braillebook

Steven Golonka, EE, Raveena Kothare, EE, Richard Lam, 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 controlled 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 replacements 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].

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. Piezoelectric 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 will produce 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 will be able to produce 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 will be inexpensive and simple to use. Detailed device specifications are listed in Table 1.

Requirement	Specification
PC connection to load text files to device	USB 2.0
Power source	Standard wall outlet
Reasonable size	50x15x12 cm^3
Intuitive device control	Buttons for ON/OFF, Menu, Yes, No, Read Line, Next Line, Previous Line
Device text storage	≈ 250 pages of text
Displays 1 line (20 ASCII characters) of text	20 Braille cells
Quick line refresh rate	≤ 5 seconds
Low-cost	≤ $400

Table 1. List of User Requirements & System Specifications.

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).

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 will make use of two stepper motors controlled by a microcontroller and a number of custom 3D printed mechanical components. The combination of these devices will provide a device that is capable of displaying a refreshable line of Braille text. The technologies that will be 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 will be more affordable for schools and households allowing blind individuals the opportunity to learn Braille.

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 will take 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 1 shows the system block diagram.

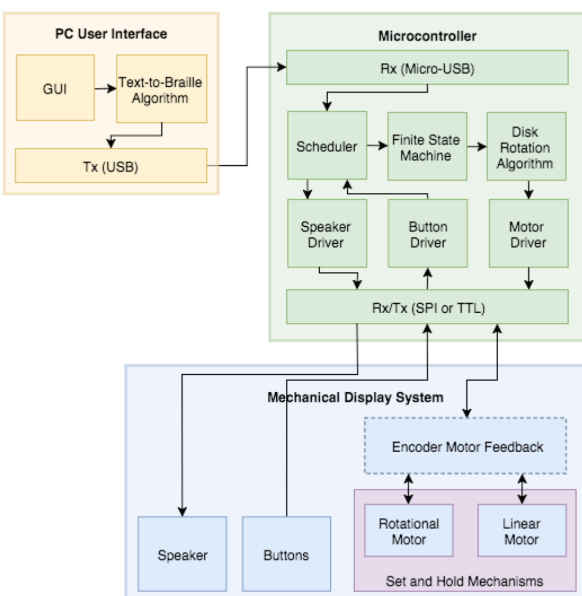

Fig. 1. System block diagram.

The PC User Interface is the mechanism for loading a text file and its associated Braille code onto the device. The PC User interface has three main components: the GUI, text-to-Braille algorithm, and binary compilation for the microcontroller. The GUI will consist of a simple front-end application that will allow a person helping the visually-impaired user to load the text file onto the device. The text-to-Braille algorithm will reside in the back-end of the desktop application, thus reducing the amount of computation required by the microcontroller.

The microcontroller manages the user input of the braillebook and controls the automation of the display and audio output. The scheduler will ensure that each process meets its deadline. The finite state machine will keep track of the positions of all the disks in the display, as well as the current place in the text file. The disk rotation algorithm will handle the way in which the motors need to move in order to orient the disks correctly. The device requires a physical connection to the computer to load the text file and instructions. We plan on having a USB-to-micro-USB connection from the computer to the microcontroller.

The mechanical display system is the physical interface with which the visually-impaired person will interact. 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. We will incorporate a speaker and buttons to allow the user to traverse through the menu, read the next or previous line, and hear 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 must be simple enough for any person to use when assisting the visually-impaired person. The GUI will require a browse file button to choose a text file on a computer. The USB debugger and binary loader will be used to implement a connection between the computer and microcontroller. We will create the GUI using Microsoft Visual Basic. We will consider setting up the device for forward compatibility with a screen-reading software, such as Windows Eyes [9] or JAWS [10]. The back-end of the PC User Interface will also handle the conversion of ASCII characters into a code that will be used to set the electromechanical display of the braillebook. Our team's programming experiences in Software Intensive Engineering and Computer System Lab courses will help in the completion of this task. We will test the functionality of this block by first checking the contents of the Braille code file after the text file goes through the text-to-Braille algorithm. Ultimately, we will check that the files are properly loaded onto the device by testing the inputs and outputs, including corner cases that we will determine in the next phase of development.

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 seven buttons, two motors, and one speaker, we need a minimum of ten I/O pins for our microcontroller. Additionally, the microcontroller must be capable of producing a pulse-width modulated signal to control the motors (at least two PWM pins). Our system will require 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 will seek the advice of Professor Holcomb in the selection of a microcontroller and design of the software architecture. We will need to learn about event listeners for a software system, scheduling of tasks, and microcontroller pins logic and power levels. In order to implement the audio functionality, we will use a peripheral chip, such as 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 will reside in the microcontroller. System awareness of the state of the display is essential to successful display of lines as the device will not have any way of "sensing" its current state; it must change its state based on information of its previous state alone. We plan on incorporating a calibration feature that will allow the user to reset the display in the event that the system does not have an accurate record of its state.

In order to test the functionality of our software system architecture, we will first test basic interaction of inputs and outputs. For example, we will check that the motors and speakers respond to button presses as appropriate. Next, we will test 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 will test 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

The microprocessor is given a number-coded version of a text file to be displayed by the braillebook. These numbers – ranging from zero to seven – correspond to the eight sides of each octagonal disk on the display. The microprocessor will iterate 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 will be used to generate a number of turns to be done on each disk such that the desired Braille text is shown. The code will determine 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.

These turn numbers will then be handled by the motor driver. The motors controlling linear and rotational motion will 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 will turn a number of steps depending on the number of turns necessary for that disk. The linear motor will turn 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 will not stop at the disk and continue to the following disk instead. The motor driver will also add 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 will be used in its development stem from an ability to develop and test efficient code.

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

E. Mechanical Display System

In 6-dot Braille, characters consists 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 2.

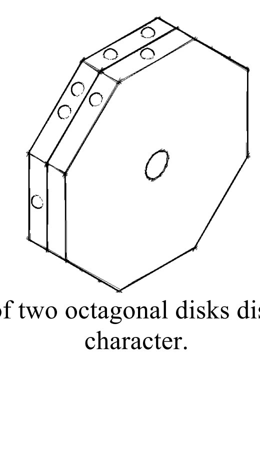

Fig. 2. Sketch of two octagonal disks displaying a Braille character.

The mechanical display is currently comprised of four main components: multiple Braille octagonal disks (Figure 3), a rotational key, a hold mechanism, and an axle (Figure 4). 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. Several buttons will be on the device including: on/off, menu, yes, no, previous line, next line, and read line. The read line button will allow 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 would be conducive to these mechanical factors. In the current version of the Braille octagonal disk, we have 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 3.

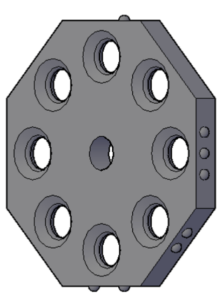

Fig. 3. Model of the Braille octagonal disk used in the braillebook design.

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 4.

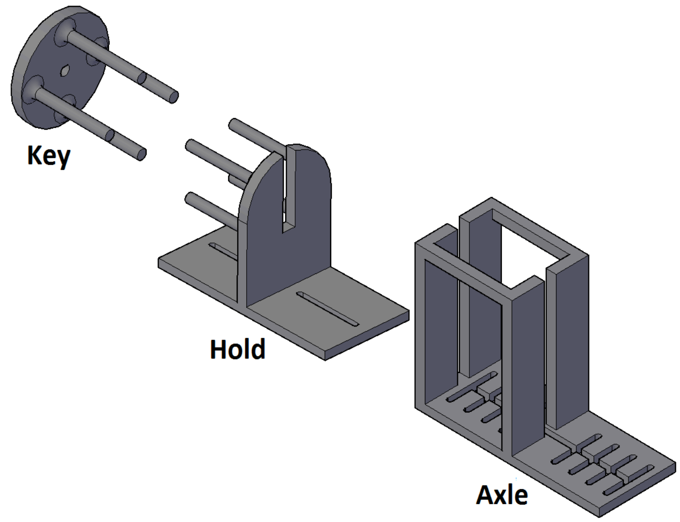

Fig. 4. Layout of the mechanical compents that make up the braillebook display.

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 devices' 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 5). 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 attached to the axle of this motor.

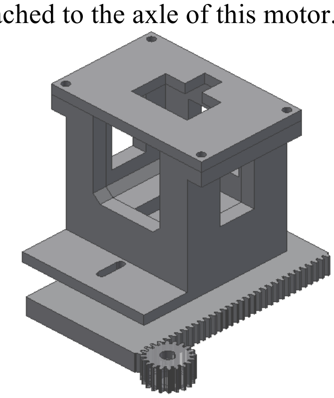

Fig. 5. Model of the rotational motor housing on top of the rack and pinion.

III. Project Management

Deliverables	Achieved
Open-box prototype of axle and key design; not scaled to specifications	Yes
Set 1 Braille cell	Yes, sets 3 cells
Control motors with microcontroller; device not responsive in real time	Yes, in real time

Table 2. List of MDR Deliverables

Our MDR goals and the status of each are listed in Table 2. Our team accomplished all MDR goals, and exceeded the latter two. Our primary goal was to decide upon and prove a mechanical design for the tactile display system. For the demo, our prototype was able to take in and display the user's input of three characters; the user was able to set the display multiple times in a row, demonstrating the validity of the Disk Rotation algorithm and the tracking of disk positions.

Our team consists of three electrical engineering majors- Rich Lam, Steve Golonka, and Raveena Kothare. The three of us meet weekly to review our progress and determine next tasks. Additionally, we meet 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 are intent on developing good communication skills within the group, as well as our professionalism in correspondence with the faculty advisors.

In the development of the MDR prototype, our team mostly worked together. Moving forward, we have determined a split of responsibilities that accounts for our technical strengths and preferences. Rich, the team manager, is the most skilled and interested in 3D prototyping, so he will be responsible for the mechanical display system; also, Rich will work on the PC user interface. Steve enjoys developing logic, so he will be 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 will work on the architecture of the microcontroller and the integration of peripherals. Once our subsystems have been mostly developed, we will work together to integrate and tune them to meet our final product specifications. The Gantt chart in Figure 6 will guide us in our progress to reach our goals for the final product.

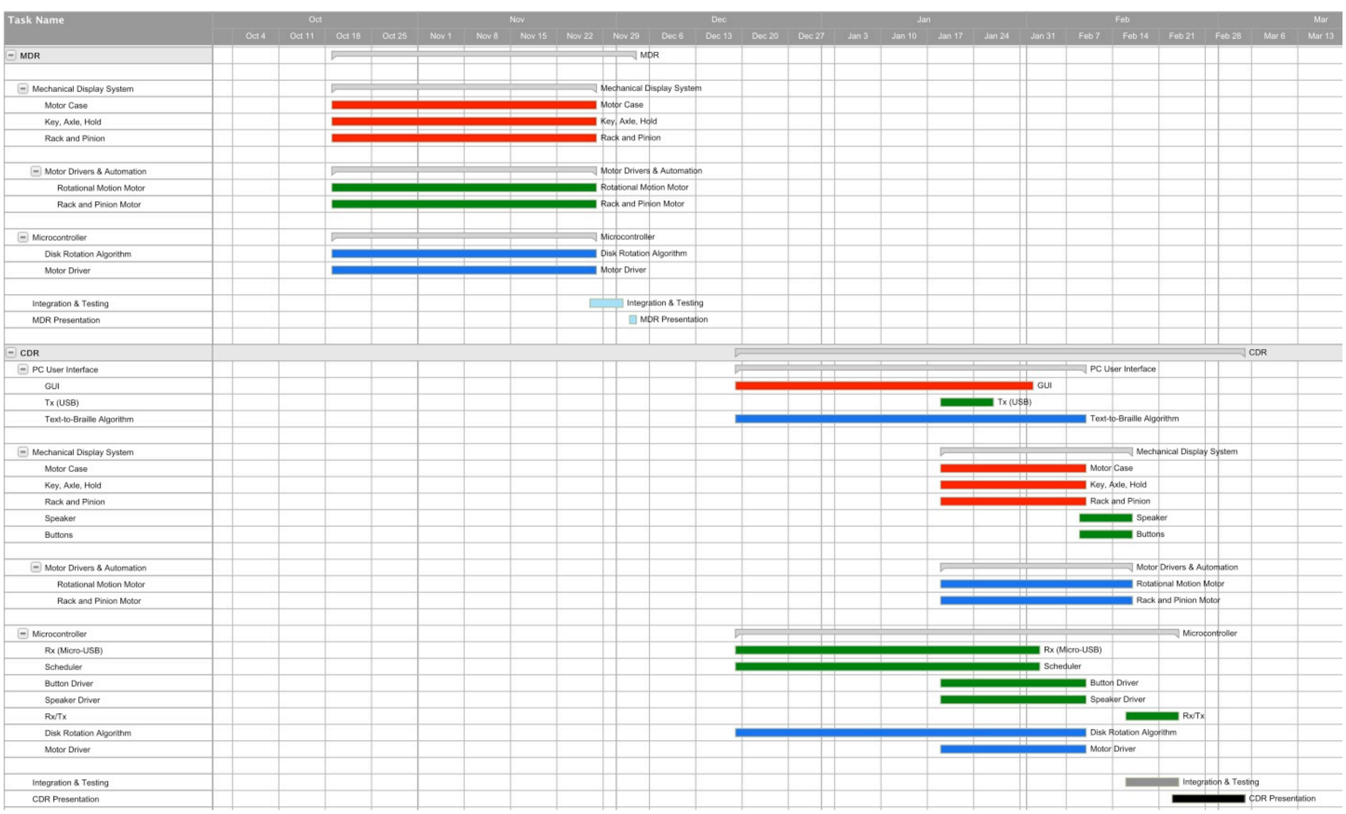

Fig. 6. Gantt Chart.

IV. Conclusion

Currently the braillebook prototype is far from the final product. 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 initial iteration of the design was capable of converting three ASCII characters entered via keyboard to the equivalent Braille characters on the display.

From here, additional functionalities will be added to the device. This includes: audio interpretation of the display, user button control, text storage, and linear motion via linear rail. In addition, the display will be extended from three characters to twenty and will be built to the design size specifications stated, the Disk Rotation Algorithm will be optimized to ensure the minimum number of rotations per refresh, and the speed at which the device sets characters will be improved.

In order to achieve the desired results we will follow the Gantt chart laid out in the project management section of this paper. In order to work through inevitable complications during the development and integration of our numerous subsystems, we will perform extensive testing and fine-tuning.

Acknowledgment

We would like to thank the following people for contributing to our progress: Professor Ganz, for putting us in contact with people from the visually-impaired community and for providing us with critical considerations for our product in evaluations; Professor Frasier, for giving us feedback, especially with motor control, in evaluations; Professor Holcomb, for helping us evaluate our choices for a microcontroller; Camille, for sharing with us her experiences as a blind person; Connor Pope, Jose LaSalle, and M5, for assisting us with 3D printing; and, Professor Goeckel, for supporting and guiding us through the entire process.

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