Indian Journal of Science and Technology, Vol 9(33), DOI: 10.17485/ijst/2016/v9i33/99593, September 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Design and Implementation of Arduino based Refreshable Braille Display Controller

Rushil Gupta*, Parikshit Kishor Singh and Surekha Bhanot

Department of Electronics and Instrumentation, BITS Pilani Pilani Campus, Pilani, India;
f2011307@pilani.bits-pilani.ac.in, parikshit_singh@pilani.bits-pilani.ac.in, surekha@pilani.bits-pilani.ac.in

Abstract

This article describes design of a controller for refreshable Braille displays used by Blind or Visually Impaired people. The controller manipulates the ASCII 64 printable character set in order to stimulate 6 dot combinations in Braille language. It is capable of transliterating English/Devanagari texts into Braille script, which is then rendered on a refreshable display. The prototype presented constitutes a portable, battery powered and stand-alone Arduino that processes pre-programmed English/Devanagari text and actuates Braille cells consisting of 6 Light Emitting Diodes representing a Braille character with the incorporation of input buttons and encoders allowing for controlled traversal of text.

Keywords: Arduino, Blind, Braille, E reader, Visually Impaired

1. Introduction

Research towards improving dissemination of knowledge has increased literacy among the blind or visually impaired empowering them to explore their challenges and abilities in today's fast paced world[1]. The needs and preferences of a learner can be described by three categories: display, control and content[2]. While innovative ideas and solutions help improve accessibility of blind people to technical education, deployment of state of the art technology, development of diverse content for the blind has also been an area of research focus since long ago[3,4].

The blind or visually impaired rely mostly on sounds and touch to communicate and navigate their environment along with Braille, which is the most common tool used by blind people to accessing knowledge. Braille is a reading and writing system that was invented in , and is a globally accepted method to describe letters, numbers and symbols in the form of dots on embossed paper or other materials. The Braille characters are displayed within a predetermined space known as a Braille cell, shown in Figure 1. Braille cells comprise of six dots fashioned in a rectangular aligned in two columns with three dots each, much like a 3×2 matrix. There are 64 combinations of Braille characters possible using the 6 dots. A 6-dot Braille pattern may represent different characters in different languages meaning that a script written for one language may have entirely different meaning in another language. All English language characters can be represented using a single Braille cell. However, different languages such as Devanagari and many Indian languages may require more than one Braille cell for complete rendition of a character since they show the usage of composite or conjugate characters that are constructed by concatenation of two or more characters with a special character known as *halati*on Devanagari[5].

Many software tools have been developed to translate embossed Braille documents into digital counterparts and vice versa, in different languages such as Mandarin, Turkish, Bangla, Odia and Devanagari[6–11]. Some researchers have also reported hardware implementations of electronic-book (e-book) reader for visually challenged. In [12] described a self-learning tool based on ontology theory and RFID for Braille code identification equipped with a voice module running on a Personal Computer (PC). Some devices that use speech recognition/synthesis and a mechanical Braille display as tactile input/output implemented on a PC along with Textbook, a lightweight

*Author for correspondence

and portable e-book reader that consists of a Braille translator, piezoelectric actuator, electronic drive and tactile display have been developed[13,14]. A self-learning tool that implements wireless technology and stimulates body Braille patterns controlled by a special Braille keyboard is also reported in the literature[15].

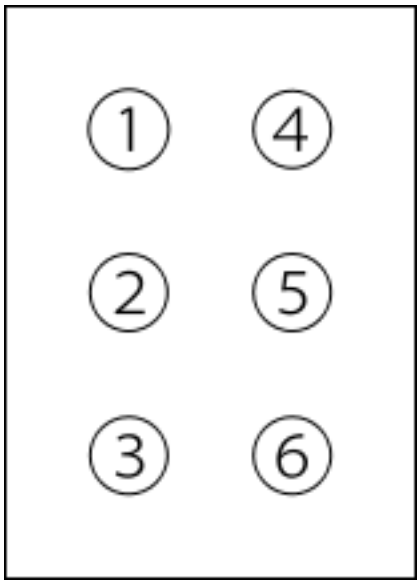

Figure 1. A braille cell.

Most of the above research implementations utilize a PC that have complex programming schemes and therefore cannot be incorporated in a portable architecture. However, fewer works have been reported that can be categorized under embedded systems and consist of Micro-controller Units (MCUs) offering lesser processing power but small footprint, drastically lesser power consumption and therefore can function in a portable form. In [16] describe an English text to Braille converter based on a ATMEGA16 MCU which controls a vibration mechanism used to stimulate Braille. In [17] report implementation of an e-book reading device based on the ATMEGA16 MCU interfaced with a Secure Digital (SD) card that reads text files written in English and features two solenoid based Braille cells running on AC Mains. In [18] present an e-book reader based on a AT89S52 MCU capable of input/output using a computer keyboard along with music recording/playback functionality.

Several devices have been reported in a Technology Resource List compiled by the National Federation of Blind (NFB United States) that is a comprehensive index of software systems and devices pertaining to Braille, speech and other blindness technology updated time to time and contains their general description, usage, features, manufacturer, and pricing information[19]. Even though these tools for Braille learning are available in variety and quality, they are produced in a limited quantity that inherently lead to high costs due to a small market base. Moreover, most of the affordable devices among these are traditional mechanical machines that have meager intelligence, information capacity and inhibit self-learning. Students must rely on their teachers to teach them Braille symbols for the various languages of the world, which imposes a major hindrance in their education of the Braille system.

In light of the discussion above, few research projects show great promise but lack some basic features and functions an e-book reader should have such as:

- Portability
- Low complexity
- Multi-lingual architecture
- Reading speed control

Hence there is considerable scope for the development of a low-cost and portable solution while boosting ergonomics and functionality. It is also noted that MCU architectures offer good flexibility at low cost for controlling a refreshable Braille display, a device that produces Braille dot patterns by electronically/mechanically raising or lowering pins to display information.

2. Arduino based Refreshable Braille Display

This paper implements a refreshable Braille display controller that utilizes Arduino-an open source, cross platform prototyping tool based on easy-to-use hardware and software based on MCU[20]. It offers following suitable features:

- Ease of Programming: It can be easily re-programmed using the desktop software provided by Arduino manufacturers allowing the user to add/remove text files using a PC.
- Peripherals: Arduino board has in-built protection circuits and a reset button that allows the user to restart the program executed by the Micro-controller at any point of time along with in-built USB implementation for easy communication with PC.
- Cross-Platform: A program made for a particular Arduino can be used on other Arduino

boards with little to no modifications. Hardware 'Shields' can also be used on all Arduino boards due to their similar designs.

- Stand-alone System: All Arduino boards have inbuilt power regulators and can be powered via a USB cable or an external power supply that may be a wall-adapter or even a battery, allowing for portability.

3. Design and Implementation

The controller for the refreshable Braille display has been designed for multiple Braille cells, and offers the following functionality and features of a basic e-book reader:

- Read a file stored in memory containing text in the pertaining language.
- Transliterate characters into Braille code.
- Display an array of character son Braille cells.
- Automatic/Manual mode selection button.
- Next/Previous character buttons for manual traversal of text.
- A Potentiometer to determine delay between successive display of lines allowing control of reading speed in automatic mode.
- File select button to jump between different files.

Figure 2 shows the various input output systems discussed connected to the controller. Next we discuss three stages of design and implementation namely: Transliteration, Software, Hardware.

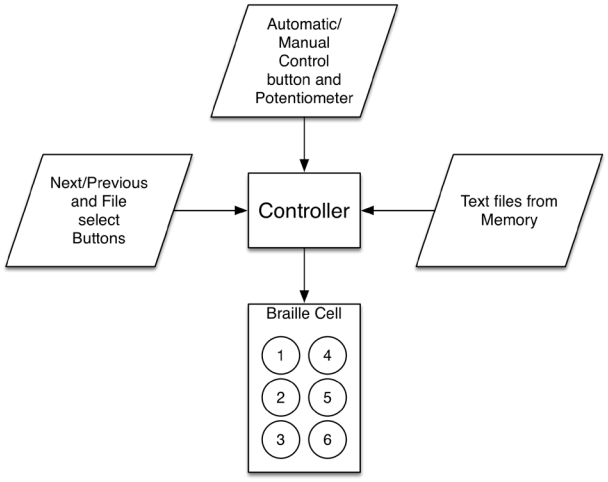

Figure 2. Controller block diagram.

3.1 Transliteration

In order to represent a Braille script constructed for a specific language, each character must be mapped to a Braille pattern, and this information needs to be stored for each language among others. However, it is easier to use a fixed intermediate scheme for storage that can correlate to both desired language and its respective Braille script, thus enabling compatibility with large number of languages. The English and Devanagarito Braille transliteration scheme employed in this paper utilize the printable 64 ASCII character set as shown in reference tables developed by Baraha Software[21].Complementing the simplicity of this device, all Indic language scan be transliterated into a Braille script, known as "*Bharati Braille*", using the Braille reference tables accepted world-wide[22].The desired text is pre-transliterated and stored in ASCII file format to be read by the Arduino board that drastically reduces processing requirements. ASCII Braille encoded documents can be produced readily using the Baraha Indian Language Editor (ILE) tool that allows users to input text in a language of preference, convert the encoding to the desired language and finally export to the corresponding Braille ASCII encoded document by the tool utilizing a phonetic keyboard converting English characters into Indian language text.

As an example, a sample line to be read in Devanagari, shown in Figure 3, taken from Article I of the Universal Declaration of Human Rights[23], is required to be transliterated into Braille script. Figure 4 shows input English characters to, and Figure 5 shows output ASCII characters from, Baraha ILE. Braille script displayed by Arduino is shown in Figure 6.

Similarly, Article I of the Universal Declaration of Human Rights in English language is shown in Figure 7. The equivalent ASCII representation is shown in Figure 8, and Braille script displayed by Arduino is shown in Figure 9.

Each file in the memory is stored in a special format by concatenating " D " for Devanagari and " E " for English to the beginning of respective files so that language can be identified and reported to the user.

सभी मनुष्यों को गौरव और अधिकारों के मामले में जन्मजात स्वतन्त्रता और समानता प्राप्त हैं।

Figure 3. Desired sample text in Devanagari script.

sabhee manuShyoM ko gaurav aur adhikaaroM ke maamale
meM janmajaat svtantrataa aur samaanataa praapt haiM

Figure 4. Input English characters to Baraha ILE.

S^9 MNU@&YO; KO G[RV [R A!IK>RO; KE M>MLE ME; J@NMJ>T @S@VT@N@TRT> [R SM>NT> @PR>@PT H/;4

Figure 5. Output ASCII characters from Baraha ILE.

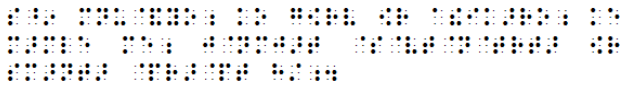

Figure 6. Braille script displayed by Arduino for Devanagari text.

All human beings are born free and equal in dignity and rights.

Figure 7. Desired sample text in English language.

ALL HUMAN BEINGS ARE BORN FREE & EQUAL IN DIGNITY AND RIGHTS4

Figure 8. ASCII equivalent for English text.

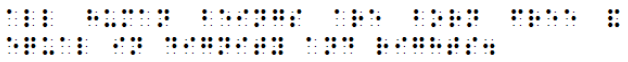

Figure 9. Braille script displayed by Arduino for English text.

3.2 Software

Figure 10 represents the program flowchart utilized by the controller to implement the features as presented in Figure 2 with bounds for file and letter variables as total number of files-1 and total number of letters in file-1, with some program blocks of the controller identified as followed:

- Store and navigate files: The controller must be able to store and read a text file as well as navigate between different files.
- Transliteration: It must be able to understand each character written in the input text and display corresponding Braille code.
- Automatic/Manual Display control: It should allow the user to control his/her reading speed by either displaying characters and traversing through the text regularly at an interval of user' choice (Automatic mode) or by traversing whenever user prompts (Manual mode).

3.3 Hardware

Figure 11 shows the complete schematic, of the Arduino based refreshable Braille display controller excluding the power supply circuit which can be a USB connection to the board, a 9V 500mA DC wall-plug adapter or a 9V alkaline battery, built using Cad Soft EAGLE Version 7.4.0 Light Edition with freeware license[24]. Generic momentary pushbuttons function as the File Select, Automatic/Manual Display mode, Previous, and Next buttons that have a small footprint and low price point. These buttons are to be used as Interrupts to the Arduino program so that corresponding action is instantly performed when pressed, along with the freedom from checking button status frequently. However, these buttons register variable counts instead of one for each press due to bad hardware which requires a de-bouncing circuit for each button to output smooth voltage levels.

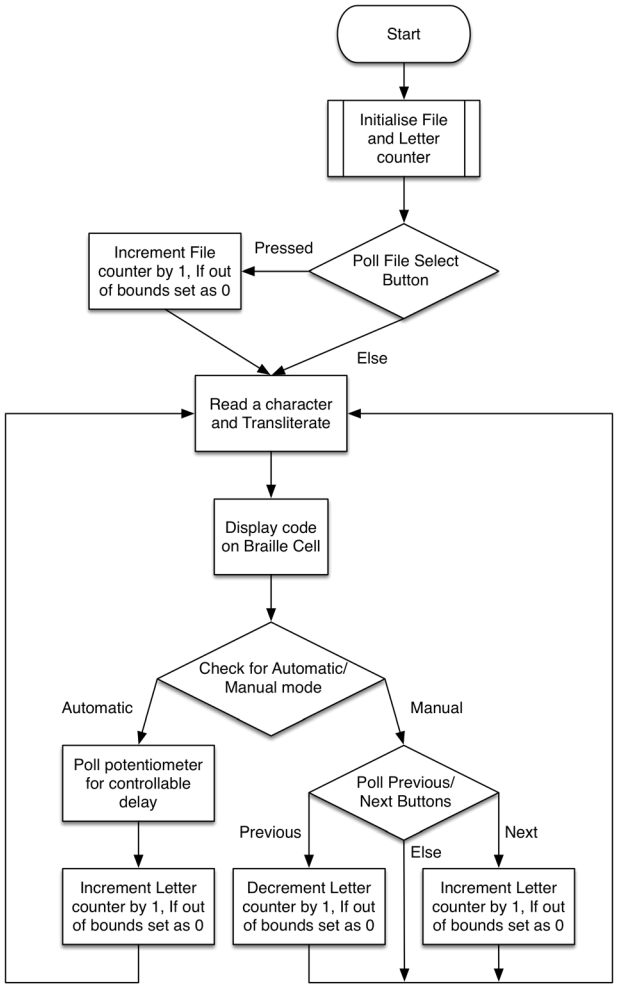

Figure 10. Program flowchart.

The current implementation utilizes two resistors, a diode, a capacitor and an inverting Schmitt trigger to de-bounce one button, highlighted in the schematic. A 10 kilo-ohm single turn rotary potentiometer is used to generate analog levels from 0 to 5V which are then mapped

by the Arduino into 1024 levels starting from 0. These levels determine the delay controller program uses to allow a Braille character stay on display before the next code is flashed on it while the Automatic mode is enabled. The Braille cell represented in the design consists of 6 Light Emitting Diodes controlled by an 8-bit Shift Register IC 74595 which not only requires just 3 inputs to control 8 outputs, but two or more registers can be daisy chained together to control more than 8 outputs (8 x number of registers) meaning that adding more Braille cells is feasible and easy while maintaining the current architecture. The current prototype utilizes three Braille cells.

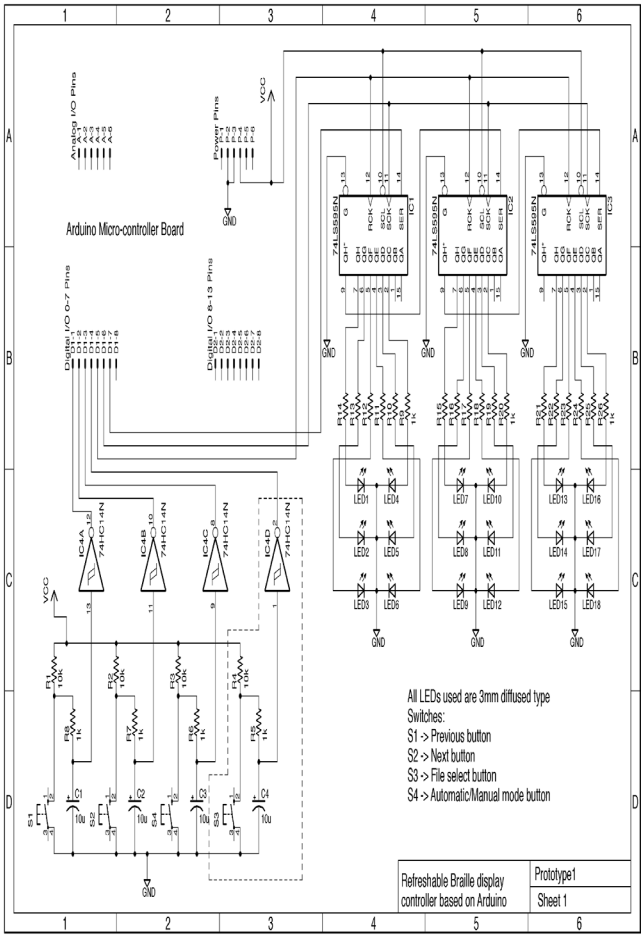

Figure 11. Controller schematic.

4. Conclusion

A simple and cost effective controller for refreshable Braille displays has been implemented using Arduino MCU that features:

- Display scheme on multiple Braille cells achieved using the desired number of shift register ICs (74595) in a daisy chain configuration with only 3 inputs from the MCU (one Braille cell per IC).
- Transliteration function to convert input ASCII coded text to 8-bit format of shift register output.
- Program functions to control traversal of digital text as well as switching to separate files
- Automatic/Manual traversal of text allowed with reading speed control.
- Hardware de-bouncing of momentary pushbuttons used as interrupts to MCU program routine for robust reactions.

Future recommendations include addition of external storage for more text files as well as speech input-output system to enable self-learning mechanisms.

5. References

1. Dias MB, Teves EA. Towards technology with a global heart through compassionate engineering. Proceedings of Raising Awareness for the Societal and Environmental Role of Engineering and (Re)Training Engineers for Participatory Design (Engineering4Society); 2015. p. 96–100.
2. Köhlmann W, Lucke U. Alternative concepts for accessible virtual classrooms for blind users. Proceedings of 15th International Conference on Advanced Learning Technologies; 2015. p. 413–17.
3. Mikułowski D, Brzostek-Pawłowska J. Problems encountered in technical education of the blind, and related aids: Virtual cubarythms and 3D drawings. Proceedings of IEEE Global Engineering Education Conference (EDUCON); 2014. p. 995–8.
4. Bauwens B, Evenepoel F, Engelen JJ. Standardization as a prerequisite for accessibility of electronic text information for persons who cannot use printed material. IEEE Transactions on Rehabilitation Engineering,.1995 Mar; 5(1):84–9.
5. Dasgupta T, Basu A. Automatic transliteration of indian language text to braille for the visually challenged in India. Information Technology in Developing Counteries. International Federation for Information Processing Working Group 9.4 and Centre for Electronic Governance Indian Institute of Management. 2009 Oct; 19(3):1–10.
6. Wang C,Wang X, Qian Y, Lin S. Accurate Braille-Chinese translation towards efficient chinese input method for blind people. Proceedings of 5th International Conference on Pervasive Computing and Applications (ICPCA); 2010 Dec. p. 82–7.

7. Onur K. Braille-2. Automatic interpretation system. Proceedings of 23th Signal Processing and Communications Applications Conference (SIU); 2015 May. p.1562–5.
8. Nabiyev V, Akgül Ö, Braille T. Translation system. Proceedings of IEEE 18th Signal Processing and Communications Applications Conference (SIU); 2010 Apr. p. 856–9.
9. Hossain SA, Biswas LA, Hossain MI. Analysis of Bangla-2-Braille machine translator. Proceedings of 17th International Conference on Computer and Information Technology (ICCIT); 2014 Dec. p. 300–4.
10. Parvathi K, Das JK, Samal BM. Odia Braille: Text transcription via image processing. Proceedings of International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE); 2015 Feb. p. 138–43.
11. Shreekanth T, Udayashankara V. A two stage Braille character segmentation approach for embossed double sided Hindi Devanagari Braille documents. Proceedings of International Conference on Contemporary Computing and Informatics (IC3I); 2014 Nov. p. 533–8.
12. Tang J. Using ontology and RFID to develop a new Chinese Braille learning platform for blind students. Expert Systems with Applications. 2013 Jun; 40(8):2817–27.
13. Araki M, Shibahara K, Mizukami Y. Spoken dialogue system for learning Braille. Proceedings of 35th IEEE Annual Computer Software and Applications Conference; 2011 Jul. p.152–6.
14. Velázquez R, Preza E, Hernández H. Making eBooks accessible to blind Braille readers. Proceedings of IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE); 2008 Oct. p. 25–9.
15. Ohtsuka S, Tomizawa T, Hasegawa S, Sasaki N, Harakawa T. Introduction of a wireless body-Braille device and a self-learning system. Proceedings of IEEE 2nd Global Conference on Consumer Electronics (GCCE); 2013 Oct. p. 407–9.
16. Dharme VS, Karmore SP. Designing of English text to braille conversion system: A survey. Proceedings of 2nd International Conference on Innovations in Information Embedded and Communication Systems; 2015 Mar. p. 1–6.
17. Kulkarni A, Bhurchandi K. Low cost e-book reading device for blind people. Proceedings of International Conference on Computing Communication Control and Automation; 2015 Feb. p. 516–20.
18. Xiaoli H, Tao L, Bing H, Qiang C,Qiang X, Qiang H. Electronic reader for the blind based on MCU. Proceedings of International Conference on Electrical and Control Engineering, Wuhan; 2010 Jun. p. 888–90.
19. Technology Recource List [Internet]. [Cited 2015 Sep 25]. Available from: https://nfb.org/technology-resource-list.
20. What is Arduino? [Internet]. [Cited 2015 Sep 03]. Available from: https://www.arduino.cc/en/Guide/Introduction.
21. Baraha – Braille support [Internet]. [Cited 2015 Sep 30]. Available from: http://baraha.com/help/Baraha/braille.htm.
22. National Library Service for the Blind and Physically [Internet]. [Cited 2014 Jul 09]. Available from: https://www.loc.gov/nls/reference/guides/brailleliteracy.html.
23. The Universal declaration of human rights [Internet]. [Cited 2015 Oct 25]. Available from: http://www.un.org/en/universal-declaration-human-rights/index.html.
24. EAGLE Freeware [Internet]. [Cited 2015 Nov 04]. Available from: http://www.cadsoftusa.com/download-eagle/freeware/.