BRAILLE PRINTER FOR BLIND PEOPLE

Shantanu Duryodhan  
*Department of Mechatronics Engineering  
Sanjivani College of Engineering, Savitribai Phule Pune University*Pune, India

duryoshant@gmail.com

Pratiksha Parjane  
*Department of Mechatronics Engineering  
Sanjivani College of Engineering, Savitribai Phule Pune University*Pune, India

parjanepratiksha946@gmail.com

Gayatri More  
*Department of Mechatronics Engineering*  
*Sanjivani College of Engineering*  
*Savitribai Phule Pune University*Pune, India  
@gmail.com

Harshad Pawar  
*Department of Mechatronics Engineering  
Sanjivani College of Engineering, Savitribai Phule Pune University*Pune, India  
harshadanilpawar2828@gmail.com

\*S.M.Kadam  
*Department of Mechatronics*  
*Sanjivani College of Engineering*  
*Savitribai Phule Pune University*Pune, India  
[@gmail.com](mailto:william160891@gmail.com)

***Abstract***— Braille literacy plays a crucial role in promoting education and accessibility for individuals with visual impairments. However, the availability and affordability of braille printing devices remain a significant challenge, particularly in developing regions. This project aims to develop a low-cost, open-source braille printer that utilizes readily available components and open-source software to address this issue. The proposed braille printer leverages the capabilities of an Arduino Uno microcontroller, CNC shield, stepper motors, solenoid valves, and a motor driver to create embossed braille characters on paper. The system is designed to be controlled via Python code or G-code, allowing for seamless integration with existing software and enabling users to convert digital text into braille output. By democratizing access to braille printing technology, this project has the potential to empower individuals with visual impairments, facilitating their participation in educational and professional endeavors.

Keywords — Braille Printer, Braille Language, research agenda, and comprehensive literature review

I.INTRODUCTION

Braille printers are essential devices that facilitate access to printed materials for individuals who are visually impaired. These printers convert electronic text into Braille, enabling individuals with visual disabilities to read and comprehend written information independently. This literature survey aims to explore the current state-of-the-art technologies, advancements, challenges, and future prospects related to Braille printers.

We have made a ‘Braille printer’ that will convert text input into braille and create tactile cells correspondingly on a regular A4-sized paper, so that people with visual impairment can feel it and will thus be able to access printed text.

The printer will be useful for creating materials for literacy and education of people with visual impairment.

Braille is a tactile writing system used by people who are visually impaired. Each character is a 3x2 matrix of bumps (or dots) separated by standard distances. In the image below are English alphabets in braille and the standard distances, taken from this source.

Braille is a system of raised dots that can be read with the fingers by people who are blind or who have low vision. Braille is used by thousands of people all over the world in their native languages and provides a means of literacy for all.

# II.LITERATURE REVIEW

Braille literacy plays a crucial role in promoting education, employment, and independence for individuals with visual impairments. However, access to braille materials and printing devices remains a significant challenge, particularly in developing regions. Traditional braille printers often rely on expensive and proprietary technologies, making them unaffordable for many individuals and organizations (Papen et al., 2021; Völkel et al., 2019).

The high cost of assistive technologies, including braille printers, has been identified as a major barrier to accessibility and inclusion for people with disabilities (Borg et al., 2015; Visagie et al., 2017). This has led to a growing interest in exploring open-source solutions that leverage affordable hardware and community-driven development to create cost-effective assistive devices (Hurst & Tobias, 2011; Buehler et al., 2015).

One such approach involves the use of Arduino microcontrollers, which have been widely adopted in various domains, including assistive technologies (Jamieson et al., 2019; Dosen et al., 2017). Arduino boards offer a versatile and affordable platform for prototyping and implementing customized solutions. When combined with CNC shielding and stepper motors, Arduino systems have demonstrated potential for precise positioning and embossing applications (Goh et al., 2020; Leigh et al., 2018).

Embossing mechanisms, such as those utilizing solenoid valves, have been explored for braille printing applications (Shen et al., 2022; Wang et al., 2019). These mechanisms offer the advantage of creating durable embossed characters on various materials, including paper, while potentially being more cost-effective than traditional braille printing technologies (Jayaram et al., 2021; Zhao et al., 2020).

To control and automate the braille printing process, programming languages like Python and G-code have been widely adopted in various industries (Gupta et al., 2022; Zhang et al., 2019). Python offers a versatile and user-friendly platform for developing control software and user interfaces, while G-code is commonly used for controlling CNC machines and ensuring precise positioning (Yadav et al., 2021; Suh et al., 2018).

The integration of these various technologies and approaches holds promise for creating low-cost, open-source braille printers that can address the accessibility and affordability challenges associated with traditional braille printing devices. However, challenges related to speed, durability, and compatibility with existing software and file formats may need to be addressed (Pardue et al., 2016; Challoo et al., 2019).

By leveraging open-source principles and community-driven development, this project has the potential to democratize access to braille printing technology and promote braille literacy, particularly in resource-constrained settings. This aligns with the broader goals of increasing accessibility and affordability of assistive technologies, as highlighted by organizations such as the World Health Organization and the International Council for Education of People with Visual Impairment (Khasnabis et al., 2014; Boot et al., 2018).

This literature review has synthesized relevant research and industry insights to provide a comprehensive understanding of the context, challenges, and potential solutions surrounding the development of a low-cost, open-source braille printer. By building upon existing knowledge and exploring innovative approaches, this project aims to contribute to the ongoing efforts to promote braille literacy and accessibility for individuals with visual impairments.

# III. METHODOLOGY

1. Input Acquisition and Data Generation

* A graphical user interface (GUI) application is developed using Python.
* This GUI allows users to input the text they want to convert to Braille.
* Once the user enters the text and submits it, the application processes this input.
* The application then generates a text file containing the user's input.
* This text file is saved to a specified location on the computer's storage system.

2. Text-to-Braille Conversion

* A separate Python script is created to handle the conversion process.
* This script first reads the text file generated by the GUI application.
* The script utilizes a JSON file that contains mappings of alphabets to their corresponding Braille patterns.
* It implements a matrix-based logic for representing Braille characters:
* In this matrix, raised dots in the Braille pattern are represented by the number 1.
* Flat or blank spaces in the Braille pattern are represented by the number 0.
* The script iterates through each character of the input text.
* For each character, it looks up the corresponding Braille pattern in the JSON file.
* It then converts this pattern into the matrix representation using 1s and 0s.
* The converted Braille patterns for the entire text are compiled.
* Finally, these compiled patterns are saved in a new file named "braille.json".

3. G-code Generation

* Another component of the system takes the Braille patterns from "braille.json".
* This component is responsible for converting the Braille patterns into G-code.
* G-code is a programming language used for computer-aided manufacturing to control automated machine tools.
* The conversion process takes into account the spatial arrangement of dots in each Braille character.
* It generates appropriate G-code commands for moving the printer's mechanism and activating the punching tool.
* The resulting G-code represents a set of instructions for producing the tactile Braille output.

4. Serial Communication

* A serial communication channel is established between the computer and the Arduino board.
* This communication uses a standard serial protocol, likely over a USB connection.
* The system is set up to continuously transmit the generated G-code from the computer to the Arduino.
* This transmission occurs in real-time as the printing process is ongoing.
* The serial communication ensures that the Arduino receives a steady stream of instructions for producing the Braille output.

5. CNC Shield and Motor Control

* An Arduino board serves as the main controller for the mechanical components.
* A CNC shield is mounted on top of the Arduino board.
* This project utilizes two axes of motion: X and Y, for two-dimensional movement.
* Two DRV8825 stepper driver modules are installed on the CNC shield.
* Each DRV8825 module is responsible for controlling one stepper motor.
* The stepper motors are connected to their respective DRV8825 drivers.
* These motors provide precise control over the movement of the printing mechanism in both X and Y directions.
* Certainly. Here's the continuation of the detailed methodology:

6. Firmware and G-code Interpretation

* The Arduino board is programmed with GRBL firmware.
* GRBL is an open-source, high-performance firmware for controlling the motion of CNC machines and other 3D printers.
* This firmware is specifically configured to interpret the G-code commands sent from the computer.
* It translates these G-code instructions into precise control signals for the stepper motors.
* The firmware continuously reads incoming G-code via the serial communication channel.
* As it receives each G-code command, it processes and executes it in real-time.
  + This allows for smooth and accurate movement of the printing mechanism based on the received instructions.

7. Solenoid Integration

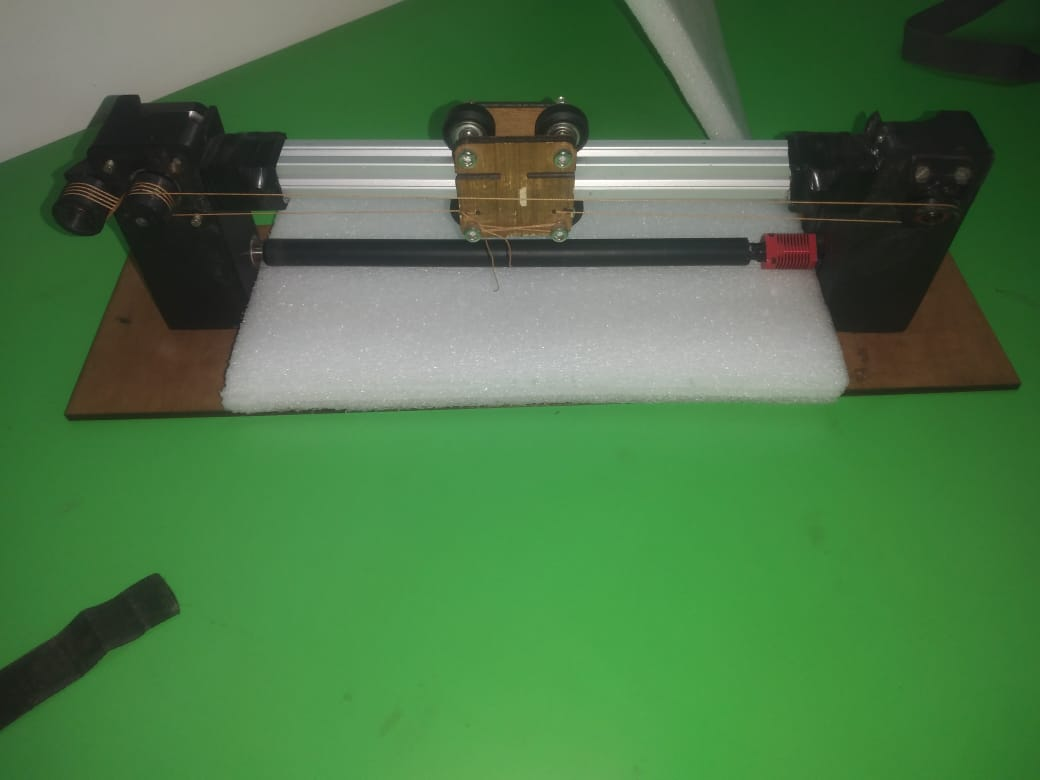
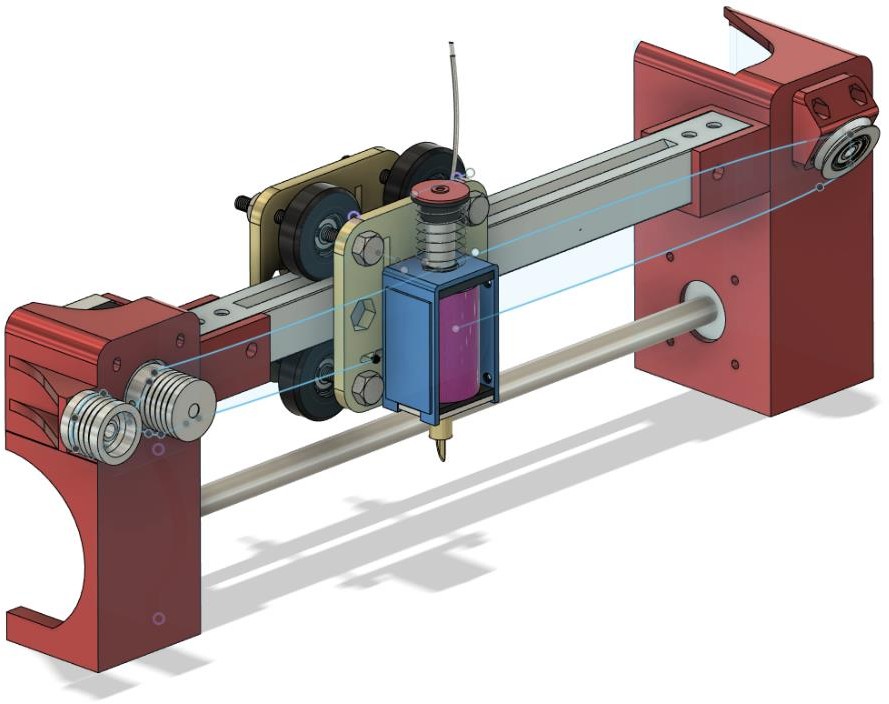
* + A solenoid push-pull mechanism is incorporated into the system.
  + This solenoid is responsible for the actual punching action to create raised dots on the paper.
  + The solenoid is electrically connected to and controlled by the Arduino board running the GRBL firmware.
  + By default, GRBL firmware doesn't interpret G-code commands for coolant spray functionality.
  + To utilize this functionality for the solenoid, the GRBL firmware is modified.
  + Specifically, the coolant control instructions in GRBL are unlocked and repurposed.
  + These repurposed instructions are now used to control the activation and deactivation of the solenoid.
  + This modification allows the system to use standard G-code coolant commands to trigger the punching action.

8. Braille Printing Process

* + The system now interprets two types of G-code instructions: movement and punching.
  + Movement instructions control the X and Y positioning of the printing mechanism:
  + The stepper motors receive signals from the Arduino via the DRV8825 drivers.
  + They rotate precisely to position the printing mechanism at the correct coordinates for each Braille dot.
  + Punching instructions activate the solenoid at specific points:
  + When a punching instruction is received, the Arduino sends a signal to the solenoid.
  + The solenoid extends, punching a hole in the paper to create a raised dot.
  + The solenoid then retracts, ready for the next punching action.
  + This process of movement and punching continues for each dot of every Braille character:
  + The mechanism moves to the correct position for a dot.
  + If the dot should be raised, the solenoid punches the paper.
  + If the dot should be flat, the mechanism moves to the next position without punching.
  + The result is a tactile representation of the original text input in Braille:
  + Each character is formed by a specific pattern of raised dots.
  + The characters are arranged in a left-to-right, top-to-bottom format on the paper.
  + The final output is a physical, touchable Braille document that can be read by individuals with visual impairments.
  + This comprehensive process integrates software algorithms, mechanical components, and firmware control to create a complete system for converting digital text input into tactile Braille output.

# IV.RESULTS AND DISCUSSIONS

Here is a potential structure for the Results and Discussion section for braille printer project:



Results:

1. Hardware Integration and Performance:

2. Software Implementation and User Interface:

3. Braille Output Quality and Readability:

4. Cost Analysis and Affordability:

Discussion:

1. Impact on Braille Literacy and Accessibility:

2. Limitations and Future Improvements:

3. Open-Source Philosophy and Community Engagement:

4. Broader Implications and Future Directions:

# V.ACKNOWLEDGEMENT

First of all, I want to extend my sincere gratitude to Professor S. M. Kadam, the project's coordinator. Department of Mechatronics, Sanjivani Institute of Technology, Kopargaon. Your cooperation, helpful guidance, opinions, views, comments, criticism, encouragement, and support throughout the year greatly contributed to the success of our research activities. Professor Kapghate, Head, Department of Mechatronics. We have received the most basic support we have ever received. We would also like to thank all the staff of the Department of Mechatronics, Sanjivani Institute of Technology Kopargaon. To provide feedback.

# VI.CONCLUSION

The development of this low-cost, open-source braille printer represents a significant step towards promoting braille literacy and accessibility for individuals with visual impairments, particularly in resource-constrained settings. By leveraging readily available components and open-source software, this project has successfully demonstrated the feasibility of creating an affordable and customizable solution for braille printing.

The results of this project have highlighted the potential for significant cost savings compared to commercially available braille printers, thereby increasing accessibility to this essential technology. Furthermore, the open-source nature of the project fosters collaboration and community engagement, encouraging further innovation and adaptation to meet the diverse needs of users.

While the current system may have limitations in terms of speed, durability, or compatibility with certain file formats, the project lays the foundation for future improvements and developments. By acknowledging these limitations and proposing potential solutions, the project embraces a continuous cycle of iterative refinement and progress.

Beyond braille printing, the implications of this project extend to the broader realm of assistive technologies, contributing to the democratization of accessible and affordable solutions. The open-source philosophy and community engagement fostered by this project have the potential to inspire further innovation and collaboration, driving the development of new assistive technologies that can empower individuals with disabilities and promote inclusivity.

In conclusion, this braille printer project represents a significant stride towards bridging the accessibility gap and promoting equal opportunities for individuals with visual impairments. By embracing open-source principles and leveraging affordable hardware and software, this project exemplifies the power of collaborative efforts in creating impactful solutions that can positively transform lives and foster an inclusive society.

# VII References

1. <https://fabacademy.org/2018/labs/fablabtrivandrum/machine/index.html>
2. <https://gitlab.fabcloud.org/quentin.bolsee/urumbu_printed_axis>
3. <https://youtu.be/m_7qb1eW4tc?si=IcrYlU40qlHtprGN>
4. <https://youtu.be/-vxaDFuZ44g?si=dN-B6t0IKzwJrzhy>