DRISHTI—A Gesture Controlled Text to Braille Converter

Vineeth Kartha, Dheeraj S. Nair, Sreekant S., Pranoy P. and Dr. P. Jayaprakash
Department of Electrical and Electronics Engineering
Government Engineering College Barton Hill
Thiruvananthapuram, India
Email: vineethkartha@ieee.org, dheerajsnair@ieee.org,sreekantsomasekahran@gmail.com,
pranoyprasannan@gmail.com, jpavithran2011@gmail.com

***Abstract*—The need for devices for aiding the physically challenged has been on demand. A text to Braille converter is intended to aid the blind to interact with computers at workplaces and homes. Though several such devices are available, the cost is a limiting factor. Here we introduce a low cost gesture controlled device that enables blind to read through digital files in PDF, ODT or DOC formats just like an ordinary Braille text.**

I. Introduction

Blind people face a great difficulty in receiving computer education due to the lack of low cost technological support. Facilities for the sightless have been organized in different places for providing reading facilities to the blind. These centers maintain archives of reading materials (literature, science, etc.) in the form of Braille coded texts. However, such centers suffer from a number of practical difficulties such as severely constrained economic resource. As the Braille pages are rather thick and contain fewer words, a normal printed book turns out to be a voluminous Braille text. Keeping in view the limited space the libraries can afford, archival storage of a considerable number of books turns out next to impossible. The Braille printing system being very expensive, in developing and underdeveloped countries there are few Braille presses producing a very limited output. These problems, coupled with the tremendous urge of the blind people to have educational support, form the basic motivating factor to develop a low cost Braille Reading System, which can at least partially alleviate some of these difficulties. Considerable work has been done to provide different types of reading systems for the blind. Many of them are available commercially. The system outputs vary from displaying the pattern of the English character onto a pin array to providing synthesized speech outputs corresponding to the text read. But the cost of such systems are prohibitive in a country like India. For example, the cost of such devices range in the order of three thousand dollars. A low cost system was developed for libraries in India [1] called MULTIREAD.

II. About Braille Script

Braille script was invented by Louis Braille (1809–1852). It is writing system which enables blind and partially sighted people to read and write through touch. It consists of patterns of raised dots arranged in cells of up to six dots in a 3×2 configuration. Fig. 1 shows a Braille cell and the Braille script for English alphabets. There are a number of different versions of Braille:

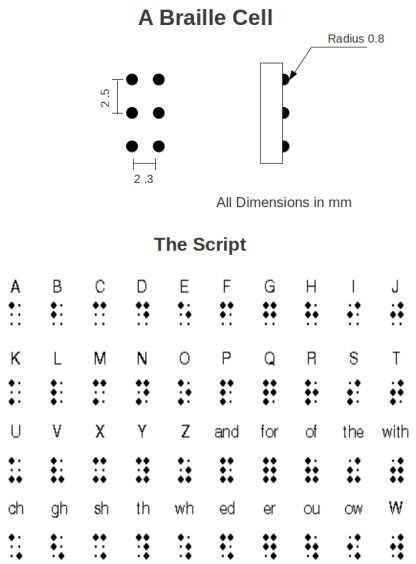

Fig. 1. Braille Script

- Grade 1 which consists of the 26 standard letters of the alphabet and punctuation. It is used by people who are first starting to read Braille.
- Grade 2 which consists of the 26 standard letters of the alphabet, punctuation and contractions. The contractions are employed to save space.
- Grade 3 which is used mainly in personal letters, diaries, and notes. It is a kind of shorthand, with entire words shortened to a few letters.

III. Block Diagram

Fig. 2 shows the basic block diagram of the device. The device consists of a microcontroller which is the heart of the

This work was funded by Center for Engineering Research and Development, Govt. of Kerala

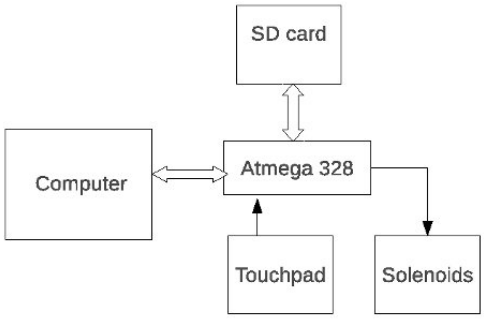

Fig. 2. Block Diagram

device. The conversion of characters take place in this device, at present we have started the work with Atmega 8 chip as it was available to us but the chip had to be changed to Atmega 328 due to memory constraints of Atmega 8. The code of Atmega 8 is compatible with Atmega 328, so switching the chip will not hinder the progress made so far. The tactile display is made of six solenoids that represent the Braille characters, the device will be having only a single Braille cell. Here we use the principle of a dot-matrix printer [2], [3]. The touchpad is interfaced to the device so that the user can navigate through the documents using gestures like forward stroke, backward stroke, up or down movements. The two click buttons of the touchpad should provide the necessary keypad functions like ENTER, or close. The scanner or webcam can be interfaced at a later stage, These devices have no direct connection to the device, they are connected to the PC and with the help of an Optical Character Recognizing software the textbooks or letters available as hard copy can be converted to digital format. The USB drives or SD card are directly interfaced to the device so that the device can act as a standalone reader without a computer and it can be carried around just like a book [4].

A. Atmega328

It is a high performance, low power AtmelAVR 8-Bit microcontroller. Its advanced RISC Architecture has 131 powerful instructions—most single clock cycle execution, 32×8 general purpose working registers, up to 20 MIPS throughput at 20MHz, 32KBytes of in-system self-programmable flash program memory, 1KBytes EEPROM, 2KBytes Internal SRAM. Available in 28 pin DIP.

B. Touchpad

A touchpad is a pointing device featuring a tactile sensor, a specialized surface that can translate the motion and position of a user's fingers to a relative position. Touchpads operate in one of several ways, including capacitive sensing and conductance sensing. The touchpads use an internal PS/2 connection.

IV. The Braille cell

Drishti contains a single Braille cell thereby reducing the cost of the device and the size, in the present project we have used solenoids to make the Braille cell, but use of modern piezoelectric Braille cells will reduce the size and power dissipation further. Here we used six solenoids placed in a 3×2 matrix. Since the size of solenoids where a bit big, we had to use a special arrangement on the solenoid plunger as in Fig. 5(b). Fig. 3(a) shows the view of the Braille cell.

A. Solenoids

It is basically a coil and a ferromagnetic substance (plunger), aligned in such a way that a magnetic force is developed on the plunger on the passage of an electric current through the coil. This force pushes a the plunger, creating a linear motion. For our project we use the custom made solenoids. The solenoids were tested to find out the electrical parameters and to check the suitability for the device.

The electrical characteristics of the solenoid are

- Voltage $6V$ DC
- Current $0.7A$
- Inductance $0.6mH$
- Resistance 40Ω

The physical characteristics are:

- Height 50mm
- length 20mm
- Breadth 15mm
- Force 1-2 N
- Push type

V. Circuit Diagram

The circuit is the basic interfacing of the six solenoids and the touchpad to the Arduino. We use a $16Mhz$ crystal oscillator as the the clock circuit. The touchpad pins are connected to the Atmega328 pins. An 8-bit shift register 74164 is used to interface the solenoids so that the number of output pins is increased. The processor and shift register can provide only around $40mA$ of current but the solenoid needs current in the range of $1 - 2A$. So an interfacing circuit is to be used. A transistor of nominal rating is used as a switch, here we used TP31A. A solenoid is a coil of wire with a magnetic core. This is virtually identical to a large inductor, so it should not be surprising that they have inductance, i.e., once a current is moving in the solenoid it will attempt to continue moving that current. This can be fatal to your digital device when it switches off the solenoid and the solenoid creates a voltage across its leads large enough to either move the current, arc through the air, or burn through a semiconductor. The solution is to provide an easy and safe path for the current to flow until it dissipates the energy stored in the magnetic fields of the solenoid. The schematic shows a bypass diode. When the solenoid is on or idle this diode will not conduct. Its top terminal will be at a higher or equal voltage. When the solenoid is turned off and tries to continue forcing current downward, this current can flow back up through diode

(a) Side view

(b) Solenoid arrangement

Fig. 3. The braille cell

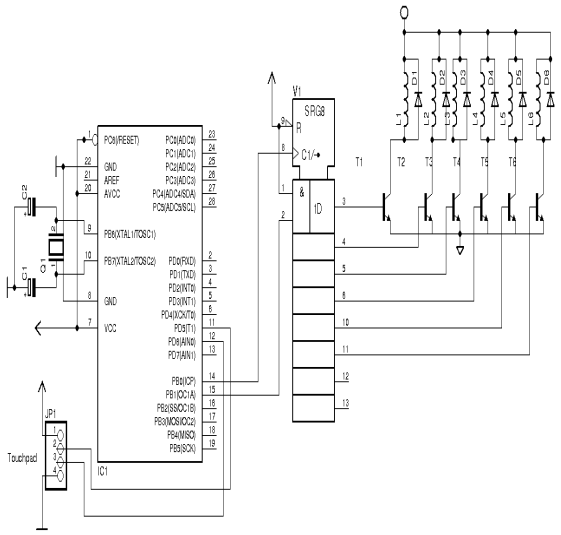

Fig. 4. Circut Diagram

until the energy is dissipated. This diode must be able to pass whatever current passes through the solenoid and have a reverse breakdown voltage of at least the power supply voltage. In certain cases an optoisolator is used to separate the processor from the solenoid circuit, in order to protect it.

VI. Software

The software has two components,

- The microcontroller code using the Arduino platform
- Software on the computer using processing platform, poppler utils and unoconv

A. Arduino Code

The Arduino IDE is a cross-platform application written in Java. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to run programs on a command-line interface. The Arduino IDE comes with a C/C++ library which makes many common input/output operations much easier. Arduino programs are written in C/C++.

The code for Drishti consists of the Braille library where each letter of English alphabet is stored as a hexadecimal number as shown in Table I and it is sent to a shift register, based on the input from the touchpad. With the touchpad the reader can make a forward movement or a backward movement to read the next character or previous character respectively. Also using the buttons of touchpad operations like opening or closing of files can be performed. The microcontroller also checks for any memory card available and if available looks for `.txt` files in them.

B. Processing Code

Processing is an open source programming language and integrated development environment (IDE) built for the electronic arts and visual design communities with the purpose of teaching the basics of computer programming in a visual context. The language builds on the Java programming language, but uses a simplified syntax and graphics programming model. The processing code of Drishti does the work of opening, converting and transmitting files to the microcontroller through the serial interface. It receives feedback from the touchpad interfaced to the microcontroller, and does actions like sending the next character, or previous character, opening or closing files, etc.

C. Poppler Utils

Poppler (or libpoppler) is a free software library used to render PDF documents [7], [8]. It is the most common such library on GNU/Linux systems, and also available for windows and is used by the PDF viewers of the open source GNOME and KDE desktop environments. Its development is supported by `www.freedesktop.org`.

The project was started by Kristian Hogsberg with two goals in mind:

TABLE I
HEXCODE FOR BRAILLE LIBRARY

SYMBOL.	S1	S2	S3	S4	S5	S6	HEXCODE
A	H	L	L	L	L	L	0x01
B	H	H	L	L	L	L	0x03
C	H	L	L	H	L	L	0x09
D	H	L	L	H	H	L	0x19
E	H	L	L	L	H	L	0x11
F	H	H	L	H	L	L	0x0B
G	H	H	L	H	H	L	0x1B
H	H	H	L	L	H	L	0x13
I	L	H	L	H	L	L	0x0A
J	L	H	L	H	H	L	0x1A
K	H	L	H	L	L	L	0x05
L	H	H	H	L	L	L	0x07
M	H	L	H	H	L	L	0x0D
N	H	L	H	H	H	L	0x1D
O	H	L	H	L	H	L	0x15
P	H	H	H	H	L	L	0x0F
Q	H	H	H	H	H	L	0x1F
R	H	H	H	L	H	L	0x17
S	L	H	H	H	L	L	0x0E
T	L	H	H	H	H	L	0x1E
U	H	L	H	L	L	H	0x25
V	H	H	H	L	L	H	0x27
W	L	H	L	H	H	H	0x3A
X	H	L	H	H	L	H	0x2D
Y	H	L	H	H	H	H	0x3D
Z	H	L	H	L	H	H	0x35

1) To provide PDF rendering functionality as a shared library, in order to centralize maintenance effort
2) To go beyond the goals of Xpdf, and integrate with functionality provided by modern operating systems

The poppler library represents a complete implementation of ISO 32000-1, the PDF format standard, and is the first major free PDF library to support its forms and annotations features.

D. Unoconv

Universal Network Objects (UNO) is the component model used in the OpenOffice.org and LibreOffice computer software application suites. It is interface-based and designed to offer interoperability between different programming languages, object models and machine architectures, on a single machine, within a LAN or over the Internet [6].

Users can implement or access UNO components from any programming language for which a language binding exists. Complete UNO language bindings exist for C++, Java, Python and Tcl. Bindings allowing access, but not writing, to components exist for StarOffice Basic, OLE Automation and the .NET Common Language Infrastructure.

VII. SYSTEM REQUIREMENTS

It works on linux and windows platform. In windows presently it detects only pdf and txt files whereas in linux it can detect pdf, doc,odt and txt.

A. Linux or Unix

- A Personal Computer with any Linux or UNIX OS
- Poppler Utils
- unoconv
- Java runtime environment.
- A RS232 port or drivers for USB-Rs232 adapters.

B. Windows

- A Personal Computer with Windows 98 or newer versions.
- Poppler Utils
- Java runtime environment.
- A RS232 port or drivers for USB-Rs232 adapters.

Also a Micro SD card with txt files can also be used.

All files to be read are to be stored in a folder *Drishti* within the Documents folder of the user. On running the program all files of formats mentioned above will be converted to txt format and it is these txt files that Drishti reads. The user just has to scroll using the touchpad and feel the braille cell.

If the user decides to read from the memory card just insert it in the slot and push the reset button. a green led is provided on top to check if the micro sd card has been detected. Fig VII-B and Fig 5 shows the top and side view of device Drishti.

VIII. A REAL WORLD TEST

We first checked the device to identify its maximum operating speed and this was done by providing a variable frequency supply to the solenoids which simulated the the navigation on the touchpad.The speed so obtained was 108 words per minute. The device was first tested by verifying that the patterns displayed corresponding to a character. Each character was sent from the computer terminal to the device through the RS232 port, and patterns were observed. A visually challenged person identified the characters. Then a text was saved in the computer and was read by the blind person. From the observations we concluded that the device was suitable for beginners in braille but it does affect the speed of experienced braille users. In the real application a speed of 90-100 words per minute could be obtained. And time is needed for the user to get familiarized with the device. A suggestion made by a person who tested this device was that by including upto 5 braille cells the speed problem can be solved. Also since the device we made was a prototype it had imperfections in the leveling of the braille cell. But overall the device was a success and we were able to demonstrate that a low cost electronic braille display can be made. With a few more modifications and by following precise manufacturing techniques an efficient and low cost text to braille display can be made.

IX. CONCLUSION

The project has achieved the set targets. Further enhancements are possible on the device. The use of piezoelectric Braille cell will further improve speed of the device and also reduce the power consumption. The present device is noisier

(a) Top view

(b) Side View

Fig. 5. Drishti

due to the solenoids using piezo-electric braille cells will reduce the noise too. Another development possible is to find suitable batteries that can be used to power up the device so that it can be used even without AC mains. Integrating USB drives in place of Micro SD cards is also a possible expansion on the hardware side. On the software side using optical character recognizers opens a new arena for the visually challenged to learn and read. The Braille displays are superior to text to speech synthesizers as it promotes Braille literacy. This project is intended to benefit the visually challenged in developing and under developed countries, where Braille presses are rare and unapproachable for the common man.

ACKNOWLEDGMENT

We would like to thank the members of the Electrical and Electronics department of Govt. Engineering College Barton Hill, for their guidance and support. We express our sincere gratitude to Center for Engineering Research and Development (CERD) for funding the program. We also thank the members of Society for Promoting Alternatives in Computing and Employment (SPACE), Vellayambalam; Insight, Vellayambalam and Kerala Federation of the Blind for their support and help.

REFERENCES

[1] A. Basu, P. Dutta, S. Roy, and S. Banerjee, "A PC-based Braille library system for the sightless," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 1, Mar. 1998.

[2] N. Sriskanthan and K. R. Subramanian, "Braille display terminals for personal computers," *IEEE Trans. Consum. Electron.*, vol. 36, no. 2, May 1990.

[3] C. P. Grossner, T. Radhakrishnan, and A. Schena, "An intelligent Braille display device," in *IEEE Micro*, vol. 6, no. 3, Jun. 1986, pp. 43–51, ISSN : 0272–1732, DOI: 10.1109/MM.1986.304676.

[4] R. Velazquez, H. Hernandez, and E. Preza, "A portable eBook reader for the blind," in *32nd Annual Int. Conf. of IEEE EMBS*, Buenos Aires, Argentina, Aug.31–Sep.4 2010.

[5] S. Supriya and A. Senthilkumar, "Electronic Braillepad," in *Int. Conf. on Contr., Autom., Commun. & Energy Conservation*, Jun.4–6 2009, pp. 1–5, ISBN: 978-1-4244-4789-3.

[6] D. Wieers, "unoconv - linux man page," http://dag.wieers.com/home-made/unoconv/.

[7] "Poppler (software)," http://en.wikipedia.org/wiki/Poppler_(software).

[8] "Poppler," http://poppler.freedesktop.org/.