Sense Solutions
School of Engineering Science, Simon Fraser University
8888 University Drive, Burnaby, BC, V6A 1S6
ensc440-sensesolutions@sfu.ca

October 14th, 2010

Dr. Andrew Rawicz,
School of Engineering Science
Simon Fraser University
8888 University Drive
Burnaby, British Columbia
V5A 1S6

RE: ENSC 440 Project Design Specifications for Cost Effective Braille Embosser

Dear Dr. Rawicz,

With passion and commitment, the four engineers in Sense Solutions are devoted to creating a practical, affordable Braille embosser for the visually impaired. Our product will give people a chance to own a personal Braille embosser at a minimum cost for everyday use. Please find the attached design specifications for a cost effective Braille embosser.

This document aims to outline the technical aspects of the system that will provide the functionality previously outlined in the Functional Specification for the *Elementium*. The design specifications take into account various tests performed in order to produce a quality, cost effective design.

Sense Solutions consists of four innovative and skilled engineers who have various valuable backgrounds. All of us feel obliged to apply our knowledge and experience to develop products that will not only be useful and competitive, but also will benefit the society as a whole. We are Brendan Fairs (CEO), Yiran Du (COO), Heedong Park (CFO) and Rio Li (CTO). If you have any questions or concerns, please feel free to contact us at ensc440-sensesolutions@sfu.ca.

Sincerely,

Brendan Fairs
Sense Solutions, Chief Executive Officer

Enclosure: Design Specifications for Cost Effective Braille Embosser sense solutions

Elementium

Cost Effective Braille Embosser

DESIGN SPECIFICATIONS

Management Team:	Yiran Du Brendan Fairs Rio Li Heedong Park
Contact:	ensc440-sensesolutions@sfu.ca

Submitted To:	Dr. Andrew Rawicz - ENSC 440 Mr. Mike Sjoerdsma - ENSC 305 School of Engineering Science Simon Fraser University
Issued Date:	November 17, 2010
Revision:	1.0

Executive Summary

The design specification for the *Elementium* Personal Braille Embosser provides comprehensive detail on the design and implementation of the *Elementium*. These design specifications are based on testing and research in order to find viable solutions that correspond to the Functional Specifications outlined in a previous document.

There are many key components to the design, ranging from mechanical, to electrical, to software. The physical embossing of the Braille dots onto a piece of paper, as well as the paper feed itself are the key design pieces to creating this product. A complex software interfacing module was created in order to accept user input and translate this into a format useable by the embosser itself. To combine this software with the mechanical embosser, a microcontroller was programmed to control all of the necessary components.

For the mechanical actuation, it was decided through vigorous testing to use electromagnetic solenoids in conjunction with servo motors and custom designed embossing pins. By using the solenoids in a linear fashion to block corresponding pins inside an embossing case, and using the servo motors for both paper feed and linear movement of this case, we were able to produce the desired embossment.

The microcontroller unit which controls the mechanical actuation is connected to the client PC through a USB connection, and receives it's commands from a user controlled GUI.

For each of the above design choices, group testing and research had been done in order to choose the appropriate technique and/or part. This document will explain our design choices, as well as outline some of the alternatives that were tested and not chosen.

Table of Contents

List of Figures

List of Tables

Glossary

USB	Universal Serial Bus
GUI	Graphical User Interface
LED	Light Emitting Diode
Grade I Braille	Braille ouput is letter by letter – small words are spelled out entirely
DPDT	Double Pole Double Throw
ASS	American Standard Sign
ANSI	American National Standard Institute
CBA	Canadian Braille Authority
CSA	Canadian Standards Association
PC	Personal Computer (Windows)
MAC	Macintosh Computer (Linux, Unix, Apple OS)
UEB	Unified English Braille (Code)
ASCII	American Standard Code for Information Interchange
UART	Universal Asynchronous Receiver and Transmitter
ICSP	In Circuit Serial Programming

1.0 Introduction

Sense Solution's innovative *Elementium* is a convenient, cost-effective Braille Embosser. The product contains several parts: user-friendly software which can convert digital text into Grade 1 Braille, a data cable that transmits information between PC and the embosser, a power cable that provides power supply to the embosser. The Braille Dots dimension used is from the American Standard Sign, which is most generally used across North America. [1] *Elementium* provides an ideal way for visually impaired to read and write in Braille. This document outlines the design specifications for various components of the device.

1.1 Scope

The following pages illustrate the design details for the functions as stated in *Functional Specifications*, and the design choices which were made with emphasis on safety, usability and cost-effectiveness. The primary focus of this document is on the prototype requirements, but further simplification and optimization for the final products are also included for reference.

1.2 Intended Audience

- Development and Test Engineers: To be used as a guide line to design and verify the functions and requirements.
- Usability Engineers: To be used as a reference to determine if the product meets the corresponding requirements.
- Marketing: To be used as a reference to evaluate its market value and assist advertisement

2.0 System Specification

Elementium will be able to translate text into Grade 1 Braille and emboss the contents onto paper. Users can choose any content they want to print into a text file and print them out. By pressing the print button on the provided software interface, *Elementium* will initialize and start printing Braille onto paper. The user can stop printing at any time by pressing the cancel button, and then *Elementium* will stop and dispense the paper out and reset to the initial state. *Elementium* is compatible with both PC and MAC and requires a USB connection and the household power supply to run.

3.0 Overall System Design

In this section, each component of the overall system will be presented. More detailed information on the mechanical, electrical, and user interface will be presented in the respective sections following this overview.

3.1 High-Level System Design

The figure below shows each component of the Braille Printing System and their relations. Direction of flow of information is illustrated by arrow heads.

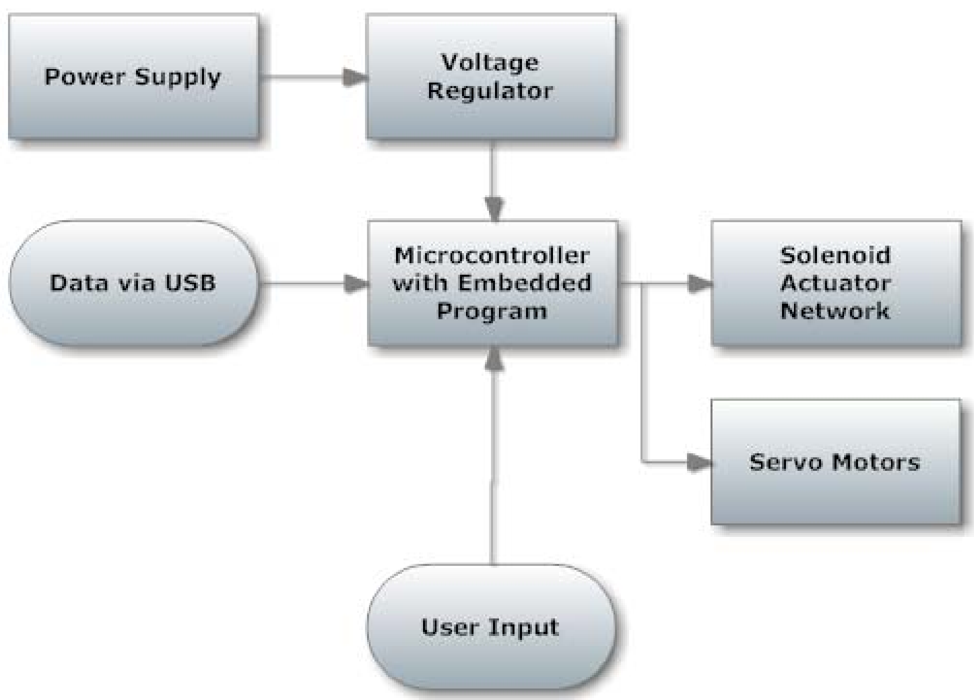

Figure 1: High-level system diagram

The translation software developed will accept digital text input and will convert them to digital binaries which form Braille Dots Matrix. Each line of the converted digital binaries will then be carried to the microcontroller via USB connection when user prompts the print command in the software application which accompanies the device and is installed on a PC. The microcontroller will then process the digital input and it will send electrical signals to corresponding actuators that will determine the embossing of the Braille dots on a paper. The microcontroller will also manage positioning of a paper during the printing process. The system will respond to user inputs such as stop/resume and cancel printing. Figure 2 below shows this process in a graphical representation, the process is from left to right.

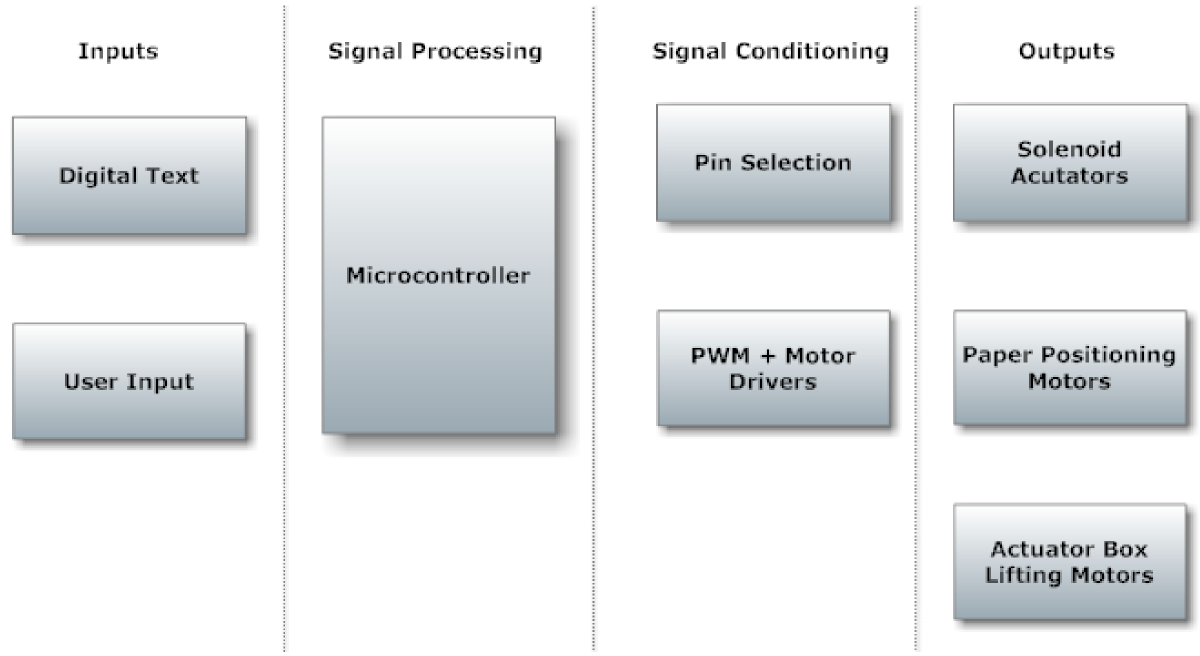

Figure 2: Block diagram of system stages

3.2 Mechanical System Design

The main objective of the mechanical system is the ability to actuate corresponding Braille dots, position the paper, and finally emboss the paper. Several matters are considered in building this sub system.

- Dot Placement
- Dot Design
- Dot Actuation

3.2.1 Dot Placement and Design

As mentioned in the Functional Specifications, the dots will conform to the standard American Braille Dot size specifications, as shown in Figure 3 below. This gives the constraints that the actuation mechanism must be designed within the dimension specified.

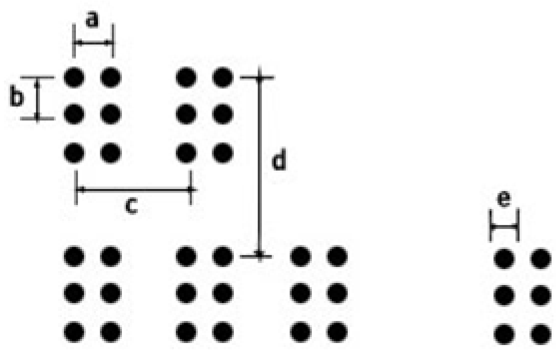

Figure 3: Standard Braille cell dimensions

The unit for the dimensions in the table below is mm. This is the standard from American Standard Sign, which is the most generally used in North America.[1]

Horizontal dot to dot	Vertical dot to dot	Cell to cell	Line to line	Dot base diameter	Dot height
a	b	c	d	e	
2.28 mm	2.28 mm	6.09 mm	10.16 mm	1.5 mm	0.5 mm

Table 1: Size of the Standard Braille cell dimensions

Due to the very small dimensions of the Braille cell formation, the mechanical actuation for embossing needs to be functional within a small space, since the dots need to be so close together. This suggests that the components themselves must be quite small, to ensure that the full printing mechanical parts remains within the small dimensions.

To maintain the accuracy of the Braille dots and spacing, the embossing head of the actuator needs to be hemispherical in shape, and within the size constrains set by the size requirements, and also paper positioning must be precise. There are few other limitations on the construction of the embossed Braille dots, which allows for much creativity when designing the embossing system.

3.2.2 Dot Actuation

The device will employ mechanical actuation to configure an array of embossing pins to press against on the paper to emboss Braille dots. Several methods of actuation have been researched and also through several modifications in the design, the most suitable solution was carefully carried out.

- Actuation using miniature solenoids
- Actuation using a magnet with an electromagnet pairs
- Use actuators to fix position of selected emboss headers

Actuation using miniature solenoids is one of solution of the current impact type Braille printer in market. However, due to the fact that Braille cell dimension is very small, customized miniature solenoids are required to use them as embossers. In order to build such solenoids with enough impact strength to emboss a paper, they require significant current supply and manufacturing cost. Also, such current draw will cause the solenoids to heat up quickly. Therefore, the method was not pursued as a viable solution.

Another method proposed using magnets with miniature electromagnets pairs. An experiment has been conducted, and it is found that more emboss strength can be applied on a paper with the magnets and electromagnets pairs. The highly magnetized rare earth type magnets used in the experiment shows that high magnetic force can be realized when polarities are generated to the electromagnet by supplying current to them. However, each unit of the electromagnet is relatively expensive. Hence, obtaining multiple units of them will exceed our projected cost limit. Also, high current must be applied to the electromagnets to generate enough magnetic force to emboss a sheet of paper by the magnets attached with the print head. Therefore, this method also was not pursued as a viable solution.

Our choice of microcontroller board can supply a maximum 50mA of current through each digital output pins. This is the quarter the amount of both miniature solenoids and electromagnets needed to be operated to generate enough emboss strength on a paper. A current amplifier can be used in this case to amplify the supplied current from the digital output pin to the level of current supplied from a power adapter. However, by considering that our device requires many units needed to emboss a line of Braille dots on a paper, those options were not pursued as a viable option.

To overcome the discussed limitation, a viable design solution has been proposed. The idea is to actuate the emboss pins using the solenoids indirectly. The emboss pins that are corresponding to each Braille dot in a line will be placed in a plastic shaped with an indentation in vertical direction. There is uniform space between tail of each pin and plastic cast. When the tail of the emboss pin does get fixed by the actuation of the solenoid plunger in horizontal direction, the emboss pin will also be fixed in position. Then, when the emboss pin is pressed against a paper surface, the paper will be embossed. Otherwise, when the emboss pin does not get fixed by the actuation of the solenoid plunger in horizontal direction, the emboss pin will be displaced from the default position when the emboss pin is pressed against a paper surface, thus it will not emboss on a paper. More details will be presented in the subsequent sections that will follow in later in this paper.

3.2.3 Embossing

All components used in the above mechanism will be structured in an enclosed plastic case. The case will have enough mass to apply emboss pressure on a paper surface when placed on a sheet of paper. A mechanism is needed to lift the case to select the pins, before embossing on the paper, and then place onto the paper afterward. There can be many different mechanical solutions to give to the case this described linear motion. We chose to use servo motors to give the case the linear motion by considering that they are readily available and proven to work. Therefore several methods are proposed to translate the rotational torque generated from servo motors to the linear motion.

A method has been proposed using rack gear or worm gear to translate rotation to linear motion. It is the most reliable mechanical solution to translate rotation into linear motion. However, obtaining the right size of the gears and suspending the case requires another supporting structure. Hence, it may require many different components and complicated process to build.

Another method has been proposed using pulley or lever, but building a mechanical system using them is also considered as quite complicated process.

Finally proposed and considered as viable method is using servo motors with arm to lift the case directly and the movement of the case will be guided along physical constraints that surrounds the case. In general, servo motors are bi-directional motors. We can rely on the proposed method; because the mass of the case alone can apply enough emboss pressure on a paper surface. Otherwise, the method using gears needed to be used so that torque generated from motors to assist emboss.

3.2.4 Paper Positioning

For general computer printers use a stepper motor with rollers to move the paper in the exact increment needed to ensure a continuous image is printed. [4] Our device will use a servo motor to control rollers to move the paper in the exact increment needed to ensure a continuous Braille dots embossing. Our choice of the microcontroller unit has ability to control a servo motor by sending a pulse width modulated signal directly to the servo motor driver and perform precise degree rotation over specified period. In this way, the paper will advance to the exact position of each line.

3.2.5 Paper Feeding for Continuous Printing

Many conventional Braille printers use special Braille paper in continuous form. (Tractor feed) [5] For general computer printers use a set of rollers to pull a sheet of paper in from the feeder and advance the paper when the printer head assembly is ready for another pass. [4] We will implement the similar feeding system to our device using an obsolete printer parts. More details of the mechanical system will be discussed in *Section 4.0*.

3.3 Digital System Design

The main feature of the digital system is the ability to process incoming signals and respond accordingly. Few things needed to be considered for our device design, including:

- Management of input and output lines and their respective signals
- Implementing control logic and algorithm

3.3.1 Control Unit

The main objective of the digital system is to control the embossing mechanisms explained in the previous sections. This hardware must be able to accept pre-processed inputs from the PC software via USB, and process them further into a line of Braille dots to select emboss pins and control movement of the case enclosing the emboss pins to emboss onto a sheet of paper.

There are various manufacturers and specifications available for microcontroller units, thus it was necessary to weight and compare the specifications among them and choose a suitable microcontroller unit to achieve the control of our device. Since our device requires controlling many solenoids and servo motors concurrently, it was best to choose a microcontroller unit that satisfies all the control needed and connectivity. Also, it was better to choose a microcontroller unit that can be connected to electronics components without accommodating any intermediate circuitry. More details of the digital system will be discussed in *Section 5.0*. [3]

3.4 Power System Design

The design of the power system determines the rest of the device in means of power supply, thus is crucial for defining the power consumption of the device when embossing is in progress. The mechanical design of our device suggests that many electronic components needed to be powered during the embossing. Thus, a power adapter, also known as a transformer, is required to supply power from an external power source to the microcontroller, solenoids, and servo motors to supply current more than what can be delivered through PC via USB connection.

For our device, the microcontroller unit is solely responsible for powering the entire device electronic components. The embossing process needs power if the algorithm defined in the microcontroller directs output signal. Hence, no continuous power supply is required. Also, the microcontroller unit has ability to ensure a constant power supply via voltage regulator it consists. [3] More details of the power system will be discussed in *Section 6.0*.

3.5 User Interface

The user interface consists of the software application for pre-processing digital texts into binaries represents Braille dots, and user controlled buttons for changing state of the device. The device enclosure or casing will also be properly implemented.

3.5.1 Software Application

In general, computer printers provide a software driver. Although microcontroller units with fast processing power are available these days, most of the image mapping process for pixel generation cannot be accomplished without a computer process. The printer receives the processed binaries that contain information of each pixel and the microcontroller unit in the printers will control and send corresponding electric signals to the electronic units to print on a paper accordingly. [4]

The same scheme will also be applied to our device, hence a software application will also be provided with the device. The software application will be developed in an interface that allows users to interact with the application directly without typing any command. It will receive digital texts from users, and then it will process the text into binary representation of Braille dots. Then, they will be transferred to the microcontroller unit to be interpreted and control the electrical components accordingly. To meet the process requirements, 3 different modules needed to be implemented for the software application. They are:

- Communication
- Translation
- Graphical User Interface

3.5.2 Buttons

The user input comes by way of the buttons located on the device. Through these buttons, the user can stop the printing, resume the printing, and cancel the printing.

They will be placed on an area where they will not be inadvertently pressed, and must also be placed on a logical manner. The size of the buttons will not be too small, and not be too flat, but appropriate that user can manage to prompt the buttons.

4.0 Mechanical System

There are three distinctive mechanical designs will be implemented in our product:

- Paper positioning and feeding
- Select emboss pins
- Emboss on a paper

For each mechanism, several methods have been researched and tested empirically for the best suitable solution. As briefly outlined several methods in *Section 3.0*, the best suitable solution will only be presented in this section.

4.1 Paper Positioning and Paper Feeding Mechanism

Our device will use a servo motor to control rollers to move a paper in the exact increment needed to ensure a continuous Braille dots embossing. Figure 4 below shows graphical concept of the motion of the system in front view. Our choice of the microcontroller unit has ability to control a servo motor by sending a pulse width modulated signal directly to the servo motor driver and perform precise degree rotation over specified period. In this way, the paper will advance to the exact position of each line.

Most conventional Braille embossers use special Braille paper in continuous form (Tractor feed) [5]. Designing and structuring the explained mechanism requires a high precision manufacturing environment. Exact dimension of gears and rollers are not readily available in market, and develop a structure using the components are very hard to achieve with resources currently available to us. Hence, we decided to use parts from an obsolete printer for both paper positioning and paper feeding combined with the servo motor to control the gear network of the printer part. The paper positioning and feeding structure will combine with the emboss components explained later of this section in detail.

The emboss surface generates friction when the roller moves a paper on the surface, and this may result in incorrect positioning of the paper while embossing is in progress. Hence, tracks have been formed to assist paper movement without much friction generated. Note that the motor and roller is not structured together with the track. The graphics does not represent the mechanism in exact.

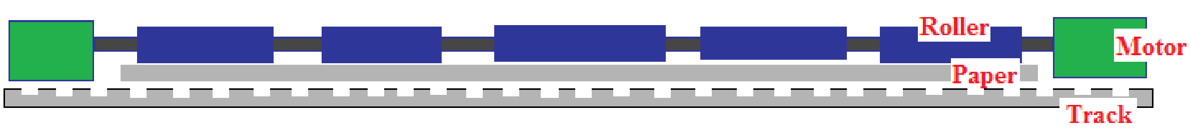

Figure 4: Graphical concept representation of the paper positioning in front view

4.2 Emboss Pins Selection Mechanism

Using the solenoid plunger directly as embossing pins found to having several constraints. The first constraint was the size of the miniature solenoid. Table 2 outlines the exact dimension specifications of the miniature solenoids. With the diameter of 12.5 mm, the distance between plungers of solenoids when placed them together is much larger than the distance between each Braille dots. Some of conventional impact types of Braille printers use customized solenoids with the size that falls within the dimension constraints of the Braille cell. However, designing and producing such solenoids will be costly and requires mechanical precision to manufacture.

Our aim is to produce a cost effective Braille printer. Hence, cost of constructing such solenoids will exceed our aimed cost.

The second constraint was that high current is required to drive the solenoids to produce enough emboss pressure on a paper. A standard letter sized paper has significant surface tension and in order to emboss on the paper, a significant strength from the solenoid plunger is required, which requires high current. There are general methods to strengthening a solenoid magnetic force. However, high current is ultimately required to drive a solenoid plunger to have enough strength to emboss on a paper. The microcontroller unit used for our digital system can only supply limited amount of current to the solenoid. The current can be amplified using a current amplifier, but this will increase complexity of the product design, and produces more constraints, hence considered as not viable solution.

The third constraint was the high temperature heat dissipation when high current is supplied to the solenoids.

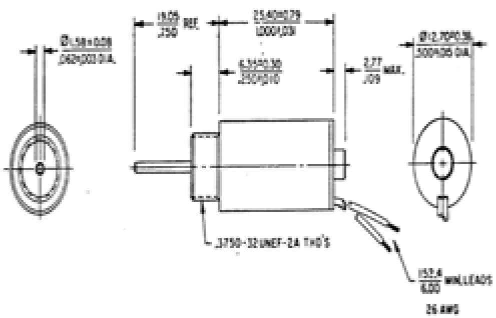

Figure 5: Solenoid actuator dimension schematics (inches)

Size(mm)	Operating Voltage(V)	Duty Cycle	Resistance(Ohms)
12.5X 12.5 X 25.0	12	1/4	44 Ohms

Table 2: Solenoid actuator specifications with electrical characteristics

The idea is to actuate the emboss pins using the solenoids indirectly. Figure 6 below shows graphical concept of the motion of the system in front view. The emboss pins that are corresponding to each Braille dots in a line will be placed in a plastic shaped with an indentation in vertical direction. There is uniform space between tail of each pin and plastic cast. When the tail of the emboss pin does get fixed by the actuation of the solenoid plunger in horizontal direction, the emboss pin will also be fixed in position. Then, when the emboss pin is pressed against a paper surface, the paper will be embossed. Otherwise, when the emboss pin does not get fixed by the actuation of the solenoid plunger in horizontal direction, the emboss pin will be displaced from the default position when the emboss pin is pressed against a paper surface, thus it will not emboss the paper.

In this way, we can manage to emboss all Braille dots of a line concurrently while small current is supplied to the solenoids and also ensure embossing pressure applied onto the paper.

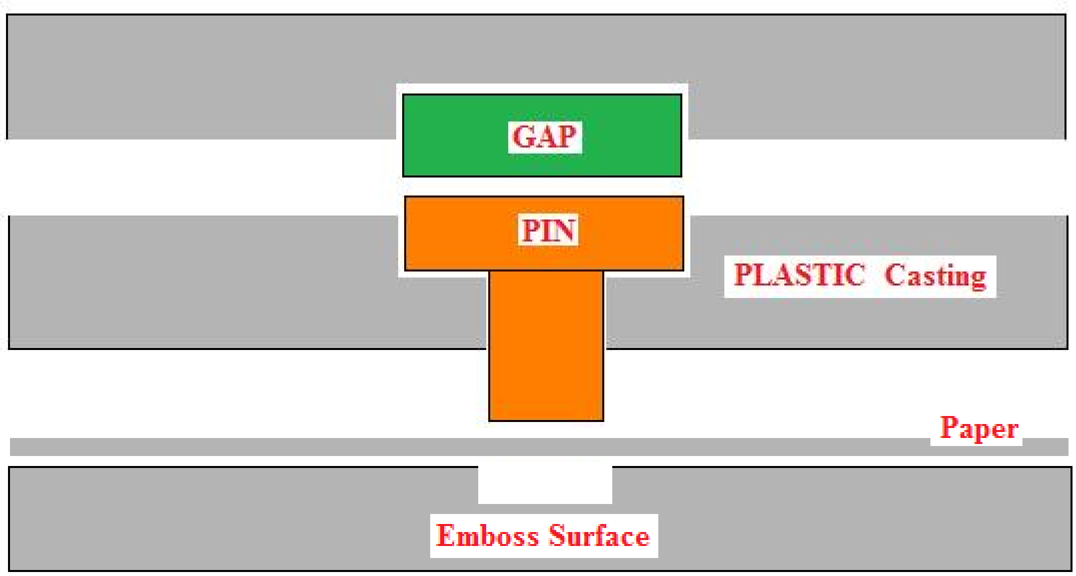

Figure 6: Graphical concept representation of the embossing unit in front view

Precise embossing can be obtained when the emboss surface is pre shaped in the exact size of hemisphere, rather than the pressure absolve surface is used as proposed in the *Functional Specifications*. Figure 7 below shows graphical concept of the motion of the system in side view.

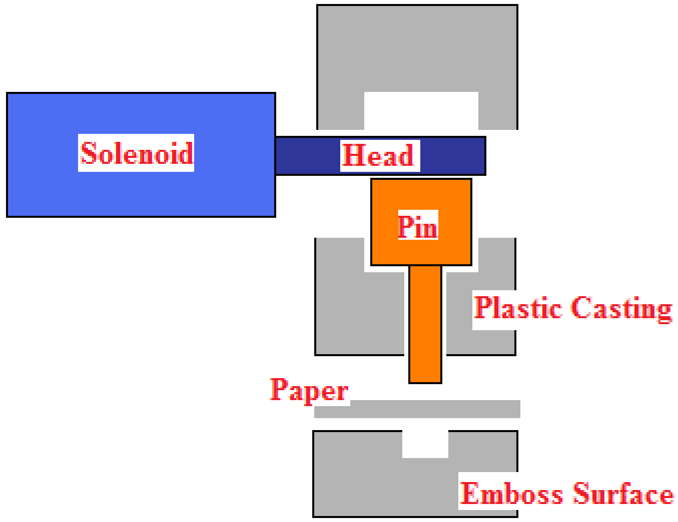

Figure 7: Graphical concept representation of the embossing unit in side view

After empirical testing has been conducted, the emboss unit design meets the requirements and constraints and is therefore the best suitable option available. The solenoids consists a return spring, so while deactivated, the plunger will be return to its default position. The whole line of Braille dots will be form using the above configuration for each emboss unit and they will enclosed in the plastic casting.

4.3 Embossing Mechanism

In the previous section, it has been mentioned that the design of the selection of the emboss pins using the solenoids will be enclosed inside the plastic casting. Figure 8 below shows graphical concept of the motion of the embossing mechanism from front point of view. All the components enclosed in the plastic casting will be considered as an actuator box except the emboss surface. The bidirectional motor and the casts will control linear motion of the actuator box to lift up or down. Specification of the bidirectional servo motors are presented in Table 3 and it shows that it can produce enough torque to lift the actuator box, which weighs less than a kilogram. Also, the components that lifting the actuator box will be shielded with PVC tape to prevent degrading plastic case of the box due to friction. While the motors lifting the box, the emboss pins selection process explained in previous section will be performed, after completing the process the motors will rotate to opposite direction to put the box on a paper. The mass of the actuator box applies enough pressure to on a paper to emboss with the emboss pins selected when placed on the emboss surface. After embossing a line of Braille dots, the actuator box will be lifted again by the motors with lifting arm and all emboss pins will return to its default position and the pin selection process will be performed again.

Size(mm)	Operating Voltage(V)	Output Torque(Kg/cm)	Operating Speed(deg/sec)
22.8 X 11.6 X 20.6	4.8 – 6.0	1.4	0.10/60

Table 3: Servo motor specifications with electrical characteristics

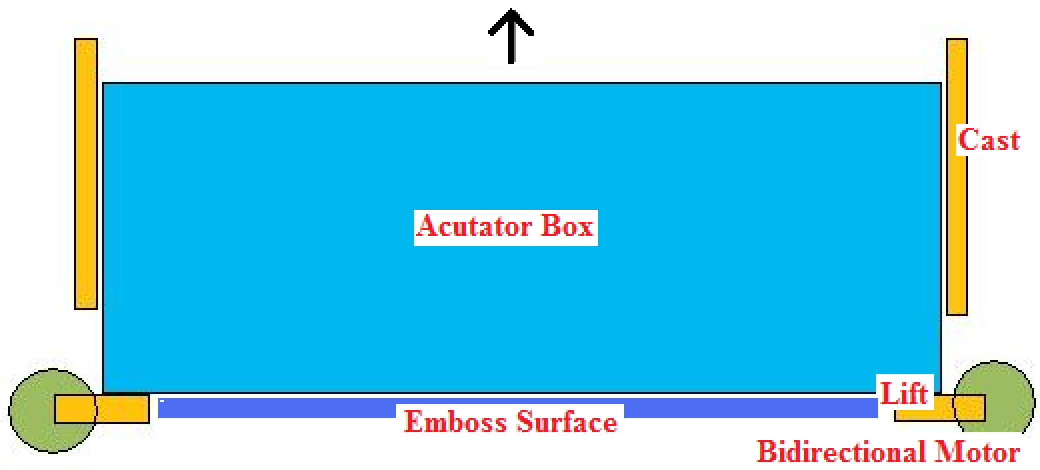

Figure 8: Graphical concept representation of the actuator box movement in front view

5.0 Digital System Design

The device requires a control unit to control the solenoid and motors and also to provide the desired response for each user command.

5.1 Microcontroller Unit (MCU)

The main feature of a microcontroller unit is the ability to control the number of output signals concurrently and pulse width modulated signal for servo motors. Also, it is required to accompany a hardware unit for communication such as USB to receive data processed from a personal computer. It is also required to accompany a power control unit to deliver reliable voltage and current to number of output pins to control electrical components. Also, it needed to accommodate electrical safety features as well to fulfill the electrical safety requirements specified in the *Functional Specifications*.

Several microcontroller units were considered to meet the ability to control and process the level of complexity of our system algorithm. Also, we considered a microcontroller unit that accompanies all units mentioned. Our choice was using the Arduino microcontroller board which interfaced with the Atmel ATMega microcontroller. It has 54 digital input/output pins (of which 14 can be used as pulse width modulated outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz internal clock, a USB connection, a power jack, an ICSP header, and a reset button.

5.1.1 Required I/O lines

The microcontroller unit is required to receive data from a personal computer and user control signals from the user interface. Then, it is required to route or process the received signals and data accordingly to the corresponding electrical components. We estimate these tasks would require the number of I/O pins outlined in Table 4.

I/O Signals	Number of lines
Start/Resume Cancel	1 Interrupt line 1 Interrupt line
Power Supply(Vcc and Gnd)	2 lines
Dots Select	40 lines
Servo Motors	4 lines
Total	48 lines

Table 4: I/O pin requirements for micro controller unit

5.2 Serial Communication

Computers can exchange bits of information serially or in parallel. In applications where it is necessary to have one computer talk to another, the most commonly used communication method is serial communication.

Therefore it is no surprise that serial communication is the method used to send data between the microcontroller unit and a computer. Information is sent to and from the computer and the microcontroller interprets the information and converts them to electrical signals by setting the corresponding pins.

Our particular choice of microcontroller unit has four serial ports and one of them communicates bit information at a time via USB connection. Also, the microcontroller unit has ability to setup a stable connection in both Windows and Macintosh platform with speed control.

5.3 User Controls

Our particular choice of microcontroller unit has total of 5 interrupt lines to determine which control option the user has selected from the user interface. As stated in the *Functional Specifications*, our device will have two different user input controls beside the power button. The first interrupt line will be active when user selected the stop button, and it will be low again when the same button has been selected (resume). The second interrupt line will active when cancel has been selected, in this case, the microcontroller will be reset to initial state, and the interrupt line will be low again afterward. The flowchart in Figure 9 shows how the microcontroller will control the user selected options.

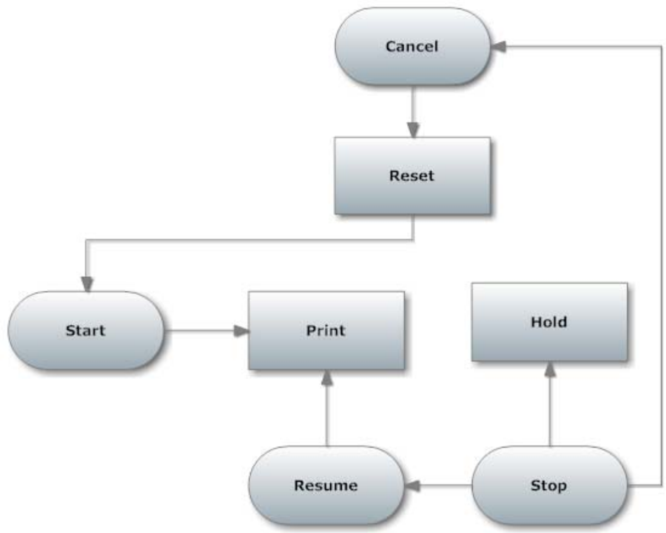

Figure 9: Flowchart of user input and its response (circular rectangle (user input) and rectangle (microcontroller control))

6.0 Power System

The device requires a power adapter to supply high current to the system. The power consumption requirement of each electrical components used in the device are outlined below in Table 5. Please note the value for each solenoid is the maximum current that can be supplied from each digital pin of the particular microcontroller unit. The general current requirement of the solenoid is higher than the value presented.

Components	Voltage(V)	Current(mA)
Microcontroller Board	3.3 – 5	10
Solenoid Actuators	5 – 12	40-50
Servo Motors	4.8 – 6.0	40-50

Table 5: Electrical requirements

The power sources and their supply capacities are outlined below in Table 6. The microcontroller unit interface has voltage regulation circuitry to ensure safe power control and accommodates circuit breaker mechanism to prevent excess current flow. [2]

Sources	Voltage(V)	Current(mA)
PC via USB	12	500
Power Adapter	12	3000

Table 6: Power supply

7.0 User Interface

The user interface unit is a set of buttons that allows the user to communicate with the device and also includes graphically interfaced user software that allows the user to input digital texts to emboss corresponding Braille dots on a paper.

7.1 Software Design

There will be Braille Translation software included with the device. It is developed using Java integrated development environment as graphical user interface to allow users to paste digital texts into the application directly to print into Braille dots representation. The software consists of 3 modules:

- Communication
- Translation
- Graphical User Interface

7.1.1 Communication

The microcontroller and the computer software follow the serial communication protocol specified in the state machine representation in Figure 10.

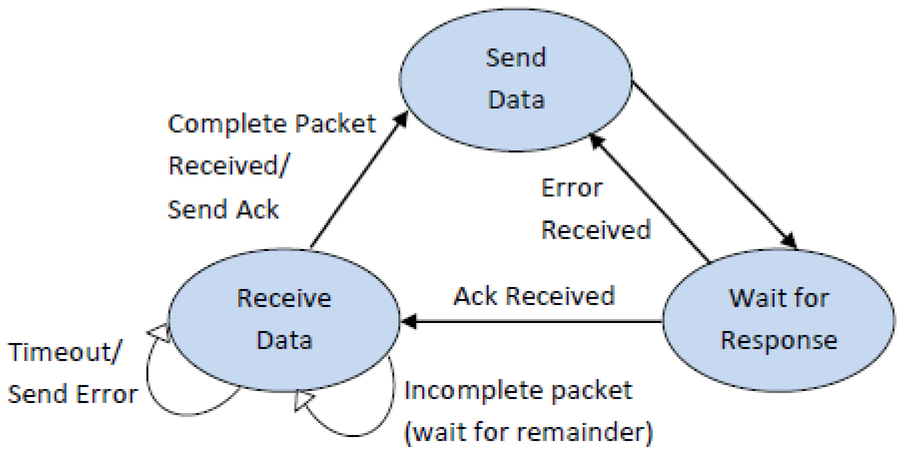

Figure 10: Serial communication protocol

Initially, the microcontroller is in the send state, and the computer is in the receive state. The microcontroller will start the communication by first send an acknowledge bit to the serial port and computer will check for the acknowledge bit in the serial port when notified by the serial port. When the computer software application gets notified, it will receive the acknowledge bit. If the bit is matched with the pre-specified acknowledge protocol, then it will deliver packet of data to the serial port each bit at a time. Once it delivered the packet of data, then it will not send any more data to the serial port until it read either error or acknowledge bit with notification from the serial port.

The first packet of data from the software application will include following data:

- Start of packet indicator bit (Character '!')
- First line of Braille dots representation in binaries (Three lines of binaries per Braille character line)
- End of packet indicator bit (Character '*')

Then the microcontroller will check each bit that it reads from the serial port, and if the read byte is equivalent to the start of packet indicator character, then it will start saving to a buffer to be processed further until it reads the end of packet indicator. The number of Braille dots in each line consists of 40 binaries. If the end of packet indicator is not read within that number then the packet is corrupted and required to receive the packet again. This will let the microcontroller to send error bit to the serial port (Character 'E'), otherwise it will send the acknowledge bit to the serial port (Character 'A').

The software application will terminate the serial communication in two different instances. It will terminate the serial application when the end of data indicator is read from the serial port which was sent from the microcontroller after sending the last packet of data with the end of data indication. The microcontroller will receive it first from the computer, and then when it reads the indicator then it will send the end of data indicator back to the computer.

Also, if the software application has not received the acknowledge bit via serial port in a specific duration (Time out) or the microcontroller sends the cancel indicator(Character C), then the communication will be terminated. However, the user will be able to reestablish the serial communication upon prompting print function.

7.1.2 Translation

The objective of the translation module of the software application is to convert digital text into Braille dots representation in binary format. A character of the Braille dots representation is in 2 by 3 matrix dot formation. Our device is capable of printing 20 characters per line. Then the number of dots per line in Braille dots representation is 40 dots. Therefore three lines of 40 dots is equivalent to a line of characters in Braille dots representation. The translation module of the software application will then perform the following sequences of process. The non ASCII characters should first be removed by user manually before input into the text field of the graphical interface. It is planned to implement the character parse and filtering feature, if time permits.

1. Accept digital text and save it into a buffer. (A 'space' considered as a valid ASCII

character)
2. Read the buffer contents, and then convert each character into corresponding Braille dots representation.
   A. Since each Braille character is in 2 by 3 matrix format, each position can be represented by a binary string with each binary number represents position of each dots and their usage.
   B. For example, Character 'A' can be converted to Braille representation in binary format as '000001 + 100000'; the '000001' indicates that the character is an uppercase, and '100000' indicates character 'a'
3. Reposition the binaries to generate each line of 40 dots, and then add start of the line and end of line indicator character to the line, and then finally save to a string array.
4. Every 45^{th} line will include end of page indicator character next to the end of line character to notify microcontroller.

7.1.3 Graphical User Interface

The purpose of the graphical user interface development is to allow users to interact with programs directly without typing any command. Netbeans, a platform framework for Java desktop applications and also an integrated development environment for developing with Java, provides simple developing interface to develop a graphical user interface.

The following implementations will be done for the graphical user interface.

1. Text field accepts digital text content from user by simply paste on the field.
2. 'Browse' for file button to locate a text file contains digital text content.
3. 'Print' button to start all processes mentioned in previous sections.

7.2 Buttons

There will be 2 NO-DPST (Normally Open – Double Pole Single Throw) type push buttons to control the Braille printing: Resume/Stop, Cancel. A toggle switch will serve to change the power supply of the device.

The buttons will be located to a place where a person will have easy access. User should not be able to activate buttons accidentally in any of the various positions by which the device is expected to be used.

7.2.1 Push Buttons

All buttons will be directly connected to the corresponding interrupt line on the micro controller interface. When any button is pushed, a hardware interrupt is generated and the microcontroller will enter an interrupt service routine and determine which button was pressed, thus executing the predefined actions. The microcontroller interface provides 5 separate pins for interrupt lines, thus an independent integrated electronics are not necessary.

A button de-bounce compensation algorithm has to be used to properly read the state of the button. This will be performed through a small delay defined in the microcontroller process and through the physical characteristics of the buttons. No connections to either ground or supply voltage via a pull-down or pull-up resistor is required to prevent the floating voltage effect when the button is not depressed.

7.2.2 Toggle Switch

The power button requires persistence of state (either On or Off). It shall be implemented using a toggle switch. It will be placed apart from the push buttons, but will be oriented such that a user can easily determine the state without any further assistance.

8.0 Test Plan

The device system is composed of 3 main divisions: mechanical, digital and software. The mechanical division involves the physical selection unit, positioning unit and embossing unit. The digital system can be further subdivided into 2 subparts: microcontroller unit, user controls unit. Each unit of system will be tested individually before implement them together with other units.

8.1 Electrical Testing

All electrical components will be tested first before testing a unit of system. Only components without any deficiency will be used as the parts of the device system. It is essential that those components are tested properly so that integrated device system will produce correct results.

After integrating the required electrical components with the digital and mechanical system, empirical testing on current draw for operation, voltages across the servo motors and the solenoids, heat dissipation, sensitivity, and strength, and other required testing will be conducted to ensure they all meets the requirements specified in the *Functional Specifications*.

8.2 Mechanical System Testing

To ensure that the emboss pins selection design functions as expected, we will the design thoroughly. Each pin will be tested for proper reset and consistency. The emboss surface will be tested that a paper positions as defined. Also, we will make sure that the plastic case with embossing pins moves guided path and check paper gets embossed properly when placed on the paper.

Finally, we will check any components or system degrading possibilities, and if determined, then the design will be modified to ensure long reliability.

8.3 Software System Testing

The serial communication module will be tested to ensure the packet of data transfer between a computer and the microcontroller unit works accordingly to the defined protocol in both the software application and the embedded program.

Also, the translation module will be tested to ensure the digital texts get translated to correct Braille dots in binaries. This will simply tested by observing the call back value of the string binaries and compare them with the expected values.

Graphical unit interface can be easily tested. Two tests will be conducted for the developed interface. First, digital text content will be inserted in the text field of the interface and let the program to return the text to see if the text gets returns. Second, the 'Print' button will be pressed and check that it will start embossing.

8.4 Microcontroller Testing

The general electrical testing has been done by the manufacturer of the particular microcontroller unit of our choice. Therefore, we will not be testing the electrical requirements.

However, to verify that the control system works, we will simulate the selection of emboss pins for several lines of Braille dots, then emulate the selection of the emboss pins through a set of LEDs that can show the pins selection.

The user controls unit will also be tested to verify that the interrupt service routine enters the state preprogrammed corresponding to the button pressed.

9.0 Conclusion

The design specifications above, list out all the design considerations needed to produce the prototype of our device. The ease of use, as well as the cost effectiveness of our product makes it a viable product for the market. By conforming to all of the standards and functional specifications outlined previously, a prototype will be produced by the end of the design cycle, with final production requirements easily implementable after that point.

10.0 References

[1] Canadian Braille Authority (CBA), 2005, “Canadian Braille Authority (CBA)”, http://www.canadianbrailleauthority.ca/index.php.

[2] Canadian Standards Association, 2008, “Standards - CSA - Making Standards Work for People and Business”, http://www.csa.ca/.

[3] Arduino, 2010, “Mega 2560 Series”, http://arduino.cc/en/Main/ArduinoBoardMega2560

[4] Wikipedia, 2010, “Computer Printer”, http://en.wikipedia.org/wiki/Printer_(computing)

[5] Wikipedia, 2010, “Braille Embosser”, http://en.wikipedia.org/wiki/Braille_embosser