
Haptic User Interface Phase II

Michigan State University College of Engineering

Partnered with

MSU Resource Center for Persons with Disabilities



FINAL REPORT

April 24, 2013

ECE 480 Design Team 6

Team Members

Eric Bell
James Hunter
Kristen Kirchhoff
Bin Tian
LeRonn Wilson

Sponsors

Mr. Stephen Blosser
MSU RCPD

Combined with:

Marathon Oil, Chrysler, and MSU Artificial Language Laboratory

Facilitator

Dr. Virginia M. Ayres
Associate Professor

*Department of Electrical and Computer Engineering,
Michigan State University*



Design Team 6 Technical Report

Executive Summary

The Michigan State University (MSU) Resource Center for Persons with Disabilities is currently conducting research, in conjunction with Marathon Oil, Chrysler, and the MSU Artificial Language Laboratory, for an alternative haptic user interface for visually impaired individuals. The devices on the market today are too expensive for the average consumer. In particular, current prices make it difficult for blind students to purchase devices that would help overcome challenges in mathematics and science courses. ECE 480 participates in the creation of a novel refreshable Haptic User Interface that enables blind students to “see” technical images and graphs from a computer screen and enhance their learning experience overall. A unique magnetic design enables device manufacture at a very competitive price. An initial Phase I prototype was developed in Fall 2012. ECE 480 Design Team 6 has built upon the Phase I design and successfully implemented Haptic User Interface Phase II with latched non-vibrating pins and Braille-compatible resolution.

Design Team 6 Technical Report

Acknowledgments

There are many people who were essential to the success of Design Team 6.

Mr. Stephen Blosser: He introduced the team to the field of assistive devices and demonstrated how everyday tasks taken for granted can be extraordinarily difficult for persons with disabilities. His vast knowledge and drive to see Team 6 succeed, led to a successful product. Without the assistance generously provided by Mr. Blosser, Team 6 would not have a final product. Mr. Blosser kindly offered to assist the team with creating and winding the solenoids. With the use of his lathe, Team 6 was able to create precise solenoids for a professional prototype. Mr. Blosser also generously offered to build Team 6 an enclosure for the device. Mr. Blosser also allowed the team to borrow two old computer power supplies for the project. It was a great learning experience and a pleasure to have Mr. Blosser and the Resource Center for Persons with Disabilities (RCPD) as sponsors.

Dr. Virginia M. Ayres: As Design Team 6's facilitator, Dr. Ayres showed the extent of her knowledge and patience when assisting Team 6 throughout the semester. Team 6 would like to thank Dr. Ayres for her continuous help in maintaining well executed presentations and technical writings. Dr. Ayres also played a large role in keeping the team on track and ensuring the team always had a back-up plan if the initial plan were to fail. She was able to show the team that failures can often lead to success.

Mr. Gregg Mulder and Mr. Brian Wright of MSU ECE Shop: Without the assistance of these two gentlemen, Team 6 would not have succeeded. They graciously helped with the printing of the 3D model, along with cleaning and preparing the object for usage.

Dr. Timothy Grotjohn: The team is grateful for the instruction provided in this course. The knowledge for how to handle the project, as well as real-world scenarios added to the team's ability to flourish. Feedback received on technical writings, presentations, and the demonstration was found very helpful when preparing for a successful project.

Mr. Gregg Motter: Team 6 would like to thank Mr. Motter for all of his astute lectures. By preparing Team 6 with a lecture on how to interview your customers, the team was able to have an efficient discussion with Mr. Blosser and the blind student testers. By applying necessary skills learned in Mr. Motter's lectures, Team 6 was able to convert spoken language to primary and secondary requirements.

MicroGroup: Team 6 would like to thank MicroGroup for generously donating hypodermic tubing for the use of the Phase II Haptic User Interface. This donation is greatly appreciated and added to the success of the product.

Design Team 6 would also like to extend gratitude to the following:

Professor Erik Goodman for educating the class on project management and Microsoft Project, Professor Fang Peng for his informative lecture on power supply selection and circuit protection, Mr. Steven Noll for his sacrifice of time to guest lecture on intellectual property, Mr. Garth

Design Team 6 Technical Report

Motschenbacher's discussion on engineering job search, Dr. Asif Naseem for his inspirational lecture and tricks of the trade, Mr. Karl Gude for his material on creating an effective poster and presentation, and Mrs. Roxanne Peacock for all of her assistance and kindness in ordering and purchasing materials.

Table of Contents

Executive Summary	1
Acknowledgments.....	2
Chapter 1: Introduction and Background.....	5
1.1 Michigan State University - RCPD.....	5
1.2 Design Project Overview	5
1.3 Current Devices Available	6
1.4 Phase I Summary	7
1.5 Phase II Objectives	8
Chapter 2: From Concept to Solution	8
2.1 Function Analysis	8
2.1.1 FAST Diagram.....	9
2.1.2 House of Quality	10
2.1.3 Gantt Chart.....	10
2.2 Conceptual Designs	11
2.2.1 Proposed Design 1 – Increase Solenoids and Pins.....	12
2.2.2 Proposed Design 2 – Smaller Solenoids and Pins	12
2.2.3 Proposed Design 3 – Magnetic Latching	13
2.3 Final Design Solution	14
2.4 Preliminary Budget	14
Chapter 3: Technical Justification	15
3.1 Hardware Design Efforts	15
3.1.1 Prototype I.....	15
3.1.2 Prototype II	16
3.1.3 Prototype III	18
3.1.4 Solenoid Driver Circuitry	19
3.2 Hardware Implementation	22
3.2.1 Creating Coils	22
3.2.2 Soldering Components.....	25

Design Team 6 Technical Report

3.2.3 Integration of Neodymium Magnets	27
3.2.4 HEF4794B Shift Register	27
3.2.5 Pin Tower and Wedge Designs.....	28
3.3 Software and Interface Design.....	30
3.3.1 PCB Creation and Design	30
3.3.2 Hex Level Shifter: MC14504B.....	33
3.3.3 Initial Shift Register and Arduino Testing.....	34
3.4 Software Implementation.....	35
3.4.1 Arduino Software Phase II.....	35
3.4.2 Graphical User Interface from Phase I.....	37
3.4.3 Graphical User Interface - Phase II.....	43
Chapter 4: Final Product Evaluation.....	44
4.1 Phase II HUI	44
Chapter 5: Final Cost, Schedule, and Conclusions	45
5.1 Final Cost	45
5.2 Schedule.....	46
5.3 Future Enhancements.....	47
5.4 Conclusion	49
Appendix 1: Design Team Composition and Responsibilities	50
LeRonn Wilson: Team Manager.....	50
Bin Tian: Documentation Preparation	51
Eric Bell: Webmaster	51
Kristen Kirchhoff: Presentation Preparation.....	52
James Hunter: Lab Management	53
Appendix 2: References.....	53
Appendix 3: Technical Attachments.....	55

Chapter 1: Introduction and Background

1.1 Michigan State University - RCPD

The Resource Center for Persons with Disabilities (RCPD) office was originally created during the 1971-1972 academic semester, with the goal of providing equal access to Michigan State University (MSU) for all students. The RCPD office provides services to MSU students, employees, and visitors to ensure that they have equal access to all MSU services and facilities. The mission of the RCPD office is to lead Michigan State University in maximizing ability and opportunity for full participation by persons with disabilities.

The functions of the RCPD office are summed up by using the following acronym:

Assess and document disability, academic, and workplace needs
Build and facilitate individual plans for reasonable accommodations
Link individuals with technology, education, and resources
Extend independence through auxiliary aids, disability-related information, and self-advocacy

RCPD at Michigan State values full integration of all persons with disabilities throughout all University programs and services. Further information can be found on the RCPD website <https://www.rcpd.msu.edu/>.

1.2 Design Project Overview

The ECE 480 Design Team 6 for Spring Semester 2013 consists of a motivated group of individuals eager to learn about and help the visually impaired. None of the team members have had any prior experience working with blind individuals, so this design project served not only as an opportunity to design a useful product, but as a great learning experience overall.

Design Team 6 Technical Report

Successful implementation of the cost-competitive refreshable haptic display will enable Jordyn, an MSU sophomore computer science major, and potentially other blind students across the globe like her, to interact and alter drawings and other graphic materials. By allowing the blind to “see” graphs and images in real-time they will have yet another tool to enable them to further their studies. This device will be extremely useful for physics and mathematics materials. The device will be in conjunction with the specificity and needs of the Michigan State University RCPD. It will also take the direct feedback and demands of current blind students and faculty of MSU. Above all else, the goals of the device are for it to be the first commercially affordable device for the average person that is user friendly and productively useful.

1.3 Current Devices Available

Current machines used to produce graphic images for blind students are time consuming and expensive. The amount of time it takes to transfer images to paper or Braille reader so the student can feel it through a Braille style layout makes it difficult to produce the multiple images that would be needed for mathematics and science classes. The devices on the market, which implement graphs and drawings in Braille format, are expensive (\$5,000 - \$10,000). Figure 1.3a shows the Alva 544 Satellite Braille Display, which costs \$6,295. Figure 1.3b shows the PowerBraille Display, which runs from \$4,495 - \$10,550 depending on the resolution of the device. The design of the MSU RCPD refreshable Haptic User Interface (HUI) uses different technology than the ones currently on the market.



Figure 1.3a: Alva 544 Satellite Display



Figure 1.3b: PowerBraille Display

Design Team 6 Technical Report

1.4 Phase I Summary

An initial implementation HUI was developed during the Fall 2012 semester by a prior ECE 480 group. This will be referred to as the Phase I design. The Phase I design was a refreshable HUI, achieved by using solenoids. The magnetic force induced by the solenoid on the metal pins is responsible for displacing the pins through a casing to display the image. The Phase I design was portable and featured rows of pins that would rise to represent portions of an image on a computer screen.

The HUI was driven by computer software designed to open any image in a grayscale format and then allow the user to move the mouse cursor over the image to select portions of the image to be represented by the device pins. The technology utilized in this design consisted of custom software, a modern USB interface, AC/DC and DC/DC power converters, as well as microcontrollers and custom-made solenoid circuits.

The Phase I design achieved the stated goals of the project, but also presented many opportunities for improvement. The design met with mixed reviews from volunteer blind student reviewers. One of the major opportunities for design improvement involves the pins themselves, which, when driven by the solenoid, vibrated significantly. The users did not enjoy this and stated that it made the image difficult to discern. Additionally, the resolution on the display was fairly low, making it difficult to interpret an image. Furthermore, there was significant heat dissipation from the device. Finally the portability could be improved by using compact and lightweight ergonomics.

The Phase I did review positively overall as innovative and was an excellent starting point from which to build upon. Significantly, the Fall 2012 group was successful in keeping the cost of the project low; their total cost of design was only \$110.



Figure 1.4: Phase I final product

1.5 Phase II Objectives

The team's immediate sponsor, Mr. Stephen Blosser from the RCPD, tasked Team 6 to enhance the functionality of Phase I's HUI to better meet the needs of the visually impaired individuals that will use the product. The primary requirements of Phase II were to increase the resolution and eliminate the vibration of the pins. Secondary requirements included reducing the amount of heat that dissipates from the device, implementation with smaller pins that coincide with the size of Braille dots, and reduction in the height that pin travels. Team 6 set out to design and build a HUI that fulfilled the stated objectives, thus advancing the development of a successful product that will positively impact blind individuals' learning experiences.

Chapter 2: From Concept to Solution

2.1 Function Analysis

Following an informative discussion with the project sponsor Mr. Blosser about the objectives for this semester's project, Team 6 decided that it would be best to also receive feedback from the intended customer to get a better understanding of what is important to them. The team asked Mr. Al Puzzuoli, a blind RCPD faculty member, to test the Phase I

Design Team 6 Technical Report

device and express his concerns. Mr. Puzzuoli stated “...the pins trembled perceptibly as I touched them” and “(the pins) were spaced widely enough so that there was not a lot of definition to the images.” He concluded that “Both of these issues combined made for a somewhat murky experience.”

2.1.1 FAST Diagram

After conducting the interviews with Mr. Blosser and Mr. Puzzuoli, the team developed a FAST diagram to clarify the necessary stages required to meet the customers’ and the sponsor’s functional needs. This diagram, shown in Figure 2.1.1, was useful for categorizing important elements, eliminating unnecessary items, and mapping the requirements to ensure efficient functionality desired by the team.

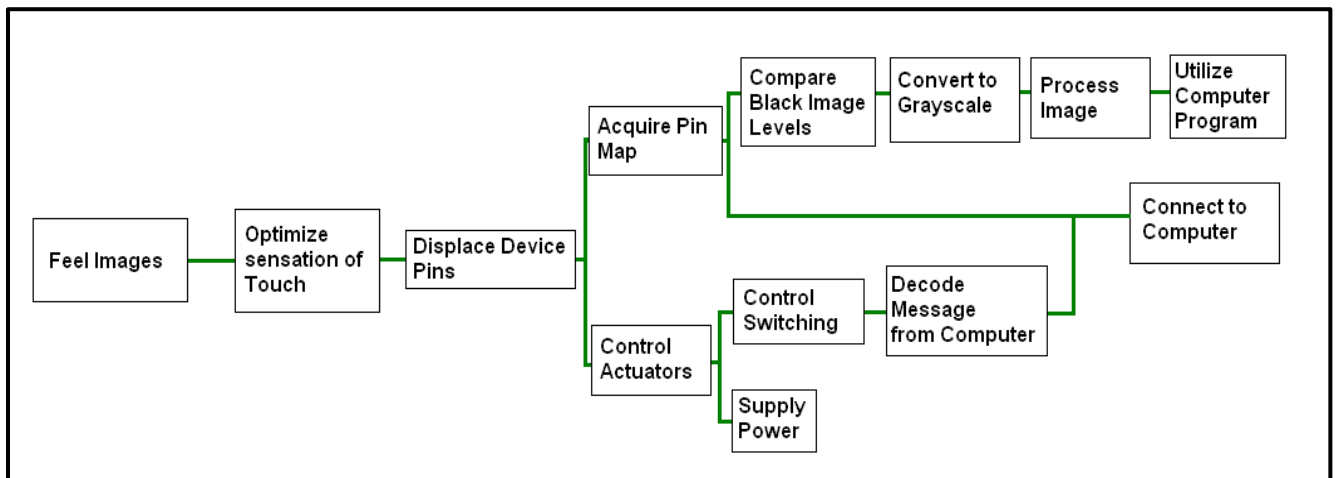


Figure 2.1.1: FAST diagram

To achieve the objective of allowing visually impaired users to “feel” images, the FAST diagram outlines paths needed to succeed. Beginning from the far left and progressing to the right, a series of prior goals was identified. Beginning from the far right and progressing to the left demonstrates the necessity of each task. The FAST diagram clarified sequence of individual task that Team 6 needed to accomplish to allow visually impaired users to feel an image by the physical displacement of pins controlled by a computer program.

Design Team 6 Technical Report

2.1.2 House of Quality

Six Sigma is a systematically driven approach that companies use to strategically prioritize their product and reduce the possibility of error in manufacturing and service. One of the fundamental methods in Six Sigma is a House of Quality diagram. Team 6 utilized this approach, diagram shown in Figure 2.1.2, in order to identify key design parameters including the increased resolution and decreased pin vibration specified by the sponsor and customers to focus on while building various components of the Phase II HUI.

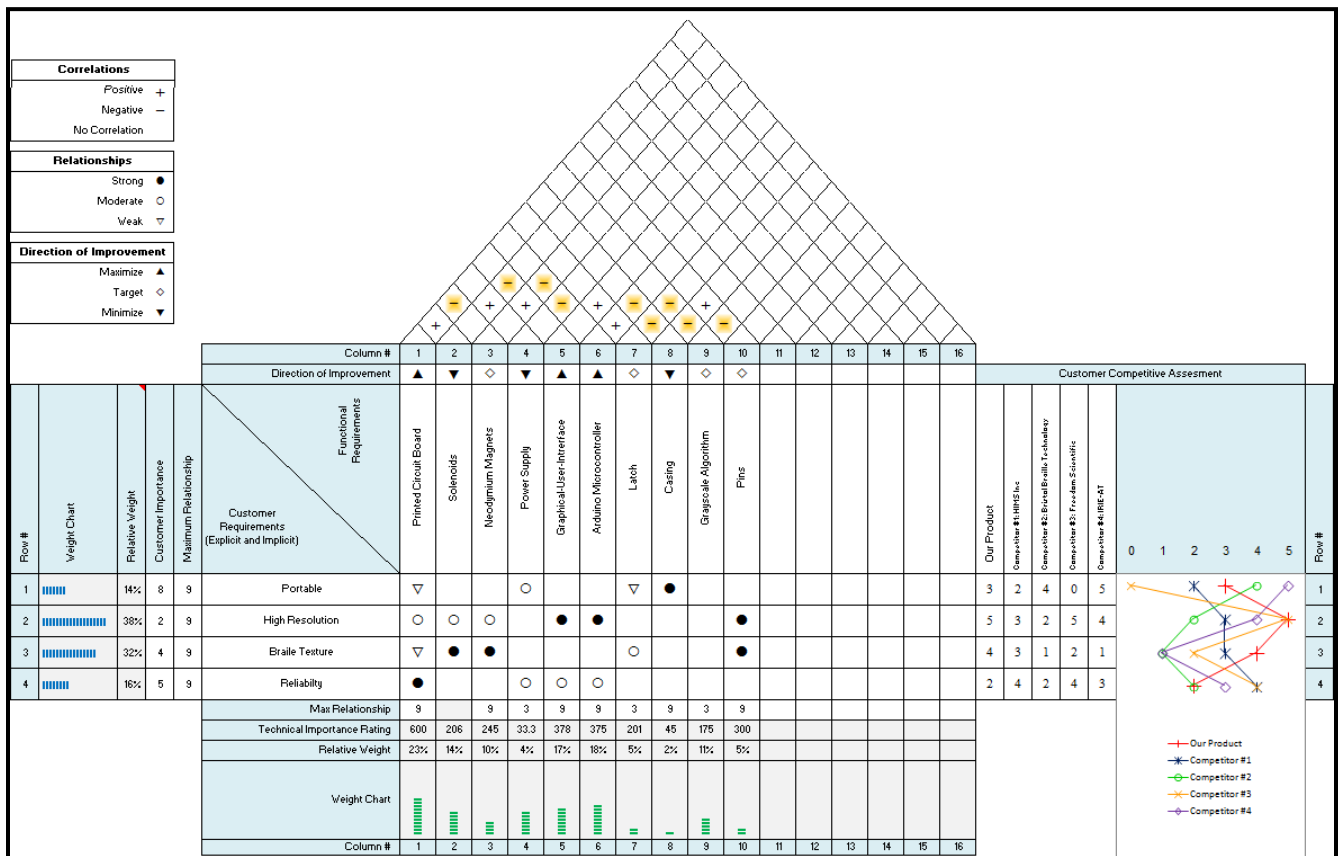


Figure 2.1.2: House of Quality diagram

2.1.3 Gantt Chart

Team 6 illustrated a project schedule using a Gantt chart. The Gantt chart demonstrates the start and finish dates of the terminal elements and summary elements of the project. Terminal elements and summary elements comprise the work breakdown structure of the

Design Team 6 Technical Report

project. It was important to maintain critical path when designing the Gantt chart. Team 6 began by generating a formal definition of the Phase II project with goals, deliverables, constraints, and assumptions. By identifying the project start and end dates, mandatory milestones, and including report signoffs a successful Gantt chart was developed. Team 6 utilized the Gantt chart extensively throughout the semester to maintain a schedule and ensure all goals and deliverables were met. Figure 2.1.3 displays Team 6's Gantt chart.

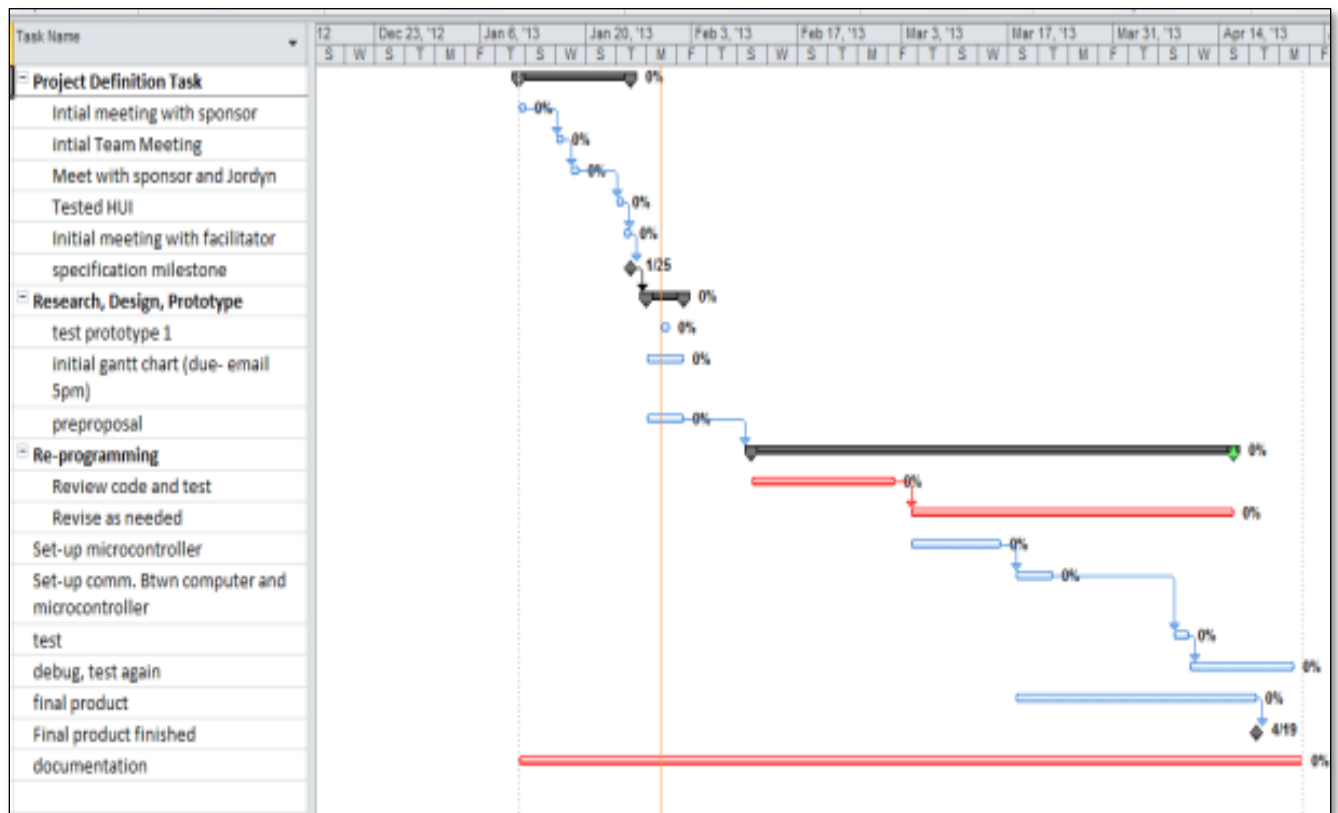


Figure 2.1.3: Phase II Gantt chart

2.2 Conceptual Designs

A discussion of proposed designs as well as the executed design is provided. All approaches are potentially revisit-able under future circumstances. This section records the Team 6 exploration of three conceptual designs, with their advantages and potential pitfalls.

Design Team 6 Technical Report

2.2.1 Proposed Design 1 – Increase Solenoids and Pins

To address the resolution issue, a first approach was to explore inclusion of more solenoids and pins in the device. Team 6 customer interview identified that one factor in the perception of poor resolution in the Phase I design was due to the fact that the pins were placed farther apart than what a Braille-literate individual is used to feeling. The quickest solution to that problem would be to insert solenoids between two rows of existing solenoids to essentially double the resolution. Figure 2.2.1a shows the Phase I rows of the device and how they were laid out and Figure 2.2.1b shows the Design 1 solution to increase the resolution.



Figure 2.2.1a: Current row design

Figure 2.2.1b: Inserting solenoids between existing rows to increase resolution

Although this option increased the resolution, there were multiple disadvantages. First, this design would not resolve the vibration issue, which is a primary requirement. With the pins no longer in matrix, interfacing with software will be an issue. Uniform matrices are easier to write software for, as input for the Arduino Leonardo, a microcontroller used to interface the device with the computer. Table 2.2.1 outlines the advantages and disadvantages to Proposed Design 1.

Pros	Cons
Increases resolution	Doesn't address vibration problem
	Pins no longer in matrix, interfacing with software issue

Table 2.2.1: Proposed Design 1 Evaluation

2.2.2 Proposed Design 2 – Smaller Solenoids and Pins

The second design focused on size reduction for the pins and solenoids. In Phase I there were 32 solenoids and pins. The proposed Phase II design would implement an eight-by-

Design Team 6 Technical Report

eight matrix of pins using 64 solenoids and pins. In order to fit 64 solenoids inside the device, the diameter of the solenoid needs to decrease from 8mm to either 6mm or 4mm. This addresses a problem identified by the student testers. The 8 mm pins made determining the graphic images on the display difficult because this is not a standard Braille dot size. By making the pins smaller, they would become similar to the size of Braille dots, and the device would produce a familiar sensation to the user and thus be more comfortable. Proposed Design 2 resolved the resolution issue. Further analysis showed that this change could positively impact the vibration issue. The pins do not have to protrude as high as they did in the Phase I prototype, therefore potentially decreasing the vibration that occurred when the pins moved. Software modification would be needed to accommodate the 64 pins and solenoids design, but as matrix uniformity is maintained, this did not present an inherent difficulty. Several aspects identified during analysis of Proposed Design 2 were incorporated into Proposed Design 3 and adopted. Table 2.2.2 details the strengths and weaknesses of Proposed Design 2.

Pros	Cons
Increases resolution	Could still have some vibration
Decrease some vibrations	Software modification
More compact	

Table 2.2.2: Proposed Design 2 Evaluation

2.2.3 Proposed Design 3 – Magnetic Latching

Proposed Design 3 explicitly addressed the vibration issue. This design consists of an array of magnetic bars that could create an eight-by-eight matrix (64 pins). The higher pin matrix addresses the resolution issue. The use of magnetic bars will allow the pins to latch when in use, instead of constantly moving up and down. This addresses the vibration issue. The bars would have small holes drilled in them in order to allow the pin to glide up and down as needed. In order to raise the pin when needed (i.e. to feel the graphic image output from the computer) a pulse needs to be sent to both bars. Therefore

Design Team 6 Technical Report

two consecutive pulses are sent to the device from a power source. These pulses charge the bar; setting the polarity so the magnet is attracted to the bar. This allows for the pin to protrude the surface and the user to feel the graphic image. The magnetic latch of the pins will eliminate the vibration of the pin. Without the vibration from the pin constantly rising and falling, the amount of heat dissipated will decrease. The execution of 64 pins, the resolution will be enhanced. This design will lead to a more compact product. Software modification will need to be made to complement this design. Table 2.2.3 details the strengths and weaknesses of Proposed Design 3.

Pros	Cons
Increases resolution	Software modification
Eliminate vibrations	
More compact	
Reduce heat dissipation	

Table 2.2.3: Proposed Design 3 Evaluation

2.3 Final Design Solution

Proposed Design 3 was selected for initial investigation. The latch design, implemented by the magnetic bars, allows higher resolution and eliminates the vibration that currently exists with Phase I HUI, therefore meeting the Phase II design requirements and addressing the two primary requirements identified by users. This design also enables the pin size to a standard Braille dot, while reducing the stroke that the pin travels. The reduction of heat emitted when powered is achieved by the combination of a smaller pin and decreasing the stroke that the pin travels.

2.4 Preliminary Budget

The preliminary budget was based on the implementation of Proposed Design 3. Team 6 determined that neodymium magnets used to induce a magnetic force on the pins to either raise or drop the pins have strength. An Arduino Leonardo microcontroller is used to interface the computer software with the device itself. The hypodermic tubing was

Design Team 6 Technical Report

generously donated by MicroGroup for the use of the haptic project. Minor components provided free of charge from the MSU ECE shop. A completed final circuitry design is fabricated onto a printed circuit board (PCB). A power supply is required to run the device. Table 2.4 shows a breakdown of the HUI Phase II costs at the preliminary estimate stage. This estimate proved accurate except for a later need to purchase specific copper wire for the solenoids.

Product	Cost
Neodymium magnets	\$100
Arduino Leonardo microcontroller	\$30.00
Hypodermic tubing	\$0
Transistors	\$0
Resistors	\$0
Capacitors	\$0
Power supply	\$30
PCBs	\$50
Casing	\$50
Total Cost	\$260

Table 2.4: Preliminary budget

Chapter 3: Technical Justification

3.1 Hardware Design Efforts

3.1.1 Prototype I

Design 3 was first implemented as Prototype I. Mr. Blosser assisted in the design and creation of the magnetic latch prototype. He demonstrated the latching device's ability to raise the pin when pulses were sent and drop the pin when the polarity was reversed. Prototype I was very basic and demonstrated the use of one pin built from a stack of magnets. Team 6 then performed additional tests of the single magnet stack pin prototype. To test the prototype we used the power supply provided in lab and connected two banana-to-banana connectors to the ports of the power supply. We clipped one of the

Design Team 6 Technical Report

alligator clips of the prototype to one of the banana ends. Then to complete the circuit we briefly touched the other alligator clip to the banana connector. This momentary pulse consistently raised the pin. Prototype I enabled identification of alignment issues with the magnet stack pin. These were resolved using a single magnet design. Work with Prototype I also first identified the desirability and feasibility of using smaller diameter magnets. The smaller magnets would be the shape of the pin and enable an eight-by-eight array design. Decreasing the diameter of the magnets decreases the distance between the pins. This reduction creates a device that is competitive with the resolution of existing products. Also a reduction in pin size is consistent with how Braille is written: Braille characters are close together, so the reader is able to read in a fluid motion.

Experiments with Prototype I further revealed that the magnets alone did not exert enough force to withstand the force of the user's finger. In order to increase the force, a solenoid would need to be added. This could easily be implemented using the new single pin magnet design. Prototype I is shown as a schematic in Figure 3.1.1a and in its experimental implementation in Figure 3.1.1b.

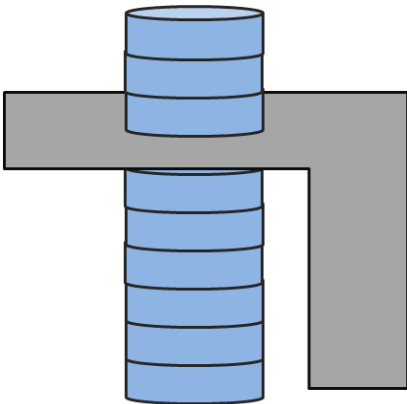


Figure 3.1.1a: Magnet Schematic

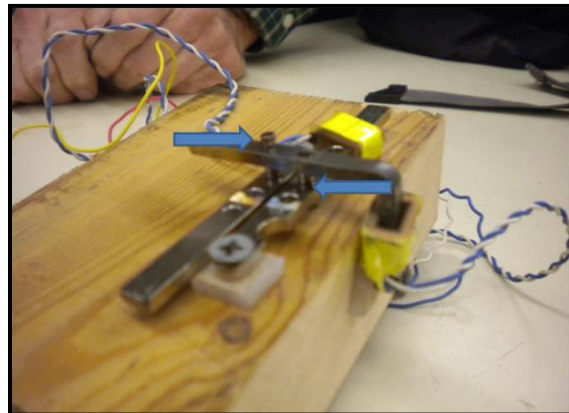


Figure 3.1.1b: Prototype I

3.1.2 Prototype II

Prototype II used a single pin neodymium magnet with a solenoid within the magnetic latch configuration. With the addition of a solenoid, the design proved to be effective.

Design Team 6 Technical Report

To build the solenoid, 0.062 inch diameter hypodermic tubing was used as the casing and copper wire was wound onto it. A cylinder magnet was placed within the tubing. Figure 3.1.2a shows a schematic of the solenoid. The polarity reversing concept to raise and lower pins was demonstrated for Prototype II. The experimental Prototype II is a schematic in Figure 3.1.2a and in its experimental implementation in Figure 3.1.2b. Figure 3.1.2c illustrates the concept of reversing of the polarity to either raise or drop the pin.

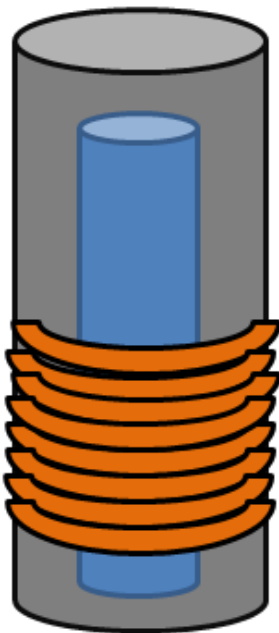


Figure 3.1.2a: Solenoid Schematic

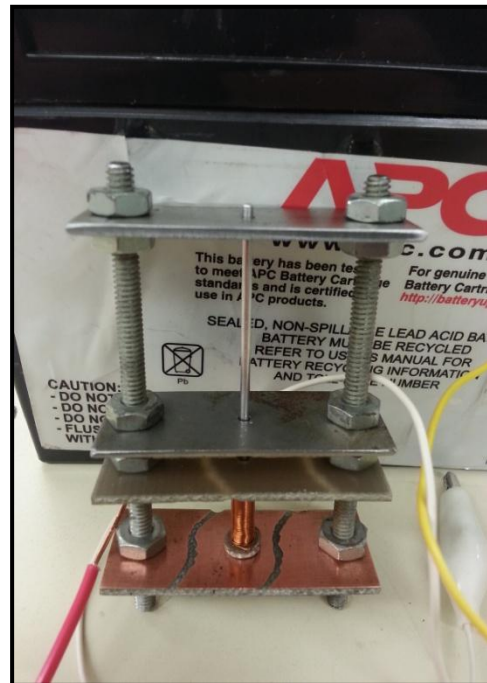


Figure 3.1.2b: Prototype II

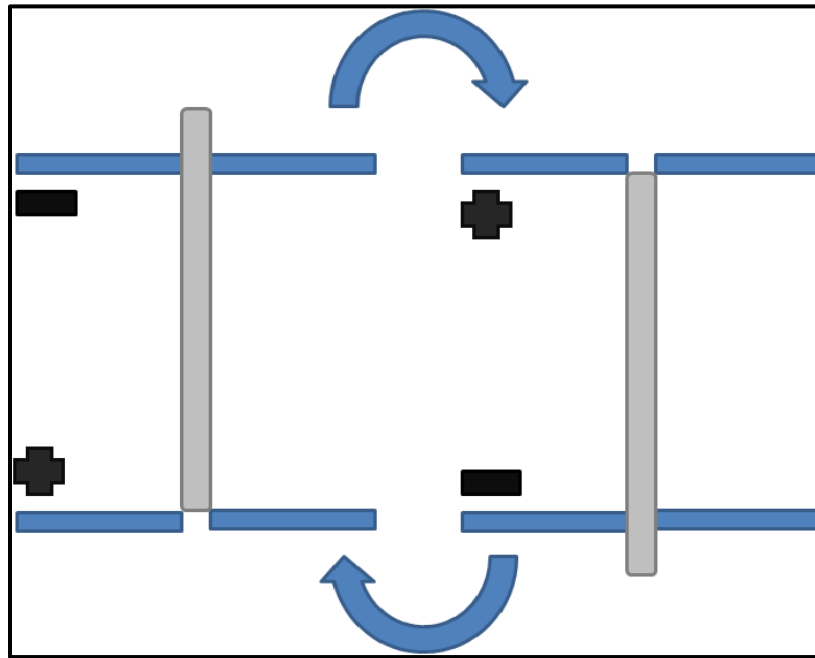


Figure 3.1.2c: Polarity diagram

3.1.3 Prototype III

After testing Prototype II, some concerns were still unresolved. Even with the additional force from the solenoid, if the user applied too much pressure to the pin, it was still susceptible to dropping. In order to create a more durable device, Team 6 concluded that either more current could be sent to the solenoids or a mechanical device would need to be implemented in order to keep the pins from dropping. Increasing the amount of the current would lead to the solenoids dissipating more heat. More heat could cause other components to burn out and could eventually make the device unsafe. A better and safer direction would be to implement a mechanical device to lock the pins in place. Team 6, in collaboration with the sponsor, determined to use a wedge to drive the pins up. The wedge would be attached to the magnet that is inserted in the solenoid. The solenoid creates an electromagnet that drives the magnet and wedge. The wedge then locks under the pin and ensures that the pin will not drop, no matter how much force is applied by the user's finger. The solenoids, neodymium magnets, and hypodermic tubing from Prototype II were all found acceptable for use in Prototype III. A wedge for each solenoid was added and columns to house the pins were designed and created. Prototype

Design Team 6 Technical Report

III is shown as a schematic in Figure 3.1.3a and in its experimental implementation in Figure 3.1.3b.

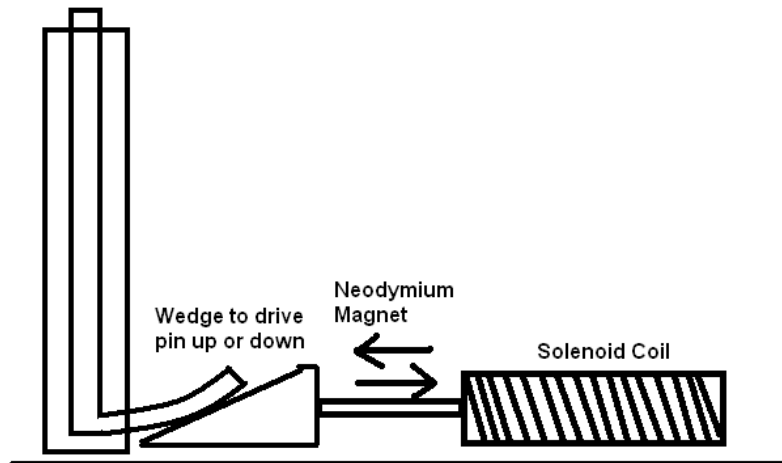


Figure 3.1.3a: Prototype III schematic

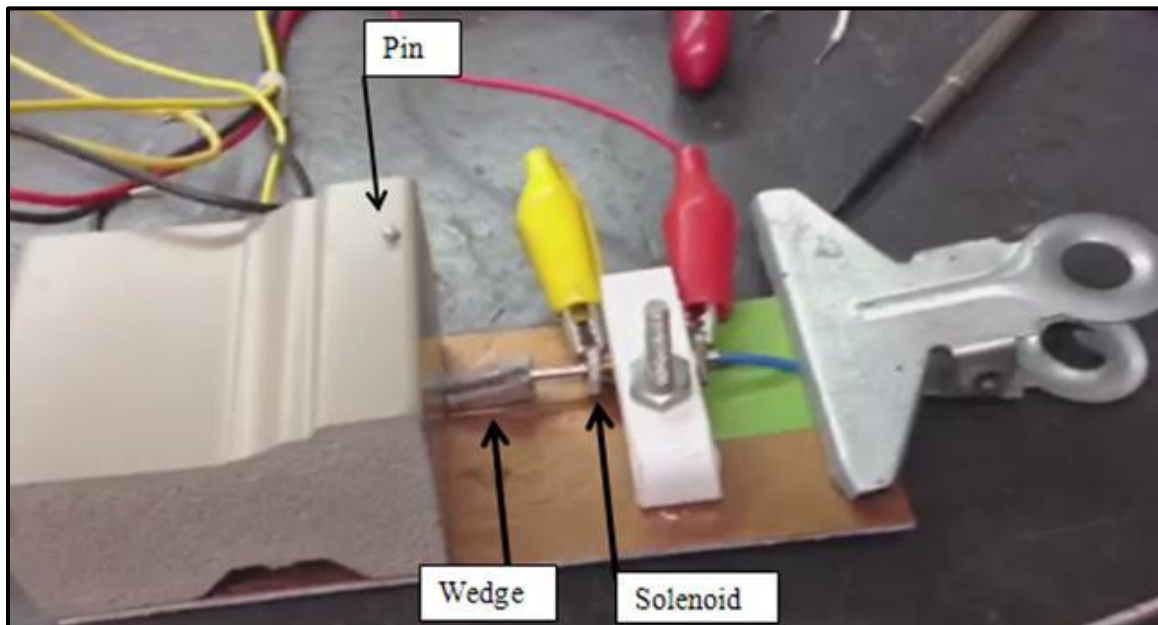


Figure 3.1.3b: Prototype III

3.1.4 Solenoid Driver Circuitry

In order to drive the solenoid, a circuitry design needs to be employed. Figure 3.1.4a shows a model and simulation of the original circuit used to power each solenoid for the

Design Team 6 Technical Report

Phase I HUI. The model and simulation were created in TI TINA. The output of the circuit is a solenoid, which acts as an inductor. The vibrations that were created each time the pin raised and dropped are seen in the output signal.

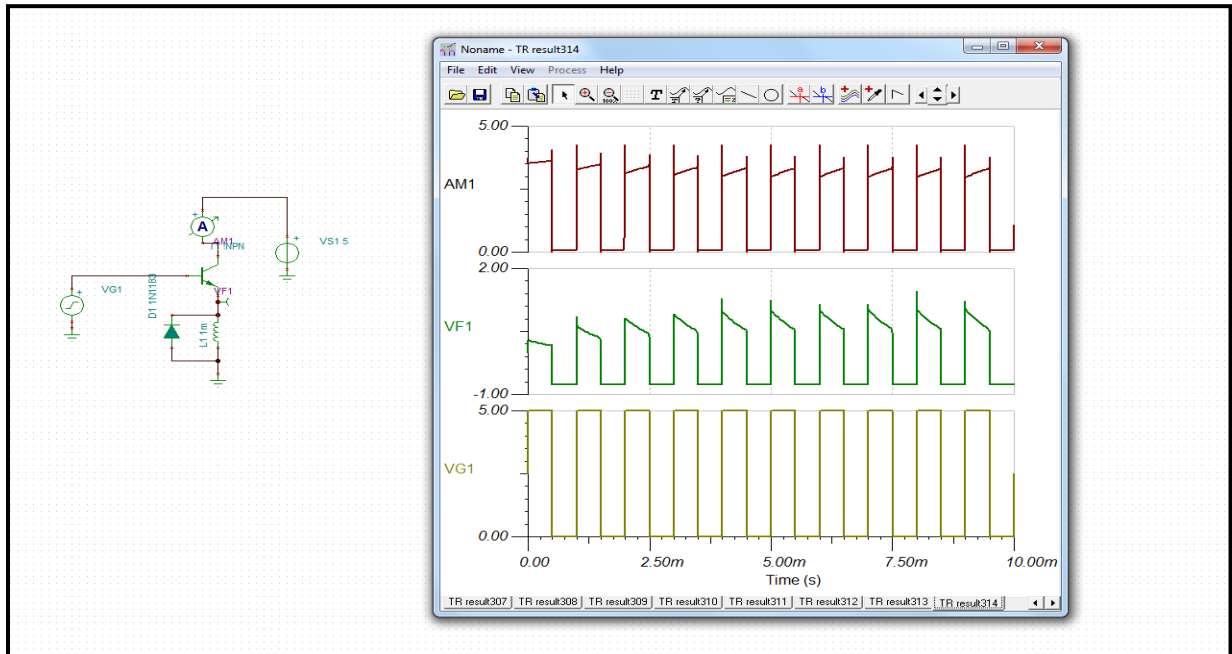


Figure 3.1.4a: Phase I solenoid driver

Team 6's first Phase II circuitry design included a push-pull circuit. The push-pull circuit was implemented in order to have short pulse width, which will create better control and will eliminate the vibration issue. Furthermore, it decreased the heat dispersion, since the pulse width of each pulse had decreased. Figure 3.1.4b is a simulation of this design, along with the circuit schematic. However, a problem identified with this design is that the current and voltage signals did not match. The second issue traced to a fundamental limitation that with a higher inductance, the voltage and current signals decay with time.

Design Team 6 Technical Report

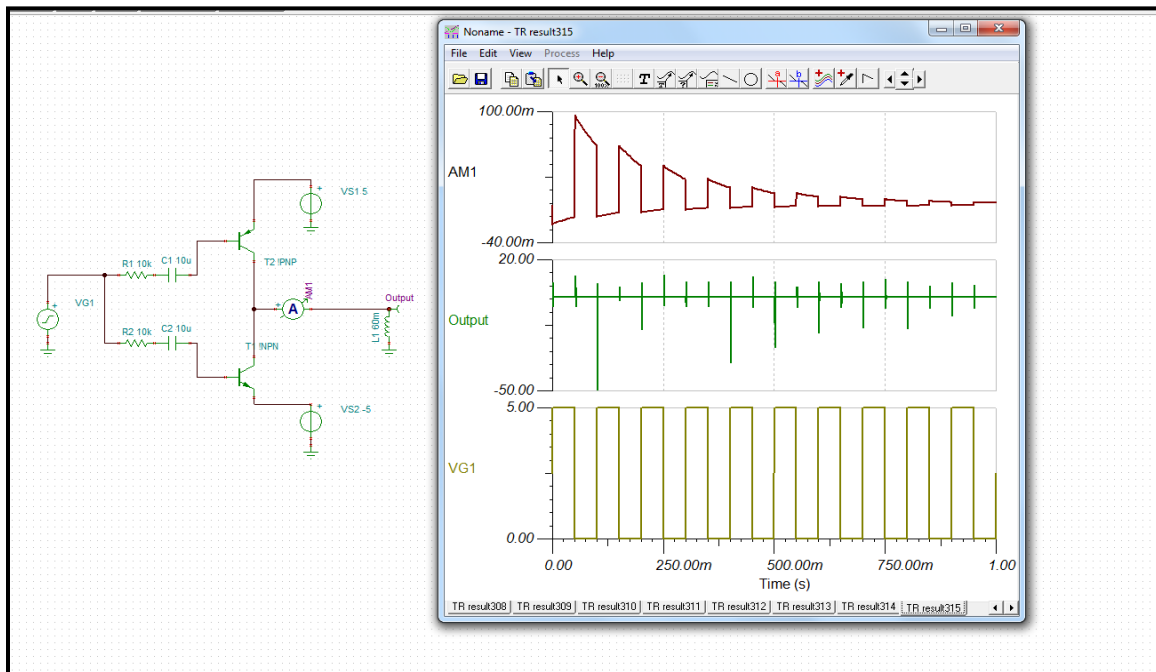


Figure 3.1.4b: First circuitry design

To correct these issues, a revised Phase II circuitry design was created. This design included the addition of a voltage comparator and a differentiator. When the input signal rises, the output voltage is a short positive pulse corresponding to the current in the solenoid. Therefore, the magnet in the solenoid is forced to rise up based on the polarity. When the signal is on its falling edge and the current changes direction, the solenoid switches polarity and allows the pin to drop down. The voltage comparator and differentiator also solved the problem of decaying voltage and current signals. Figure 3.1.4c illustrates the success of the addition of the voltage comparator and differentiator to the push-pull circuit.

Design Team 6 Technical Report

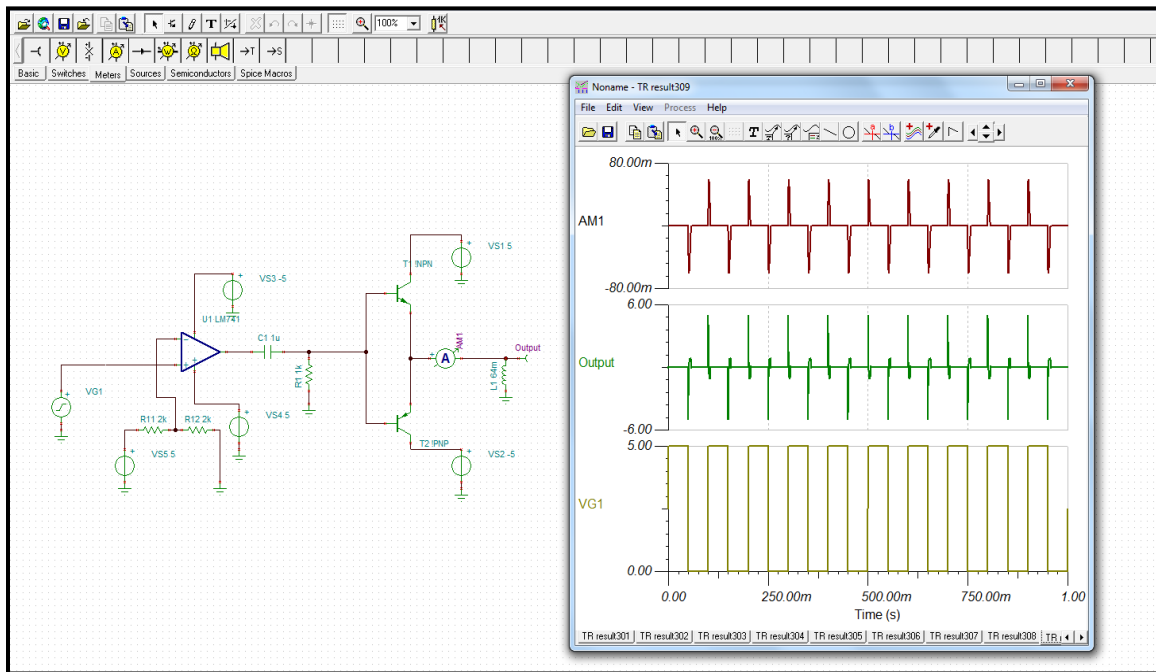


Figure 3.1.4c: Revised driver circuitry

3.2 Hardware Implementation

3.2.1 Creating Coils

The design for this project required the use of solenoids. Simply, a solenoid is a coil wound tightly in a packed helix where it creates a magnetic field. Figure 3.2.1a shows a simple illustration of a solenoid.

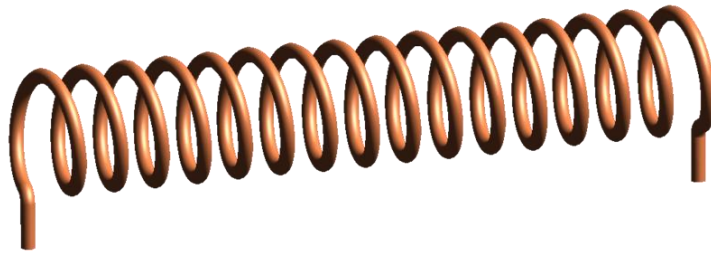


Figure 3.2.1a: Schematic of a solenoid

To verify that the solenoids would provide the proper displacement force Team 6 needed to investigate magnetic field properties of solenoids. Figure 3.2.1b is a diagram of what the magnetic field looks like for an ideal solenoid.

Design Team 6 Technical Report

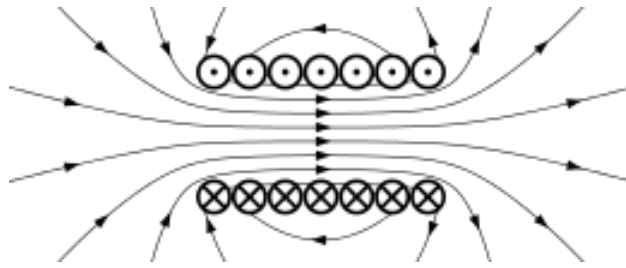


Figure 3.2.1b: Magnetic field created by a solenoid

Before calculating the strength of the solenoid's magnetic field, the magnetic flux density needs to be estimated. Using the following equation, the magnetic flux in a solenoid can be calculated.

$$\Phi = \mu_0 * \frac{N * i * A}{l}$$

With the magnetic flux estimated, the magnetic force exerted by the solenoid on the pin can be calculated using this equation below.

$$F = (N * i)^2 * \mu_0 * \frac{A}{2g^2}$$

Where μ_0 is the magnetic constant, N is the number of turns of wire on the solenoid, i is the current through the solenoid, A is the cross sectional area, and g is the length of the gap between the solenoid and the pin.

For this project Team 6 created and wound its own solenoids. While solenoids could have been ordered through a company, the pricing for solenoids was expensive for the budget. It was also felt that the flexibility to experiment might be desirable. Fortunately, Team 6's sponsor was able to offer access to a unique mini-lathe designed and built by himself, which was ideal to coil the solenoids quickly and efficiently.

Before the copper wire could be wound, a spool had to be created. In collaboration with the sponsor and his equipment, Team 6 were able to cut metal tubing, Figure 3.2.1c, and

Design Team 6 Technical Report

fit each end with a circular cut out of printed circuit board. Upon completion of the spools, the mini-lathe was used to wind each coil with copper wire.

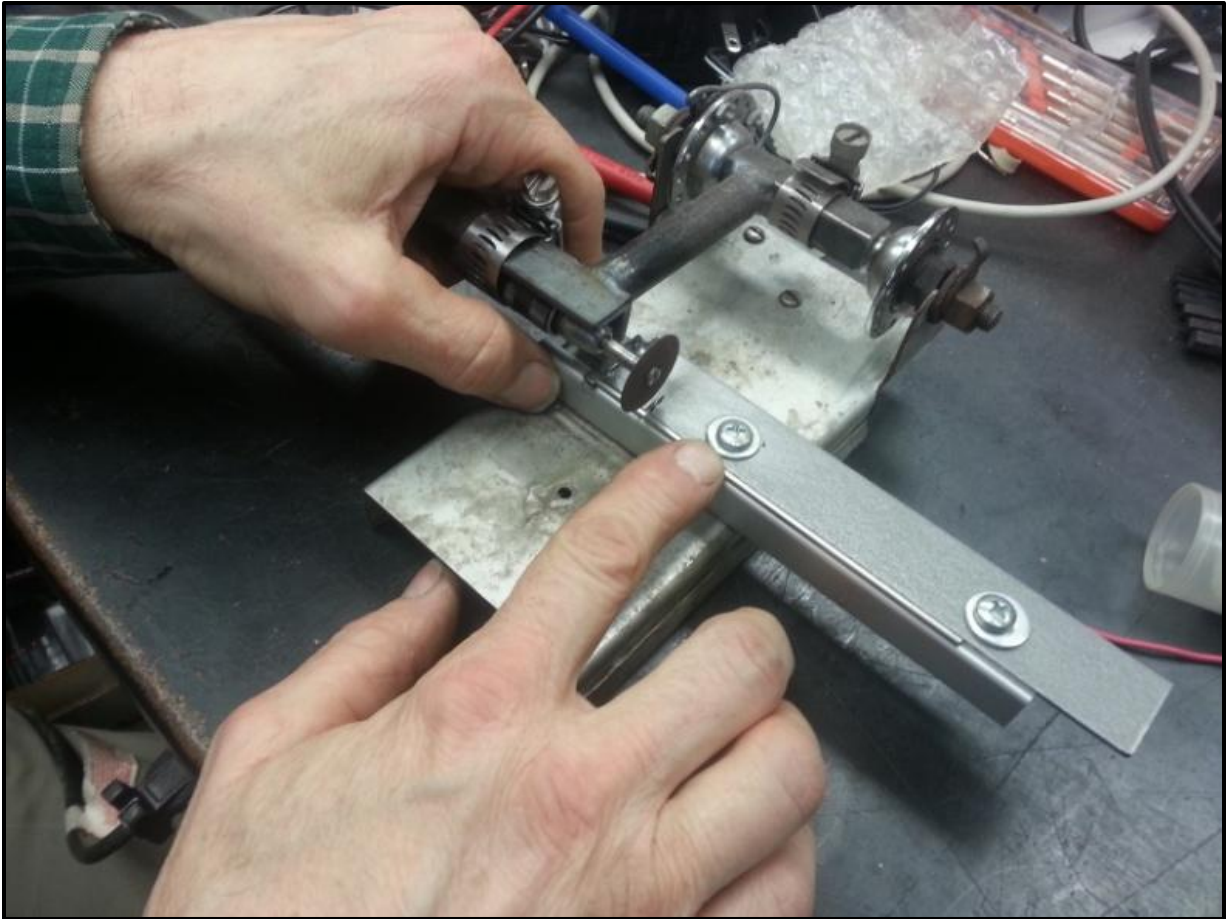


Figure 3.2.1c: Cutting solenoid core

Figure 3.2.1d shows the winding of a coil. In total, 24 solenoids were created and each solenoid had approximately 350 turns of copper wire.



Figure 3.2.1d: Solenoid winding

Each of these solenoids was then soldered onto one of three total PCBs. Each PCB has eight solenoids.

3.2.2 Soldering Components

After every part for the PCB was ordered, and the solenoids were finished, the team began soldering the components onto the PCBs. The most difficult challenge during soldering was the solenoids. Without actual through-hole or surface mount pins on the solenoids, the team experimented with several different methods. First, Team 6 attempted to solder a thick copper wire on the wedge of the solenoid to act as a through-hole pin. However, the soldering proved to be difficult and the decision to switch to

Design Team 6 Technical Report

stripped proto-board wire was made. That wire was difficult as well, so Team 6 decided to simply solder the solenoids onto the board directly, as shown in Figure 3.2.2a. This proved effective.

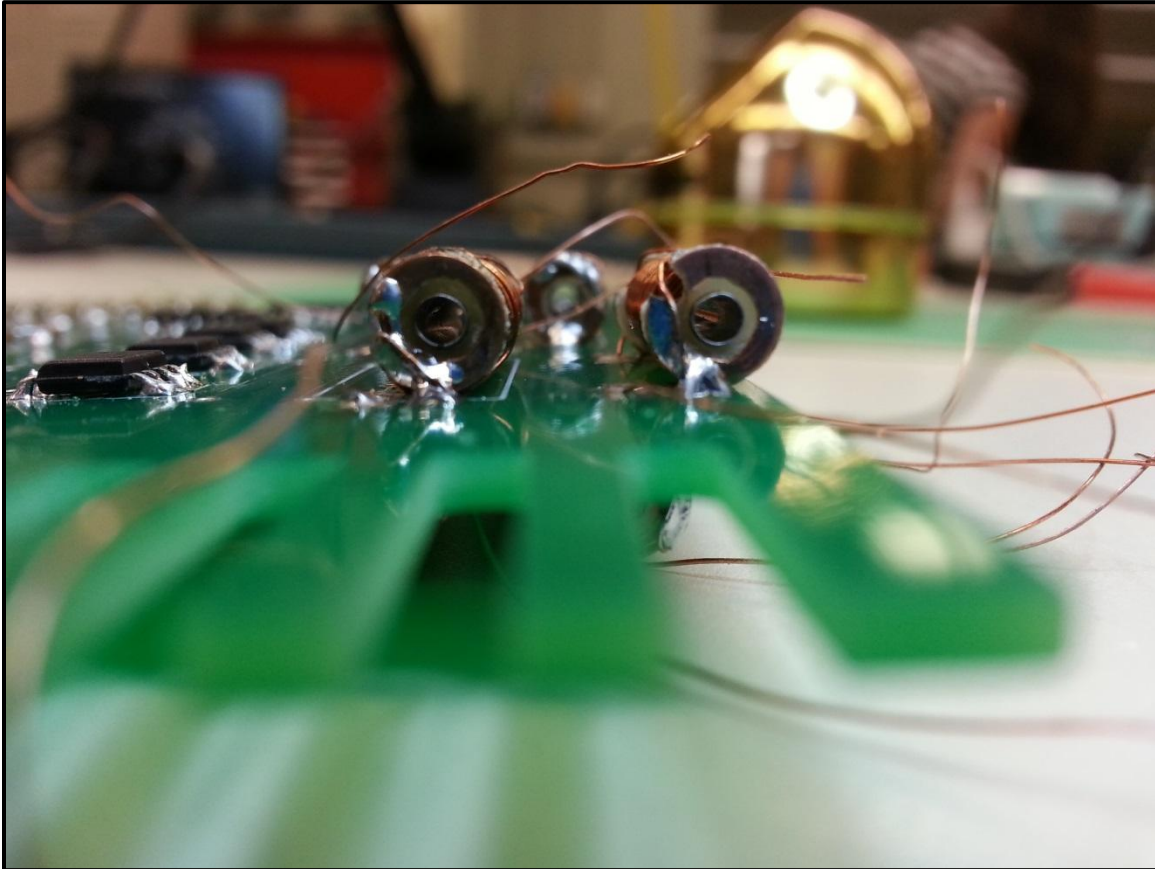


Figure 3.2.2a: The ends of the solenoids were soldered directly onto the PCB

After every piece was soldered in, testing was done at once to ensure proper connections. Figure 3.2.2b shows one of the final PCBs with soldered components.

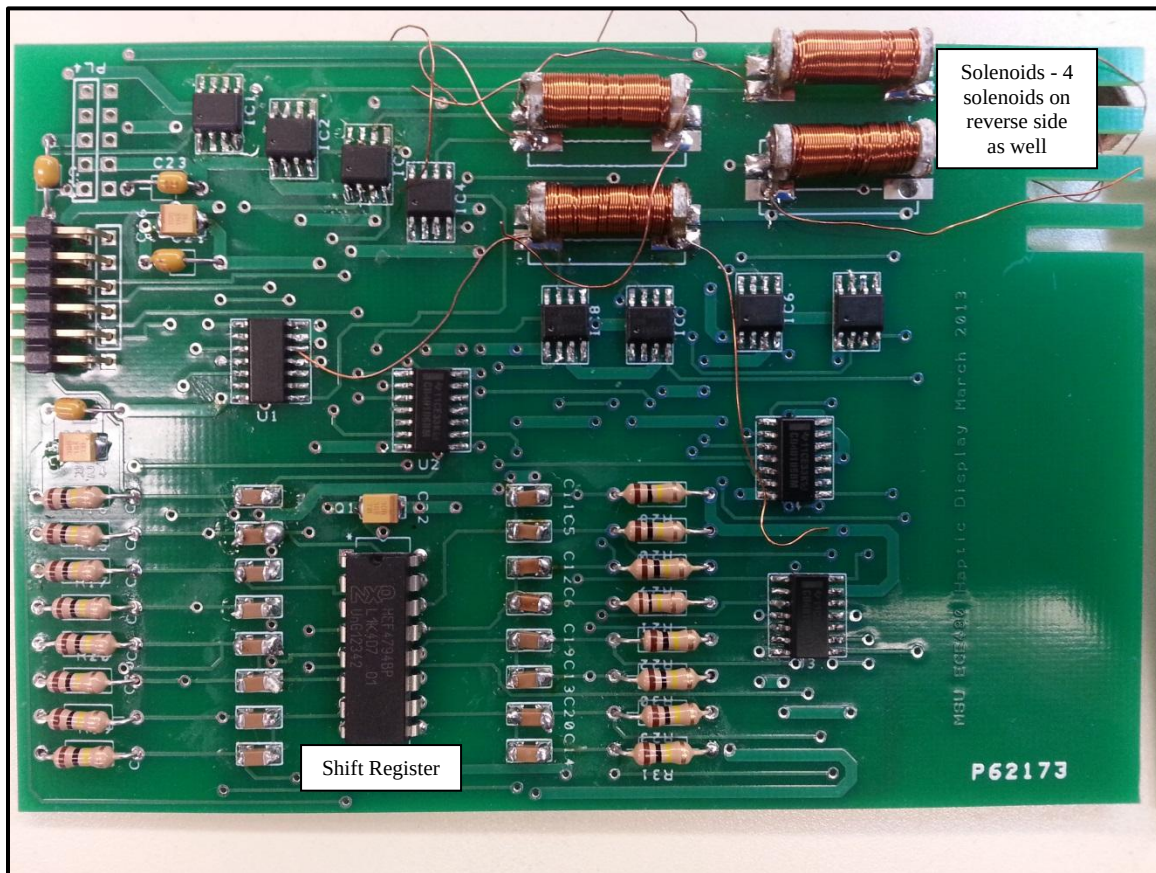


Figure 3.2.2b: Completed PCB

3.2.3 Integration of Neodymium Magnets

A critical component of the final product was neodymium magnets. Neodymium magnets are extremely high energy products that are ideal for compact designs. Each neodymium magnet is implemented in conjunction with a solenoid. By sending current through a solenoid, a magnetic field is generated that drives the magnet forward or backward. The horizontal force from the magnet, along with the wedge at the tip of each magnet, drives the pins up vertically with a strong enough force that it can withstand the pressure of the user's finger. At the end of each solenoid, rubber stoppers are attached to prevent the magnets from retracting too far back when the magnetic field reverses polarity.

3.2.4 HEF4794B Shift Register

The HEF4794B shift register was chosen for the design project. The HEF4794B was chosen due to its ability to use 12V logic from a 5V or 3.3V source, as well as its

Design Team 6 Technical Report

capability to store data. This shift register is an 8-stage shift-and-store register LED driver. It has a storage latch associated with each stage for strobing data from the serial input to parallel outputs. The data in each shift register stage is transferred to the storage register when the strobe (STR) input is set high. The data in the storage register will appear as an output whenever the output enable (EO) signal is also set high. The HEF4794B will buffer the serial data out on to the Arduino Leonardo board.

3.2.5 Pin Tower and Wedge Designs

The hardware components which hold the tactile pins and the wedge which lift the pins were created by using 3D Modeling software called NX 8.0. The models were designed in a modular way so that they could be easily modified and adjusted for future designs. The software allowed for very exact measurements and modeling, which was required in order for our design to function correctly.

The design of the tactile pin towers consists of twenty-four towers of various heights, which have a hollow well throughout. In the bottom portion of the front, there is an access port which allows for the pin to be inserted, and also allows the pin to be vertically adjusted by the wedges which are attached to our circuit-driven solenoids. The wedges were designed at different heights in order to allow for the solenoids to be stacked on circuit boards, rather than side-by-side. This created a configuration where we could place each pin closer together, which significantly improves the resolution of our display. Figure 3.2.5a is a 3D model of the tactile pin tower.

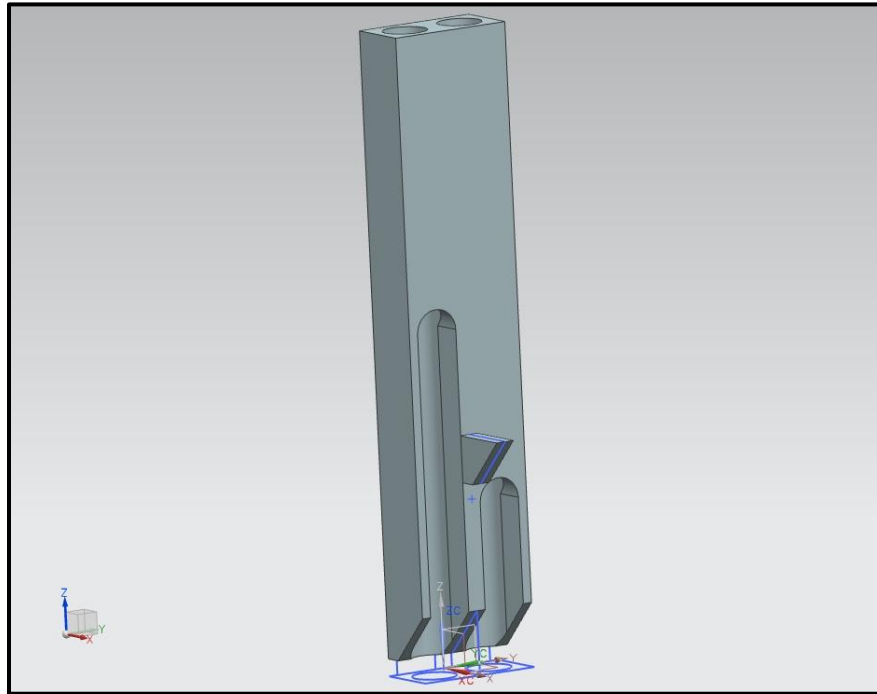


Figure 3.2.5a: 3D model of single tower

The towers were designed to be attached to a solid piece which will hold the circuit boards in place, and will also provide stability to our design. Figure 3.2.5b is the final design of the pin towers.

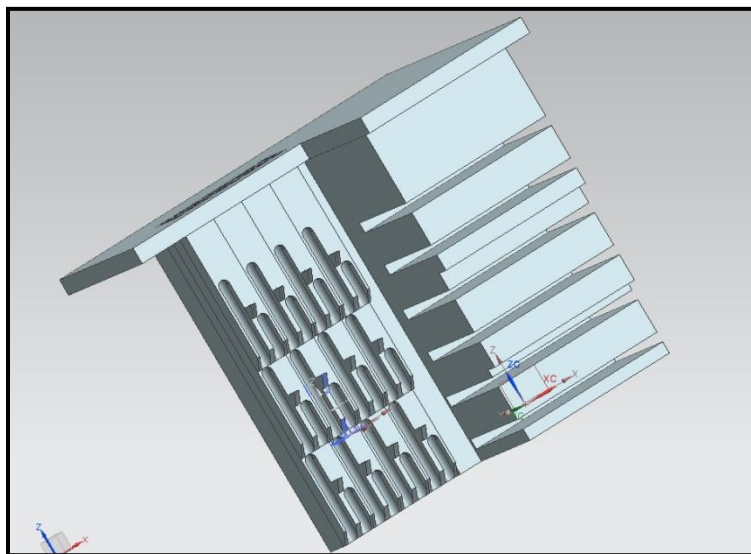


Figure 3.2.5b: 3D model of pin towers

Design Team 6 Technical Report

The wedges which vertically adjust the tactile pins were designed with walls on either side of the wedge so that the pins would not become dislodged. This will make our design much sturdier and less likely to malfunction. Additionally, the wedges were created with a hole in the back, so that the neodymium magnets could easily be inserted to drive the wedges. Figure 3.2.5c is a 3D model of the wedge.

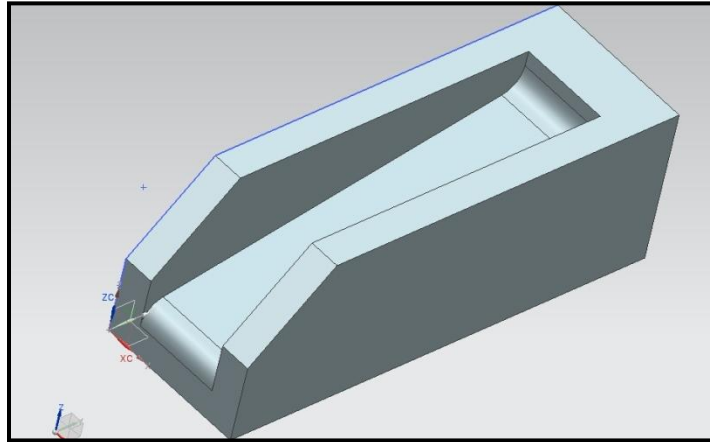


Figure 3.2.5c: 3D model of wedge

3.3 Software and Interface Design

3.3.1 PCB Creation and Design

Printed Circuit Boards (PCBs) allow for a professional way to integrate electronic components onto a single board. Prototype designs on bread boards can easily be transferred into a PCB design. PCBs eliminate wires and save considerable space by allowing components to be mounted directly onto the surface of the board. Because there are no physical wires, each PCB needs to be designed specifically for the purpose of the circuit with predetermined paths created by the designer. Once a circuit schematic is drawn up and ready for implementation, a user can select one of many available software programs to create the layout of the PCB.

This project uses a circuit that utilizes a shift register, differentiator, and push-pull MOSFETs. Each are described below briefly.

Design Team 6 Technical Report

Shift Register

Shift registers are simply flip-flops that share a clock. The output of each flip-flop is connected to the input of the succeeding flip-flop in the chain, resulting in a circuit that shifts the bit in the array by one.

Differentiator

A differentiator is a circuit that is designed so that the output of the circuit is approximately directly proportional to the rate of change of the input. This design uses a passive differentiator, which consists of resistors and capacitors only. A passive differentiator circuit is a four-terminal network which consists of two passive elements. It is a first-order high-pass filter.

MOSFET Push-Pull Amplifier

MOSFET Push-Pull Amplifiers use the N Channel MOSFET to amplify the positive parts of an input signal and the P Channel MOSFET to amplify the negative half.

The circuit for an individual solenoid utilizing all of these components is shown in Figure 3.3.1a.

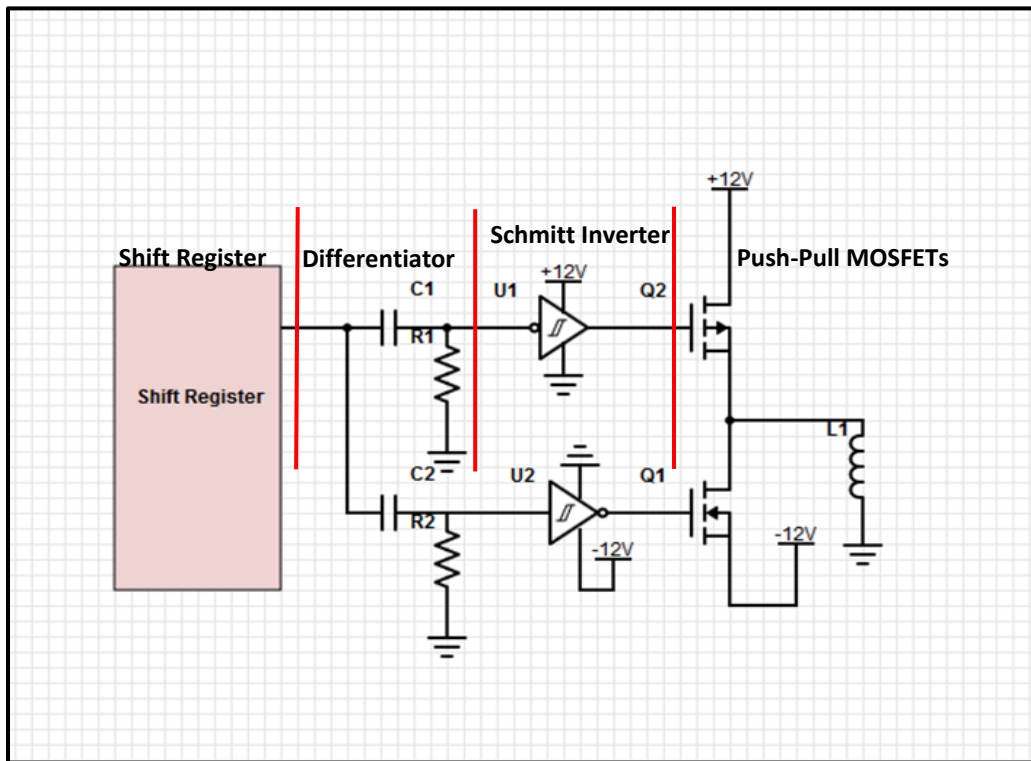


Figure 3.3.1a

The digital signal from the Leonardo Arduino board is sent to the shift register. The shift register has eight output ports which connect to one solenoid each. The signal from the shift register travels through the differentiator and then the push-pull MOSFETs. If the signal is a rising edge then the differentiator will create a positive pulse. This pulse will cause the MOSFET to “push” the wedge, driving the wedge forward and raising a pin. If the signal from the shift register is a falling edge, then the differentiator will create a negative pulse. This pulse will “pull” the wedge back and lower a pin.

Each PCB has eight solenoids, and so there are eight of the circuits on each board. The total circuitry for one PCB is shown in Figure 3.3.1b.

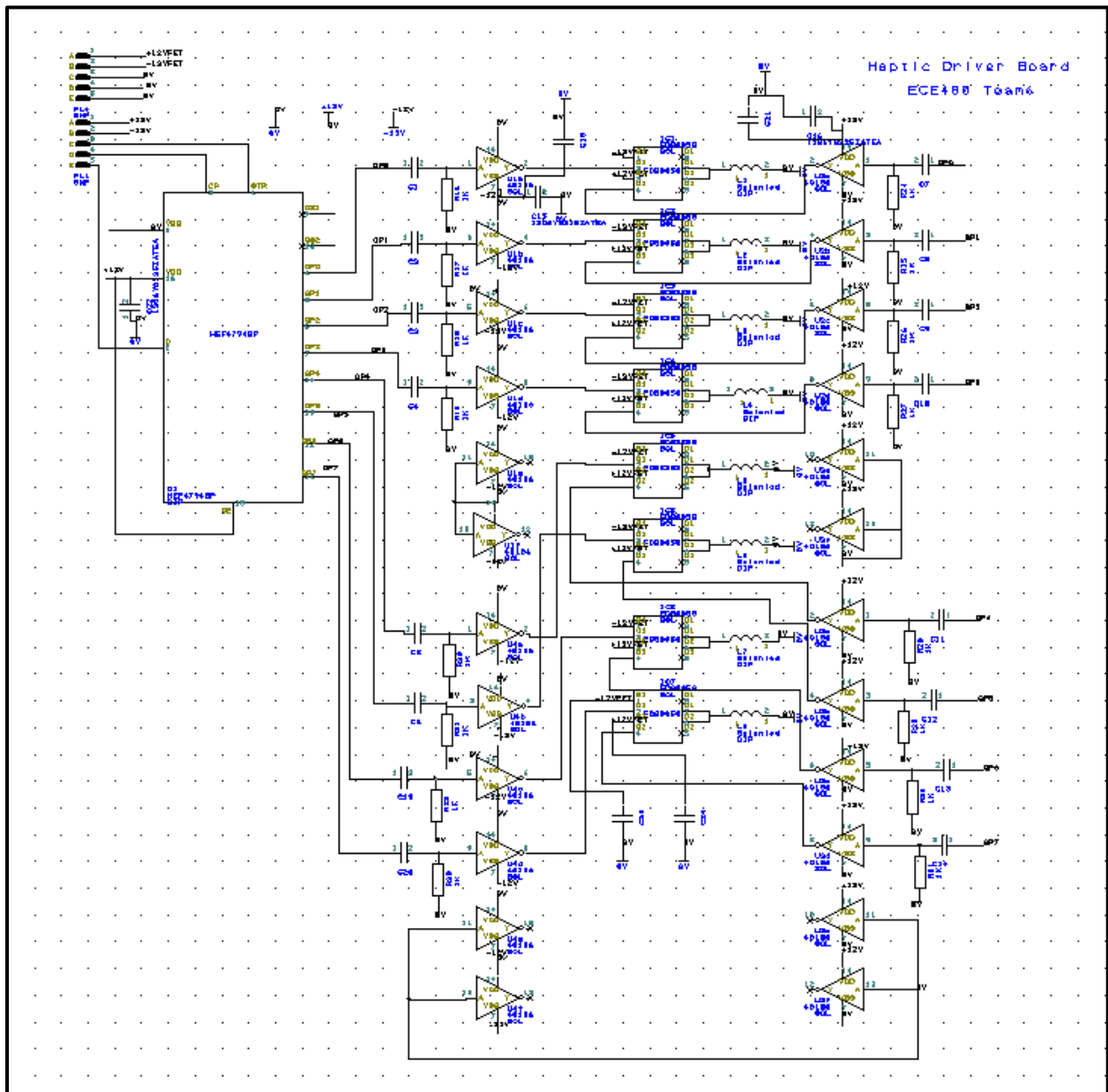


Figure 3.3.1b

3.3.2 Hex Level Shifter: MC14504B

The MC14504B is a hex non-inverting level shifter using CMOS technology. The level shifter will shift 12 V supply voltage to 5V CMOS logic. A control input allows interface from CMOS to CMOS at one logic level to another logic level. The MC14504B is used for Phase II HUI in order to down translate from high to low voltage (12V $\rightarrow$ 5V).

Design Team 6 Technical Report

3.3.3 Initial Shift Register and Arduino Testing

The Arduino code that was written to turn on and off eight test LEDs is named Shift_LEDs. The code is based on the timing diagram and the logic table from the datasheet of the HEF4794B. When the clock pin transitions from low to high, the shift register reads the state of the data pin. The data is shifted and stored in the storage register when the strobe input is high. The data in the storage register appears at the outputs whenever the output enable (EO) signal is high.

In order for the Arduino Lenoardo to communicate to the HEF4794B shift register, the input ports of the shift register need to be initialized and be created as outputs of the Arduino. To turn the test LEDs on and off, an array of 1s and 0s was established. Each term in the array represents a LED position. If the bit passed to the port is a 1, then the LED will turn on. If it is a zero, then the LED will turn off. Next, the pins must be initialized. This authorizes the data pin, strobe pin, clock pin, and output enable pin to be outputs. A function to pulse the clock was also written. The function will send a low, then high, then low pulse to the HEF4794. This will tell is to take a single bit (either a 1 or 0) from the array above. The HEF4794 looks for the front of the pulse when it goes from low to high. It then accepts the low or high input from the data pin into the shift register.

The main loop of Shift_LEDs steps through the values in the previously established array. The data is written to each output pin of the Arduino ($O_7 - O_0$) with either a 1 or 0 (high or low) with each pulse of the clock. An if-statement is created, so once the array is finished being iterated, the output enable pin is set low, to momentarily turn off the outputs. Then a high pulse is sent to the strobe input and will cause the HEF4794B to move the 1s and 0s from the shift register to the storage register. Once the clock pulses again, the outputs are turned back on and the strobe input is sent low. Figure 3.3.3 illustrates the circuit and code working.

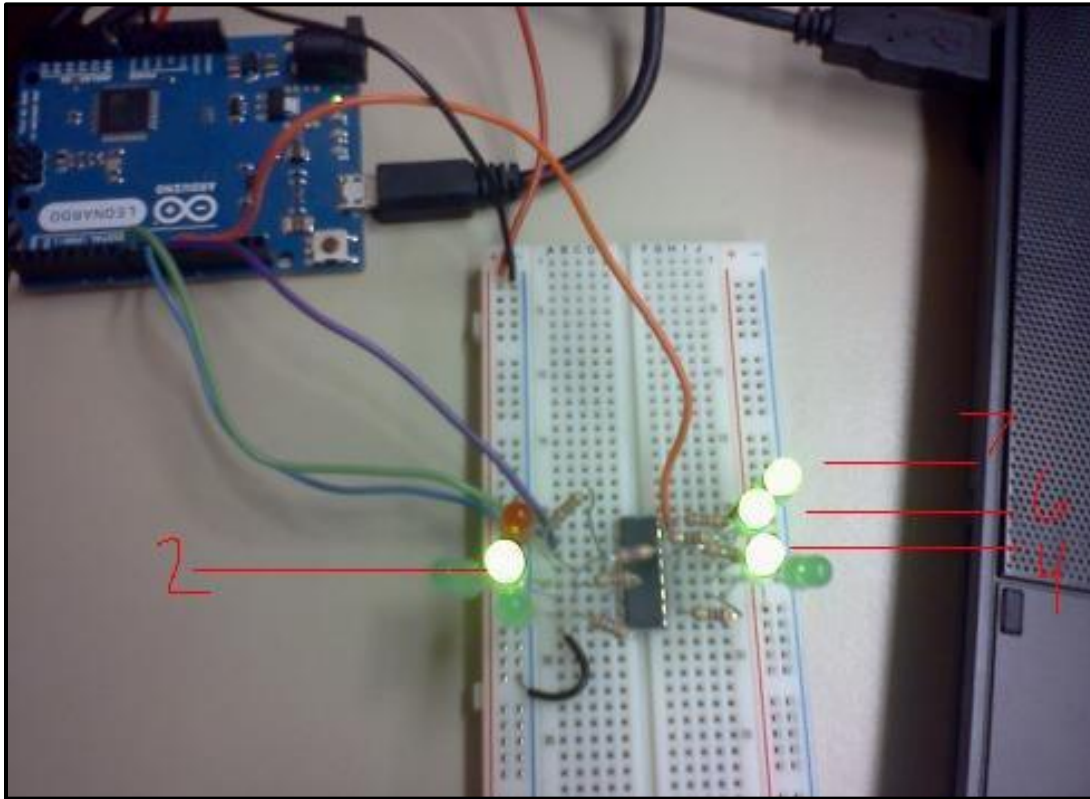


Figure 3.3.3: Circuit and code implementation. Outputs on: O_7 , O_6 , O_4 , and O_2 .

3.4 Software Implementation

3.4.1 Arduino Software Phase II

After successful trials with the initial Arduino software and eight LED test circuit, the software was altered to correspond with the project with its 24 pin display. The new software needed to program three HEF4794B shift registers, in order to power a total of 24 LEDs. For initial testing and debugging purposes, test LEDs were again used and represent the pins of the final Phase II product. The Phase II HUI successfully displayed three graphic images: a triangle, a line, and a rectangle. Further testing showed that it can easily display ordinary Braille. Integrated graphics and word capability is of great benefit for the customer, so this is an important Team 6 Phase II development. The Phase II HUI successfully displayed MSU, ECE, and 480 in Braille.

For the LEDs to correspond to the pins on the display, Team 6 had to view the order the solenoids on the PCBs are wired. For design ease the solenoid eight and seven on each

Design Team 6 Technical Report

board are switched when comparing to the pin display on the product. Figure 3.4.1a displays this. When writing the software it was important to remember that the arrays written in the software are in a reverse order from the arrays established on the device. For instance on the device the far left pin is known at position 0, but when coding the array the first element must begin with position 7. Figure 3.4.1a also displays this process.

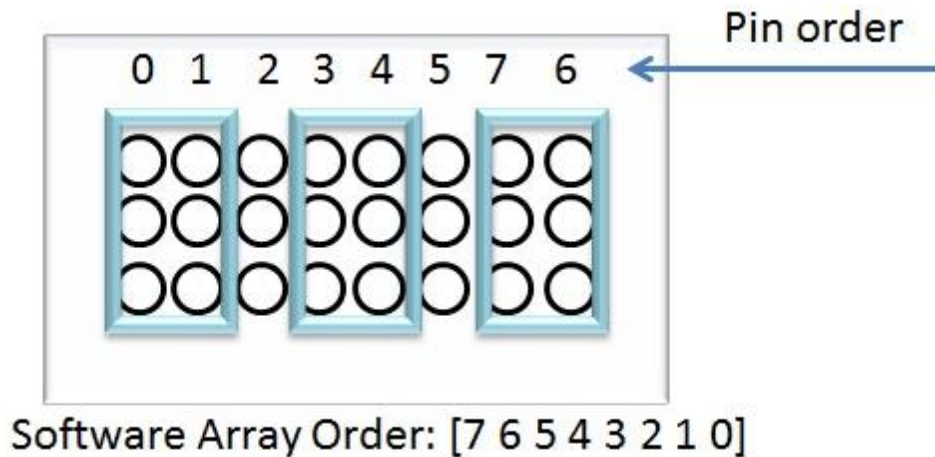


Figure 3.4.1a: Pin order in comparison to array order within Arduino software

Since the software was tested with a sample circuit of LEDs on a proto-board, the LEDs were not arranged in a grid similar to pin-out display of Phase II. Figure 3.4.1b is a schematic what the Phase II HUI would look like when displaying MSU.

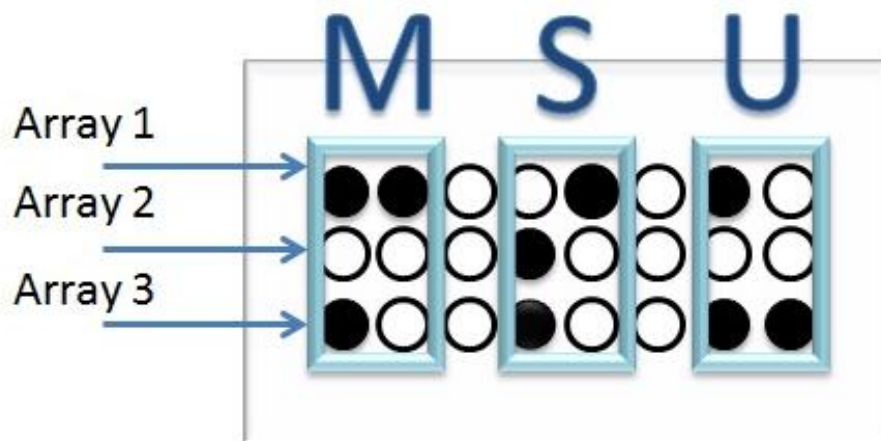


Figure 3.4.1b: MSU in Braille – pin display

Design Team 6 Technical Report

Figure 3.4.1c is the output of MSU on the sample circuit. Since each array is a row, it is important to recognize that each shift register is not a Braille character. One Braille character is a combination of one array from each shift register.

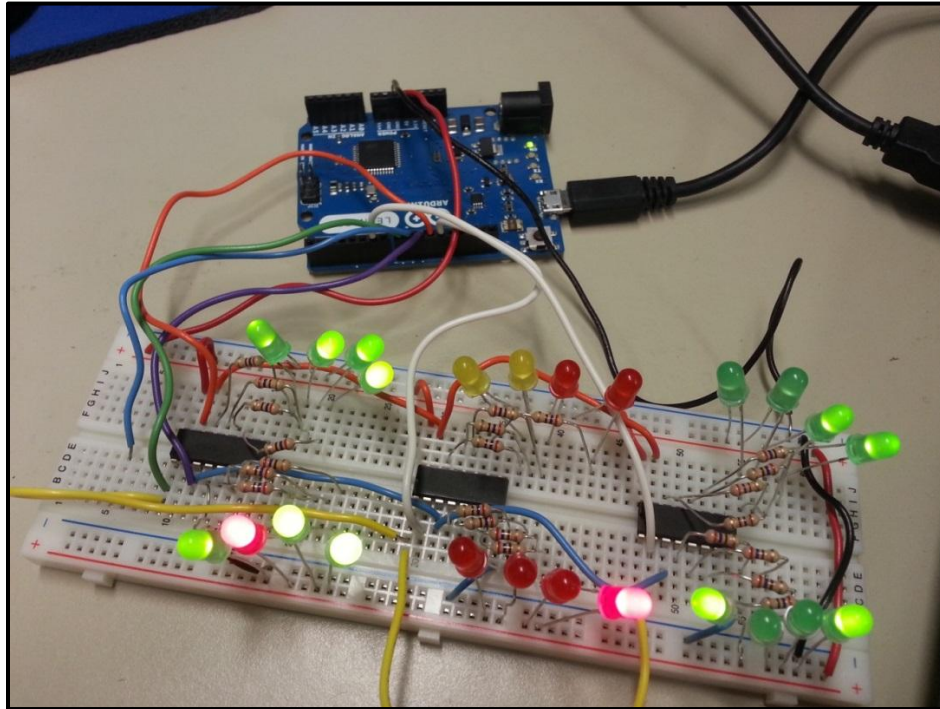


Figure 3.4.1b: MSU in Braille

The Arduino software also generates a clock function that is utilized when reading data (each array) and writing the data to the outputs of the shift registers. The Arduino software is interfaced with a graphical user interface created by Team 6 for the Phase II HUI.

3.4.2 Graphical User Interface from Phase I

Phase I of the Haptic User Interface concluded at the end of the Fall Semester 2012. The team did an outstanding job of creating a Graphical User Interface (GUI) to be implemented with the project for future designs. The interface was coded in the computer language C#. It featured a drop-down menu for selecting the ports to be used, a menu for uploading images to be converted to grayscale and used for implementation, and a “Training Mode” feature that tests the image and ensures the proper pins should be

Design Team 6 Technical Report

raised at the appropriate places. Figure 3.4.2a shows the GUI in its basic form upon initial opening.

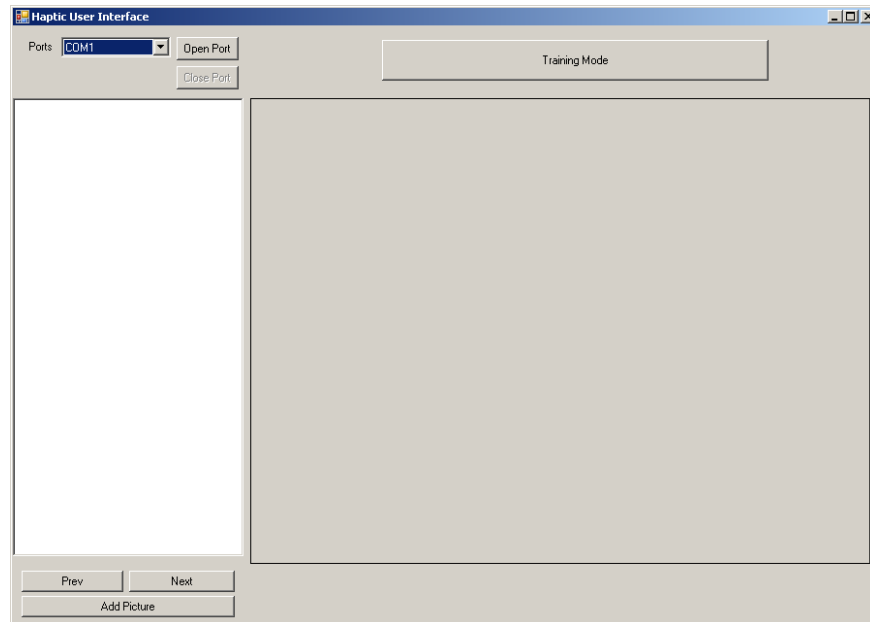


Figure 3.4.2a: Phase I GUI

The GUI is a great tool to help Team 6’s intended customer. However, due to time constraints and a changing of goals outlined earlier in the report, Team 6 did not utilize this GUI for final implementation. This is not to say it is no longer needed, but that it will have to be implemented in a future phase of the project. Although it is not used for this project, the following summary of the GUI provides a breakdown of each part and how it can be used for future implementation:

Ports:

The top-left portion of the GUI features a drop-down menu called “Ports”. What this does is establish a connection between the GUI and a device. The GUI automatically determines if a port is available and selects the top option by default. By selecting the down arrow you can select more ports if they are available. Clicking the “Open Port” button opens the port for communication to the device through the selected option. Figure 3.4.2b demonstrates through some

Design Team 6 Technical Report

code how this is accomplished and Figure 3.4.2c demonstrates the use of the drop-down menu.

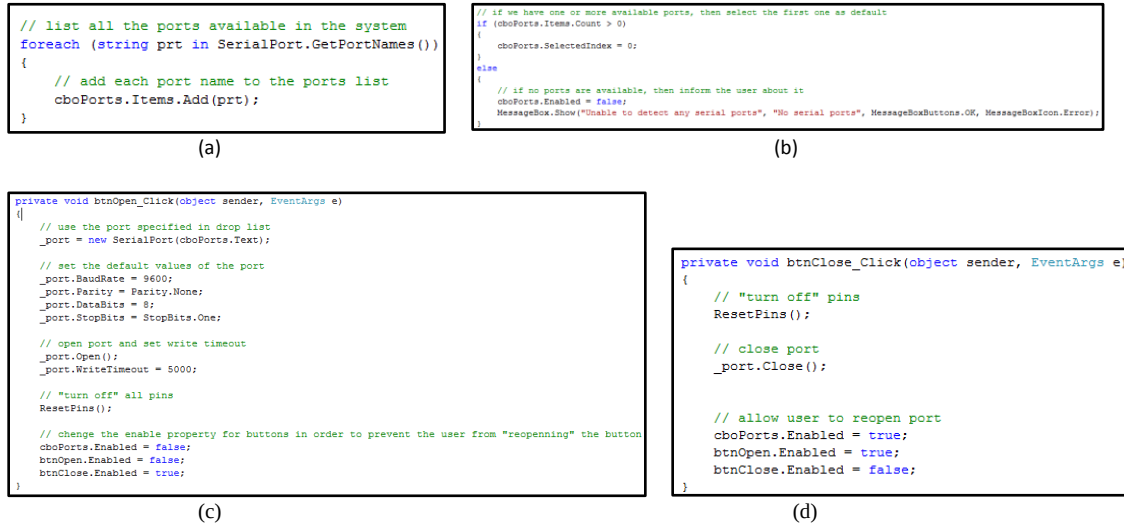


Figure 3.4.2b : (a) Shows the user every port that is available; (b) Selects the first port available as default if multiple ports exist, and outputs an error message if no ports are detected; (c) Upon clicking the “Open Port” button this code is executed; (d) Upon clicking the “Close Ports” button this code is executed

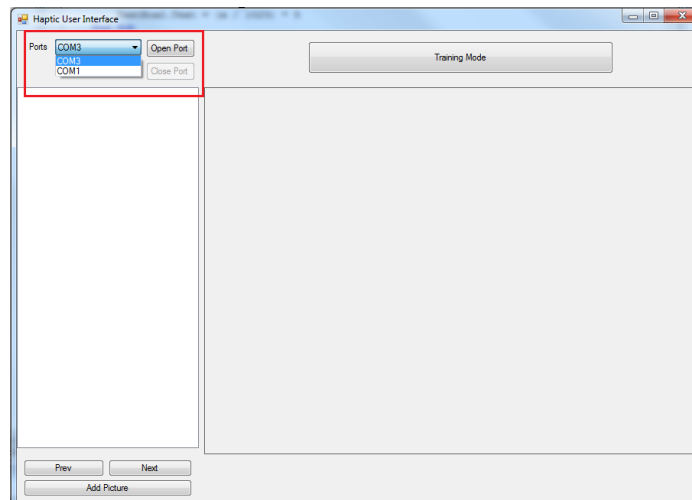


Figure 3.4.2c

Uploading Images:

In the bottom-left of the GUI, there is a button called “Add Picture”. This button is programmed to add images and, through a different portion of the code, convert that image to grayscale. Upon clicking the button a display window pops up prompting the user to select an image of pre-determined formats. Once the image

Design Team 6 Technical Report

has been selected, converted to grayscale, and loaded, it will display in the box on the right. Figure 3.4.2d shows through the code how this is accomplished and Figure 3.4.2e shows the pop up window that prompts the user to select an image.

```
private void btnAdd_Click(object sender, EventArgs e)
{
    // display open file dialog and set its defaults
    OpenFileDialog fd = new OpenFileDialog();
    fd.InitialDirectory = "%USERPROFILE%";
    fd.Filter = "Image Files (*.jpg;*.png;*.bmp;*.tiff)|*.jpg;*.png;*.bmp;*.tiff";

    // if the user selected a file
    if (fd.ShowDialog() == DialogResult.OK && fd.FileName != null)
    {
        // attempt add file to list
        try
        {
            // if the user did not add the file before
            if (!lstImages.Items.Contains(fd.FileName))
            {
                // add file to list and select it
                lstImages.Items.Add(fd.FileName);
                lstImages.SelectedIndex = lstImages.Items.Count - 1;
            }
            // otherwise, if the user did add the file before
            else
            {
                // just select the file
                lstImages.SelectedIndex = lstImages.Items.IndexOf(fd.FileName);
            }
        }
        catch (Exception ex)
        {
            // display error to user
            MessageBox.Show(ex.Message, "Unable to open file", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Figure 3.4.2d : Upon clicking the button the user is prompted to select a file that only ends in the appropriate file format. Following that, the program attempts to add the file to the list. If it is successful, the image is loaded in the GUI. If it is not successful, an error message displays.

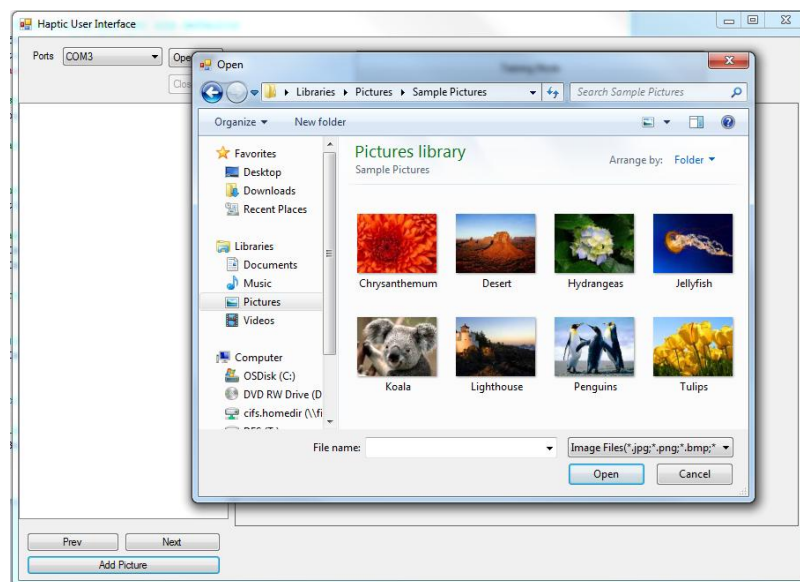


Figure 3.4.2e : This menu pops up after the “Add Picture” button is pressed

Design Team 6 Technical Report

Upon adding the image to the list and converting the image to grayscale, more images can be added as needed. Doing so creates a list in the white box on the left that you can navigate through by clicking the “Prev” and “Next” buttons. This is demonstrated in Figures 3.4.2f and 3.4.2g.

```
private void btnNext_Click(object sender, EventArgs e)
{
    // if we reached the end of the list, then loop around
    if (lstImages.SelectedIndex == lstImages.Items.Count - 1)
        lstImages.SelectedIndex = 0;
    // otherwise, select the next item
    else
        lstImages.SelectedIndex++;
}
private void btnPrev_Click(object sender, EventArgs e)
{
    // if we reached the beginning of the list, then loop around
    if (lstImages.SelectedIndex == 0)
        lstImages.SelectedIndex = lstImages.Items.Count - 1;
    // otherwise, select previous item
    else
        lstImages.SelectedIndex--;
}
```

Figure 3.4.2f

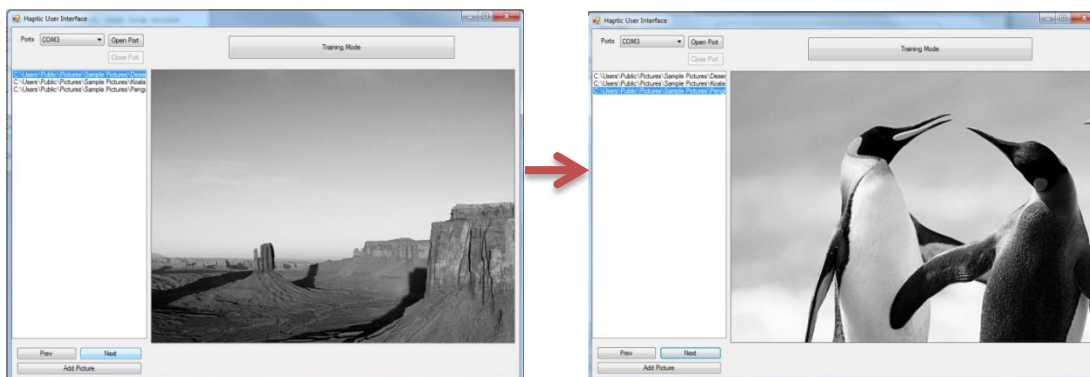


Figure 3.4.2g : After loading multiple images a list appears on the left. Clicking “Next” or “Prev” cycles through these images

Pin Display/Mouse:

Phase II HUI uses the mouse to “feel” an image. When the pointer of a mouse hovers of a dark portion of an image, the GUI sends a signal to a device to raise pins to signify that this is part of the image that should be felt by the user. The “Training Mode” button in the top-middle of the GUI provides a great example of what this should look like on a physical device. Upon clicking the button, a new window pops up with rows of blocks that simulate pins. When the mouse cursor

Design Team 6 Technical Report

hovers over the image and moves around the image, the pins change color representing which pins would be raised on a physical device. Figures 3.4.2h and 3.4.2i demonstrate this through code and visuals, respectively.

```
private void picDisplay_MouseMove(object sender, MouseEventArgs e)
{
    // make sure there is an image loaded in the picture box
    if (picDisplay.Image != null)
    {
        int grayIdx = 0; // index of gray values array
        Bitmap bmp = new Bitmap(picDisplay.Image); // copy the grayscale picture to memory

        int xPixel = 0; // stores the x location of the pixel
        int yPixel = 0; // stores the y location of the pixel
```

```
...
// loop through each module row
for (int modRowCtr = 0; modRowCtr < MODULES_DOWN; modRowCtr++)
{
    // for each module row, loop through each module column
    for (int modColCtr = 0; modColCtr < MODULES_ACROSS; modColCtr++)
    {
        // reset the gray index to which will send out "new" data
        grayIdx = 0;

        // for accessed module, loop through each of its pin rows
        for (int pinRowCtr = 0; pinRowCtr < RPM; pinRowCtr++)
        {
            // for each pin row, loop through each pin column
            for (int pinColCtr = 0; pinColCtr < CPM; pinColCtr++)
            {
                // attempt to read pixel at mouse location
                try
                {
                    // figure out pixel location depending on which module we are reading
                    // and which pin in that module we are reading
                    xPixel = e.X + pinColCtr + (modColCtr * CPM);
                    yPixel = e.Y + pinRowCtr + (modRowCtr * RPM);

                    // store the gray value of that pixel (R = G = B)
                    grayValues[grayIdx] = bmp.GetPixel(xPixel, yPixel).R;
                }
            }
        }
    }
}
```

(a)

```
// sends an "off" signal to all pins
private void ResetPins()
{
    // fill data with "white" values (or off)
    for (int i = 0; i < RPM * CPM; i++)
    {
        grayValues[i] = 0xFF;
    }
}
```

(b)

```
private void picDisplay_MouseLeave(object sender, EventArgs e)
{
    // reset pins when mouse leaves picture
    ResetPins();
}
```

(c)

Figure 3.4.2h : (a) Ensures an image is loaded, and then assigns pixel values to the image. It loops through each row and column and attempts to read the pixel location. (b) Sends a "white" value to the pins, which turns them off (c) If the mouse leaves the display window it resets the pins, turning them off

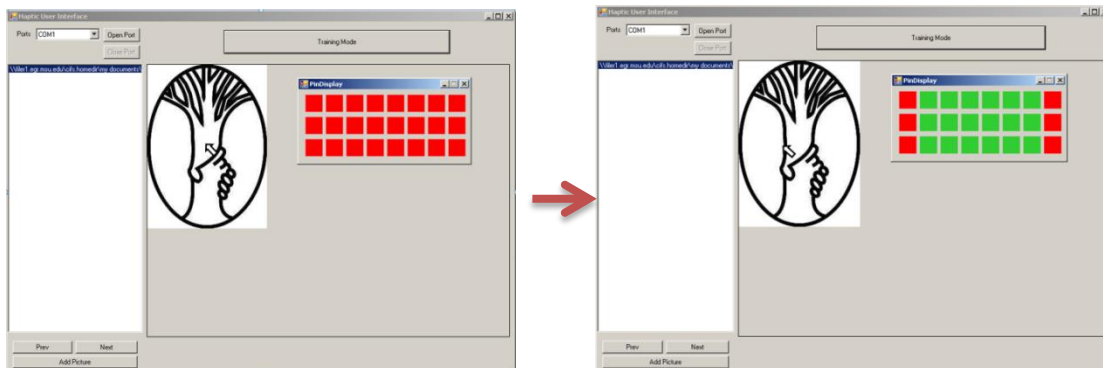


Figure 3.4.2i : On the left, the mouse is in all white space so the Pin Display shows all red pins, or "down" pins. On the right, the mouse hovers above the black outline of the tree so the Pin Display shows a portion of green pins, or "up" pins.

Design Team 6 Technical Report

3.4.3 Graphical User Interface - Phase II

The Graphical User Interface (GUI) that was developed to interact with the Haptic User Interface Phase II was primarily designed to establish a connection with the device and to demonstrate the capability of the device. The program is capable of scanning the USB ports on the host computer to determine in which port the device is currently attached, and then will establish a communication link with the device. Once the connection is established, the program is capable of sending commands to the device which direct it to display various Braille characters or various shapes. The commands are chosen by the user, and are sent by utilizing the GUI on the computer to click on the corresponding button. Figure 3.4.3 displays the Phase II GUI.

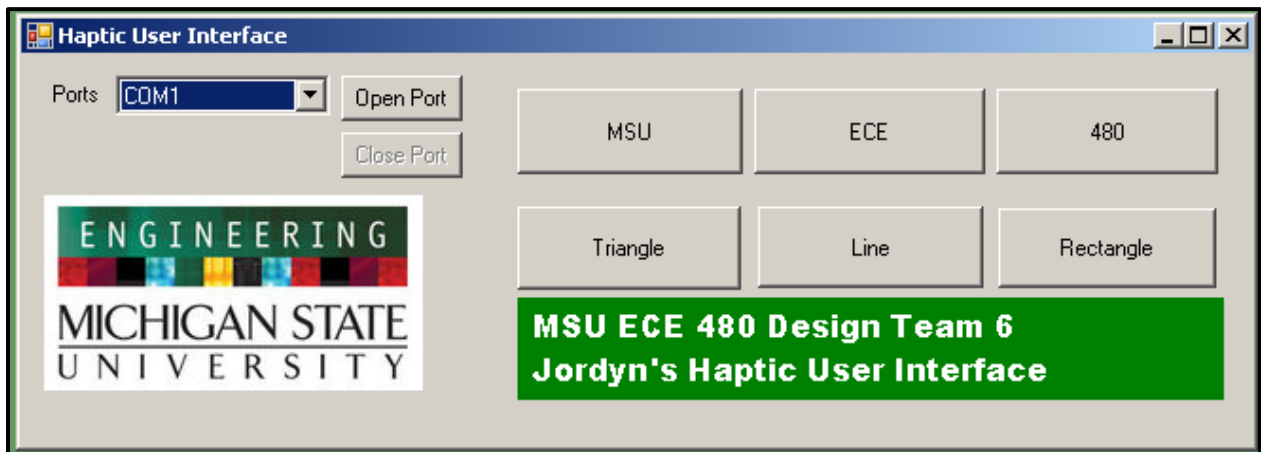


Figure 3.4.3: Phase II GUI

The program is written in the Microsoft Visual Studio 2008 environment, and the code is written in the C# programming language. Additionally, the program utilizes the Microsoft .NET framework for serial communications. An example of the code to detect open USB ports for the graphical user interface is shown in the following figure:

```
public frmMain()
{
    InitializeComponent();
}

private void frmMain_Load(object sender, EventArgs e)
{
    // list all the ports available in the system
    foreach (string prt in SerialPort.GetPortNames())
```

Design Team 6 Technical Report

```
{
    // add each port name to the ports list
    cboPorts.Items.Add(prt);
}

// if we have one or more available ports, then select the
first one as default
if (cboPorts.Items.Count > 0)
{
    cboPorts.SelectedIndex = 0;
}
else
{
    // if no ports are available, then inform the user
about it
    cboPorts.Enabled = false;
    MessageBox.Show("Unable to detect any serial ports",
"No serial ports", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
```

The program communicates with the Arduino Leonardo, which is firmware-programmed by using the Arduino IDE software. The source code which was written for use with the software was written in the C programming language. The software includes references to which ports of the microcontroller are attached, and also direct the Arduino Leonardo to communicate with the shift registers in order to display the correct characters or shapes.

Chapter 4: Final Product Evaluation

4.1 Phase II HUI

The Phase II HUI consists of an eight by three matrix, resulting in 24 pins. Each pin is driven by a wedge that is powered by a solenoid. To add to the product's versatility, Mr. Blosser suggested the final product to display both Braille and graphic images. This makes the Phase II HUI like no other product on the market, enhancing the accessibility of visual impaired individuals.

The Phase II HUI's Arduino code articulates the arrays needed to display MSU, ECE, 480, a triangle, a line, and a rectangle. This code in conjunction with the Phase II GUI

Design Team 6 Technical Report

represents the software aspect of Phase II. When a specific button is selected on the GUI, for instance MSU, the GUI compiles the code needed to output MSU onto the device. The Phase II HUI's hardware components consist of the pin tower to house the 24 pins, the solenoids to drive each wedge, and the PCBs which contain the circuitry needed to power the solenoids.

To power the Phase II HUI, the sponsor generously donated two computer power supplies from his lab. The two supplies are utilized to send a +12V and -12V to power the solenoids to either drive the pin up or retract it.

Chapter 5: Final Cost, Schedule, and Conclusions

5.1 Final Cost

Team 6 was allocated with the budget of \$500 for the semester. In order to build the Phase II HUI many components were needed. To ensure that Team 6 would remain productive and used designated project time efficiently, extra components were ordered. Table 5.1a outlines the total expenses made by Team 6.

Quantity	Item	Cost
	31 Gauge Copper Wire (3140ft)	\$ 110.90
15	3D Printing - Ferrules	\$ 7.38
1	3D Printing - Tower	\$ 70.87
24	3D Printing - Wedges	\$ 18.00
1	Arduino Leonardo Microcontroller	\$ 21.95
12	Capacitors (thru-hole)	\$ -
	Casing	\$ -
100	Ceramic Capacitors (surface mount)	\$ 8.30
3	HEF4794B Shift Register	\$ 4.80
6	Hex Shifter	\$ 4.80
	Hypodermic Tubing	\$ -
30	MOSFET	\$ 22.20
50	Neodymium Magnets	\$ 25.50
2	Power Supplies	\$ -
4	Printed Circuit Boards	\$ 132.00
48	Resistors	\$ -
24	Schmitt Trigger	\$ 8.90
12	Tantalum Capacitors (surface mount)	\$ -

Design Team 6 Technical Report

	Grand Total	\$ 446.80

Table 5.1a: Total budget spent

Since not all components were used to create Phase II HUI, Table 5.1b displays the total costs to build the final design.

Quantity	Item	Cost
1850 ft.	31 Gauge Copper Wire	\$ 65.31
12	3D Printing - Ferrules	\$ 5.94
1	3D Printing - Tower	\$ 70.87
24	3D Printing - Wedges	\$ 18.00
1	Arduino Leonardo Microcontroller	\$ 21.95
12	Capacitors (thru-hole)	\$ -
	Casing	\$ -
48	Ceramic Capacitors (surface mount)	\$ 3.98
3	HEF4794B Shift Register	\$ 4.80
1	Hex Shifter	\$ 0.80
	Hypodermic Tubing	\$ -
24	MOSFET	\$ 17.76
24	Neodymium Magnets	\$ 12.24
2	Power Supplies	\$ -
3	Printed Circuit Boards	\$ 99.00
48	Resistors	\$ -
12	Schmitt Trigger	\$ 4.45
12	Tantalum Capacitors (surface mount)	\$ -
	Grand Total	\$ 325.10

Table 5.1b: Cost to build one Phase II HUI

With a total cost of \$325.10, the Phase II HUI is 93.5 % - 96.75% less than similar devices on the market (\$5,000 - \$10,000). Since the Phase II HUI can be utilized as a Braille and graphic image display; it is like no other product on the market.

5.2 Schedule

To begin Team 6's project, each member researched last semester's design of a Haptic User Interface. After meeting Jordyn, a visually impaired computer science sophomore at MSU, Team 6 began to understand what needed to be improved to meet all expectations. After weekly meetings with Mr. Blosser, three different prototypes were tested. The first two prototypes had some success with one prototype utilizing magnetic latching and the

Design Team 6 Technical Report

other prototype added solenoids to drive the pins up or down due to a pulse that was sent to change polarity. Both prototypes worked but still needed improvements to fulfill the team objectives. From the two previous prototypes, a third prototype was developed near the beginning of March. This prototype utilized a mechanical device in order to wedge the pins in place. Prototype III worked efficiently and met all primary and secondary requirements bestowed upon Team 6. Following the schedule specified in the Gantt chart, Team 6 was able to remain on schedule and accomplish the objectives set at the beginning of the semester.

5.3 Future Enhancements

Phase II represents an important advance with more phases to come. The purpose of this project was to build upon Phase I and create a product worth pursuing for future teams. While Phase I had many successes, especially being the first prototype of its kind, Phase I's largest success was creating a GUI for images to be "felt" by visually impaired individuals. Other successes included utilizing solenoids and enabling pins to be raised and lowered.

For Phase II, Team 6 focused most of their effort on stabilizing the rapidly vibrating pins from Phase I and improving the resolution of the images. By creating several prototypes and identifying and resolving issues, Team 6 has successfully managed to stabilize the pins. Furthermore, by reducing the space between each pin, Team 6 has dramatically increased the resolution to be felt. Lastly, the device has an added functionality to double as a Braille device, something Phase I did not have.

Due to time constraints and the vast difference in hardware from Phase I's HUI, Team 6 was unable to implement the GUI from Phase I. This GUI allows the user to input a graphic image and display on the device. Since Team 6's hardware design is different from Phase I, Team 6 cannot just simply connect the device to execute the GUI without several changes. If Team 6 had a few more weeks to alter and debug the software for

Design Team 6 Technical Report

Phase I's GUI, it is believed that the software would have been able to be implemented in Phase II HUI.

For future stages of this device, teams should have two primary goals:

1. Increase resolution further by implementing more PCBs and solenoids
2. Implement a fully functional GUI for images to be uploaded and “felt”
3. Design and build more compact power supplies

To accomplish the first goal, future teams should utilize the circuit created by Team 6 this semester. Ordering more PCBs and components, and creating more solenoids, will enable future teams to increase the overall pin count from a total of 24 pins to 48, 64, or higher. This will quickly and efficiently increase the resolution of the device. The most difficult part of increasing the resolution at that point is soldering the components and winding the solenoids. Table 5.3 displays how much a future team would spend if the team built a HUI with 64 pins.

Quantity	Item	Cost
4934 ft.	30 Gauge Copper Wire (3140ft)	\$ 174.16
32	3D Printing – Ferrules	\$ 15.74
1	3D Printing – Tower	\$ 70.87
64	3D Printing – Wedges	\$ 48.00
1	Arduino Leonardo Microcontroller	\$ 21.95
32	Capacitors (thru-hole)	\$ -
1	Casing	\$ -
128	Ceramic Capacitors (surface mount)	\$ 10.62
8	HEF4794B Shift Register	\$ 12.80
1	Hex Shifter	\$ 0.80
	Hypodermic Tubing	\$ -
64	MOSFET	\$ 47.36
64	Neodymium Magnets	\$ 32.64
2	Power Supplies	\$ -
8	Printed Circuit Boards	\$ 264.00
128	Resistors	\$ -
32	Schmitt Trigger	\$ 11.87
24	Tantalum Capacitors (surface mount)	\$ -
	Grand Total	\$ 710.81

Table 5.3: Proposed budget of 64 pin HUI

Design Team 6 Technical Report

For the second goal, future teams should take the GUI from Phase I, which was outlined in section 3.4.2 and fully implement with the Arduino software. By successfully creating a connection between the software, future teams will be able to upload any image, have it converted to grayscale, and then have it shown on the device. Also, with an increased resolution, the device will more likely resemble the image on the screen.

For the third goal, future teams would need to design power supplies that would be able to continue to drive the existing pins or additional pins (i.e. resolution increased). Future teams could also look into purchasing smaller power supplies that would power the device, but it would be less expensive to design and build their own.

As more time and effort is put into the GUI and implementing it with the device, increasing resolution by adding more PCBs and solenoids, and making power supplies more compact the device will become ready for mass distribution.

5.4 Conclusion

Design Team 6 was presented with the objective of improving upon the Phase I HUI, a tactile display that would allow visually impaired individuals to feel an image. To complete this task, Team 6 divided the objectives into two primary and secondary requirements after consulting the sponsor and customers. Primary requirements included a higher resolution and elimination of the vibrating pins. Secondary requirements were to reduce the amount of heat the device dissipates, decrease the size of the pins to coincide with Braille dot size, and decrease the stroke of the pin. With the objectives determined, Team 6 set out to design a device that would fulfill all requirements and be a large improvement from Phase I. The team accomplished this objective using innovative and cost effective technology, such as solenoids. By utilizing the solenoids' electromagnetic forces to drive a 3x8 matrix of pins, Team 6 was able to build the Phase II HUI. To eliminate the vibration of the pins, Team 6 implemented a mechanical device to lock pins after they were driven up. The mechanical device was a wedge that's end was attached to a neodymium magnet. The neodymium magnet then slides into the solenoid. By sending

Design Team 6 Technical Report

current through the solenoids, the wedges were then driven forward. Once the wedges were driven forward, the pins were lifted and locked into place.

To complement the innovative hardware design, Team 6 design and programmed a GUI to offer six selections to the user. Once the selection was chosen, the GUI compiled the software written for the Arduino Leonardo microcontroller to write to the shift registers' outputs. The output displayed of the Phase II HUI depends on the button selected by the user.

This versatile device can be easily enhanced and is built to be expanded upon. By adding more pins the resolution can vastly be improved. Additional software can also be added to make the device display images that are not pre-determined. Design Team 6 succeeded in producing a Haptic User Interface with better resolution, no vibration, less heat dispersion, and is consistent with the size of Braille dots. Team 6 was able to produce this product for the price of \$325.10. This cost is significantly lower than other devices on the market that are upwards of \$10,000. If the product were to go to mass production, it could easily be available to most visually impaired individuals. This achievement is the most important accomplishment of Design Team 6.

Appendix 1: Design Team Composition and Responsibilities

LeRonn Wilson: Team Manager

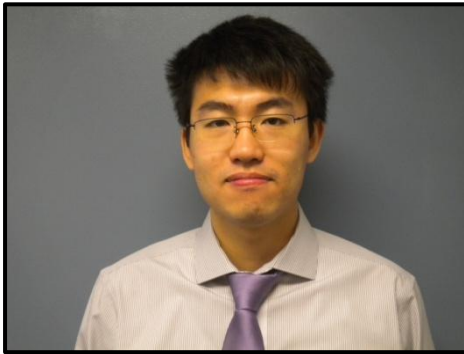


Mr. Wilson had a solid all-around understanding of each major component of the prototype and how phase II was progressing. With Mr. Wilson frequent presence in the lab, he had a supporting role in many of the design aspects, including designing and creating the solenoids, soldering the components to the PCB, and performed testing on the boards. Also, with his experience in mechanical design, Mr. Wilson worked closely with the team's sponsor in developing a new haptic user interface design. This paved the way for the software group to have a prototype for testing.

Design Team 6 Technical Report

After the team had a conceptual design, Mr. Wilson was actively involved in the implementation of the solenoids. Mr. Wilson technical contributions were building the spools to wind the copper wire coils and wound the spools with the coils using a lathe machine. Also, he assisted in inserting the neodymium magnets in the wedges to help pins drive up or down. Once the solenoids were created, Mr. Wilson assisted with soldering the solenoids on the PCB and did testing to make sure each solenoid worked efficiently. After testing, the team observed that the pins will have a lot of force going forward and backward. The magnet will move forward in the solenoid so the pins can exhibit a Braille display but Mr. Wilson added rubber stoppers at the back of each solenoid to prevent the magnet from shooting to far backward which would cause a safety hazard and some pins not to work.

Bin Tian: Documentation Preparation



Mr. Tian's technical tasks in the project mainly focus on design research and design implementation. At the beginning of the project, he involved in designing early prototypes of mechanical displays. Based on the characteristics of haptic interface display, he came up with several conceptual designs. He researched into solenoid drivers by building passive differentiators and push-pull current amplifier. Mr. Tian used TI

TINA software to simulate and optimize circuit performance. With simulation results, he involved in building proto-boards at the early stage and testing circuitry designs. In the PCB design process, he used PCB Artist, a software program that creates circuitry schematics and layouts to be used to fabricated PCBs. Meanwhile, he did troubleshooting of fabricated two layer PCBs. His other contribution was helping the group wound solenoids and programming the Arduino microcontroller board.

Eric Bell: Webmaster



Mr. Bell's responsibilities included the design of hardware components used within the device, and he also assisted with tasks such as soldering of circuit boards, was instrumental in the programming of the graphical user interface for the device, and additionally was responsible for all design and upkeep of the team web site. Mr. Bell spent a significant amount of time designing the hardware components that allow for the pins to be raised and lowered via the circuit-driven solenoids,

Design Team 6 Technical Report

and also designed the other small components used such as the ramp-like components which drive the pins. These were all created using the NX 8.0 software package, and were molded out of plastic using 3D printing technology. The design of the hardware components required significant thought and planning with regards to ensuring that all components were sized correctly. The design also needed to be designed in conjunction with the spacing of components of components on the PCB board in order to ensure that everything worked correctly.

Additionally, Mr. Bell utilized Microsoft Visual Studio software to write source code using the C# programming language in order to develop a graphical user interface to interact with the Arduino Leonardo Microcontroller. The software utilizes the Microsoft .NET framework in order to create a USB-Serial connection to the Arduino Leonardo, after which it simply sends certain commands to the microcontroller to allow it to react accordingly. Mr. Bell also aided in the completion of the circuit boards by spending time soldering components on to the PCB boards. The team web site was also designed and programmed by Mr. Bell. He also was responsible for the upkeep of the web site. The web site consists of a number of pages which include general background and information about the project, project updates, important documents, photos, and team information.

Kristen Kirchhoff: Presentation Preparation



Ms. Kirchhoff was tasked the position of writing the Arduino software, helping create the solenoids that drive the pins, and assisted with the soldering of the components on the printed circuit boards. By being involved in software and hardware aspects of the project, Ms. Kirchhoff has a great understanding of how the Phase II HUI works. In order for the Haptic User Interface to communicate with the computer, the Arduino Leonardo microcontroller is needed to buffer serial data. The

Arduino software programmed the shift registers used in the device. The software initializes the inputs of the shift register to outputs on the Arduino board. The outputs on the shift register are the pins and they are either sent a one or zero to drive the pin up or retract it. A clock function also needed to be written in order to read data and then write data to the outputs. The final Arduino software contained six separate functions that correspond to either Braille characters or a graphic image.

When producing the solenoids, Ms. Kirchhoff was involved in all stages, beginning with the building of the spool to winding the copper wire. Each spool was individually cut, drilled, and press-fitted before solenoid was coiled. Upon completion of all solenoids, the team needed to then solder them onto each of the printed circuit boards. With eight solenoids per board and three boards, a total of 24 solenoids were needed. Besides the

Design Team 6 Technical Report

soldering of the solenoids, several other components needed to be soldered in order for the boards to function. The components varied from thru-hole resistors and capacitors to surface mount chips.

With the completion of the Arduino software and the solenoids, the Phase II HUI was on its way to be completed for final testing and debugging. After numerous test trials and when debugging was no longer needed, the Phase II HUI was ready to be presented to Team 6's sponsor and customers.

James Hunter: Lab Management



Mr. Hunter was initially tasked with analyzing and improving the Graphical User Interface from Phase I and implementing it for Phase II. However, as the project progressed and the focus changed from needing the GUI for the project to work properly, to stabilizing the pins and improving the resolution, Mr. Hunter's focus changed to overall construction of the device. He was in the lab consistently working on reports, presentations, or hardware construction.

Mr. Hunter assisted in almost every aspect of the project, from construction of the coils, testing of parts, and soldering of components, to final assembly of the device. One of his largest contributions came from his soldering skills where he soldered the majority of the components onto the PCBs. This proved to be difficult as he had no experience in surface mount components. He also soldered the majority of the solenoids, another difficult process due to the fact the solenoids were hand-wound and had neither through-hole pins nor surface mount pins. In addition to the soldering, Mr. Hunter assisted in winding the solenoids with other team members using the sponsor's lathe. Additionally, he designed and soldered the hex inverter proto-board. He also helped with testing the parts to ensure connectivity and proper functionality.

Appendix 2: References

[1] Arduino Tutorial: <http://arduino.cc/en/Tutorial/ShiftOut>

[2] Arduino Tutorial: <http://www.arduino.cc/en/Guide/Introduction>

[3] Gillespie, Richard B., and Sile O'Modhrain. "The Moose: A Haptic User Interface For Blind Persons with Application to the Digital Sound Studio." Center for Standard University: Computer Research in Music and Acoustics, Oct. 1995. Web. 18 Jan. 2013.

Design Team 6 Technical Report

- [4] "Haptic Technology." *Wikipedia*. Wikimedia Foundation, n.d. Web. 3 Feb. 2013.
- [5] HEF4794B Datasheet:
<http://html.alldatasheet.net/html-pdf/17782/PHILIPS/HEF4794B/247/1/HEF4794B.html>
- [6] Hex Shifter Datasheet:
<http://pdf1.alldatasheet.com/datasheet-pdf/view/174211/ONSEMI/MC14504BCPG.html>
- [7] MacLean, Karon. "Haptics and the User Interface." *Haptics and the User Interface*. Georgia Tech, n.d. Web. 15 Jan. 2013.
- [8] "Magnetic Domains." *Magnetic Domains*. Wikipedia, n.d. Web. 25 Mar. 2013.
- [9] Miller, Tim, and Robert Zeleznik. "Haptic User Interfaces." *Haptic User Interfaces*. Brown University, n.d. Web. 28 Jan. 2013.
- [10] MOSFET Datasheet:
www.fairchildsemi.com/ds/FD/FDS8858CZ.pdf
- [11] "Multiple HEF4794 LED Drivers." *Class 27*. N.p., n.d. Web. Mar.-Apr. 2013.
<http://www.khazar.com/academics/portal/ucsc/2010winter/art22/class27.php>
- [12] "Neodymium Magnet." *Wikipedia*. Wikimedia Foundation, n.d. Web. Jan.-Feb. 2013.
- [13] Schmitt Trigger Datasheet:
www.ti.com/lit/ds/symlink/cd40106b-mil.pdf
- [14] "Solenoid." *Wikipedia*. Wikimedia Foundation, n.d. Web. Feb.-Mar. 2013.

Appendix 3: Technical Attachments

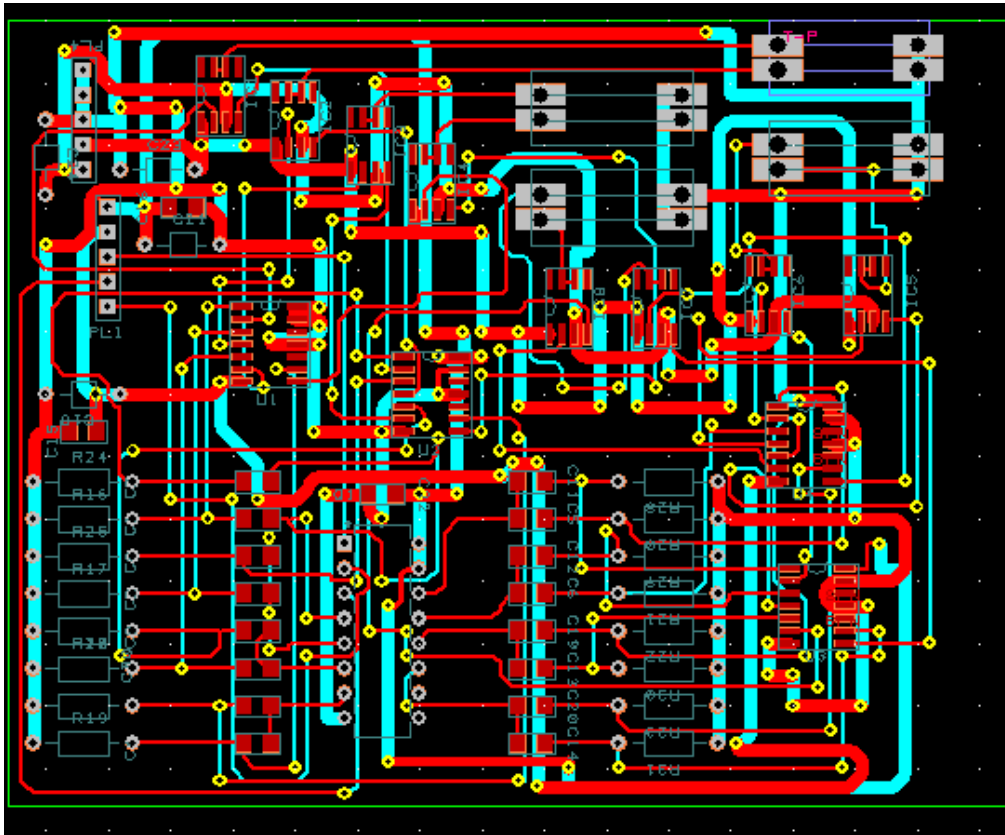


Figure 1: PCB layout