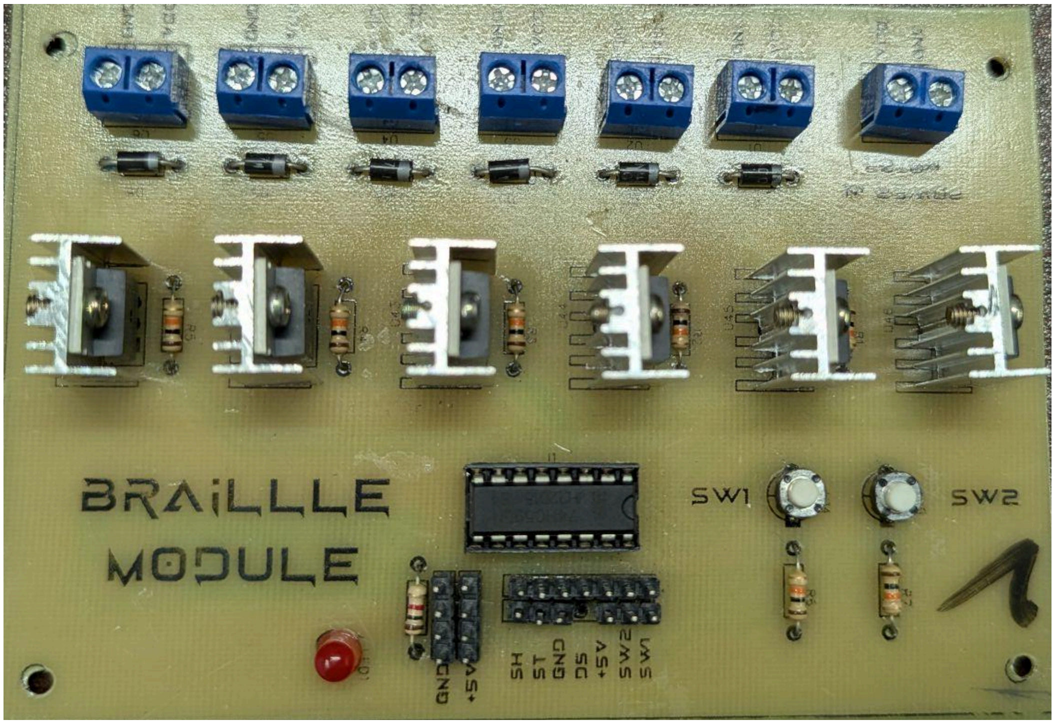

Figure 24: Fabricated PCB

Primary PCB Design Considerations:

1. Compact design which can be utilized for scaling and higher number of braille cells in future work
2. Ease of fabrication
3. Keeping low voltage signals separate from high current lines
4. Ensuring grounds are connected

PCB Features:

1. Logic level N-channel MOSFETs
2. Flyback diodes
3. Power input and output terminals
4. GPIO connection pins
5. Shift Register
6. Heat sink for MOSFETs

Reliability and Thermal Efficiency

We paid attention to mechanical aspects for reliability. As all the MOSFETS had external heatsink to dissipate the excess heat generated at the high speed switching during actuation process. To accommodate all the external heatsinks the MOSFETS were placed in a vertical array while maintaining proper distance between each of them. Furthermore, we used a Star Grounding design, which means all the grounds from the Raspberry Pi, switch, MOSFET, solenoids and shift registers meet at a single heavy duty point. This can cause false trigger, flickering and raspberry pi to misread in the actuation process.

3D Printing

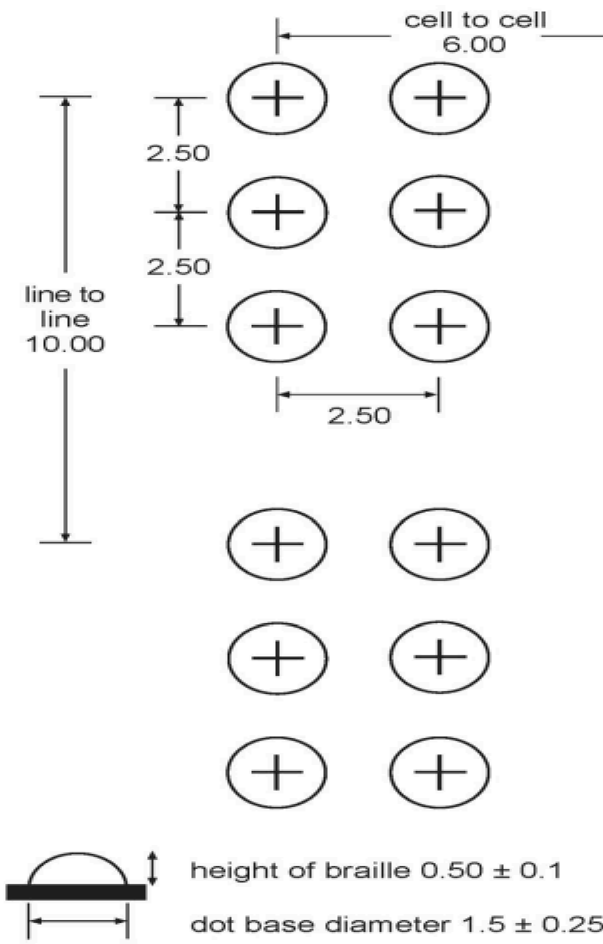

Figure 25: Standard Braille Dimensions

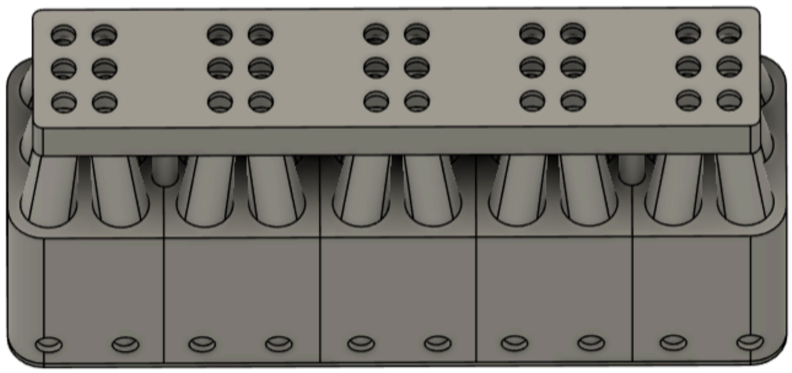

Figure 26: 3D visual of proposed Braille module

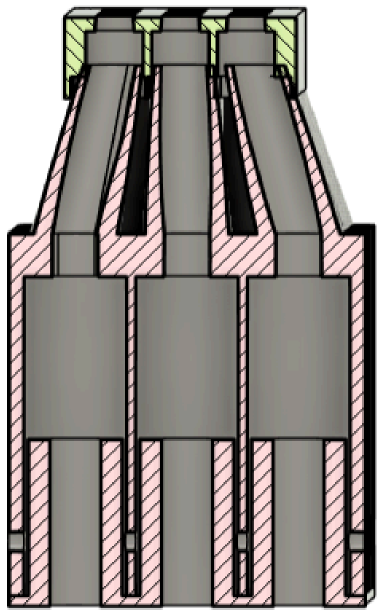

Figure 27: Cross section of braille module.

Our main goal was to build the 3D model by ourselves but the module was too complex for beginners like us so we decided to do the work with the help of professionals, due to maintaining the standard braille dimension it was also challenging for them. The enclosure and internal mounting framework were designed using CAD tools and fabricated through low-cost 3D printing. The Dot space was also tilted to ensure the Braille Dimension.

3D Design Considerations:

1. Standard compliant Braille dot spacing
2. Structural strength
3. Properly connecting solenoids to Braille dots using rods
4. Compact and ergonomic
5. Assembly considerations

Final Setup:
Firstly, we connected the power wires to every PCB. A buck converter was used to increase the current provided to the solenoids. Then we soldered wires to the solenoids and connected them to the PCB solenoid power output. The next part was the most challenging part which was connecting the solenoids to the mechanical part. This part proved to be quite difficult due to the fact that the rod and the Braille dot would get stitched easily. Moreover, the section responsible for holding the solenoids prevented the solenoids from actuation due to lack of space. The entire system is controlled by Raspberry Pi, so we used GPIO pins for MOSFET activation. Python was used to extract text from PDFs and EasyOCR features.

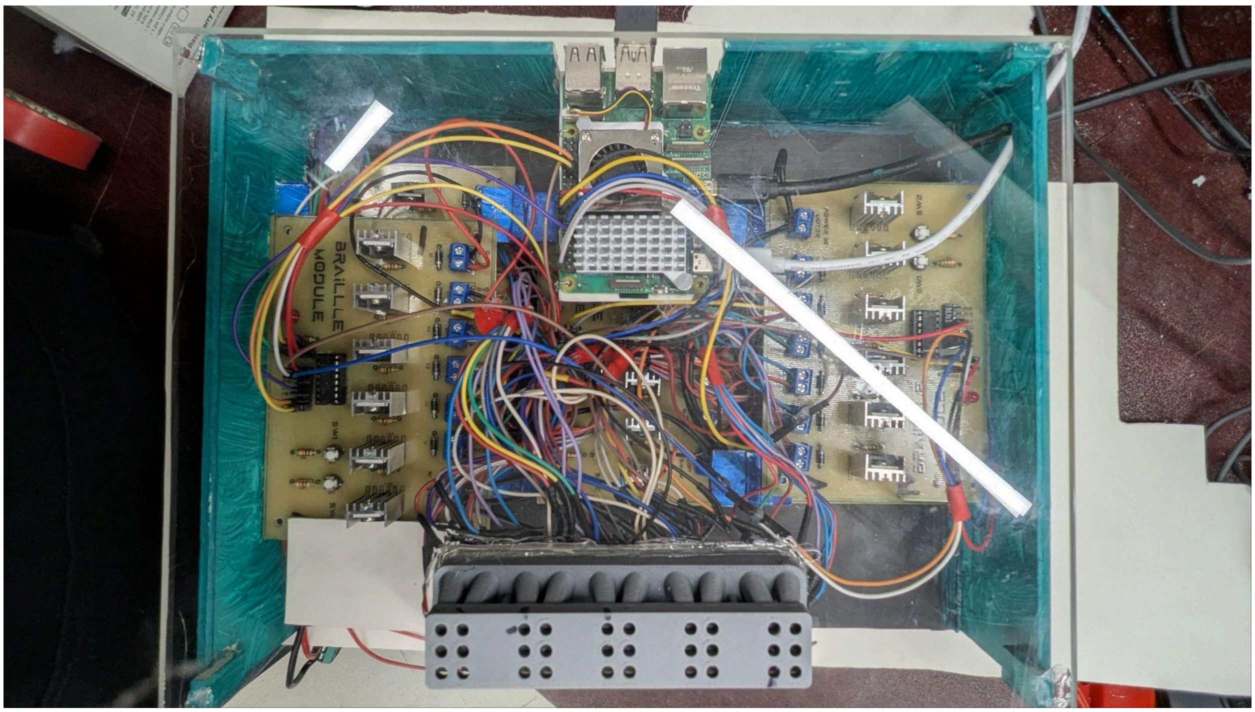

Figure 28: Final Setup

5.3 Evaluate the solution to meet desired need

5.3.1 Challenges faced in meeting the desired need
The desired need of our project is to have reliable Braille dot actuation for both PDF and image processing. Moreover, it needs to have consistent timing and free of software or hardware mishaps. We did not get our expected results due to two main issues. They are improper Braille dot actuation and unsynchronized solenoid actuation during GPIO cleanup. These two issues were major hurdles in the way of achieving our expected results.

5.3.2 Improvements to the design

We needed to address the two issues separately since they were caused by different domains. Our steps are described below.

Improper Braille dot actuation:

This issue was the culmination of two problems, first being the rods getting stuck to the walls of the mechanical part which meant the dot could not properly actuate or retract, secondly the solenoid would often not actuate at all even when powered properly because the limited and tight space for the solenoid. This meant the solenoid shaft got stuck with the wires. We took initiative to solve both of these problems.

1. Reducing the radius of the rod and the dot: Since we had to maintain the standard for Braille dot space, the rods were placed slightly at an angle to ensure the standard was met. However this was the main reason for the rod getting attached to the walls of the structure. We reduced the thickness of the rod which allowed smoother actuation with far lower risk of getting stuck. We also reduced the rod radius to further reduce the chance of this issue.
2. Solving solenoid space issue: The part responsible for holding the solenoids was deep and the solenoid had to be inserted completely. This meant there was not enough space to keep the shaft and the wires untangled. To fix this issue, we cut off a portion of the mechanical part. Now only the core body of the solenoid was being held by the mechanical design and the shaft and the wires were untangled. This prevented any issues with solenoid actuation.

Inconsistent timing for solenoid actuation:

We first noticed this issue when we tested out the PCB. We observed that after solenoid actuation according to the Braille map, the solenoids would keep actuating without any signal and in a random pattern. We were quite confused by this at the beginning. This problem is caused by floating state caused by the shift register. The solution to this is making sure GPIO is not cleared after word display. The shift register held the value at 0. Therefore the problem was solved by modifying the code.

5.3.2 Evaluate the effectiveness of design changes

Due to changes we made to our designs, we observed a significant rise in the accuracy of the results. These improvements were verifiable and clear.

Mechanical Behaviour:

As previously mentioned, before the adjustments, the rod and the dot would get stuck frequently during testing. To be precise, of the 10 test runs, 6 of the printed words were inaccurate because the dots were unable to actuate or retract properly. It should be noted that since actuation and retraction were faulty, subsequent word prints would also be affected and would often be inaccurate since the dots were either stuck in actuation or retraction from the previous word. So the previous version of the design had accuracy of only **40%** which is far lower than our requirement. Another aspect is in this design, higher number of solenoid actuation is correlated with higher probability of inaccurate result. After the improvements, the accuracy stood at **86.67%** as mentioned in chapter 4 for PDF extraction. The improved dots and rods would rarely get stuck and they usually delivered correct results according to the Braille map. Through the multiple cycles of testing and in our FYDP showcase it almost always provided the correct Braille output.

Inconsistent Solenoid Timing:

As previously mentioned, this problem was solved by changing the code. After the changes, the issue was completely resolved and we saw no random solenoid actuation after the Braille display. We demonstrated this improvement to our ATC as well.

5.4 Conclusion

Our solutions targeted the two dominant issues with our design. These issues caused inaccurate results which forced us to make adequate changes. Better 3D printing with resin materials may provide superior results. In conclusion it can be said that we achieved a baseline success rate and hopefully we will be able to improve that in the future.

Chapter 6: Impact Analysis And Project Sustainability

6.1 Introduction

Access to written information remains a challenge for visually impaired people in Bangladesh. Digital Braille technologies are often expensive, limited and unavailable. Printed Braille books are expensive and difficult to produce. Moreover, refreshable Braille modules are very expensive and rare in the context of Bangladesh. Since we are utilizing easily available hardware such as Raspberry Pi 5, solenoids, MOSFETs and open source software, our project provides an accessible solution to the problem statement. This section analyzes the social, health, safety, legal and cultural impact of the low cost refreshable Braille module. We will also evaluate the sustainability of the project through established standards.

6.2 Assess The Impact Of The Solution

Societal Impact

The project increases quality of life by providing access to information for visually impaired people. It achieves that by converting commonly used text mediums like PDFs, printed documents or handwritten texts into tactile Braille output. It reduces the restriction faced by visually impaired people and empowers them by providing access to education. Therefore the low cost Braille module reduces the barrier to literacy and access to information. The system has the potential to increase participation of visually impaired people in civic life, community and professional employment and vocational training. This integrates with national efforts of disability inclusion and increases the practical availability of assistive products where resources are limited.

Health Impact

It is a well established fact that access to assistive technologies improve various aspects of health in disabled individuals. Our refreshable Braille device serves to enhance mental health in relevant areas by being a reliable source of communication. The World Health organization agrees with this perspective and approves assistive technologies as a resource to disabled people. Therefore our project aligns with national and WHO efforts to improve health and quality of life for disabled people.

Safety Impact

The safety of the device depends on both mechanical and electronic components and design choices. There are two major sections of the project which are electronic and mechanical aspects. We followed the standard safety measures for electronics. Wires were insulated to avoid short circuits and heat sinks were used to avoid overheating. The mechanical section was carefully designed to avoid obstructions and pinch points.

Legal Impact

Under Bangladesh's **Rights and Protection of Persons with Disabilities Act, 2013**, persons with disabilities are subject to rights and protections that emphasise accessibility, inclusion and non-discrimination. The law emphasizes on including disabled people in civic and professional domains. A locally built low cost refreshable Braille module is an assistive device which aims to empower visually impaired people is aligned with the national policy. The project may also follow ISO/IEC standards referenced by BSTI. If the device is integrated into institutional services, product documentation, warranties, demonstration of conformity with national procurement standards can be relevant.

Cultural Impact

This project empowers disabled people by improving their autonomy. Since the project includes handwritten text to Braille conversion, it preserves the culture of social practices of handwritten notes or other texts. Furthermore, a low cost refreshable Braille module will help to reduce the stigma of assistive technologies in classrooms, workplaces and public venues. It is important to engage with disability organizations to ensure proper cultural integration. Cultural acceptability through awareness and training campaigns will help people to accept new assistive technologies. Overall this project will hopefully be a step in the right direction for disability inclusion.

6.3 Evaluate The Sustainability

For any engineering project, one of the most important aspects is to judge sustainability. The SWOT analysis of this project provides the sustainability assessment and demonstrates how it is a viable sustainable solution to the lack of affordable refreshable Braille module and how it aligns with global standards of Sustainable Development Goals(SDGs).

SWOT Analysis

Table-9: SWOT Analysis

Strengths	Weaknesses
1. Cost-effective compared to commercially available refreshable Braille displays 2. Use of open source software and easily available hardware materials 3. Focus on accessibility and inclusivity	1. Long durations of usage can cause solenoid overheating 2. Mechanical components are subject to wear and tear 3. Limited support for languages other than English
Opportunities	**Threats**
1. Scalable and sustainable assistive technology. 2. Potential candidate of government and NGO support in mass production 3. Supports UN SDG-4 Quality Education	1. Competition from commercially available assistive devices 2. OCR processing may require frequent updating to stay relevant 3. Lack of policy level support can restrict adoption in institutions

6.4 Conclusion

Overall, it can be said the project is a sustainable solution to the core problem. It is an effective and affordable system to improve literacy. From the above discussion, we can confidently say that our project fulfils the criteria of SDG-4 Quality Education and will help to bring literacy to the underprivileged communities.

Chapter 7: Engineering Project Management.

7.1 Introduction

In an engineering project, proper project management plays a critical role in the project's success. It needs to be ensured that the project is completed within the defined time, budget, quality and scope. Unlike conventional projects, engineering projects require management and knowledge of multidisciplinary components such as hardware design, software development, system integration, testing, and user-centered validation. We used proven methods of project management such as WBS and Gantt charts to ensure we were on track for success. With the help of our ATC members in every step, we tried to achieve maximum productivity.

This section of the report focuses on the planning methodology, development phases, risk identification and mitigation strategies and project progress evaluation.

7.2 Define, plan and manage engineering project

The project definition is accompanied by the core issue: the lack of a low cost accessible Braille module for the visually impaired community in Bangladesh. Therefore the primary objective is to design and implement an automated Braille module that is capable of converting digital and handwritten text to Braille output. After defining the project and identifying functional and non-functional requirements, we focused on project planning which mostly occurred in FYDP-P. We divided the project in manageable sections and defined workflows for FYDP-P, FYDP-D and FYDP-C respectively. Gantt charts and weekly progress tracking were used to ensure the project was progressing according to the time constraints. Moreover, we identified the risks and came up with solutions for each issue.. We also divided workload and worked parallelly when possible.

Table-10: Workload

Responsibilities				
Alvee	Mubtasik	Asif	Hafizur	Shabab
Project Planning	Code Debugging	Component acquisition	Final hardware setup	Project planning
Prototype building	Prototype building	Prototype building	Wire connection and testing	Draft report writing
PCB Design	EasyOCR integration	3D Printing fabrication	Mechanical calibration	Poster and flyer
Testing and evaluation	Circuit debugging and corrections	Power supply design	Solenoid mounting and alignment	Component acquisition

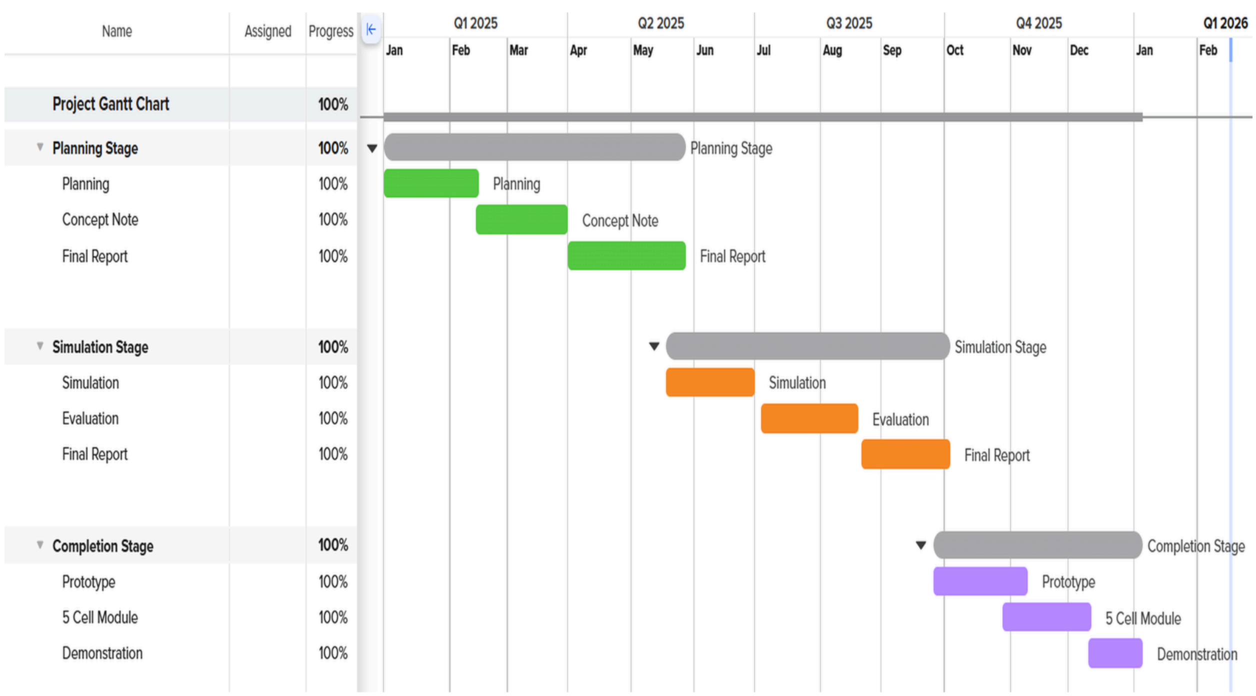

Figure 29: Gantt Chart

7.3 Evaluate project progress

7.3.1 Project Planning of EEE499P/EEE400P

At the beginning of our FYDP-P, we proposed multiple project ideas. After some time, the refreshable Braille module idea was accepted by our ATC. We decided to work on the concept note and the final proposal which included the identification of multiple design approaches. We also finalized the problem statement, scope, objectives, requirements according to the guidelines of our ATC. Each member worked on their respective part and we completed our final proposal and delivered our FYDP-P presentation to the panel. The Gantt chart of FYDP-P shows the timeline of this semester.

Table-11: FYDP-P Planning

Section	Name Of Task	Week
Planning	Background Research, Problem Statement	1-2
	Objective, Constraints	3
	Functional and Non-functional requirements	4
	Multiple design approaches	5
Concept Note	Applicable Codes and Standards	6
	Methodology, Gantt Chart	7
	Expected Outcome	8
Final Report	Budget, Ethical Consideration, Risk management, Risk Mitigation	9-10
	Slides and Presentation	11
	Peer evaluation	12

7.3.2 Project Planning of EEE499D/EEE400D

The goal of FYDP-D is to provide comparisons between different design approaches using simulation tools and evaluating the best approach. To achieve this, we identified the IT tools needed with the help of our ATC. We used various modern engineering tools for our purpose. Using the simulation tools, we determined that the electromagnetic approach is suited for us. We integrated Braille mapping to our code and successfully completed a software demonstration.

Table-12: FYDP-D Planning

Section	Name Of Task	Week
Simulation	IT Tool Identification	1
	Developing Circuits	2-3
	Troubleshooting	4-6
Comparisons	Generating Graphs	7
	Evaluating graphs and corrections	8
	Determining the best design	9
Final Report	Report Writing	10
	Slides and Presentation	11
	Peer evaluation	12

7.3.3 Project Planning of EEE499C/EEE400C

We faced significant issues during the beginning of FYDP-C due the lack of certain materials needed for the electromagnetic braille design. We realized ferrite core and 1.5 mm neodymium magnets are not easily available. With discussions with our ATC we decided to shift to the solenoid design. The concept is essentially the same as the electromagnetic design. After that we started building a one cell Braille module with 6 solenoids. We successfully completed the prototype and moved on to build the 5 cell module. We worked on the PCB and the 3D design simultaneously. Then we focused on the final setup which includes both electronic and mechanical parts. We completed the 5 cell module and successfully demonstrated our project in FYDP-C project demonstration.

Table-13-FYDP-C Planning

Section	Name Of Task	Week
Prototype	Planning	1
	Component Acquisition	2
	Circuit Building	3
5 cell module	PCB Design and printing	4-5
	3D Design	5
	Final Setup	6-9
	Troubleshooting	10
Demonstration	Poster	10
	Demonstration	11
	Peer evaluation	12

7.3.1 Risk Management and Contingency Plan

Risk management and mitigation strategies are vital to every engineering project. We need to identify what can go wrong and the risk level. One way to do this is using a risk assessment matrix which considers both likelihood and severity of a risk and assigns a level to it.

Table-14:Risk Assistment Matrix

Type of risk	Likelihood	Severity	Risk Level	Contingency plan	Members responsible
Mechanical jam (Braille dot misalignment)	High	Low	Medium	Adjusting the Braille dots and more accurate 3D printing	Asif and Hafiz
MOSFETs overheating	Medium	Medium	Medium	Correctly rated MOSFETs with heatsinks	All
Short Circuit	Medium	High	High	Proper insulation and following safety guidelines	Hafiz
Solenoid overheating	High	Medium	High	Duty cycle period limitations and cooldown time	Alvee
EasyOCR error	Medium	Low	Low	Using updated versions of software and better image processing models	Mubtasik
Power supply error	Low	Medium	Low	Using rated PSU and Buck Converter	Shabab

7.4 Conclusion

The success of this project largely depended on proper resource management. Through a proper understanding of systematic planning, progress monitoring and technical knowledge, we were able to provide a low cost solution to the lack of assistive technology focused on education.

Chapter 8: Economical Analysis. [CO12]

8.1 Introduction:

Large cost reductions were achieved by the proposed refreshable Braille reading device. Because the project is an improvement on a proven reliable product, the project Braille Reader maintains the reliability and functionality of the professional one while meeting the required dimensions set forth in the standard for Braille. The total cost breakdown for production of the first prototype (one module) was approximately 26,600 BDT per unit. The total production cost includes electronics and hardware: 6,600 BDT; power and operating system: 15,000 BDT; 3D-printed housing (structural components): 5,000 BDT. We have selected a selling price of 35,000 BDT. This gives us an ROI of 31.6% and a profit margin of 8400 BDT.

Standard commercially produced refreshable Braille displays such as the Focus 40 Blue utilize proprietary driver circuits, piezoelectric actuators, and precision mechanical assemblies; generally sold for several thousand US dollars, the majority of the visually impaired population does not have access to these devices, especially in developing countries.

In designing the new refreshable Braille module, costs were significantly reduced through the use of widely available miniature solenoids, combined with custom-designed PCB driver circuitry and additive manufacturing processes to replace piezoelectric actuators. Our choices make the project budget friendly because of low cost components and scalability.

Since we are providing a low cost solution to a significant problem in our society, we have potential for government or NGO support. This will help us to establish our project in institutions. Overall, the economic analysis confirms that the proposed design offers a highly cost-competitive, scalable, and sustainable alternative to existing commercial Braille modules, supporting broader access to digital Braille technology.

8.2 Economic Analysis

The number of visually impaired people in Bangladesh is very high and access to assistive technologies, especially the refreshable Braille display, is very few because of the high price and reliance on imports. There are no detailed national statistics on Braille literacy, but research and institutional reports show that the Braille literacy rate in visually impaired people in Bangladesh is low (usually estimated at below 10 percent), mainly because there is a poor availability of the Braille learning materials, Braille textbooks and Braille digital it devices.

Available types Commercial refreshable Braille displays which are sold in the Bangladeshi market are mostly imported and can cost in the range of 200000. Consequently, these products tend to be relatively cheap only to institutions or externally funded programs, and are not accessible to the major part of the student body and individual users.

This barrier to the economy is the solenoid-based refreshable Braille module being proposed. According to prototype implementation, the overall unit cost of production will be estimated at around 26,600 BDT consisting of electronics and hardware (6,600 BDT), power and operating system (15,000 BDT), and mechanical structure made through low-cost 3D printing (5,000 BDT). The system will attain a target selling price of 35,000 BDT, which translates to unit profit of 8,400 BDT, which is equivalent to an estimated ROI of 31.6 even with a low volume of production.

The proposed module is significantly more appropriate to the Bangladesh market both due to its 80-90 percent cost reduction with regard to the current commercial Braille displays. Local production of solenoids, specially designed PCB driver circuits, and additive manufacturing will limit use of imported expensive components and allow local assembly or local fabrication in the future. This price can render the device viable to school use by the visually impaired, universities, the NGOs, and individual learners.

Using Digital Braille access eliminates the need to rely on the expensive printed Braille books which are scarce and costly to print.

In general, the economic feasibility study shows that the proposed Braille module will be economically sound, market-relevant, and socially effective in the Bangladesh setting, which will provide an alternative to the costly imported solutions.

8.3 Cost benefit analysis

Table 15: Price Breakdown

Subsystem	**Components**	**Price**	**Cost (TK)**
Electronics and Hardware	Mosfet (IRL540n)	6000	15,600
	PCB	7000	
	Shift Register (sn74hc595n)	1400	
	Adapter	600	
	Buck Converter	600	
PU	Raspberry Pi 5	16000	16,000
Structure	3D printing	2000	2,000
Total Cost			**33,600**
Selling Price			**42,000**

8.4 Evaluation of Economic and Financial Aspects

The economic and financial viability of the proposed refreshable Braille module has been analyzed considering the cost structure, pricing, marketability, and sustainability in a long term scope in the Bangladesh environment.

A. Cost Efficiency

The suggested system is very cost-efficient in terms of cost and performance when it comes to commercially available refreshable Braille displays. The cost per unit of production is estimated to be 26,600 BDT on average and the targeted selling price will be 35,000 BDT. By comparison, commercial imports of Braille devices that are found in the local market cost on average between 200,000 and 600,000 BDT. It can be explained by the fact that the electromagnetic solenoid actuators were used, local electronic components were used and final fabrication methods are low-cost, such as additive manufacturing.

B. Financial Viability

According to the cost and pricing estimates, the proposed module will have a profit per unit of about 8,400 BDT, or a profit margin of 31.6. The margin is substantial enough to support the assembly, testing, distribution, and maintenance costs and be competitive. The positive margin shows that the system is viable financially at low volumes of production. Additional cost savings can be implemented by bulk purchase and production at local areas that could be contributing more to the bottom line or even more price cuts.

C. Market Suitability

The Bangladesh assistive technology market is one that is highly price-sensitive and has a low purchasing power by the final consumers. The reason is that the existing commercial Braille displays are expensive such that only institutions and programs funded by donors can use them.

D. Socio-Economic Impact

The rate of Braille literacy in Bangladesh is also low in most cases of less than 10% as they have little access to affordable Braille learning materials as well as to digital resources. A cheap refreshable Braille module can be used to have less dependency on costly printed Braille media and foreign assistive devices. Cost-benefit wise, the suggested system has long-term socio-economic advantages because it contributes to inclusive education, digital accessibility, and better employment opportunities among the visually impaired members of the population.

8.5 Conclusion

The work provides the layout and cost-benefit analysis of a basic, refreshable Braille module, whose purpose is to enhance accessibility to the needs of visually impaired consumers, mostly targeting cost-sensitive markets like Bangladesh.

Economic and financial evaluation proves that the suggested module is cost effective and financially feasible, and its price is considerably lower than those of imports, but it still offers sustainable profit margin. The lower price increases the viability of the market and allows more schools, non-governmental organizations, and individuals to use the assistive technology as they are now not able to do due to the expensive nature of the technology.

Socio-economically speaking, access to an affordable digital version of the Braille has the potential to facilitate better Braille literacy, inclusive education, and digital contribution of the visually impaired.

In general, the suggested refreshable Braille module is a scaled and cost-effective method of assistive technology. Future directions will be the multi-cell expansion, wireless connectivity and optimizing the long-term durability to increase functionality and impact.

Chapter 9: Ethics and Professional Responsibilities

9.1 Introduction

The development of our project which is a refreshable braille display points out several ethical considerations to abide by , things that should be kept in check while developing this device are inclusivity , equity , privacy and responsible innovation . Since this project is to develop a device for assisting vulnerable classes of people it is predominant to follow code of ethics in developing the device and ensure design benefits for users and ensure security for the users .

9.2 Identify ethical issues and professional responsibility

Inclusivity and Equity

This project is committed to ensure that the life of the visually impaired peoples is made easy and included with the society. Precisely nurture their communicating with the world since it is equally crucial to have the same volume of information to say and to read . As far as the visually impaired individuals lack access to the official and institutional places , that is where our project becomes quite instrumental in helping the individuals to read official documents, books, deeds, bonds, and such as declarations.Since individuals with vision impairments struggle to access the resources and feel inaccessible to technology, our project becomes a significant solution to the issue. We are discussing that gap in low-served areas such as rural areas in developing world where, as recent statistics provided by the world health organization, more than 90 percent of the world blind population is located, that is about 253 million people around the world with a large percentage of them being in South Asia such as Bangladesh with estimates of about 1.5 million people being blind or severely challenged in their visual functions due to reasons such as poverty and inadequate health care. We are designing a device that does not make existing disparities even greater; we are designing it with low-cost materials, such as those available as microcontrollers and 3D-printed parts, and using open-source software and hardware, which will ensure that the information is accessible to visually impaired people. Consider the scenario where a person in a remote village can read teaching resources or news in Braille without necessarily having to spend a lot of money on some gadgets. This includes supporting more than one language, such as Bengali, spoken by more than 230 million people across the globe, and implementing other Braille standardizations other than the English language, such as the Bengali code for the Braille, that includes special diacritics and scripts. It may unintentionally exclude non-English speakers in non-English speaking nations like Bangladesh which also forms a form of digital marginalization. We are hoping to collaborate with the communities in the area to develop and perfect these features, to make the device appear friendly and pertinent to the lives of the visually deficient individuals.

User Privacy and Data Handling

The privacy of the user must be taken with the highest regard and attention when we are working with something as personal as reading since it may be a medical report, a personal letter or just a favorite book. Our system turns PDFs digital and then converts them into Braille, but those files might include personal information that is very confidential about health, money or personal narrations of a person. That is why it is built on the principle to reflect nothing and no data will be stored or transmitted without the direct user consent so that the power will be placed in the hands of the individual using the data. We follow the example of other international requirements, such as the GDPR of the EU, where data minimization and user rights are prioritized, however, we also rely on the specifics of the local environment, e.g., in Bangladesh, the Digital Security Act of 2018 or the newly enacted Personal Data Protection Act (which may be fully implemented shortly, by 2026) highlight the importance of personal information protection against abuse, particularly in the field of technologies. On the physical level, it is possible to implement some sort of low-level security by embedding a simple microcontroller such as an Arduino or Raspberry Pi, which will deny any third-party entry by default by disabling internet connectivity and encrypting temporary data processing. This is the best way that the reading habits of the users remain totally confidential, not tracked or shared and thus users can access knowledge with utmost privacy, just as they would do with a normal book.

Potential Misuse and Bias

We had initiated this project to be extremely userfriendly , a tool that allows visually impaired people to start interacting with the world more on their own, yet we cannot disregard the ethical obligation to consider how this can be abused or fail to serve its purpose in case we are not mindful. Although it should be useful in nature, we must have mechanisms in place to ensure that we do not adapt to things such as unauthorized surveillance or hacks with unethical intentions, maybe by restricting the amount of code that can be modified, or by promoting community-level control over open-source forums. In an expected scenario this device might be used in official works where the user might be misguided with wrong info by tempering the device , also ill intended personal might be motivated to temper with deeds while dealing with the user so the user will be a victim of scams . More to the point, to prevent any inherent tendencies of us as designers to create a biased typology, the reality of which we may not be overly aware of due to our own life experiences, we are making early inroads to genuine engagement. This involves involving the visually impaired users throughout testing stages and taking their feedback on all aspects such as the sensitivity of the tactile responses (such as making sure that the pins do not press against the skin and cause pain) to the ergonomics of the device so it is indeed what they require but not what we think they require. As an example, the user stories taught us on the use of accessibility forums that not every person has the same degree of dexterity or learning curve, and we will be including user-adjustable settings.

Transparency and Accountability

As a team of students passionate about engineering for good, so we are all about being transparent and straight forward, and as a matter of fact, innovation is a winner when we are in control of the process including the warts. This is why we are committed to sharing our simulations, test findings, and even the shortcomings that we have experienced, such as the possible battery life constraints of low-power systems or a decrease in the accuracy with complicated PDF layouts. We will write reports on GitHub to acquire feedback. This is very much as the ethical standards of such organizations as the IEEE Code of Ethics which advises engineers to make known as soon as possible those factors that may pose a threat to the community or the environment and to put the needs of the population first, or the ACM Code of Ethics which finds such concepts as equity and nonmaleficence.

9.3 Apply ethical issues and professional responsibility

Our device is vulnerable to both physical threats and digital threats like malwares. Therefore a cybersecurity is required so that foreign intervention can't steal data or manipulate the OS system design. Maintenance service will ensure manual checking of the hardwares installed in the device and test proper functionality of PCB and solenoids actuations. The Anti Malware software will give real time data protection and the user will be protected from foreign observation and also, an emergency program will be installed where the user can law enforcement authorities for both security concerns and medical attention. Therefore the user can dial up to some limited authorities for assistance.

We are aware that any aids device such as ours, particularly one dealing with personal digital data, is vulnerable to threats whether it is a person messing with the hardware in a busy market or malicious software that finds its way into the device by a connected USB drive. In such locations as Dhaka, Bangladesh, where cyber threats are increasingly widespread (the reports of the Bangladesh Computer Council reveal more than 1,500 cases in 2025 alone, most of which involve weaker users of technology), it is of paramount importance to be able to protect against both an actual threat, such as theft or damage, and an online one, such as a virus or hacking attack. This is the reason why we have instituted sufficient cybersecurity measures early on so that no third party would be able to spy on the information, steal valuable PDFs, or tamper with the basic design of the operating system that may otherwise interfere with the Braille conversions or leak information about the user. Based on existing models, such as the NIST Cybersecurity Framework, scaled down to low resource settings, we are adding mechanisms such as secure boot operation to the microcontroller, and frequent over the air updates where feasible, and ensuring things remain simple and offline-first to reduce exposure.

9.4 Conclusion

For safety of both physical and digital threats we have designed an after sale service plan where our device will be given maintenance for popper hardware performance and for cyber security proper firewall and anti virus protection software will be developed for fixing bugs and real time protection from foreign interventions .

Chapter 10: Conclusion and Future Work.

10.1 Project summary/Conclusion

The development of an “Affordable Braille Reader for Converting PDF Text into Refreshable Braille Output”, addressing the accessibility needs of approximately 43 million visually impaired individuals worldwide, including 750,000 in Bangladesh. The electromagnetic solution, using Raspberry Pi for processing, MOSFETs for control, and magnetic solenoids for efficient dot actuation, excels in response time (48ms), power consumption, and cost, as verified through Proteus simulations, MATLAB analysis, and a weighted decision matrix .

The critical requirements such as real time feedback, portability and ISO compliance were fulfilled and Python, Proteus, MATLAB, Fusion 360, Raspberry Pi among others were used to facilitate quick prototyping. The Gantt chart represented in the project plan allowed the completion of the project by September to December 2025. The projected effects are improved literacy and inclusiveness, whereas ethical, risk, and safety issues focus on equity to users, mitigation measures, and design without hazards.

Then leading to hardware setup in December we were able to collect all the necessary components for our project with the refined 3D designed printed of our braille cells .We assembled the braille cells . Braille cells have 6 pins for combinations of one letter braille format, those pins required 6 solenoids for actuation. The solenoids are required to have power up separately with a separate PSU. On the other hand to control these 6 solenoids actuation timing we built PCB for each cell and they were programmed by Pi. So we had assembled these components and constantly tested their performance for accuracy and proper actuation of Dot pins. The hollow tubes of the dots pins required proper furnishing after testing and then the solenoid's placement below the pins had to be strong and rigid. Because after testing we discover excess heating in the solenoids which lead to impact in the braille cells design. This required proper isolation of the solenoids below the display.

In conclusion, this device requires optimization and developing the device's function aligning with global standards and users demand installing all the advanced components along the way .

10.2 Future work

This refreshable braille display has lots of future scope for development and to make it market ready for deployment it needs to be more furnished. To reduce the weight of the braille device, we need to make the PCBs a more compact design and smaller so that the overall device becomes smaller. The device requires many adjustments in design for braille cells so that mechanical errors in dots actuation are mitigated. as the braille dots are very small and their height of actuation should also be measured and controlled under the standard height for reading braille formats . The tactile force of the braille dots need to be engineered so that it meets the standards of the reader's satisfaction . Later on the design we built is for 5 characters the device needs to be engineered to 20 characters for more info to be displayed at a time . Use of solenoids made this project very cheaper so we can also look up more alternative reliable and cheap options for dots actuation . The materials for braille's physical design the device can be built with better insulating materials so that heating of the solenoids can be mitigated.

Then in case of software there are many more future scopes for this project. This refreshable braille requires a secure Anti malware software for data and system protection. Many more functions can be added to assist the blind person along with all the functions an AI voice assistant can be installed so that the blind person can command the AI assistant to work for them. In this way the blind person can also have access to the internet very easily. They can also get assistance in finding books from the systems library with the help of AI and also controlling the device.

To sum it all, the design of PCBs and software development with AI integration are future scopes for this project.

Chapter 11: Identification of Complex Engineering Problems and Activities.

11.1: The attribute of complex engineering problem (EP)

Attributes of Complex Engineering Problems

Table-16: Attributes of CE

	Attributes	**Put tick (√) as appropriate**
P1	Depth of knowledge required	☑
P2	Range of conflicting requirements	☑
P3	Depth of analysis required	☑
P4	Familiarity of issues	☑
P5	Extent of applicable codes	☑
P6	Extent of stakeholder involvement and needs	☑
P7	Interdependence	☑

11.2: Reasoning how the project address selected attribute (EP)

Our project is to assess visually impaired people in reading texts . The modern world is shifting towards digital communication technology. Therefore, this special kind of people are versatile and have easy access to technology to gain information. Our refreshable braille technology is able to convert text into braille format and also image of texts into braille format. To make our project relevant functional we had to integrate a Raspberry Pi which is a single board computer , then we had to configure it in a way that it fully fills our demand in controlling the device and also assess the user in demand activities. We had to fully understand the comfort of the stakeholders so the device had to be built with secure and protected materials along with concerns of the design to be socially acceptable and durable at the same time. Hazardous conditions had to be considered as the device should be able to perform in all kinds of zones like cold countries or warm countries with efficient accuracy in translating to braille format . Also our device should be able to stay online and perform for a long time so that the reader can get enough satisfaction in using it. That is why we had to carefully engineer the device to perform for a long time with accuracy.

11.3 Identifying the attribute of complex engineering activities (EA)

Attributes of Complex Engineering Activities

Table-17: Attributes of CEA

	Attributes	**Put tick (√) as appropriate**
A1	Range of resource	☑
A2	Level of interaction	☑
A3	Innovation	☐
A4	Consequences for society and the environment	☑
A5	Familiarity	☑

11.4 Reasoning how the project address selected attribute (EA)

At first we had to gather safe materials and ensure proper use of those components in our project. Since our stakeholders are a special type of people we had to ensure proper use of these components so that they are properly assisted , to understand their comfort we had refined our design to their advantage . On the other hand to make this socially acceptable the design had to be eye appealing along with proper functions and rigid to withstand future updates for better experience .

12.References

[1] World Health Organization, "World Report on Vision," 2019. [Online]. Available: https://www.who.int/publications/i/item/world-report-on-vision

[2] Freedom Scientific, "Focus 40 Blue Braille Display," [Online]. Available: https://www.freedomscientific.com/

[3] M. Al-Faruque, M. Shamsi, and S. Hossain, "Design of a Low-Cost Refreshable Braille Display Using SMA Wires," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 5, pp. 1045-1054, 2019.

[4] S. Kim, H. Ryu, and K. Lee, "Electromagnetic Actuation for Affordable Refreshable Braille Displays," in *Proceedings of the IEEE Sensors Conference*, 2020, pp. 1-4.

[5] OpenBraille Project, "Low-Cost 3D Printed Refreshable Braille Display," 2019. [Online]. Available: https://openbraille.org/

[6] K. Pesudovs *et al.*, "Global estimates on the number of people blind or visually impaired by cataract: a meta-analysis from 2000 to 2020," *Eye*, vol. 38, no. 11, pp. 2156–2172, Mar. 2024, doi: 10.1038/s41433-024-02961-1.

[7] "Experts: 750,000 people suffer from blindness in Bangladesh," Dhaka Tribune, Oct. 12, 2018.[Online].Available:https://www.dhakatribune.com/bangladesh/2018/10/12/experts-750-000-people-suffer-from-blindness-in-bangladesh.

[8] "THE 17 GOALS | Sustainable Development." https://sdgs.un.org/goals

[9] Ruipeng Chen, David Folio. Electromagnetic Actuation Microrobotic Systems. Current Robotics Reports, 2022, 3 (3), pp.119-126. ff10.1007/s43154-022-00081-wff. Ffhal-03758932f

[10] J. Smith and K. Lee, "Advances in Low-Cost Actuation Technologies for Assistive Devices," *J. Access. Technol.*, vol. 15, no. 3, pp. 45-60, 2023.

[11] M. Garcia and H. Omar, "Tactile Feedback Standards in Assistive Technology," *J. Inclusive Design*, vol. 8, no. 1, pp. 33-48, 2022.

[12] T. Nguyen et al., "3D Printing for Sustainable Electronics Manufacturing," *Int. J. Sustain. Design*, vol. 10, no. 2, pp. 78-92, 2023.

[13] R. Khan and S. Patel, "Energy-Efficient Microcontroller Applications in Assistive Devices," *IEEE Trans. Eng. Med. Biol.*, vol. 22, no. 4, pp. 112-125, 2024.

[14] M. Etezad, R. Joshi, and F. L. Cibrian, “Advancements in refreshable Braille display technology: A comprehensive survey,” *Displays*, vol. 90, Dec. 2025, Art. no. 103133, doi: 10.1016/j.displa.2025.103133.

[15] Smith, J., & Lee, K. (2023). Advances in Low-Cost Actuation Technologies for Assistive Devices. *Journal of Accessible Technology*, 15(3), 45-60.

[16] Khan, R., & Patel, S. (2024). Energy-Efficient Microcontroller Applications in Assistive Devices. *IEEE Transactions on Engineering in Medicine and Biology*, 22(4), 112-125.

[17] Nguyen, T., et al. (2023). 3D Printing for Sustainable Electronics Manufacturing. *International Journal of Sustainable Design*, 10(2), 78-92.

[18] Garcia, M., & Omar, H. (2022). Tactile Feedback Standards in Assistive Technology. *Journal of Inclusive Design*, 8(1), 33-48.

[19] Brown, L., & Chen, Y. (2024). Impact of Digital Inclusivity on Visually Impaired Education. *Global Accessibility Review*, 19(5), 201-215.

Appendix

Logbook:(FYDP-P)

Date	Attendees	Summary of meeting
18/2/25	Mubtasik ,Nafis ,Hafiz Asifur , Shabaab ,ATC Panel	Initial idea discussion
25/2/25	Mubtasik ,Nafis ,Asifur	Further discussion about which ideas are suitable for FYDP
2/3/25	Mubtasik ,Nafis , Hafiz , Asifur ,Shabaab ,ATC Panel	Braille idea accepted by ATC
5/3/25	Mubtasik ,Nafis , Hafiz Asifur ,Shabaab	Draft note submission
6/3/25	Mubtasik , Nafis ,Hafiz Asifur , Shabaab	Slide preparation for presentation on braille idea
9/3/25	Mubtasik , Nafis , Hafiz Asifur ,Shabaab ATC Panel	Presentation on topic
13/3/25	Mubtasik , Nafis , Hafiz Asifur	Further discussion on refining multiple approaches
24/3/25	Mubtasik ,Nafis ,Hafiz Asifur ,Shabaab	Presentation cancelled due to midterms
2/4/25	Mubtasik ,Nafis Hafiz Asifur , Shabaab	Preparing slides, final concept note, preliminary proposal writing

15/4/25	Mubtasik ,Nafis ,Hafiz Asifur , Shabaab ,ATC Panel	Progress presentation
8/5/25	Mubtasik, Nafis ,Hafiz Asifur , Shabaab.	FYDP-P Main presentation

Logbook: (FYDP-D)

Date	Type of meeting	Attending Members
June 26, 2025	Feedback on system algorithm	Mubtasik , Alvee, Asif , Shabab
July 8, 2025	Consultation and feedback about system	Hafiz , Mubtasik , Asif ,
July 15, 2025	Consultation about multiple design approach	Mubtasik , Alvee, Asif
July 29, 2025	Consultation about input process and user feedback	Hafiz , Mubtasik , Alvee, Asif
Aug 05, 2025	Consultation	Mubtasik , Alvee, Asif ,
Aug 14 , 2025	Progress presentation	Hafiz , Mubtasik , Alvee, Asif , Shabab
Aug 25 , 2025	Progress Presentation and feedback	Hafiz , Mubtasik , Alvee, Asif , Shabab
Aug 27 , 2025	Feedback on final presentation	Hafiz , Mubtasik , Alvee, Asif , Shabab
Sept 9, 2025	Consultation	Hafiz , Mubtasik , Alvee, Asif

Logbook: (FYDP-C)

Date	Type of meeting	Attending Members
Sept 29, 2025	Testing on different types of solenoids	Mubtasik , Alvee, Asif
Oct 7, 2025	Building the 1st prototype	Alvee, Mubtasik , Asif ,
Oct 15, 2025	Finalizing the code to control the solenoids.	Mubtasik , Alvee,
Oct 20, 2025	Running the Fullyfunctional Prototype	Mubtasik , Alvee, Asif
Oct 22, 2025	Order the components	Mubtasik , Alvee, Asif , Shabab
Oct 30 , 2025	Start to Build the final product	Mubtasik , Alvee, Asif ,Hafiz Shabab
Nov 25 , 2025	End of the production	Mubtasik , Alvee, Asif ,Hafiz
Dec 1 , 2025	Run the module with initial codes	Hafiz , Mubtasik , Alvee, Asif , Shabab
Dec 8, 2025	Progress Presentation to the ATC panel	Hafiz , Mubtasik , Alvee, Asif , Shabab
Dec 17, 2025	Start to write the report	Hafiz , Shabab
Dec 20,2025	Finalize the module and prepare for the final presentation	Mubtasik , Alvee, Asif ,Hafiz Shabab
Jan 5, 2026	FYDP-C final presentation and showcase	Hafiz , Mubtasik , Alvee, Asif , Shabab

Related Codes:

<u>For PDF file as input:</u>

```
import RPi.GPIO as GPIO
import time
import PyPDF2
import re
from tkinter import Tk, filedialog

# ========= GPIO SETUP =========
DATA_PINS = [12, 17, 23, 24, 25]  # DS1 → DS5 (left to right)
CLOCK = 27
LATCH = 22

# Button pins
BTN_NEXT = 5   # SW1 - next word
BTN_PREV = 6   # SW2 - previous word

GPIO.setmode(GPIO.BCM)
GPIO.setup(DATA_PINS + [CLOCK, LATCH], GPIO.OUT)
GPIO.setup(BTN_NEXT, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(BTN_PREV, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

# Initialize all output pins LOW
for pin in DATA_PINS + [CLOCK, LATCH]:
    GPIO.output(pin, GPIO.LOW)

# LETTER MAP (basic 7-segment mapping - A-Z + space)
LETTER_MAP = {
    'A': 'b00000001', 'B': 'b00000101', 'C': 'b00000011', 'D': 'b00001011', 'E': 'b00001001',
    'F': 'b00000111', 'G': 'b00001111', 'H': 'b00001101', 'I': 'b00000110', 'J': 'b00001110',
```

```
    'K': 'b00010001', 'L': 'b00010101', 'M': 'b00010011', 'N': 'b00011011', 'O': 'b00011001',
    'P': 'b00010111', 'Q': 'b00011111', 'R': 'b00011101', 'S': 'b00010110', 'T': 'b00011110',
    'U': 'b00110001', 'V': 'b00110101', 'W': 'b00101110', 'X': 'b00110011', 'Y': 'b00111011',
    'Z': 'b00111001',
    ' ': 'b00000000',  # Blank
    # Optional: add numbers if your display supports them
    # '0': 'b00111111', '1': 'b00000110', etc.
}

def letter_to_byte(letter):
    """Convert letter pattern string to integer byte value"""
    pattern = LETTER_MAP.get(letter.upper(), 'b00000000')
    return int(pattern[2:], 2)  # remove '0b' prefix

#SHIFT REGISTER OUTPUT
def shift_out_parallel(bytes_5):
    """Shift out 5 bytes in parallel to the shift registers"""
    GPIO.output(LATCH, GPIO.LOW)
    for bit in range(7, -1, -1):  # MSB first
        GPIO.output(CLOCK, GPIO.LOW)
        for i, pin in enumerate(DATA_PINS):
            GPIO.output(pin, (bytes_5[i] >> bit) & 0x01)
        time.sleep(0.00001)  # very short delay
        GPIO.output(CLOCK, GPIO.HIGH)
        time.sleep(0.00001)
    GPIO.output(CLOCK, GPIO.LOW)
    GPIO.output(LATCH, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(LATCH, GPIO.LOW)

# DISPLAY CURRENT WORD
```

```
def show_current_word():
    global current_word_index, words

    if not words:
        shift_out_parallel([0] * 5)
        return

    word = words[current_word_index].upper()
    print(f"[{current_word_index + 1}/{len(words)}] {word}")

    if len(word) <= 5:
        # Short words: left aligned + padding with spaces
        display_chars = (word + '     ')[:5]
        bytes_5 = [letter_to_byte(c) for c in display_chars]
        shift_out_parallel(bytes_5)
        # Keep displayed for a while
        time.sleep(2.5)
    else:
        # Long words: simple scrolling
        padded = word + "      "  # some spaces at the end
        for i in range(len(padded) - 4):
            segment = padded[i:i+5]
            bytes_5 = [letter_to_byte(c) for c in segment]
            shift_out_parallel(bytes_5)
            time.sleep(0.45)  # scrolling speed - adjust as needed
        # Pause at the end
        time.sleep(1.2)

# ========= PDF EXTRACTION =========
def extract_text_from_pdf(path):
    text = ""
```

```
    try:
        with open(path, "rb") as f:
            reader = PyPDF2.PdfReader(f)
            for page in reader.pages:
                t = page.extract_text()
                if t:
                    text += t + "\n"
    except Exception as e:
        print(f"Error reading PDF: {e}")
    return text

def find_words(text, min_length=1):
    # Find words containing only letters (uppercased)
    return re.findall(r'\b[A-Za-z]{' + str(min_length) + r',}\b', text.upper())

# ========= FILE SELECTION =========
root = Tk()
root.withdraw()
print("Select PDF file")
pdf_path = filedialog.askopenfilename(title="Select PDF", filetypes=[("PDF files", "*.pdf")])
root.destroy()

if not pdf_path:
    shift_out_parallel([0] * 5)
    print( No file selected - display cleared")
    GPIO.cleanup()
    exit()

text = extract_text_from_pdf(pdf_path)
words = find_words(text, min_length=1)
```

```
if not words:
    shift_out_parallel([0] * 5)
    print(" No words found in the document")
    GPIO.cleanup()
    exit()

print(f" Found {len(words)} words")

# ========= BUTTON HANDLING =========
current_word_index = 0

def next_word_callback(channel):
    global current_word_index
    current_word_index = (current_word_index + 1) % len(words)
    show_current_word()

def prev_word_callback(channel):
    global current_word_index
    current_word_index = (current_word_index - 1) % len(words)
    show_current_word()

# Register button events - only on rising edge (press)
GPIO.add_event_detect(BTN_NEXT, GPIO.RISING, callback=next_word_callback,
bouncetime=250)
GPIO.add_event_detect(BTN_PREV, GPIO.RISING, callback=prev_word_callback,
bouncetime=250)

# ========= MAIN PROGRAM =========
print("\n=== CONTROL INSTRUCTIONS ===")
print("  SW1 (GPIO 5)  → NEXT word")
print("  SW2 (GPIO 6)  → PREVIOUS word")
```

```
print("  Press Ctrl+C to exit")
print("===========================\n")

# Show the first word immediately
show_current_word()

try:
    while True:
        time.sleep(0.3)  # low CPU usage
except KeyboardInterrupt:
    print("\nProgram terminated by user")
finally:
    # Cleanup
    GPIO.remove_event_detect(BTN_NEXT)
    GPIO.remove_event_detect(BTN_PREV)
    shift_out_parallel([0] * 5)
    print("✔ Display cleared")
    # GPIO.cleanup()  #
```

For image(jpg, png, jpeg) file as input:

```
import RPi.GPIO as GPIO
import time
import easyocr
from tkinter import Tk, filedialog
import re

# ========= GPIO SETUP =========
DATA_PINS = [12, 17, 23, 24, 25]   # DS1 → DS5 (left to right)
CLOCK     = 27
LATCH     = 22

# Button pins
BTN_NEXT  = 5    # Next word
BTN_PREV  = 6    # Previous word

GPIO.setmode(GPIO.BCM)
GPIO.setup(DATA_PINS + [CLOCK, LATCH], GPIO.OUT)
GPIO.setup(BTN_NEXT, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(BTN_PREV, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

# Initialize all output pins LOW
for pin in DATA_PINS + [CLOCK, LATCH]:
    GPIO.output(pin, GPIO.LOW)

# ========= BRAILLE LETTER MAP (your current mapping) =========
LETTER_MAP = {
    'A':'b00000001','B':'b00000101','C':'b00000011','D':'b00001011','E':'b00001001',
    'F':'b00000111','G':'b00001111','H':'b00001101','I':'b00000110','J':'b00001110',
    'K':'b00010001','L':'b00010101','M':'b00010011','N':'b00011011','O':'b00011001',
    'P':'b00010111','Q':'b00011111','R':'b00011101','S':'b00010110','T':'b00011110',
```

```
    'U':'b00110001','V':'b00110101','W':'b00101110','X':'b00110011','Y':'b00111011',
    'Z':'b00111001',
    ' ': 'b00000000',
}

def letter_to_byte(letter):
    pattern = LETTER_MAP.get(letter.upper(), 'b00000000')
    return int(pattern[2:], 2)  # skip 'b0' prefix

# ========= SHIFT OUT FUNCTION =========
def shift_out_parallel(bytes_5):
    GPIO.output(LATCH, GPIO.LOW)
    for bit in range(7, -1, -1):
        GPIO.output(CLOCK, GPIO.LOW)
        for i, pin in enumerate(DATA_PINS):
            GPIO.output(pin, (bytes_5[i] >> bit) & 0x01)
        time.sleep(0.000005)   # very short delay
        GPIO.output(CLOCK, GPIO.HIGH)
        time.sleep(0.000005)
    GPIO.output(CLOCK, GPIO.LOW)
    GPIO.output(LATCH, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(LATCH, GPIO.LOW)

# ========= DISPLAY ONE WORD (your 10 sec logic) =========
def display_word(word, display_time=10.0, pause_between=0.6):
    word = word.upper()
    n = len(word)

    if n <= 5:
        # Short word
```

```
        bytes_5 = [letter_to_byte(c) for c in word] + [0] * (5 - n)
        print(f"  Showing (10s): {word}")
        shift_out_parallel(bytes_5)
        time.sleep(display_time)
    else:
        # Long word - show first 5, then remainder
        first_part = word[:5]
        bytes_first = [letter_to_byte(c) for c in first_part]
        print(f"  Showing first 5 (10s): {first_part}")
        shift_out_parallel(bytes_first)
        time.sleep(display_time)

        shift_out_parallel([0]*5)          # brief clear
        time.sleep(pause_between)

        remainder = word[5:10]             # max ~10 letters total for now
        if remainder:
            bytes_rem = [letter_to_byte(c) for c in remainder] + [0] * (5 - len(remainder))
            print(f"  Showing next part (10s): {remainder}")
            shift_out_parallel(bytes_rem)
            time.sleep(display_time)

# ========= MAIN DISPLAY LOGIC =========
def show_current_word():
    global current_word_index, all_words

    if not all_words:
        shift_out_parallel([0]*5)
        return

    word = all_words[current_word_index]
```

```
    print(f"\n[{current_word_index+1}/{len(all_words)}]  {word}")
    display_word(word)

# ========= FILE SELECTION & OCR =========
root = Tk()
root.withdraw()
print(" Select handwritten image (PNG/JPG) - can contain multiple words")
image_path = filedialog.askopenfilename(
    title="Select Handwritten Image",
    filetypes=[("Image files", "*.png *.jpg *.jpeg")]
)
root.destroy()

if not image_path:
    shift_out_parallel([0]*5)
    print(" No file selected")
    GPIO.cleanup()
    exit()

print(" Running EasyOCR (this may take 5–20 seconds on Pi)...")
reader = easyocr.Reader(['en'], gpu=False)
results = reader.readtext(image_path, detail=0, paragraph=False)

# Clean and filter words (only alphabetic characters)
all_words = []
for text in results:
    cleaned = ''.join(c for c in text.upper() if c.isalpha())
    if len(cleaned) >= 1:  # you can increase to 2 or 3 if you want
        all_words.append(cleaned)

all_words = list(dict.fromkeys(all_words))  # remove exact duplicates
```

```
if not all_words:
    print("  No valid words detected in image")
    shift_out_parallel([0]*5)
    GPIO.cleanup()
    exit()

print(f"Found {len(all_words)} words: {', '.join(all_words)}")

# ========= BUTTON HANDLING =========
current_word_index = 0

def next_word_callback(channel):
    global current_word_index
    current_word_index = (current_word_index + 1) % len(all_words)
    show_current_word()

def prev_word_callback(channel):
    global current_word_index
    current_word_index = (current_word_index - 1) % len(all_words)
    show_current_word()

# Add event detection with debouncing
GPIO.add_event_detect(BTN_NEXT, GPIO.RISING, callback=next_word_callback,
bouncetime=200)
GPIO.add_event_detect(BTN_PREV, GPIO.RISING, callback=prev_word_callback,
bouncetime=200)

# ========= START =========
print("\n" + "="*50)
print("          Controls:")
```

```
print("   SW1 (GPIO 5)  →  NEXT word")
print("   SW2 (GPIO 6)  →  PREVIOUS word")
print("   Ctrl+C        →  Exit")
print("="*50 + "\n")

# Show first word
show_current_word()

try:
    while True:
        time.sleep(0.4)  # low cpu usage loop
except KeyboardInterrupt:
    print("\nProgram terminated by user")
finally:
    # Cleanup
    GPIO.remove_event_detect(BTN_NEXT)
    GPIO.remove_event_detect(BTN_PREV)
    shift_out_parallel([0]*5)
    print("✔  Display cleared")
    GPIO.cleanup()
```

Code for approach-1 time response:

```
%% System Delay Visualization - Grouped Layout
% Reproduces the delay breakdown with grouped bars and updated solenoid timing

clear; clc; close all;

%% 1. Setup Data (in milliseconds)
words = {'1st Word', '2nd Word', '3rd Word', '4th Word'};

% Updated Delay Components
pi_delay = [5.45, 1.48, 1.48, 1.48];        % Processing Unit (Pi)
solenoid_delay = [16.32, 16.32, 16.32, 16.32]; % Updated Solenoid Delay
electronic_delay = repmat(31e-6, 1, 4);     % Circuitry/Electronic Delay

% Combine into a matrix for grouped plotting (Rows = Words, Columns = Components)
data = [pi_delay; solenoid_delay; electronic_delay]';

%% 2. Create Figure and Plot
figure('Color', 'w', 'Position', [100, 100, 1000, 600]);
% Changed from 'stacked' to 'grouped'
hBar = bar(data, 'grouped', 'BarWidth', 0.8, 'EdgeColor', 'none');

%% 3. Match Visual Style (Colors)
% Pi Delay (Blue)
hBar(1).FaceColor = [52, 152, 219] / 255;
% Solenoid Delay (Red/Orange)
hBar(2).FaceColor = [231, 76, 60] / 255;
% Electronic Delay (Green)
hBar(3).FaceColor = [46, 204, 113] / 255;

%% 4. Axes and Labels
title('Delay Component Breakdown per Word (Grouped)', 'FontSize', 16);
ylabel('Delay (ms)', 'FontSize', 12);

% Set X-axis labels
set(gca, 'XTickLabel', words, 'FontSize', 11);

% Adjust Y-axis for visibility
ylim([0, 20]); % Set limit slightly above 16.32 for headroom
```

```
grid on;
ax = gca;
ax.GridLineStyle = '--';
ax.Layer = 'top';

%% 5. Add Value Labels on Top of Each Individual Bar
% We loop through each set of bars (Pi, Solenoid, Electronic)
for k = 1:size(data, 2)
    % Get the x-coordinates of the center of each individual bar in the group
    xPoints = hBar(k).XEndPoints;
    yPoints = hBar(k).YData;

    for i = 1:length(yPoints)
        % For the extremely small electronic delay, we display more decimals
        if yPoints(i) < 0.01
            labelStr = sprintf('%.6f ms', yPoints(i));
        else
            labelStr = sprintf('%.2f ms', yPoints(i));
        end

        text(xPoints(i), yPoints(i) + 0.2, labelStr, ...
            'HorizontalAlignment', 'center', ...
            'VerticalAlignment', 'bottom', ...
            'FontSize', 9, ...
            'Rotation', 45);
end

%% 6. Legend
legend({'Pi Delay (ms)', 'Solenoid Delay (ms)', 'Electronic Delay (ms)'}, ...
    'Location', 'northeast', 'FontSize', 10);

% Final Clean up
set(gca, 'Box', 'off');
```