Braillebook

Richard Lam, EE, Steven Golonka, EE, Raveena Kothare, EE

Abstract—**With the advent of text-to-voice and other audio technologies, Braille literacy within the blind community has declined in recent decades. In order to combat this decline and produce an alternative to expensive refreshable Braille displays that are currently available, we are designing and producing Braillebook – a low cost, dynamic Braille learning tool. The design implements pairs of 8-sided, plastic, rotating disks with faces that combine to display any 6-dot Braille character. The Braillebook can be produced at a fraction of the cost of its current market competition. The disks are driven by two stepper motors which are in turn driven by a single microcontroller. Preliminary designs proved the functionality of the system architecture; however, display refresh speed is limited by the sequential nature of the rotational positioning of the disks and is a significant drawback in comparison to current devices.**

I. Introduction

BRAILLE is a tactile writing system used by visually-impaired individuals. It is a code, comprised of raised dots that can be felt by the fingers of a reader, which is used to represent languages and symbols. Research demonstrates that Braille literacy is positively correlated with academic achievement and employment of blind individuals [1]. Specifically, of the twenty-six percent of blind people who are employed, nearly all can read Braille [2]. The low employment rates of blind individuals is a concern to the national economy as well. Since seventy-four percent of working-age blind people are unemployed and must depend on disability income benefits, the United States loses about eight billion dollars in productivity [1].

Despite the benefits of Braille literacy, today, only about ten percent of blind children are taught Braille. This is especially significant when compared to the 1950s, when about fifty percent of blind American children were taught Braille in school [3]. One of the largest factors in the decrease of Braille literacy rates is the growing prevalence of convenient technology in today's age. While technologies, such as text-to-voice translators and voice-recognition software, do certainly increase the ease with which blind people can live their daily lives, they are often improperly viewed as a replacement for Braille literacy. The National Braille Press cites research which shows that blind people who are Braille-literate possess a strong advantage in their ability to master grammar, language, math, and science [1].

R. Lam from Holden, MA (e-mail: rlam@umass.edu).
S. Golonka from North Hatfield, MA (e-mail: sgolonka@umass.edu).
R. Kothare from Belmont, MA (e-mail: rkothare@umass.edu).

Today, Braille texts printed on paper cost four to five times that of a normal book. And, electronic Braille readers cost two to eight thousand dollars each [4]. While these costs are clearly exorbitant, especially when compared to the audio technology alternatives, they are a particularly large hurdle for the blind community as its majority is unemployed. In the past decade, there have been many unsuccessful endeavors to push to market a low-cost Braille e-reader. Peizoelectric touchscreen vibration [5] and heat-driven expansion of paraffin waxes [4] are amongst the attempted approaches.

Our team believes that in order to increase Braille literacy rates, there must exist a way to learn Braille that is convenient and accessible to the blind community. For this reason, we produced a low-cost, refreshable Braille display geared as an educational tool for users seeking to learn Braille. We hope that our product, Braillebook, will contribute to a growth in productivity and quality of life of blind people.

Through research of the market, as well as the input of a blind individual, we determined a goal for the user experience and device requirements. We aim to create a device that is capable of taking a text file input, converting the characters to Braille code, and setting it on a tactile display. Also, the device produces a text-to-speech output of the words displayed on the device when prompted by the user. Ultimately, the user will be able to read a displayed line of Braille text and compare it to the audio output if desired. Most importantly, Braillebook is inexpensive and simple to use. Detailed device specifications are listed in Table 1.

Requirement	Goal	Accomplished
Display	20 Braille characters	20 Braille characters
Desktop Application	Compatible with screen reading software	Tested with Thunderbird [12]
Refreshable	5 second average refresh rate	10 second average refresh rate
Fully packaged	Closed box	Closed box with access hoods and magnets
Low Cost	< $400	< $200
Audio Feedback	Read displayed line when prompted	Integrated EMIC2 Speech Module
Intuitive User Interface	User-friendly buttons	4 arcade buttons with labels
Automatic calibration	No assistance required	IR sensor used in startup calibration

Table 1. List of User Requirements & System Specifications.

II. Design

A. Overview

To solve the problem of decreasing Braille literacy rates and corresponding decline of employed blind people, Braillebook will serve as an affordable and accessible Braille learning device. The device makes use of two stepper motors controlled by a microcontroller and a number of custom 3D printed mechanical components. The combination of these devices provides a device that is capable of displaying a refreshable line of Braille text. The technologies that we used in this device are significantly cheaper than those used in existing products which can cost in excess of $2000 [4]. Because of this reduction in price, this technology is more affordable for schools and households allowing blind children the opportunity to learn Braille. Figure 1 shows a system overview.

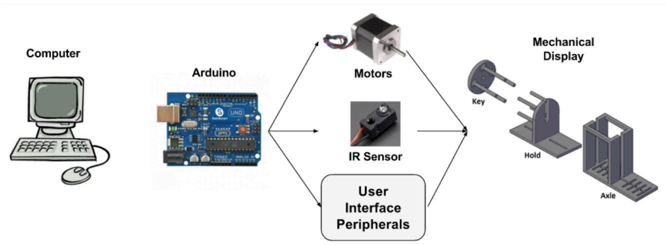

Fig. 1. System overview.

During the development of design ideas for Braillebook, a number of alternative technologies were considered including piezoelectric crystals and solenoids. These two design considerations were discarded due to inherent difficulties that could not be avoided. The cost of a design involving piezoelectric crystals would not be viable in terms of making Braillebook an affordable tool. Currently available Braille devices that use this technology are sleek in design, but also expensive. A design involving solenoids is not viable in terms of size. A set of 20 Braille characters takes up approximately eight inches. These 20 characters would require 120 individual solenoids. And, the solenoids that are able to fit are too small to generate enough force to execute the necessary mechanical function. Magnetic interference would also be a concern because of the close proximity of the solenoids. After taking these considerations into account, we decided that a design involving two motors would be best suited to make an inexpensive, refreshable Braille display. Figure 2 shows the system block diagram.

The PC User Interface is the mechanism for loading a text file and its associated Braille code onto the device. The PC User Interface is comprised of the GUI, the Text State Machine, and the COM port connection initialization. The GUI consists of a simple front-end application that allows the user to load one line at a time from the text file onto the device. Its compatibility with screen reading software, such as Thunderbird [12], has been verified. The Text State Machine is responsible for keeping track of which line is currently displayed on the device and the location within the larger text file. Lastly, the PC User Interface ensures proper connection with the device via serial port.

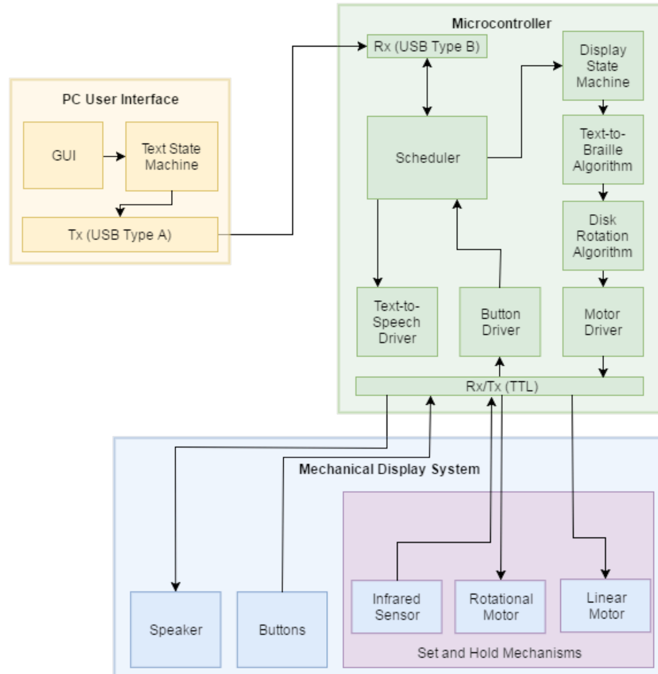

Fig. 2. System block diagram.

The microcontroller manages the user input of the Braillebook and controls the automation of the display and audio output. The scheduler ensures that each process meets its deadline. The finite state machine keeps track of the positions of all the disks in the display. The disk rotation algorithm handles the way in which the motors need to move in order to orient the disks correctly. The device requires a USB-to-micro USB connection with the computer to load the text file and instructions.

The mechanical display system is the physical interface with which the visually impaired person interacts. The mechanical display is comprised of four main components: Braille octagonal disks, a rotational key, a hold mechanism, and an axle. The Braille octagonal disks are placed adjacent to each other to form the display consisting of multiple Braille characters. The rotational key is used to set the disks in the correct orientation, while the hold mechanism maintains the disk orientation. The axle is used as a pivot point on which the disks rotate. A speaker and buttons allow the user to traverse the lines of the text file and to listen to the currently displayed text.

B. PC User Interface

The PC User Interface is required to load a text file onto the device for reading. The interface is compatible with screen-reading software, such as Thunderbird [12], Windows Eyes

[9], or JAWS [10]. The GUI was created using Microsoft Visual Studio [13]. Our team's programming experiences in Software Intensive Engineering and Computer System Lab courses helped in the completion of this task. Figure 3 provides a screenshot of the graphic user interface of the Windows application.

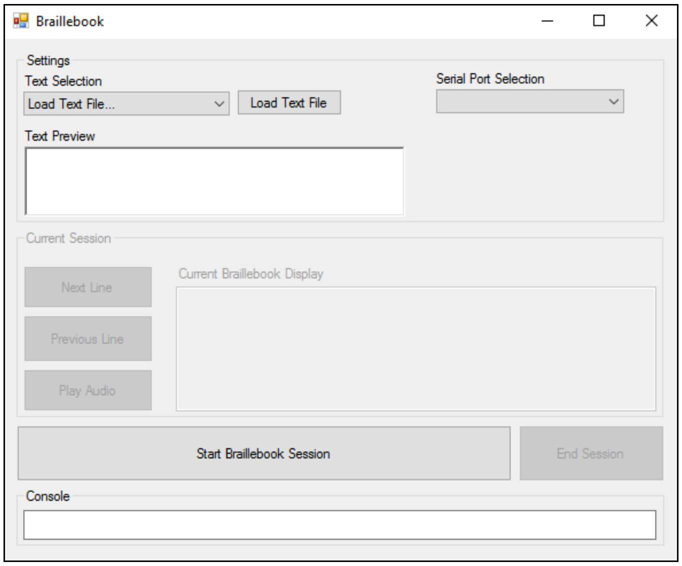

Fig. 3. Braillebook Windows application.

C. *Microcontroller Architecture*

The microcontroller block encompasses the software architecture and the handling of the peripherals. The primary concern of this block regards the processing of the button inputs and the output to the motors and speaker. To account for four buttons, two motors, and one speaker, we needed a minimum of seven I/O pins for our microcontroller. Additionally, the microcontroller is capable of producing a pulse-width modulated signal to control the motors (at least two PWM pins). Our system requires UART and SPI connections.

Two of the team members have worked with ST and mbed microcontrollers. While the former is good for real-time performance with peripherals, it is more expensive than the latter, which is good for prototyping, but lacks technical support. We sought the advice of Professor Holcomb in the selection of a microcontroller and design of the software architecture. Ultimately, we chose the Arduino Uno [8] due to the abundance of available resources and ease of integration with our chosen peripherals. In order to implement the audio functionality, we used the EMIC2 Text-to-Speech module [11]. Synchronization of all tasks is a high priority in the design of this technical block. Also, the finite state machine, which keeps track of the display status at any given point in time, resides in the microcontroller. System awareness of the state of the display is essential to successful display of lines as the device does not have any way of "sensing" its current state; it must change its state based on information of its previous and next states alone.

In order to test the functionality of our software system architecture, we first tested the basic interaction of inputs and outputs. For example, we checked that the motors and speakers respond to button presses as appropriate. Next, we tested the timing of the communication to the motor modules, which is critical to the performance of the overall product - the linear and rotational motion must be synchronized precisely. Finally, we tested the finite state machine by running several iterations of user commands and checking corner cases to verify state-awareness of the system.

D. *Motor Control and Algorithms*

Upon startup of the device, a calibration sequence is performed in order to bring the device to the correct position to begin operation. This calibration is done through the use of a break-beam infrared sensor which communicates with the microcontroller to alert the device when it is in the correct position.

The microprocessor is then given a 20 character text file to be displayed by the Braillebook. The microcontroller converts the characters into a number code using the text-to-Braille algorithm. These numbers – ranging from zero to seven – correspond to the eight sides of each octagonal disk on the display. The microprocessor iterates through these numbers at the command of the user. These changes to the state of the display are tracked by a finite state machine which passes the current state of the display and the desired state of the display to the Disk Rotation Algorithm. These inputs are used to generate a number of turns to be done on each disk such that the desired Braille text is shown.

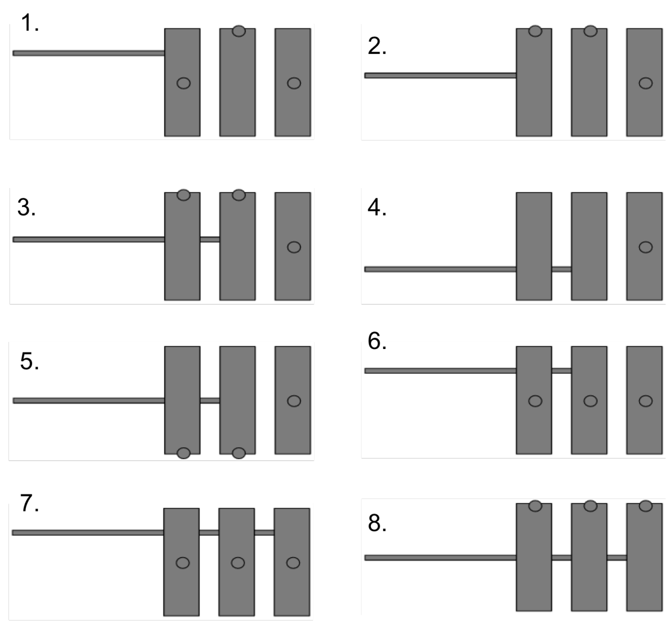

Fig. 4. Disk Rotation Algorithm visualization for 4-sided disks.

The code determines the most efficient number of spins for each disk, accounting for both forward and backward rotation of the bipolar stepper motor that controls the key mechanism.

Figure 4 demonstrates the steps generated by the algorithm for a simple case with three disks, each of which only has four sides.

These turn numbers are then handled by the motor driver. The motors controlling linear and rotational motion alternate in order to spin each disk the specified number of times. The number of steps performed by the stepper motors is also determined by the motor driver. The rotational motor turns a number of steps depending on the number of turns necessary for that disk. The linear motor turns a set number of steps for each disk except in the event that a disk does not need to be spun. In this case, the linear motor does not stop at the disk and continue to the following disk instead. The motor driver also adds necessary delays between motor actions to ensure that jams in the device do not occur due to overlapping motion of the two motors.

The Disk Rotation algorithm can be tested easily by manually inputting test cases and checking the turn numbers it outputs. By feeding these outputs to a simulation of the display, the functionality can be determined. Once the rotational motor is operational, the output of the algorithm can be fed to the motor to determine if the outputs being generated are correct. Because this subsystem's functionality is dependent on code, the techniques that we used in its development stem from an ability to develop and test efficient code.

The motor control was fully tested once the electromechanical display was assembled. However, by measuring the degree of rotation of the motor shaft, it was possible to ensure that the motors were behaving independently as expected before the display was complete. To achieve the required functionality of this subsystem, it was imperative to become familiar with the topic of motor control and implementing motor drivers with a microprocessor.

E. *Mechanical Display System*

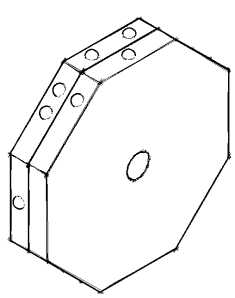

Fig. 5. Sketch of two octagonal disks displaying a Braille character.

In 6-dot Braille, characters consist of six raised dots, which represent a Standard English alphabetical character with three rows and two columns. Using two octagonal disks with a different combination of dots on each side, pairs of faces can represent single characters as shown in Figure 5.

The mechanical display is comprised of four main components: multiple Braille octagonal disks (Figure 6), a rotational key, a hold mechanism, and an axle (Figure 7). In order to refresh the Braille surface, the mechanical display requires a key which sequentially rotates each individual octagonal disk to the correct position while the hold mechanism maintains the orientation of the disks. Once an individual disk is in the correct orientation, linear motion from the second motor moves the key to the next disk for reorientation. This allows each disk to be set serially until all the disks are in the correct position for reading. The read line button allows the text to be read out loud through a speaker on the display.

The advances in 3D printing enabled quick prototyping for each of the components. Previous knowledge of CAD paired with experiences of physical interfaces from an Electronics II project greatly influenced the technical production of these parts. Moreover, CAD software such as AutoCAD and AutoDesk Inventor were excellent tools in modeling the components. The opening of the new Digital Media Center at the WEB Dubois Library and access to the Marcus 5 3D printer opened many opportunities for different designs for the mechanical display. However, extensive testing for each design was still required in order to perfect the seamless motion for the display.

The mechanics of rotational motion and linear motion required a disk design that is conducive to these mechanical factors. The final design of the Braille octagonal disk has nine holes, eight of which correspond to the faces of the disks, to allow a key and hold mechanism to be inserted from the side and actuate the motions needed to refresh the display. The hole in the middle of the disk is used as an opening for the axle around which the disks are spun. An example of the Braille octagonal disk is shown in Figure 6.

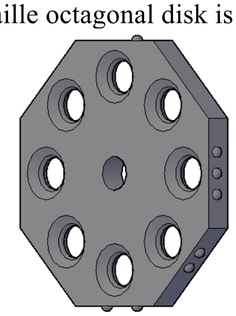

Fig. 6. Model of the Braille octagonal disk used in the Braillebook design.

A key was developed to be inserted from the side of the disk to rotate an individual octagonal disk. The hold mechanism moves in a linear motion along with the key at a constant distance of less than the width of a Braille octagon disk (.2 in). This allowed enough space between the key and hold to set an individual disk without the two mechanisms conflicting with each other. The axle was mounted such that it would not be in the way of the linearly moving key and hold. The placement of each these components can be seen in Figure 7.

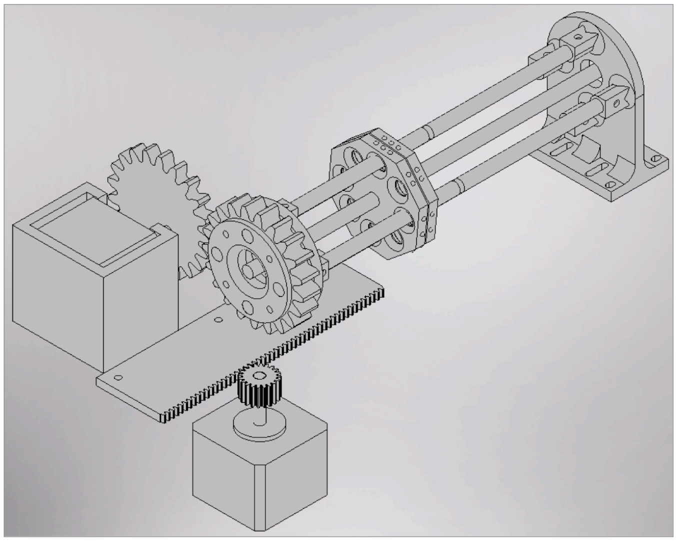

Fig. 7. Layout of the mechanical compents that make up the Braillebook display.

The key and hold mechanism are both mounted to a sliding bar such that they are always at an equal distance from one another. This allows the device's linear position to be controlled by a single motor. This linear positioning is handled by a stepper motor which produces motion through the use of a rack and pinion (refer to Figure 8). A second stepper motor, which supplies the rotational motion that sets the disks' positions, is housed on top of the rack and pinion. The key is driven by a gear that is attached to the axle of this motor.

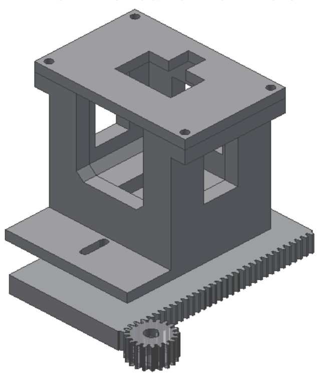

Fig. 8. Model of the rotational motor housing on top of the rack and pinion.

III. Results

Item	QTY	Per Piece	Per 1000
Arduino Uno	1	$5.65	$2.65
Arcade Button	3	$2.00	$0.27
EMIC2 Chip	1	$60.00	$20.00
Hinge	2	$2.50	$0.19
IR Sensor	1	$1.95	$1.36
Metal Rod	2	$2.00	$1.00
Motor Driver	1	$19.95	$13.96
Plastic Parts	1	$20.00	$10.00
Power Button	1	$5.93	$4.60
Power Supply	1	$9.99	$6.86
Slider	1	$7.37	$5.37
Speaker	1	$1.95	$1.36
Stepper Motor	2	$14.00	$5.00
Wooden Case	1	$20.00	$10.00
Total		$195.79	$89.35

Table 2. Bill of materials.

As affordability was one of our largest goals with this device, we are very content with our final cost of materials. We are positively encouraged by our numbers, itemized in Table 2, that this device could be manufactured on a larger scale and sold as a final product for less than a tenth of the cost of the market competition.

Linear Motion

- 100 steps per inch
- 20 steps per disk
- 800 steps to traverse entire display
- ~0.1 seconds per 20 steps

Rotational Motion

- 1.8 degrees per step
- 25 steps per turn
- 4 turns maximum required
- ~0.125 seconds per 25 steps

Line Refresh Statistics

- Best case refresh time: 4 seconds
- Worst case refresh time: 24 seconds
- Average refresh time: ~10 seconds

Fig. 9. Performance measurements.

Finally, we tried our best to characterize the performance of our device. Given our estimations, as shown in Figure 9, we are content with the functionality.

IV. Project Management

Our team consists of three electrical engineering majors- Rich Lam, Steve Golonka, and Raveena Kothare. The three of us met weekly to review our progress and determine next tasks. Additionally, we met once a week with our faculty advisor, Professor Dennis Goeckel, to get his input on our approaches and progress. As primarily a group of friends, the three of us benefit from a comfortable and enjoyable work environment. Furthermore, we were intent on developing good communication skills within the group, as well as our professionalism in correspondence with the faculty advisors; we are content with the extent to which we met these goals.

For the initial stages of prototyping, we mostly worked together on all subsystems. However, after we verified functionality of the initial prototype, we determined a split of responsibilities that accounted for our technical strengths and preferences. Rich, the team manager, is the most skilled and interested in 3D prototyping, so he was responsible for the mechanical display system. Steve enjoys developing logic, so he was responsible for the Disk Rotation and Text-to-Braille algorithms, in addition to the motor control. Raveena has experience with embedded-C systems, so she worked on the architecture of the microcontroller and the integration of peripherals; additionally, she developed the PC User Interface. Once our subsystems were mostly developed, we worked together to integrate and tune them to meet our final product specifications.

V. Conclusion

Currently, the Braillebook prototype is a functional device capable of converting text files to readable 6-dot Braille. However, before introducing Braillebook to the marketplace, further optimizations would be required. These include: additional layers of feedback to track the devices position at every move, Bluetooth connectivity to allow the user to use a mobile app instead of needing a PC, increased library for use with the Text-to-Braille Algorithm, and further increasing the precision and efficiency of the mechanical display. Getting to this point required a large number of design considerations to be tested and confirmed as reasonable. Once the mechanical design was decided upon and the components were acquired, the logic and algorithms that drive the motions of the motors were developed. After the testing of each subsystem, the prototype was assembled. This design is capable of converting 20 ASCII characters transmitted from the PC User Interface into the equivalent Braille characters on the display.

Additional functionalities of the device include: audio interpretation of the display, user button control, and compatibility with screen reading software. The Disk Rotation Algorithm is optimized to ensure the minimum number of rotations per refresh, and the linear motor control has been optimized in order to avoid unnecessary delays during each refresh.

In order to achieve the desired results, we followed a Gantt chart detailing each of our responsibilities for the design of the device. To work through inevitable complications during the development and integration of our numerous subsystems, we performed extensive testing and fine-tuning.

Acknowledgment

We are very grateful for the contributions of the following people: Professor Ganz, for putting us in contact with people from the visually-impaired community and for providing us with critical considerations of product usability; Professor Frasier, for giving us feedback about motor control; Camille, for sharing with us her experiences as a visually-disabled person; Connor Pope, for assisting us with 3D printing; James Golonka, for providing us with construction advice and equipment; and, Professor Goeckel, for supporting and guiding us through the entire process. Thank you, everyone!

References

[1] "The Need for Braille?," *NBP*. [Online]. Available at: https://www.nbp.org/ic/nbp/Braille/needforBraille.html. [Accessed: Nov-2015].
[2] "Reading lessons," *The Economist*, 2015. [Online]. Available at: http://www.economist.com/news/science-and-technology/21654564-tablet-computers-blind-people-are-development-reading-lessons. [Accessed: Oct-2015].
[3] "Joining the dots," *The Economist*, Jan-2014. [Online]. Available at: http://www.economist.com/news/international/21595475-new-technology-could-save-writing-system-blind-joining-dots. [Accessed: Oct-2015].
[4] J. Cassidy, "The missing budget Braille e-reader," *BBC News*, Sep-2014. [Online]. Available at: http://www.bbc.com/news/technology-27243376. [Accessed: Oct-2015].
[5] J. Flatley, "Researchers develop Braille for vibrating touchscreen devices," *Engadget*, 2009. [Online]. Available at: http://www.engadget.com/2009/03/31/researchers-develop-Braille-for-vibrating-touchscreen/. [Accessed: Dec-2015].
[6] Adafruit, Stepper Motor Data Sheet, 2016. [Online]. Available at: https://www.adafruit.com/product/324.
[7] Adafruit, Motor/Stepper/Servo Shield for Arduino, v2, 2016. [Online]. Available at: https://www.adafruit.com/products/1438.
[8] Arduino, Arduino Uno Documentation, 2016. [Online]. Available at: https://www.arduino.cc/en/Main/ArduinoBoardUno.
[9] Window Eyes. [Online]. Available at: http://www.gwmicro.com/window-eyes/
[10] JAWS Screen Reader. [Online]. Available at: http://www.freedomscientific.com/Products/Blindness/JAWS
[11] EMIC2 Text-to-Speech Module. [Online]. Available at: https://www.parallax.com/product/30016
[12] https://www.mozilla.org/en-US/thunderbird/. [Accessed: April-2016].
[13] https://www.visualstudio.com/. [Accessed: April-2016].