

Multimodal Interaction and Performance Validation of a Raspberry Pi-Based Grade 1 Braille Notetaker

Shaharil Mad Saad and Tan Nian Sien

Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru Johor, Malaysia

DOI: <https://doi.org/10.47772/IJRISS.2026.100800857>

Received: 5 September 2026; Accepted: 10 September 2026; Published: 19 September 2026

ABSTRACT

Electronic Braille notetakers can support independent reading and writing, but their interaction logic must coordinate chorded input, refreshable output, file operations, cursor feedback, and audio prompts within a compact device. This paper presents the design and functional validation of a Raspberry Pi 4-based Grade 1 Braille notetaker developed as an extension to the UTM eBraille platform. Fifteen mechanical switches provide six-dot Perkins input and command navigation, while a 24-cell Braille module, speaker, custom main circuit board, and Braille-cell interface board form the output and interconnection layers. The software is organized into file manager, reading, editing, trash, settings, and search states. Bidirectional conversion routines translate keyboard chords into text and map text characters, capitalization indicators, numeric indicators, and cursor positions to Braille-cell data. Functional tests verified basic characters, capitalized letters and words, numbers, cursor movement, text insertion, save reminders, and file-property presentation. A repeated entry test used the ten-character sequence “abcdefghij” over 20 trials. Mean entry time was 0.3233 s per character, equivalent to 3.1134 characters/s, and mean character accuracy was 97.5%. The results demonstrate that a state-based multimodal interface can combine essential note editing and file management on an embedded Braille platform. Current limitations include text-only files, uncontracted English Braille, an external speaker, and the absence of participant-based usability evaluation.

Keywords: assistive technology, Braille notetaker, embedded system, multimodal interaction, Raspberry Pi, refreshable Braille

INTRODUCTION

Access to writing tools is as important as access to reading materials for blind learners and professionals. Conventional Perkins Braille writers provide reliable six-key chorded entry but produce embossed paper and do not inherently support searchable digital files, editing, copying, or electronic storage [10]–[13]. Refreshable displays and commercial electronic notetakers add digital interaction, although their cost, platform dependence, and feature complexity motivate continued development of configurable embedded alternatives [14]–[18].

The UTM eBraille platform was initially developed to present digital materials such as the Qur'an, textbooks, and dictionaries. The project reported in the source thesis extended that reading-oriented platform with a note-taking workflow. The central design problem was to make several operations—Braille entry, text conversion, tactile display, cursor movement, file navigation, recovery of deleted items, searching, and audio feedback—behave coherently through a limited number of physical controls.

This paper reframes the project around multimodal interaction and performance validation. Its contributions are: (i) a state-based interface spanning six operational modes; (ii) bidirectional mapping between Grade 1 Unified English Braille and plain text; (iii) a tactile cursor represented by blinking dots 7 and 8; and (iv) quantitative characterization of chorded entry over 20 repeated trials. The evaluation is restricted to the evidence recorded in the thesis and does not claim clinical effectiveness or usability with blind participants.

Design Context and Requirements

Braille entry and electronic note taking

A standard Braille cell represents a character through six primary dots, while electronic displays may provide dots 7 and 8 for computer notation and interaction feedback [3]–[5], [20]. Perkins-style entry uses simultaneous key combinations corresponding to the six dots. Unified English Braille additionally relies on contextual indicators: one capital indicator affects the following letter, two indicators capitalize a word, and a numeric indicator changes the interpretation of subsequent letter patterns [19], [20]. These rules require the input algorithm to retain state rather than translate each chord independently.

Commercial devices reviewed in the thesis ranged from compact readers to feature-rich notetakers [15]–[18]. The target prototype was therefore specified to provide a smaller core workflow: open and navigate plain-text files; enter and edit Grade 1 English Braille; save changes; view file properties; search; restore or permanently remove files; customize sound and sorting; and present text through a refreshable Braille line. The scope excludes contracted Braille and file formats other than TXT.

Table 1: Operational states and implemented functions.

Mode	Principal functions
File manager	Navigate folders/files; view properties; create, rename, copy, cut, paste, delete, open, and search.
Reading	Display file content line by line and enter editing mode.
Editing	Enter text; move the cursor by character or word; mark, copy, cut, paste, save, and exit.
Trash	Inspect deleted items; restore them or delete them permanently.
Settings	Enable/disable sound and select sorting field and order.
Search	Enter a keyword and navigate matching folders or files.

SYSTEM DESIGN AND METHODOLOGY

Hardware architecture

The prototype uses a Raspberry Pi 4 Model B as the controller. A custom main circuit board links the processor, a DC–DC converter, the input switches, and the output interfaces. A separate Braille-cell board connects the refreshable module. Fifteen mechanical switches implement Dot 1–Dot 6, Enter, Backspace, Space, Escape, Shift, and four directional commands. A speaker provides auditory confirmation and status information. The assembled system is shown in Figure 1.



Figure 1: Completed Raspberry Pi-based Braille notetaker prototype.

Table 2: Prototype hardware and functional roles.

Hardware element	Role
Raspberry Pi 4	Executes the Python control, conversion, file-management, and output routines.
15 mechanical switches	Provide six-dot chorded entry and navigation/editing commands.
24-cell Braille module	Presents one constrained line of tactile characters and cursor feedback.
Main and Braille-cell PCBs	Interconnect the controller, converter, switches, and display module.
Speaker	Provides optional spoken or audible interface feedback.

State-based interaction architecture

The main loop reads switch states, determines the active operating mode, performs the requested operation, and refreshes tactile or audio output. Directional keys provide consistent list and line navigation. Shift combined with dot or command keys invokes less frequent actions, reducing the number of dedicated controls. Examples include Shift+Dots 4–6 for rename, Shift+Dots 4–5 for a new file, Shift+Dot 2 for copy, Shift+Dot 3 for cut, Shift+Dot 1 for paste, and Shift+Backspace for delete.

In reading mode, text is converted to Braille patterns and divided into rows of at most 24 cells before transmission. Figure 2 summarizes the thesis data path: character patterns are converted to cell codes, rearranged for the output electronics, and sent as a 24-cell data frame. In editing mode, the same translation operates in reverse for chorded input. The cursor is indicated by raising and lowering dots 7 and 8 at 0.5 s intervals, preserving the character pattern while providing a local tactile position cue.

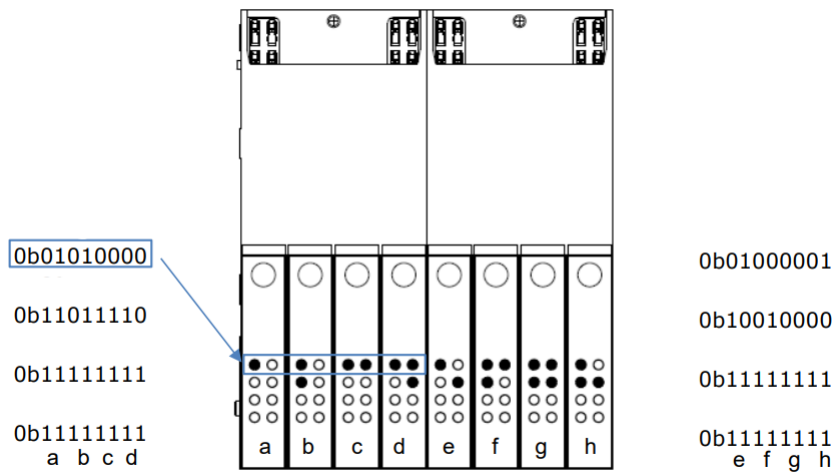


Figure 2: Organization of character codes into a 24-cell Braille output frame.

Functional and performance tests

Functional verification compared the Braille-cell output with terminal data for representative Grade 1 constructs: lower-case text, letter and word capitalization, numbers, two cursor positions, inserted text, an unsaved-change prompt, and file properties. These cases exercise both translation rules and navigation state. The performance test required entry of the sequence “abcdefghij”. Time for ten characters and the resulting output were recorded for 20 trials. Character accuracy was calculated as the number of correct positions divided by ten, expressed as a percentage. Time per character and characters per second were derived from each trial before their means were reported.

RESULTS AND DISCUSSION

Translation and interaction verification

The lower-case sequence “abc def” was represented correctly on the tactile cells and in the terminal-generated data frame (Figure 3). Capitalization tests showed a single capital indicator before the relevant letter in “Ghi Jk” and a double indicator before the fully capitalized word “LMNOP”. The number test inserted the numeric indicator before “123 45”, preventing the shared Braille patterns from being interpreted as letters. Together, these cases confirm that the converter retained the contextual state required by Grade 1 Braille rather than performing an isolated one-pattern/one-character substitution.



Figure 3: Representative lower-case output on the Braille cells and Raspberry Pi terminal.

Cursor tests placed the blinking dots at the eighth cell and then at the fifth cell after three left commands. Text entry appended “Tan” at the end of the selected line, while the exit routine displayed “Save or not?” when changes had not been stored (Figure 4). File-property navigation successfully exposed the filename, modification time, access time, type, and size on the tactile line. These observations verify the intended control paths, but they measure technical function rather than user comprehension or task efficiency.

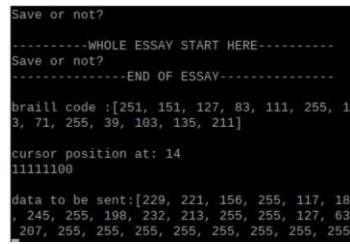


Figure 4: Unsaved-change reminder on the Braille cells and terminal.

Entry speed and character accuracy

Across 20 trials, 17 reproduced all ten characters, two contained one incorrect character, and one contained two incorrect characters. Thus, 195 of 200 entered character positions were correct, giving a mean accuracy of 97.5%. The recorded mean time was 0.3233 s per character, corresponding to 3.1134 characters/s. Individual ten-character completion times ranged from 2.8135 s to 4.0995 s. Table 3 presents the complete trial-level evidence.

Table 3: Twenty repeated character-entry trials.

Trial	10-char time (s)	s/char	char/s	Output	Accuracy (%)
1	3.0714	0.3071	3.2559	abcdefghij	100
2	3.0440	0.3044	3.2851	abcdefghij	100
3	4.0995	0.4099	2.4393	abadefggij	80
4	3.4706	0.3471	2.8814	abcdefghij	100
5	3.2259	0.3226	3.0999	abcdefghij	100
6	3.2501	0.3250	3.0768	abcdefghij	100
7	3.4425	0.3443	2.9048	abcdafghij	90
8	3.0504	0.3050	3.2782	abcdefghij	100
9	3.1208	0.3121	3.2043	abcdefghij	100
10	3.4141	0.3414	2.9291	abcdefghij	100
11	3.3107	0.3311	3.0205	abcdefg <h,j< td=""><td>90</td></h,j<>	90
12	3.0280	0.3028	3.3025	abcdefghij	100
13	3.6160	0.3616	2.7655	abcdefghij	100
14	3.1327	0.3133	3.1921	abcdefghij	100
15	2.9794	0.2979	3.3564	abcdefghij	100
16	3.2483	0.3248	3.0786	abcdefghij	100

17	2.8135	0.2813	3.5543	abadeefghij	90
18	3.0783	0.3078	3.2485	abcdeefghij	100
19	3.0131	0.3013	3.3189	abcdeefghij	100
20	3.2513	0.3251	3.0757	abcdeefghij	100
Mean	—	0.3233	3.1134	—	97.5

The error pattern is consistent with input-event sensitivity: repeated or missed switch events can alter one or more chorded characters. The thesis identified inaccurate pressing and mechanical-switch bounce as plausible causes. Because the trials were not distributed across multiple users, the figures characterize the implemented prototype and test procedure rather than a general human typing rate. Debouncing, chord-window tuning, and controlled participant studies are therefore necessary before comparing the device with commercial Braille keyboards.

Scope, strengths, and limitations

The principal strength of the system is functional integration. A single embedded platform coordinates tactile entry, refreshable output, audio feedback, editing, search, and recoverable file deletion. The consistent use of directional and Shift-modified commands also limits the physical control count. The measured accuracy indicates that the prototype can capture rapid sequential chords under the reported test conditions, while the successful indicator and cursor cases show that the software handles essential context beyond basic alphabet mapping. Four constraints remain. Only plain-text files and Grade 1 English Braille are supported. The speaker is external and increases size. Storage is local to the Raspberry Pi SD card without cloud synchronization. Most importantly, the evaluation did not include blind participants, learning time, workload, comfort, discoverability, or extended-note tasks. Future work should add robust hardware and software debouncing, contracted Braille, common office formats, an integrated speaker, backup or synchronization, and user-centred evaluation with representative participants.

CONCLUSION

A multimodal Grade 1 Braille notetaker was implemented on Raspberry Pi 4 using fifteen mechanical inputs, a 24-cell refreshable Braille module, custom interface boards, and audio output. Its state-based software supported file management, reading, editing, trash recovery, settings, and search. Functional tests verified lower-case text, capitalization, numbers, cursor feedback, text insertion, save reminders, and file properties. In 20 repeated ten-character trials, the prototype achieved 0.3233 s per character, 3.1134 characters/s, and 97.5% character accuracy. These results establish technical feasibility for an embedded note-taking extension to the UTM eBraille platform. They do not yet establish user usability; participant-based validation and broader format and Braille support remain the most important next steps.

ACKNOWLEDGMENTS

The authors thank Universiti Teknologi Malaysia and the project supervisor for the facilities, technical guidance, and constructive feedback that supported this study.

REFERENCES

1. S. Kumari, P. Angnani, A. Akole, Y. Bhamare, and Z. Naikwadi, "Enhanced Braille Display: Use of OCR and Solenoid to Improve Text to Braille Conversion," in INCET, 2020, doi: 10.1109/INCET49848.2020.9153996.

2. F. L. M. Chew et al., "Estimates of visual impairment and its causes from the National Eye Survey in Malaysia (NESII)," PLoS One, vol. 13, no. 6, 2018, doi: 10.1371/journal.pone.0198799.
3. J. John, "Recognition of Documents in Braille," arXiv:1709.09875, 2017.
4. N. R. J. Hynes, D. J. Jebaraj, J. S. Kumar, S. Immanuvel, and R. Sankaranarayanan, "Portable electronic braille devices—An overview," AIP Conf. Proc., vol. 2142, p. 140018, 2019, doi: 10.1063/1.5122531.
5. S. Ramachandran, Gururaj, K. N. Pallavi, and N. Rajan, "Text to Braille Converting Communication Device for the Visual and Hearing Impaired Persons," in ICCCI, 2021, doi: 10.1109/ICCCI50826.2021.9402590.
6. S. N. Rao, C. D'Silva, and V. Parthasarathy, "Evaluation of a smartphone keyboard for the visually challenged," in ICECCT, 2017, doi: 10.1109/ICECCT.2017.8117933.
7. M. Baker, E. M. Nightingale, and S. Bills, "An Editing Process for Blind or Visually Impaired Editors," IEEE Trans. Prof. Commun., vol. 64, no. 3, pp. 275–287, 2021, doi: 10.1109/TPC.2021.3090722.
8. S. Dharshan and Shaharil, "Development of Graphical User Interface (GUI) for Teaching Braille in Class for the Blind People," Procedia Undergraduate Mechanical Engineering Research, 2020.
9. Universiti Teknologi Malaysia, "What is eBraille?" Available: <https://engineering.utm.my/mechanical/ebraile/what-is-ebraile/>.
10. C. Judi, "The Perkins Braille Turns Seventy," American Council of the Blind, 2022.
11. Perkins School for the Blind, "Perkins Braille," 2021. Available: <https://brailler.perkins.org/>.
12. WeCapable, "Perkins Braille: Base of all Modern Braille Keyboards," 2022.
13. Perkins School for the Blind, "Why the Perkins Braille," 2021.
14. American Foundation for the Blind, "Electronic Notetakers (Braille)," 2020.
15. Orbit Research, Orbit Reader 20 User Guide, ver. 2.1, 2022.
16. American Printing House for the Blind, Braille Trail Reader LE User's Guide, 2021.
17. SELVAS Healthcare, BrailleSense 6 User Manual, ver. 1.6, 2021.
18. HumanWare, BrailleNote Touch Plus User Guide, 2022.
19. S. Hong, L. P. Rosenblum, and A. F. Campbell, "Implementation of Unified English Braille by Teachers of Students with Visual Impairments in the United States," J. Vis. Impair. Blind., vol. 111, no. 6, pp. 543–556, 2017.
20. C. Simpson, Ed., The Rules of Unified English Braille, 2nd ed. International Council on English Braille, 2013.
21. Johnson Matthey Piezo Products GmbH, "32-Pin Braille Module Data Sheet."
22. Raspberry Pi Trading Ltd., "Raspberry Pi 4 Computer Model B Datasheet," 2021.
23. XP Power, "DC-HVDC Converter Datasheet," 2020.
24. Harman Kardon, "NEO Portable Bluetooth Speaker," 2023.