Digital Braille Watch

Nick Anderson – Team Leader
Chandresh Singh – BWIG
Taylor Milne – Communicator
Kyle Jamar – BSAC

December 7, 2010

Client: Holly and Colton Albrecht
Advisor: John Puccinelli

University of Wisconsin – Madison : Biomedical Engineering

Table of Contents

Abstract

The Braille language has provided the visually impaired with a way to read and write for many years, yet no device exists that allows the visually impaired to read the time in Braille. In order to tell time, the visually impaired currently rely on either talking or tactile watches. However, talking watches are disruptive, while tactile watches are difficult to read and fragile. Since the current methods are inadequate, a Digital Braille Watch was designed. The final prototype uses four rotating disks, each set beneath four pins. The top of each disk has both raised and recessed sections. If a pin is on the raised section, it will be pushed slightly above the watch surface, and if a pin rests on the recessed portion, it will remain flush with the surface of the watch. By rotating a total of 165 degrees, these disks can raise and lower the pins to form all ten numerals in the Braille number system. The final prototype included an Arduino Duemilanove USB board, four servo motors, four disks, sixteen pins and springs, and various circuitry components. The design was contained in an acroylonitrile-butadien-styrene (ABS) case. Initial testing was completed, confirming the benefits of our Digital Braille Watch Design. Future work will involve machining the disks and pins out of a stronger material, as well as integrating gears into the design to lower the range of motion required from the servos. An alternative power supply that is smaller should be considered, and visually impaired individuals should test the device in order to ensure its effectiveness.

Background

Problem Statement

Currently, people with a visual impairment must rely on one of two methods in order to tell the time independently. However, talking watches disrupt the surrounding environment and draw attention to its user, while tactile watches are fragile and challenging to read. Our goal is to design a digital Braille watch capable of displaying the time using the standard Braille numbering system. This watch should display military time, be accurate and reliable, silent, and easy to read.

Braille Basics

The Braille language is the universally accepted form of written communication for the visually impaired. It utilizes a system of dots arrayed in a three row by two column grid. Raised dots are then located in any of the six positions, displaying different letters, numbers, and symbols based on their configuration.

In order for this method of communication to be accurate and precise, universal specifications have been developed. Each dot must have a base diameter of 1.44 millimeters (0.057 in.) while being 0.48 millimeters (0.019 in.) in height. Within each individual grid, the dots must be at least 2.34 millimeters (0.092 in.) apart, measured center-to-center, and each individual character should be a minimum of 6.22 millimeters (0.245 in.) away from the neighboring character. [1]

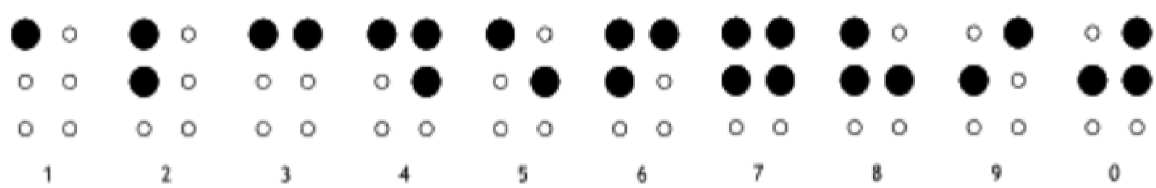


Figure 1 – Braille numerals 0-9
Image courtesy of PharmaBraille:
http://www.pharmabraille.co.uk/braille-alphabet.html

To simplify matters for this design, the numbers 0-9 are represented using only the top four dots of the three by two grid (Figure 1). Instead of having to manage an oblong, rectangular three by two grid, the design of the watch is much simpler and requires only a two by two grid to display any number.

The distance between two Braille pins must be at least 2.34 millimeters apart, since this is the minimal distance required to distinguish between two points with the fingertip. The sense of touch is triggered by mechanoreceptors that are stimulated once deformation of the skin occurs, due to contact, and causes these receptors to send information to the central nervous system. In order to discriminate between two points, a receptor must remain unstimulated between the two stimulated receptors. Without the presence of an unstimulated receptor, the brain would perceive the contact of the two points as one stimulus. These receptors are distributed all over the body; however, they exist in some areas in higher concentration. This distribution of these touch receptors are represented by a homunculus diagram, which shows that these receptors are present in a large amount in the fingertips, making them more sensitive to touch. It was important to consider these sensory limitations when designing the watch. [2]

Current Methods

There are two main categories of watch products currently on the market for the visually impaired: talking watches and analog tactile watches. Talking watches function by verbally relaying the time to the user whenever the user presses a button (Figure 2). This method is very effective in communicating the time; however, it can be very disruptive and draws attention to the user. Analog tactile watches on the other hand are silent (Figure 3). They function much like traditional analog watches, except in this case in order to tell the time the user must touch the face of the watch to feel where the hands are located. There are also raised markings on the watch that indicate the positions of the numbers; however, there is no standard format for these markings and they vary from product to product. Our client has informed us that these watches can be difficult to read and come with a learning curve when first used. Also, the hands of these watches are exposed while the user is telling the time, and therefore they can be easily broken or damaged.

In addition to these currently available watch products, there is a watch that has recently been designed called the Haptica Braille Watch (Figure 4). This design features a set of 16 rotating disks that circulate Braille dots in and out of the display to assemble the desired Braille numerals. Each disk contains a single Braille dot that is moved in concert with the other disks to display the time in Braille. This concept was created by David Chavez in 2008. Chavez is not an engineer and has not created a prototype for his design. [3]

Design Criteria and Considerations

Design Specifications

Our clients for this project are Holly and Colton Albrecht. Colton is Holly's visually impaired son. Together, they came up with the idea for the Digital Braille Watch. As such, the project will be created in accordance to their wishes and specifications. Their main requirements are that the design is able to correctly display the current time in standard Braille, utilize military time, and operate without any noise. The watch must not be dangerous to the user, so moving parts and electronic components must be contained properly. It has to be accurate within the minute whenever it is connected to a power source. Holly doesn't require that our prototype is any particular size; rather, she is looking for a proof of concept. However, the prototype should be designed so that it would be possible to scale down to watch-size in the future. For more information on the product design specifications, see Appendix A.

Figure 2 – Talking watches verbally communicate the time
Image courtesy of Independent Living Aids, LLC:
http://www.independentliving.com/prodinfo.asp?number=756480

Figure 3 – Visually impaired touch the hands of the tactile analog watch to tell the time
Image courtesy of Auguste Reyond:
http://watchluxus.com/braille-watches-by-auguste-reyond

Figure 4 – Haptica Braille Watch design by David Chavez
Image courtesy of Tuvie Design of the Future:
http://www.tuvie.com/haptica-braille-watch-concept/

Funding

Since caring for a visually impaired child can be financially taxing, it is difficult for our client to provide funding for this project. As a result, we turned to outside sources to try to offset the financial burden on our client. Based on advice from our advisor, John Puccinelli, we wrote a budget proposal (Appendix B) to the University of Wisconsin Madison Department of Biomedical Engineering (BME) to request funding. Fortunately, the Department granted our request, allowing us to not use our clients' money.

Actuating Dots

An existing product frequently used by the visually impaired is the HumanWare's BrailleNote system (Figure 5). This device has an 18 character refreshable display and is used as a link for the visually impaired to the digital world. To form the display each Braille dot is a pin that actuates up or down through a flat surface (Figure 6). This gives a familiar Braille configuration similar to feeling bumps on a sheet of paper. [4]

Figure 5 – HumanWare's Braille Note is used by the visually impaired to read computer screens
Image courtesy of HumanWare
http://www.anu.edu.au/disabilities/atproject/

After searching for possible mechanisms to drive the pins up and down, it was found that many solutions have been proposed. Some of these included attaching the pins to a spring mechanism or an electromagnetic solenoid. These mechanisms are highly complicated and mechanically intricate. For the prototype, 16 actuating mechanisms would be required to display the four numerals, making for a complicated product. This complexity would make it exceedingly difficult to both create a prototype and to eventually be able to scale it down to wristwatch size. Also, since the pins would be driven up and down many times, this mechanism would require a large power source. Due to these reasons, this design was not included in the design matrix.

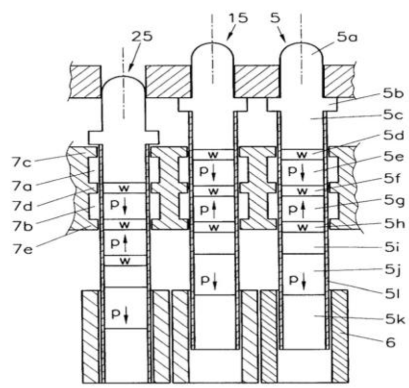


Figure 6 – Actuating dots mechanism
Image courtesy of Litschel, Dietmar, and Schwertner

Rotating Disks

This design is based off of the prototype last spring's Braille Watch design group created. Their design uses eight rotating disks to form the required Braille numerals. Each disk has four raised dots, which can be configured to form the top or bottom half of the character cell (Figure 7). [5]

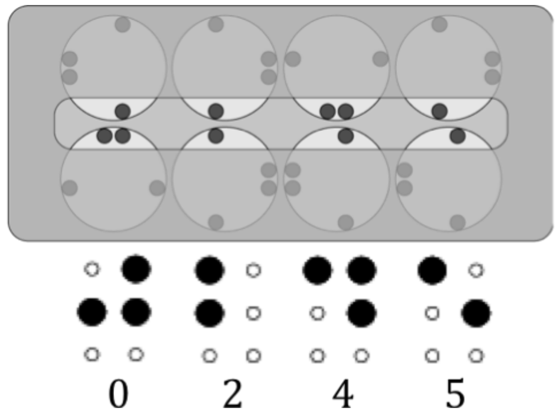


Figure 7 – By a rotation of 90º, 180º or 270º, the disks can display the correct time
Image Courtesy of Spring 2010 Braille Watch Team

This design met the client's requirements and is the first existing Braille time-keeping device. However, many downfalls exist with this design. As can be seen in Figure 8, the prototype is much too large to fit on the wrist of a user, thus, the size must be cut down significantly. Also, this device uses eight moving parts, leading to high power consumption. Although the eight rotating disks is an improvement over the sixteen solenoids of the previous design, ideally the number of moving parts could be cut further. With this design, there is not much opportunity to significantly cut size and power use. Finally, due to the mechanical nature of this design, the Braille numbers were often difficult to read. The Braille display relied heavily on the accuracy of the servo motors used to rotate the disks.

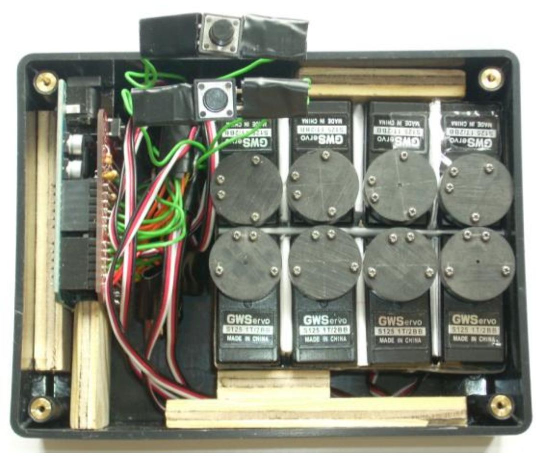


Figure 8- Internal view of rotating disk prototype
Image Courtesy of Spring 2010 Braille Watch Team

Sliding Plates

The Sliding Plates design is rather unique from previous design concepts. It consists of eight plates that lay paired up along the watch, with each pair creating a single Braille digit. Every plate can slide up or down within the face of the watch, revealing one, two or no dots as necessary (Figure 9).

An advantage to this design is its potential to be small. The plates would be thin, which corresponds with making a light and ergonomically friendly watch. Additionally this model is less likely to have alignment errors that could cause user confusion. Each Braille dot would be spaced at a standard distance relative to the others. Nonetheless, there are drawbacks. This design contains complications with regards to the mechanism driving it. While possible, sliding the plates back and forth to specified positions becomes intricate, making it challenging to create. Also the Sliding Plates design still has eight moving parts, a problem shared with last semesters design. The numerous moving parts puts this design's power requirements well above that of either of the existing methods.

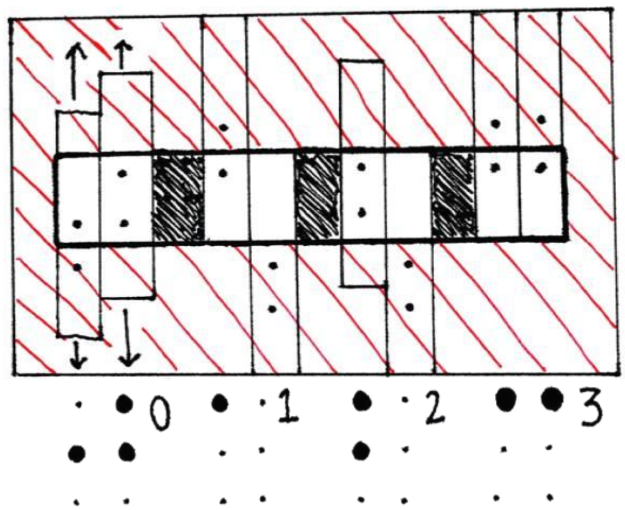


Figure 9 – Plates slide up and down to display the correct time

Disk and Pins

The Disk and Pins design, much like its name suggests, consists of disk and pin sets. Four disks are located beneath the watch surface, one for each Braille numeral. Above each disk four pins are positioned so that they rest on the disks surface. The portion of the disk against which the pins rest has both raised and recessed sections (Figure 10). If a pin is on the raised surface, it will be pushed slightly above the watch plane, and if not, the top of the pin will remain

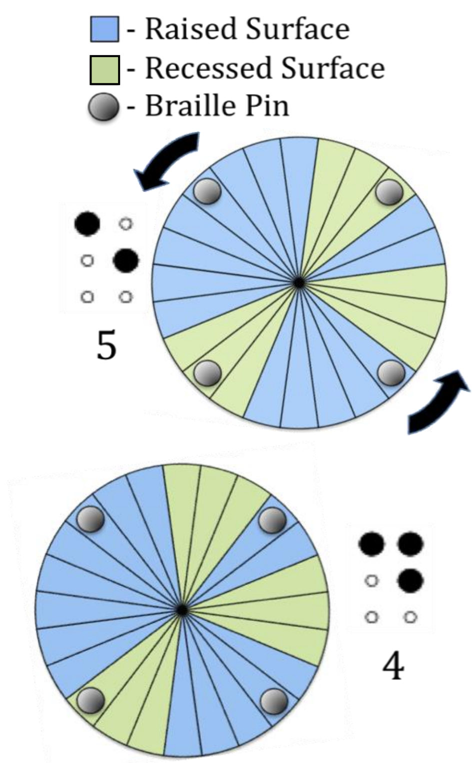


Figure 10– The raised and recessed surfaces on the disk cause different numbers to be displayed

flush with the surface of the watch (Figure 11). When the disk rotates to different positions different combinations of pins are raised. In this way all necessary numbers can be displayed.

A clear benefit to this design is that only four moving parts are needed. This cuts down on the energy necessary to run the watch. Smaller servos than those which last semester's group used can be utilized, since the disk would not have to turn more than 165 degrees to achieve numbers zero through nine. Not only does this design permit smaller components, but it also removes ambiguity of the Braille number display. The pegs stay in place, causing the Braille dots to remain aligned. There is little room for alignment error in the disk placement, but the benefits of this design outweigh the minimal downfalls.

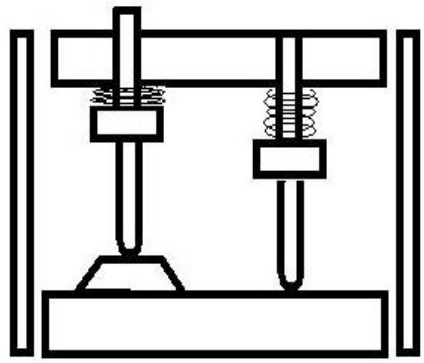


Figure 11: The raised surface on the disk pushes the Braille pin to the surface of the watch

Design Evaluation

The rotating disks, sliding plates and disk and pins designs were evaluated on a weighted one to ten scale for a variety of design criteria (Table 1). The most important

Table 1 – The design matrix displays the design evaluation on a scale of one to ten (one = poor, ten = excellent) and is weighted on a variety of design criteria.

Weight	Design Aspects	Rotating Disks	Sliding Plates	Disk and Pins
0.05	Prototype Cost	6	7	9
0.15	Aesthetics	7	8	9
0.25	Ergonomics	7	8	9
0.05	Safety	9	9	10
0.10	Durability	8	8	8
0.15	Accuracy	10	10	10
0.15	Design Simplicity	9	9	7
0.10	Scalability	5	10	9
1	**Total**	**7.70**	**8.65**	**8.80**

criteria were given more weight in the matrix and include ergonomics, aesthetics, accuracy, and design simplicity. These aspects were determined to be the most important design characteristics since they are critical in terms of the ease of use and effective functionality of the final product.

Ergonomics was weighted most heavily in the design matrix since the two current methods and the previous teams' designs were deficient in this area. Also, we feel that ergonomics is the most important criteria in functionality and success of a watch design. The rotating disk design is too large and lacks potential to be scaled down to a typical watch size. The sliding plates would be a better design ergonomically; however, the disks and pins offer an efficient way to produce a pocketsize watch. Aesthetics also was weighted heavily because the watch shouldn't draw any attention to its user. One of the most important features is the watch's ability to reliably and accurately display the time. The design must also be simple in order to minimize cost, increase durability, and enhance performance. After evaluating the designs' pros and cons, it was determined that the disk and pins design scored the highest and, therefore, was selected as the design to pursue.

Prototype

Servos

Considering that the design required 165 degrees of rotation from the servo, HS-55 sub-micro servos (Figure12) were chosen due to their small size and ability to rotate 180 degrees. The ability to chose such a small servo, measuring 0.89 inches by 0.45 inches by 0.94 inches, greatly reduced the size of the prototype when compared to previous designs. The HS-55 is also power efficient due to their pulse-width modulation. This means that the servo requires a short electric pulse for a duration of 600 to 2400 microseconds in order to rotate the desired angle. Furthermore, these servos can be simply controlled using a microcontroller. [6]

Figure 12 – Four HS-55 sub-micro servos were used to rotate the disks
Image courtesy of Servocity: http://www.servocity.com/html/hs-55_sub-micro.html

Disks

The four disks had to be designed and manufactured with high precision. To easily transition between the recessed and raised surfaces, a slope of 55 degrees was used to transfer between the heights. By using this combination of raised and recessed surfaces seen in Figure 10, all necessary numbers can be displayed. In order to manufacture these disks, a three dimensional printer made available to the team by the BME Department was used (Appendix C). By uploading CAD drawings to the printer, it was able to build the disks out of ABS plastic with a high degree of precision and accuracy.

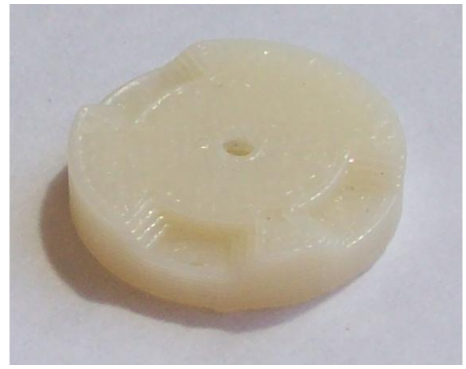


Figure 13 – Four Braille disks were manufactured using the 3D printer.

Pins

The sixteen pins were also made out of ABS using the 3D printer. In order for the pins to be held in place and to support a spring that will push against the underside of the top of the watch, the pins were designed with a circular platform. This platform provides a surface for the spring to rest on and also prevents the pin from falling out if the watch is overturned. The pins were designed with a rounded bottom in order to easily slide up and down the slopes on the disk. The top of the pin was rounded as well in order to provide a user-friendly interface.

Figure 14 – Sixteen Braille pins were needed to display the time.

Springs

In order to hold the pins inside the casing, springs were placed on the pins, keeping the pins from rising when in the recessed position. Originally, springs that were 0.25 inches long with a spring constant, *k*, of 3.8 pounds per inch were used.

Figure 15 – Springs were cut to a fraction of their original length in order to provide the desired force.

However, with these springs, it was calculated that the total force exerted on the top of the casing, while all pins were in the recessed position, was 15.2 pounds. This force made it difficult to assemble the watch and threatened its integrity. In order to eliminate this force, the springs were cut down to a length, which allowed them to be uncompressed while resting in the lowered position in the casing (Figure 15). This permitted the watch to be assembled without difficulty and drastically lowered the forces created from the pins raising and lowering.

Programming

The prototype was programmed using an Arduino Duemilanove USB Board (Figure 16). Arduino is an open-source computing platform based on an input/output board. It functions using an ATMega328 microcontroller that implements the wiring programming language. Wiring is an open source form of Java specifically used for electronics with input/output boards. By downloading the Arduino-0021 and Wiring-0022 programs, code could then be uploaded to the Arduino Board via an USB cord[7].

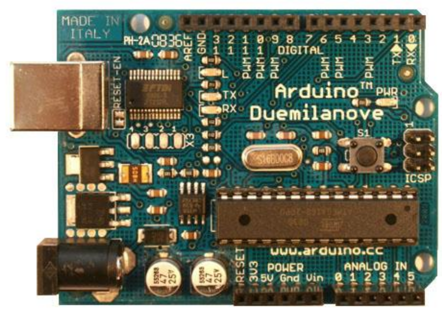


Figure 16 – The Arduino Duemilanove will be used to program the watch
Image courtesy of Arduino:
http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove

The selected timing chip was a DS1307 Real Time Clock Module (Figure 17). This chip easily interfaces with the Arduino when connected in the configuration seen in Figure 18. By downloading the code found in Appendix D, the timing chip was initially programmed so that it contained the correct time and date. This clock module was helpful for this project because after it was initially set, it kept the correct time and date, accurate to within a second. It

Figure 17– DS1307 Real Time Clock Module
Image courtesy of Active Robots:
http://www.active-robots.com/products/components/real-time-clock-module.shtml

is internally powered, so it is able to function even when the Arduino is not connected to a power source.

The code for the final prototype interfaced four of the HS-55 sub-micro servos with the Real Time Clock Module. Using the same configuration seen in Figure 18, the Arduino received the correct date and time from the timing chip and sent the appropriate signal to each of the four servos. This signal rotated each servo to the desired angle, forming the configuration that displayed the correct numerals.

In order to achieve the desired functionality, two buttons were also attached to the Arduino. The first button changed the mode of the watch, while the other updated the display. The watch has three modes: the first shows the time in hours and minutes, the next shows only the minutes and seconds, and the final mode shows the month and day. When the update button is pushed, the updated information of the respective mode is displayed. The servos do not move unless this button or the mode button is pressed, preventing the constant movement of the servos and saving power. The complete code of the prototype can be seen in Appendix E.

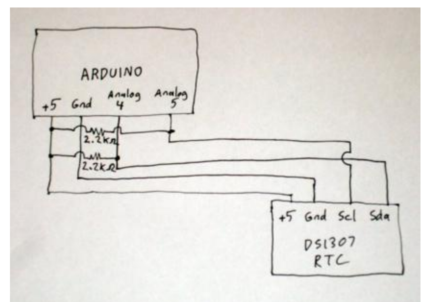


Figure 18– Circuitry for timing chip
Image courtesy of Hobby Robotics: http://www.glacialwanderer.com/hobbyrobotics/?p=12_schem.png

Circuitry

The constructed circuit contained seven: a timing chip, two buttons and seven servos. Much of the circuitry was determined by looking at example projects. Figure 18 demonstrates the wiring used for the timing chip, while Figure 19 is a representation of how each of the buttons was configured. Each of the servos contained a ground, power and input wire. The power wire was connected to the 5V pin on the Arduino, while the ground wire was connected to the

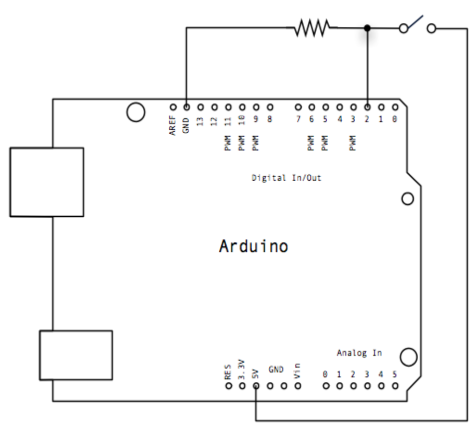


Figure 19– Circuitry for reset button
Image courtesy of Arduino: http://arduino.cc/en/uploads/Tutorial/button_schem.png

ground pin. The input wire for each servo was connected to a different input pin on the Arduino.

Casing

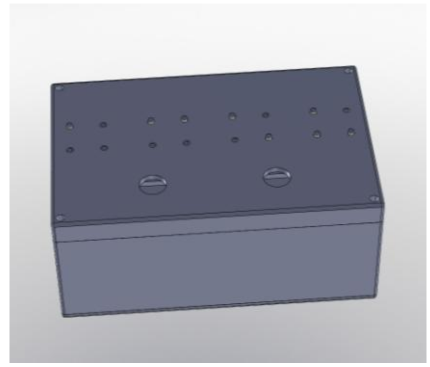


Figure 20– CAD of final assembled casing.

The casing of our watch, which housed everything besides the Arduino controller, was made out of ABS using the three dimensional printer. It involves three separate pieces that are screwed together to form the final case. The bottom portion consists of 3.325 inches by 2.145 inches by 0.063 inches plate that has four bracket areas into which the servos fit. This serves to lock the servos onto the base plate so they are centered directly below the pins and aids in laying out the interior of the watch. The middle portion of the casing covers the servo and disk apparatus and has four grids of two by two holes for the bottom of the pins to rest in. The top of the case has another set of holes for the tops of the pins to raise and lower through as well as two holes for the buttons. Overall the casing measures 3.325 inches by 2.145 inches by 1.425 inches.

Assembly

Due to the casing design, the assembly of the watch was straightforward. The servos were attached to the casing using Gorilla Glue to anchor them into the pre-assigned slots on the base layer of the casing. Due to an alteration in our design, the casing was not tall enough to accommodate the servo and disk apparatus. The original idea was to screw the disks directly into the servo, removing the servo arm attachment that came with the servo. Upon testing this, it was evident that the disks were not perfectly level, so instead the disk was mounted on top of the servo arm, which was then attached directly to the servo. This added approximately 0.063 inches of height to our servo assembly, requiring that spacers be used between the bottom and middle portions of the casing assembly. This provided an added benefit; the wiring from the servos could exit the watch through this gap, eliminating the need for an additional hole in the side of the casing.

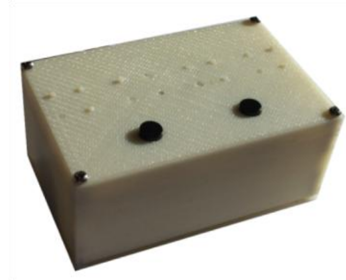


Figure 21 – Assembled Digital Braille watch prototype.

Modifications needed to be performed on many of the printed parts. Initially the sixteen holes in the casing, in which the pins were going to rest, were too small. To solve this problem, these holes were enlarged with a #51 drill bit, 0.067 inches in diameter, and the pins were sanded down. The slopes of the disk were also smoothed out using a circular micro-file. This prevented the base of the pins from catching on the rough surface of the disk. In addition, four holes were drilled in the middle casing in the center of the two by two dot grids since the screws used to secure the disks to the servos rubbed against the bottom of the casing, preventing the disks from rotating properly.

After the bottom and middle portions of the case were secured with screws, the pins were placed into the holes in the middle portion of the casing. The springs that had been cut in half in order to reduce their resistance were then placed on each pin, and the top portion of the casing was attached, completing the assemble of the watch. The complete assembly of the prototype can be seen in Figures 21.

Testing and Results

In order to gain useful feedback on the relative merits of the design, a survey was sent to the Wisconsin Center for the Blind and Visually Impaired. Colton attends this school, so several of his teachers were contacted to ask if they or the students would be willing to answer a few brief questions about the Digital Braille Watch project. The questions gauged the user's familiarity

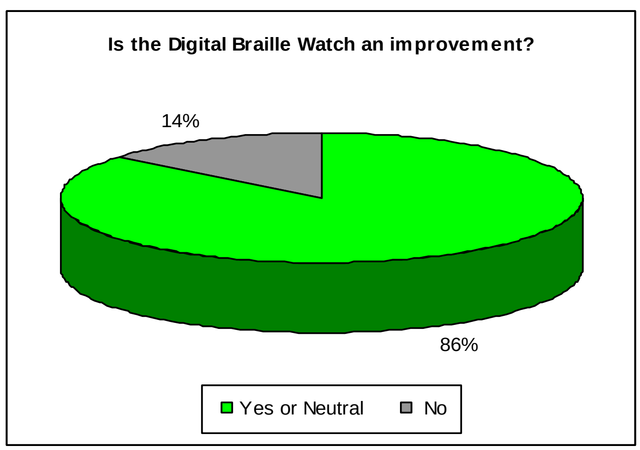


Figure 22– Pie chart demonstrating that the Digital Braille Watch is an improvement upon existing devices.

with current time-telling methods, and they were asked to rate their level of agreement with several statements regarding the Digital Braille Watch and the competition. The full attachment sent to the school can be seen in Appendix F.

All the respondents said that they were familiar with either the tactile analog or audible watch devices. While there were mixed responses regarding whether or not the current methods are adequate, only 14.3% of those surveyed thought that the Digital Braille Watch would not be an improvement upon the existing time-telling methods (Figure 22). When asked to comment on the design, one of the respondents stated, "I strongly encourage the development of a digital watch for those whose personal needs or interests would welcome such a device." This further demonstrates the need for a watch similar to the prototype that has been created.

Management and Planning

At the beginning of the semester, the Gantt chart found in Appendix G was created as a work and time management tool. It was roughly followed throughout the semester to ensure that the project was on track and deliverables were finished on time. All expenses were recorded throughout the semester as well. The projects total cost was $249.96. A detailed list of the expenses can be found in Appendix H.

Future Work

The current design for the digital Braille watch has proved successful in displaying the appropriate time with Braille digits and has been reduced greatly in size from last semester's model. There has also been a reduction in the power needed to run the watch due to the use of smaller and fewer servos. In this regard, the design requirements were achieved, but there are still improvements that can be made to optimize the design.

While this prototype is comparatively small, it is still large enough to be considered bulky as a pocket watch and impractical in size as a wristwatch. Some of the size could be eliminated by reevaluation of part placement. Currently, some of the size is simply due to forced estimates on the space necessary for the internal components. This is something that a rearrangement of parts could easily remedy. Additionally, the use of smaller servos with a more limited rotational range, in conjunction with a gear system to magnify this range, could help minimize the size even further. Furthermore, one of the most critical inhibitors in minimizing the watch size is the space that it would require to house the batteries necessary to run it.

Power consumption has been a primary in previous semesters and will continue to remain so in any future work. The various options that were considered for their size, such as coin batteries, were incapable of lasting for more than a couple of weeks because of the energy draw of the servos. Servos draw energy when active but also when idle. Not only this, but the large number of batteries needed was enough to add significant weight to the watch. A potential solution to this would be to consider a rechargeable battery that could be plugged into a power source to recharge much like a cell phone. In addition, while the real time clock module is effective in communicating the time, it also has significant drift. This means that the timing mechanism is not perfect, which can lead to an inaccurate time display after an extended period of time. A more accurate time-keeping mechanism would be to integrate a WWVB radio-receiver. The time is sent to this chip via a radio signal, ensuring that it remains accurate over long durations of time.[8]

Finally, the watch components would be better able to serve their functions if they were custom made for the purpose. This would not only help with mass production but would also help with streamlining the watch in other ways as well. Size could be further reduced if bulky and unneeded appendages on the servos were eliminated or if a different material was used, allowing thinner parts or different assembly methods.

Conclusion

A fully functional Digital Braille Watch was successfully created using a rotating disk design. The final prototype met all of the major design specifications, and, based on the preliminary survey results, it was well received by a sample of visually impaired students. The design shows promise for becoming a marketable product that addresses all of the shortcomings of the current time-telling devices available for the visually impaired. The prototype is effective in proving the functionality of the design, but in order for it to be commercially ready, there needs to be several improvements. These include minimizing the size of the design, changing the watch's power source, and enhancing its function. This innovative prototype demonstrates that it is possible to create a Digital Braille Watch that is silent, is easy to read and improves the day-to-day life of the visually impaired.

References

[1] “Size and Spacing of Braille Characters.” *Braille Authority of North America*. n.d. 15 Sep. 2010. <http://www.Brailleauthority.org/sizespacingofBraille/>.

[2] Widmaier, E., Raff, H., and Strang, K. (2008). Vander’s Human Physiology (11th ed.) New York, NY: McGraw-Hill Higher Education.

[3] "Haptica Braille Watch Concept." *Tuvie Design of the Future*. 2009. 24 Sep. 2010. <http://www.tuvie.com/haptica-Braille-watch-concept/>.

[4] Litschel, Dietmar, and Schwertner. Device for Representing Relief Items. Caretec GmbH, assignee. US Patent 6109922. 2000.

[5] "Braille Watch." *UW-Madison Biomedical Engineering Design Courses – Project Pages*. 2010. 22 Sep. 2010.

[6] “What’s a Servo?” *Seattle Robotics Society* .n.d. 30 Sep. 2010. <http://www.seattle robotics.org/guide/servos.html>.

[7] “Arduino Duemilanove.” *Arduino*. n.d. 2 Mar. 2010. <http://arduino.cc/en/Main/ ArduinoBoardDuemilanove?action=diff>.

[8] “WWVB Radio Controlled Clocks.” National Institute of Standards and Technology: Time and Frequency Division. 2010. 30 Mar. 2010. <http://tf.nist.gov/stations/radioclocks.htm >.

Appendix A: Product Design Specifications

Project Design Specifications—Digital Braille Watch

September 24, 2010

Team: Nick Anderson, Taylor Milne, Kyle Jamar, Chandresh Singh
Client: Holly and Colton Albrecht
Advisor: John Puccinelli

Problem Statement:

In order to determine the time, the visually impaired currently depend on audio or tactile analog watches. However, audible watches are disruptive, while the analog tactile watches are often difficult to read and fragile. Our goal is to develop a digital Braille watch that will efficiently display the time without the issues of the current technologies. This watch should display military time, be accurate and reliable, and utilize the standard Braille numerals.

Client Requirements:

- Digital military time display
- Silent and without vibrations
- Time in standard Braille

Design Requirements:

1) Design Requirements

a) *Performance requirements:* See Client Requirements above
b) Safety: All electronics must be contained and the watch must not contain hazardous materials
c) *Accuracy and Reliability*: The watch must accurately display military time within the
minute
d) *Life in Service*: The watch must be able function continuously while connected to a power source
e) *Shelf Life*: Not specified for prototype
f) *Operating Environment*: The device must be able to operate reliably in a dry environment
g) *Ergonomics*: The watch should not contain rough edges or loose components and the display surface should be easy to read
h) *Size*: The prototype does not need to be watch-sized but should be scalable
i) *Weight*: See Size Requirement
j) *Materials*: The device must comprise of non-toxic components
k) *Aesthetics*, *Appearance, and Finish*: The watch should be aesthetically pleasing

2) Product Characteristics

a) *Quantity*: One working prototype
b) *Target Product Cost*: $100 or less when mass-produced

3) Miscellaneous

a) *Standards and Specifications*: Must display time according to the standard Braille language
b) *Customer*: The customer would like a device that physically displays the time using Braille digits
c) *Patient Related Concerns*: None
d) *Competition*: Audible and tactile analog watches are commercially available for the visually-impaired

Appendix B: Budget Proposal

Team 34
Clients: Holly and Colton Albrecht
albrechth@firstweber.com
(608)239-2083
Advisor: John Puccinelli
jppuccinelli@wisc.edu
Team: Nick Anderson (Team Leader)
Taylor Milne (Communicator)
Chandresh Singh (BWIG)
Kyle Jamar (BSAC)

Problem Statement: In order to determine the time, the visually impaired currently depend on talking or tactile analog watches. However, talking watches are disruptive, while the analog tactile watches are often difficult to read and fragile. Our goal is to develop a digital Braille watch that will efficiently display the time without the issues of the current technologies. This watch should display military time, be accurate and reliable, and utilize the standard Braille numerals.

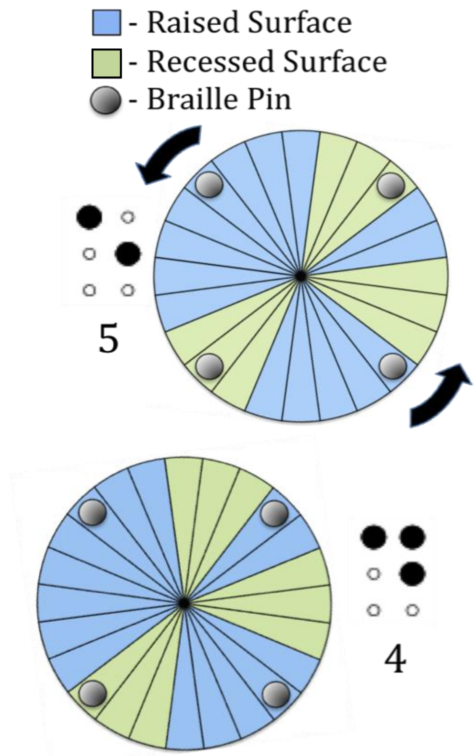


Figure A– The raised and recessed surfaces on the disk cause different numbers to be displayed

Design Description: For this design, four disks are located beneath the watch surface, one for each Braille numeral. Above each disk four pins are positioned so that they rest on the disks surface. The portion of the disk against which the pins rest has both raised and recessed sections (Figure A). If a pin is on the raised surface, it will be pushed slightly above the watch plane, and if not, the top of the pin will remain flush with the surface of the watch (Figure B). When the disk rotates to different positions different combinations of pins are raised. In this way all necessary numbers can be displayed.

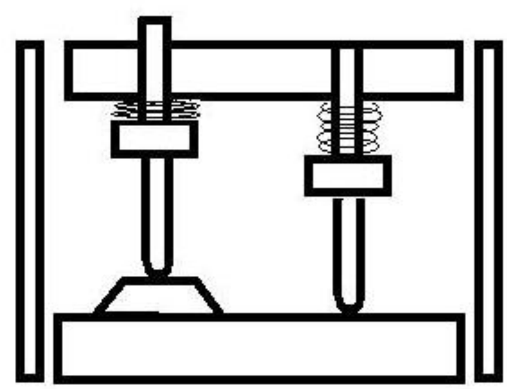


Figure B: The raised surface on the disk pushes the Braille pin to the surface of the watch

Expenses: In order to rotate the disks, four HS-50 Ultra-Micro “Feather” servos will be needed. These will be controlled by a DEV-09218 Arduino Pro Mini microcontroller. Using a DEV-08165 Arduino Serial USB Board, the Arduino Pro Mini will be interfaced with a computer and programmed. The correct time will be sent to the Arduino Pro Mini using a BOB-00099 Real Time Clock Module. Also, the watch user will be able to control the device by using two

COM-09190 Momentary Push Button Switches. In order to power the watch, twenty CR2477 3V Lithium Coin Batteries will be needed. Finally, in order to lower the pins, sixteen quarter-inch compression springs will be needed.

Part Number	Website	Quantity	Per Unit Cost	Extended Cost Without Shipping
HS-50	http://www.servocity.com/html/hs-50_ultra-micro__feather_.html	4	$14.99	$59.96
DEV-09218	http://www.sparkfun.com/commerce/product_info.php?products_id=9218	1	$18.95	$18.95
DEV-08165	http://www.sparkfun.com/commerce/product_info.php?products_id=8165	1	$20.95	$20.95
BOB-00099	http://www.sparkfun.com/commerce/product_info.php?products_id=99	1	$19.95	$19.95
COM-09190	http://www.sparkfun.com/commerce/product_info.php?products_id=9190	2	$0.50	$1.00
CR2477	http://www.batteryjunction.com/cr2477.html	20	$2.75	$55.00
Item #2001	http://hardwareproducts.thomasnet.com/item/all-categories/compression-springs/02001?	16	$3.45	$55.20
Total Extended Cost Without Shipping				**$231.01**

Appendix C: 3D Printer Proposal

Team 34
Clients: Holly and Colton Albrecht
albrechth@firstweber.com
(608)239-2083
Advisor: John Puccinelli
jppuccinelli@wisc.edu
Team: Nick Anderson (Team Leader)
Taylor Milne (Communicator)
Chandresh Singh (BWIG)
Kyle Jamar (BSAC)

<u>Problem Statement</u>: In order to determine the time, the visually impaired currently depend on talking or tactile analog watches. However, talking watches are disruptive, while the analog tactile watches are often difficult to read and fragile. Our goal is to develop a digital Braille watch that will efficiently display the time without the issues of the current technologies. This watch should display military time, be accurate and reliable, and utilize the standard Braille numerals.

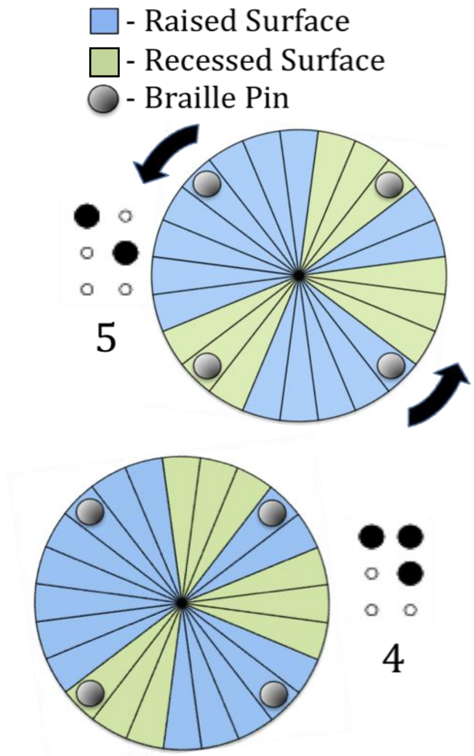


Figure I– The raised and recessed surfaces on the disk cause different numbers to be displayed

<u>Design Description</u>: For this design, four disks are located beneath the watch surface, one for each Braille numeral. Above each disk four pins are positioned so that they rest on the disks surface. The portion of the disk against which the pins rest has both raised and recessed sections (Figure I). If a pin is on the raised surface, it will be pushed slightly above the watch plane, and if not, the top of the pin will remain flush with the surface of the watch (Figure II). When the disk rotates to different positions different combinations of pins are raised. In this way all necessary numbers can be displayed.

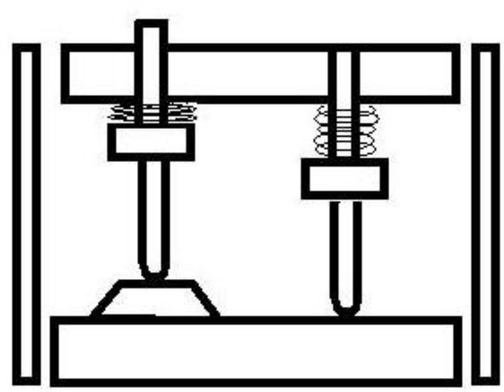


Figure II: The raised surface on the disk pushes the Braille pin to the surface of the watch

<u>Printer Needs</u>: For this design to work, precise parts are needed. We would like to use the printer to print off the Braille pins, disks, and casing that our watch requires. Due to the recent development of this design, we have yet to construct a CAD. However, approximations for the size of the watch parts can be provided. Each pin will be less than ¼ in. tall, so the sixteen pins combined will not require more than ½ in^3 of plastic. Each disk

will be approximately ¾ in. in diameter and less than ¼ in. in thickness. Thus, the total plastic required for the disks will not exceed ½ in^3. Finally, the maximum size of our casing will be is 1 in. x 3 in. x 1.5 in. This casing will have a hollow center and will not exceed 1/16 in. thick. In its complete form, this case will not contain more than 1 in^3 of plastic. In total, less than 2 in^3 of ABS will be needed to print all of our parts. Since we are working on a community outreach project, the BME department will be funding our project this semester and will provide us with the money to print the parts. CAD drawings will be provided as soon as possible.

Appendix D: Real Time Clock Module Code

```
/*
 * RTC Control v.01
 * by <http://www.combustory.com> John Vaughters
 * Credit to:
 * Maurice Ribble - http://www.glacialwanderer.com/hobbyrobotics for RTC
DS1307 code
 *
 * With this code you can set the date/time, retreive the date/time and use
the extra memory of an RTC DS1307 chip.
 * The program also sets all the extra memory space to 0xff.
 * Serial Communication method with the Arduino that utilizes a leading CHAR
for each command described below.
 * Commands:
 * T(00-59)(00-59)(00-23)(1-7)(01-31)(01-12)(00-99) -
T(sec)(min)(hour)(dayOfWeek)(dayOfMonth)(month)(year) - T Sets the date of
the RTC DS1307 Chip.
 * Example to set the time for 02-Feb-09 @ 19:57:11 for the 3 day of the
week, use this command - T1157193020209
 * Q(1-2) - (Q1) Memory initialization  (Q2) RTC - Memory Dump
 */

#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68  // This is the I2C address


// Global Variables

int command = 0  // This is the command char, in ascii form, sent from the
serial port
int i;
long previousMillis = 0;        // will store last time Temp was updated
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
byte test;

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val%10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val%16) );
}

// 1) Sets the date and time on the ds1307
// 2) Starts the clock
// 3) Sets hour mode to 24 hour clock
// Assumes you're passing in valid numbers, Probably need to put in checks
for valid numbers.

void setDateDs1307()
```

```
{

   second = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
   minute = (byte) ((Serial.read() - 48) *10 +  (Serial.read() - 48));
   hour  = (byte) ((Serial.read() - 48) *10 +  (Serial.read() - 48));
   dayOfWeek = (byte) (Serial.read() - 48);
   dayOfMonth = (byte) ((Serial.read() - 48) *10 +  (Serial.read() - 48));
   month = (byte) ((Serial.read() - 48) *10 +  (Serial.read() - 48));
   year= (byte) ((Serial.read() - 48) *10 +  (Serial.read() - 48));
   Wire.beginTransmission(DS1307_I2C_ADDRESS);
   Wire.send(0x00);
   Wire.send(decToBcd(second));    // 0 to bit 7 starts the clock
   Wire.send(decToBcd(minute));
   Wire.send(decToBcd(hour));   // If you want 12 hour am/pm you need to set
                                   // bit 6 (also need to change
readDateDs1307)
   Wire.send(decToBcd(dayOfWeek));
   Wire.send(decToBcd(dayOfMonth));
   Wire.send(decToBcd(month));
   Wire.send(decToBcd(year));
   Wire.endTransmission();
}

// Gets the date and time from the ds1307 and prints result
void getDateDs1307()
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.send(0x00);
  Wire.endTransmission();

  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

  // A few of these need masks because certain bits are control bits
  second     = bcdToDec(Wire.receive() & 0x7f);
  minute     = bcdToDec(Wire.receive());
  hour       = bcdToDec(Wire.receive() & 0x3f
  dayOfWeek  = bcdToDec(Wire.receive());
  dayOfMonth = bcdToDec(Wire.receive());
  month      = bcdToDec(Wire.receive());
  year       = bcdToDec(Wire.receive());

  Serial.print(hour, DEC);
  Serial.print(":");
  Serial.print(minute, DEC);
  Serial.print(":");
  Serial.print(second, DEC);
  Serial.print("  ");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(year, DEC);

}
```

```
void setup() {
  Wire.begin();
  Serial.begin(57600);

}

void loop() {
     if (Serial.available()) {      // Look for char in serial que and
process if found
      command = Serial.read();
      if (command == 84) {      //If command = "T" Set Date
       setDateDs1307();
       getDateDs1307();
       Serial.println(" ");
      }
      else if (command == 81) {  //If command = "Q" RTC1307 Memory Functions
        delay(100);
        if (Serial.available()) {
         command = Serial.read();
         if (command == 49) {
          Wire.beginTransmission(DS1307_I2C_ADDRESS);
          Wire.send(0x08); // Set the register pointer
         for (i = 1; i <= 27; i++) {
             Wire.send(0xff);
            delay(100);
         }
         Wire.endTransmission();
         getDateDs1307();
         Serial.println(": RTC1307 Initialized Memory");
         }
         else if (command == 50) {    //If command = "2" RTC1307 Memory Dump
          getDateDs1307();
          Serial.println(": RTC 1307 Dump Begin");
          Wire.beginTransmission(DS1307_I2C_ADDRESS);
          Wire.send(0x00);
          Wire.endTransmission();
          Wire.requestFrom(DS1307_I2C_ADDRESS, 64);
          for (i = 1; i <= 64; i++) {
             test = Wire.receive();
             Serial.print(i);
             Serial.print(":");
             Serial.println(test, DEC);
          }
          Serial.println(" RTC1307 Dump end");
         }
        }
       }
      Serial.print("Command: ");
      Serial.println(command);   // Echo command CHAR in ascii that was sent
      }

      command = 0;                 // reset command
      delay(100);
    }
```

Appendix E: Final Prototype Code

```
#include <Servo.h>
#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68

Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

const int  buttonPin1 = 11;    // the pin that the pushbutton is attached to
const int  buttonPin2 = 12;    // the pin that the pushbutton is attached to
int buttonPushCounter2 = 0;   // counter for the number of button presses
int buttonState1 = 0;         // current state of the button
int buttonState2 = 0;         // current state of the button
int lastButtonState1 = 0;     // previous state of the button
int lastButtonState2 = 0;     // previous state of the button
int onePress = 0;

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val%10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val%16) );
}

// Stops the DS1307, but it has the side effect of setting seconds to 0
// Probably only want to use this for testing
/*void stopDs1307()
{
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.send(0);
  Wire.send(0x80);
  Wire.endTransmission();
}*/

// 1) Sets the date and time on the ds1307
// 2) Starts the clock
// 3) Sets hour mode to 24 hour clock
// Assumes you're passing in valid numbers
void setDateDs1307(byte second,        // 0-59
                   byte minute,        // 0-59
                   byte hour,          // 1-23
                   byte dayOfWeek,     // 1-7
                   byte dayOfMonth,    // 1-28/29/30/31
                   byte month,         // 1-12
                   byte year)          // 0-99
{
   Wire.beginTransmission(DS1307_I2C_ADDRESS);
   Wire.send(0);
```

```
   Wire.send(decToBcd(second));    // 0 to bit 7 starts the clock
   Wire.send(decToBcd(minute));
   Wire.send(decToBcd(hour));      // If you want 12 hour am/pm you need to
set
                                   // bit 6 (also need to change
readDateDs1307)
   Wire.send(decToBcd(dayOfWeek));
   Wire.send(decToBcd(dayOfMonth));
   Wire.send(decToBcd(month));
   Wire.send(decToBcd(year));
   Wire.endTransmission();
}

// Gets the date and time from the ds1307
void getDateDs1307(byte *second,
          byte *minute,
          byte *hour,
          byte *dayOfWeek,
          byte *dayOfMonth,
          byte *month,
          byte *year)
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.send(0);
  Wire.endTransmission();

  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

  // A few of these need masks because certain bits are control bits
  *second     = bcdToDec(Wire.receive() & 0x7f);
  *minute     = bcdToDec(Wire.receive());
  *hour       = bcdToDec(Wire.receive() & 0x3f);  // Need to change this if
12 hour am/pm
  *dayOfWeek  = bcdToDec(Wire.receive());
  *dayOfMonth = bcdToDec(Wire.receive());
  *month      = bcdToDec(Wire.receive());
  *year       = bcdToDec(Wire.receive());
}
void setTime(Servo servo1, Servo servo2, int num){
  if (num == 0){
   servo1.write(157);
   delay(50);
   servo2.write(77);
  }
  if (num == 1){
   servo1.write(104);
   delay(50);
   servo2.write(132);
  }
  if (num == 2){
   servo1.write(104);
   delay(50);
   servo2.write(157);
  }
  if (num == 3){
   servo1.write(77);
```

```
    delay(50);
    servo2.write(132);
   }
   if (num == 4){
    servo1.write(77);
    delay(50);
    servo2.write(104);
   }
   if (num == 5){
    servo1.write(104);
    delay(50);
    servo2.write(104);
   }
   if (num == 6){
    servo1.write(77);
    delay(50);
    servo2.write(157);
   }
   if (num == 7){
    servo1.write(77);
    delay(50);
    servo2.write(77);
   }
   if (num == 8){
    servo1.write(104);
    delay(50);
    servo2.write(77);
   }
   if (num == 9){
    servo1.write(157);
    delay(50);
    servo2.write(157);
   }
}
void setup()
{
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  Wire.begin();
  Serial.begin(9600);

  // Change these values to what you want to set your clock to.
  // You probably only want to set your clock once and then remove
  // the setDateDs1307 call.
  second = 0;
  minute = 58;
  hour = 9;
  dayOfWeek = 4;
  dayOfMonth = 13;
  month = 4;
  year = 10;
  //setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
  servo1.attach(2);
  delay(100);
  servo2.attach(3);
  delay(100);
```

```
  servo3.attach(4);
  delay(100);
  servo4.attach(5);
  delay(100);
  servo1.write(132);  // set servo to mid-point
  delay(100);
  servo2.write(132);  // set servo to mid-point
  delay(100);
  servo3.write(132);  // set servo to mid-point
  delay(100);
  servo4.write(132);  // set servo to mid-point
  delay(100);
 }

void loop()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
  Serial.print(hour, DEC);
  Serial.print(":");
  Serial.print(minute, DEC);
  Serial.print(":");
  Serial.print(second, DEC);
  Serial.print("  ");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(year, DEC);
  Serial.print("  Day_of_week:");
  Serial.println(dayOfWeek, DEC);
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);
  if (buttonState2 == lastButtonState2){
    onePress = 0;
  }
  if (buttonState2 != lastButtonState2 && onePress == 0){
    buttonPushCounter2++;
    onePress = 1;
  }
  if ((buttonPushCounter2 % 3) == 0){
    if (buttonState1 != lastButtonState1) {
      setTime(servo1, servo2, ((hour - hour % 10) / 10));
      delay(50);
      setTime(servo3, servo4, (hour % 10));
      delay(50);
      setTime(servo5, servo6, ((minute - minute % 10) / 10));
      delay(50);
      setTime(servo7, servo8, (minute % 10));
      delay(50);
    }
  }
  if ((buttonPushCounter2 % 3) == 1){
    if (buttonState1 != lastButtonState1) {
      setTime(servo1, servo2, ((minute - minute % 10) / 10));
```

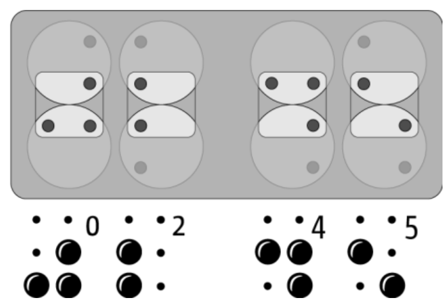
```
      delay(50);
      setTime(servo3, servo4, (minute % 10));
      delay(50);
 }
  }
  if ((buttonPushCounter2 % 3) == 2){
    if (buttonState1 != lastButtonState1) {
      setTime(servo1, servo2, ((month - month % 10) / 10));
      delay(50);
      setTime(servo3, servo4, (month % 10));
      delay(50);
  }
  }
}
```

Appendix F: Survey

Digital Braille Watch for the Visually Impaired – Survey

The purpose of this survey is to test the effectiveness and necessity of a design create by an UW-Madison Biomedical Engineering design team for their semester design project. Please read the following problem statement and design description and then fill out the following questions. Thank you!

Problem Statement: In order to determine the time, the visually impaired currently depend on audio or tactile analog watches. However, audible watches are disruptive, while the analog tactile watches are often difficult to read and fragile. Our goal is to develop a digital Braille watch that will efficiently display the time without the issues of the current technologies. This watch should display military time, be accurate and reliable, and utilize the standard Braille numerals.



The disks rotate to display the correct time.

Design Description: To eliminate the current issues experienced with the audible and tactile analog watches, a digital Braille watch has been designed. This watch displays the time in Braille numerals via disks that rotate corresponding to the correct time. This watch is silent and accurate. The watch has added functionality so that it displays the date when a button is pressed. The watch also contains an internal timing chip, so the watch will never have to be set even if the power supply must be replaced.


An example of the final design

Please answer the survey questions by clicking on the following link
http://www.surveymonkey.com/s/KX52NTJ

Appendix G: Gantt Chart

Task	September				October					November				December	
	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10
Project Research and Development															
Researching	■	■	■	■	■	■									
Brainstorming		■	■	■	■	■									
Design Matrix/Cost Estimation					■	■									
Design Selection					■	■	■								
Ordering Materials						■	■	■	■	■					
Prototyping								■	■	■	■	■	■	■	
Testing														■	■
Final Prototype															■
Deliverables															
Progress Reports		■	■	■	■	■	■	■	■	■	■	■		■	■
PDS			■												
Mid-semester Presentation						■	■								
Mid-semester paper						■	■	■							
Final Presentation														■	■
Final Paper															■
Meetings															
Client		■													
Advisor	■	■	■	■	■	■	■	■	■	■	■	■		■	■

Appendix H: Expenses

Date	Item	Per Unit Cost	Extended Cost
11/9/10	Arduino Pro Mini	$18.95	$18.95
11/9/10	Arduino Serial USB Board	$20.95	$20.95
11/9/10	Ultra-Micro "Feather" Servo (x4)	$14.99	$59.96
11/9/10	Real Time Clock Module	$19.95	$19.95
11/9/10	Push Button Switch (x2)	$0.50	$1.00
11/9/10	3V Lithium Coin Battery (x20)	$2.75	$55.00
11/9/10	Compression Spring (x16)	$3.45	$55.20
11/20/10	Arduino Pro Mini	$18.95	$18.95
		TOTAL	$249.96