IoT-Enabled Braille Display for UPI Payment Verification

Sharmad Kalpande
Mrunmayee Makh
Indian Institute of Science Education and Research Bhopal

1 Introduction

An estimated 253 million people are visually impaired: 36 million are blind and 217 million have moderate to severe visual disabilities [13] and the number of visually impaired people is expected to double by 2050 [10]. Braille is currently used by almost 6 million people in the world. India has a third of the blind population of visually impaired people, of which almost 1% are literate or use the Braille language [6].

Braille is a tactile writing system that enables individuals who are blind or visually impaired to read and write through touch. Developed by Louis Braille in the early 19th century, the system uses a series of raised dots arranged in a 2x3 grid, known as a Braille cell, to represent letters, numbers, and punctuation [7]. By running their fingers across these patterns, users can interpret text, gaining access to literature, education, and communication. Braille has evolved to include adaptations for different languages, scientific symbols, and digital interfaces, making it an essential tool for inclusivity in the modern world.

IoT has improved accessibility and changed how we interact with technology, reducing barriers for people with disabilities. Connectivity between devices enhances communication, mobility, and independent living for individuals with diverse needs. For visually impaired users, integrating IoT with Braille systems has increased independence and access to information. IoT-enabled Braille devices can receive real-time notifications, translate digital content, and assist with navigation through haptic feedback. Research has led to Braille displays that support real-time text translation and alerts. For example, Ramos-García et al. introduced a one-character refreshable Braille display using IoT to support visually impaired students in Mexico [11]. Another system combines AI and IoT to convert printed text into Braille [1]. These developments help users interact with their environment and access information more easily.

Most of the devices available are very expensive. For example, the Dot Pad, a refreshable tactile display used primarily for teaching visually impaired persons, is priced around $11,995 USD, limiting its affordability. Another device is the T-dot Braille Keyboard, which is compact and lightweight, designed as a Bluetooth-enabled Braille input device for visually impaired users, costing approximately $350 USD. Despite their effectiveness, the high prices of these assistive technologies create significant barriers, restricting widespread access and adoption within the visually impaired community. So in this work we develop a simple smart solution that help visually impaired user to very their UPI payment amounts using a refreshable braille device which is of dimension 25cm X 8cm X 7cm, and cost-effective (under $35 USD). Our major contribution are :

- Refreshable Braille device, with three braille cells.
- A dummy payment app that helps scan and send the amount to the device.

2 Related works

Recently, several studies have explored assistive technologies and tactile feedback systems to comfort the visually impaired in today's digital world. A 2024 IEEE Access paper titled "Making Digital Payments Accessible Beyond Sight" explained how accessible UPI apps are for visually impaired people[8]. It identified several limitations such as unlabeled elements and confusing navigation flows. These findings also highlighted the lack of tactile or private alternatives for blind users during UPI transactions. The 2023 DivyaPay initiative introduced a UPI app designed specifically for the disabled. It uses haptic feedback and voice-based guidance to assist the users[3]. Although beneficial, relying on sound can compromise privacy in public places. The Union Bank Sparsh Braille Debit Card (2022) allowed visually impaired users to identify their cards but it did not offer real-time transaction verification or dynamic Braille feedback[9]. Kim and Lee (2021) developed a low-cost servo-based refreshable Braille pad[4]. However, it was not specifically developed for transactions and also was not easy to carry around.nother solution, Pay Voice (2018) from Brazil, used optical character recognition (OCR) and voice to inform users of transaction values, but again lacked tactile output and maintained no privacy in crowded environments[5]. In contrast, our project provides real-time tactile Braille output for UPI transactions, addressing the limitations of privacy and dynamic feedback, which were inadequately addressed in previous systems.

Study/System	Focus Area	Output Modality	Limitations
IEEE Access (2024)	UPI app accessibility	Audio	No tactile output, privacy not ensured
DivyaPay (2023)	Voice-guided UPI system	Audio/Haptic	Audio-based, lacks tactile, privacy concern in public
Sparsh Debit Card (2022)	Braille-enhanced debit card	Tactile (Static)	No dynamic feedback or integration with UPI
Kim & Lee (2021)	Refreshable Braille display	Tactile	Not portable, not designed for transaction purposes
Pay Voice (2018)	Audio-based payment verifier	Audio	No tactile, lacks privacy, dependent on OCR

Table 1: Overview of Studies and Systems Focused on Accessibility in Financial Transactions

3 Methodology

We developed a refreshable device that displays numerical values of up to three digits in Braille format. Additionally, we created a dummy mobile application that simulates the functionality of



Figure 1: Overall workflow of our project

payment apps. This application scans UPI QR codes and transmits the corresponding transaction amount to the device for Braille output.Figure 1 illustrates the overall workflow of our project.

3.1 Mobile App and software

We developed a dummy payment application that extracts the UPI amount from a scanned QR code and transmits it to the device. The app was developed using Kotlin and Java in Android Studio[1], with XML used for UI design. Figure 2 shows the user interface of the app and the sequential steps involved in the process.

We use the ESP32 as the microcontroller in our system. The extracted transaction amount is received by the ESP32 through a BLE UART connection. The BLE UART connection allows the ESP32 to communicate wirelessly with a smartphone using Bluetooth Low Energy (BLE). The ESP32 advertises itself as a BLE device and sets up a communication channel based on the Nordic UART Service (NUS). This service defines two key channels: RX (Receive) for receiving data from the smartphone and TX (Transmit) for sending data to the smartphone. The amount is processed within the ESP32 using Thonny IDE (Integrated development environment) designed for MicroPython. MicroPython enables us to write lightweight, efficient scripts that run directly on the ESP32, allowing real-time data processing. The algorithms on the ESP32 convert the transaction amount into a format suitable for Braille output, which is then displayed on the device.

3.2 Design and Hardware

Now that the transaction amount is available in the ESP32, we require a mechanism to display it in Braille format. For this purpose, we use MG90S servo motors along with a custom-designed shaft mechanism that simulates Braille dots. Each servo motor is coupled to a vertical shafts that either protrudes or remains flat based on the Braille representation of the respective digit.

In Braille, numeric digits from 0 to 9 can be represented using only the top four dots—specifically, dots 1, 2, 4, and 5 as shown in Figure 3. Thus, we made use of only 4 motors instead of 6 motors for each digit saving the space, power and cost. So, we used 12 motors in all instead of 18 for displaying 3 numerical digits.

[1]Android Studio

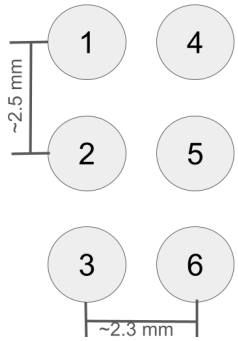


Figure 3: Illustration of Braille dots

For 3D design, we used AutoCAD Fusion 360. We designed a motor box to securely hold the servo motors in place, along with custom shafts featuring smooth, rounded tops that function as Braille dots. Additionally, we created a Braille housing box to organize and support all components, and a lid with precisely positioned Braille holes through which the dots can rise.

We chose the ESP32 as the main microcontroller due to its ample GPIOs with PWM support and built-in Bluetooth and Wi-Fi capabilities. Each servo motor is connected to a dedicated GPIO pin. Upon receiving the transaction amount from the app, the ESP32 converts it into a Braille pattern and sends PWM signals to rotate the corresponding servo shafts by 30°, raising the Braille dots for tactile reading. In Figure 4 we can see the placement of the motor for a braille cell the number represents the braille dot, dot 1 and dot 5 move clockwise 30° while dot 2 and dot 4 move counterclockwise 30°.

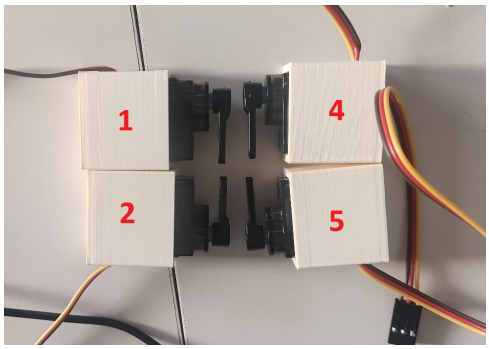
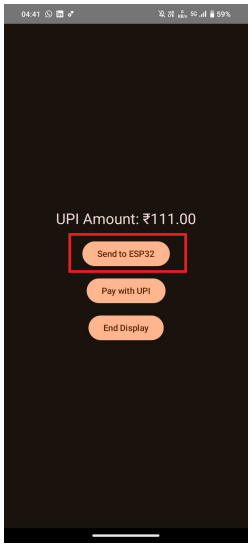


Figure 4: Positions of motors

We initially designed a compact layout for all the servo connections, powerlines and the microcontroller on a single PCB. The PCB was manually soldered with pin headers so as to match the ESP32's GPIO pinout.Although most of the components were soldered carefully, the high number of connections and tight spacing made it challenging to ensure consistent signal integrity and reliable power delivery to all servos. As a result, we had to face some issues during testing including unexpected servo movements, code-crashing, and non-responsive pins. Given the time constraints and to maintain



Step 1: Scan QR.



Step 2: Send to ESP32.



Step 3: End Display on Device.

Figure 2: Sequential steps of the dummy UPI payment app.

functional reliability for the demonstration, we eventually transitioned to a breadboard-based setup, which allowed for quicker debugging and flexible rewiring.

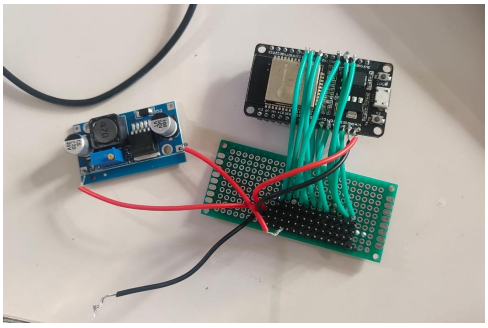


Figure 5: Soldered PCB

4 Results

After assembling all the components—including the ESP32 microcontroller, servo motors, custom shafts, and 3D-printed enclosures—we successfully built a working prototype of the device. The system accurately receives the transaction amount from the mobile application, processes it, and displays the corresponding braille representation.

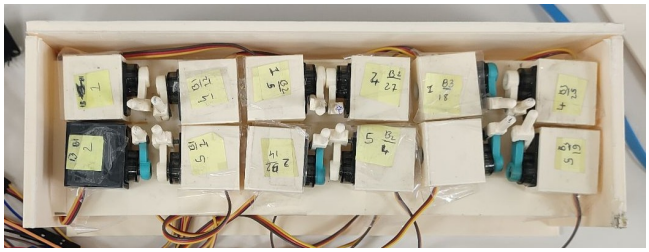


Figure 6: Internal box structure.

Figure 6 shows the internal of device and Figure 7 show the fully functional Braille device. To evaluate its performance, we conducted a small user study involving five participants. Each participant was given three random UPI amounts, which were scanned and sent to the device. The participants, unaware of the amounts, were asked to identify the values by feeling the Braille output on the device. For reference, a Braille number chart (0–9) was also provided on the screen. After completing the test, they filled out a Google Form that assessed three criteria: Usability, Accuracy, and Comfort, each rated on a scale from 1 (worst) to 5 (best). Table 2 presents the mean scores from their responses.

Table 2: Mean user evaluation scores across three criteria.

Criteria	Mean Score
Usability	4.0
Accuracy	4.8
Comfort	2.8

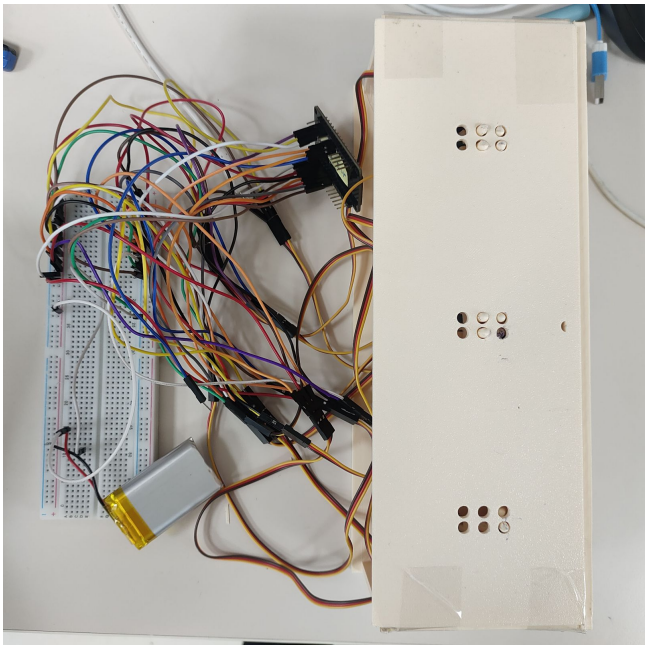


Figure 7: Complete Setup with breadboard and Esp-32

5 Conclusion

This project successfully demonstrates a working prototype of a Braille-based IoT device as from evaluation Table 2we can see that we have build our model which is usable, accuretly displays number in braille formate but lacks in comfort. This device can help visually impaired users verify UPI transaction amounts privately and independently. The system is cost-effective and addresses a clear gap in existing digital payment tools.

Despite its functionality, it has a few limitations. The use of servo motors for each braille dot makes the device bulkier and less portable. Moreover, only 3 digits can be displayed at a time which makes the system limited to only small amounts (ranging from 1 to 999). Mechanical complexity contributes to an increase in cost and space.

The future enhancements include making the system lightweight, improving portability, functionality, and user autonomy. Expanding the display to show more than three digits, including decimal points, would make the device applicable to a wider range of transactions. Additionally, integrating a Braille-based input mechanism would allow users to enter payment amounts and PIN independently, enabling a two-way secure interaction.

6 Work Division

The project was a collaborative effort. The original idea for the project was proposed by Mrunmayee which was later refined through mutual brainstorming. The mobile app development, software integration, 3D design, mechanical hardware setup—including the alignment of motors and shafts for smooth operation—assembly of the complete system, and device evaluation were all carried out by Sharmad. Mrunmayee led the hardware development which included designing the circuit, soldering the PCB and integrating it with ESP32 microcontroller. She also conducted the literature review and explored relevant research papers to support the methodology and the documentation was done by the contribution of both. Satchit and Katyayni helped in hardware, design, and 3D printing.

References

[1] Anonymous. 2023. AI and IoT-Based Braille System for Printed Material Translation. In *IEEE Xplore Conference*. https://ieeexplore.ieee.org/document/10788373

[2] Tigmanshu Bhatnagar, Nicolai Marquardt, Mark Miodownik, and Catherine Holloway. 2021. Transforming a Monolithic Sheet of Nitinol into a Passive Reconfigurable Tactile Pixel Array Display at Braille Resolution. In *2021 IEEE World Haptics Conference (WHC)*. IEEE, 409–414.

[3] Sourav Biswas, Sanjib Kumar Das, and Sumit Gupta. 2022. DivyaPay: Bridging the Digital Divide. *FOSET-12th Academic Meet 2022* (2022). https://preprint.prepare.org.in/index.php/foset/article/view/307

[4] Xu Chen et al. 2024. Development and Evaluation of Refreshable Braille Display and Active Touch-Reading System for Digital Reading of the Visually Impaired. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2024). doi:10.1109/TNSRE.2024.3363495

[5] Guilherme Folego, Filipe Costa, Bruno Costa, Alan Godoy, and Luiz Pita. 2018. Pay Voice: Point of Sale Recognition for Visually Impaired People. *arXiv* (2018). https://www.researchgate.net/publication/329705392_Pay_Voice_Point_of_Sale_Recognition_for_Visually_Impaired_People

[6] Deccan Herald. 2025. India's Braille literacy crisis. https://www.deccanherald.com/opinion/indias-braille-literacy-crisis-3339748. (Accessed on 03/11/2025).

[7] Javier Jiménez, Jesús Olea, Jesús Torres, Inmaculada Alonso, Dirk Harder, and Konstanze Fischer. 2009. Biography of Louis Braille and invention of the Braille alphabet. *Survey of Ophthalmology* 54, 1 (2009), 142–149.

[8] N. Kumar and P. Tripathi. 2024. Making Digital Payments Accessible Beyond Sight: A Usability Study of UPI-Based Smartphone Applications. *IEEE Access* (2024), 1–12. doi:10.1109/ACCESS.2023.3348840

[9] Union Bank of India. 2024. Sparsh Braille Card. https://www.unionbankofindia.co.in/en/Details/sparsh-braille-card

[10] World Health Organization. 2023. Blindness and vision impairment 2023. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. (Accessed on 03/11/2025).

[11] Ramos-García. 2023. A One-Character Refreshable Braille Display Using IoT Technology. *MDPI Proceedings* 27, 1 (2023), 11. https://www.mdpi.com/2673-4591/27/1/11

[12] Anshul Singhal, Pranay Jain, Piyush Chanana, Dhruv Jain, Rohan Paul, M Balakrishnan, and PVM Rao. 2013. Application of shape memory alloy (SMA) based actuation for refreshable display of braille. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 55911. American Society of Mechanical Engineers, V004T08A053.

[13] Braille Today. 2025. Braille Today 2025. https://museelouisbraille.com/en/le-braille-aujourd-hui. (Accessed on 03/11/2025).

[14] Barbara Vsepic, Abdurrahman Ghanem, and Stephan Vogel. 2015. BrailleEasy: one-handed braille keyboard for smartphones. In *Assistive Technology*. IOS Press, 1030–1035.