CHAPTER 1

INTRODUCTION

Refreshable Braille Displays aim at building a bridge between digital world and the visually impaired. With this device, a blind person can access digital information faster without the traditional and annoying methods like speech synthesis.

1.1 Description

Refreshable Braille Display (RBD) is a device that displays braille script with the help of metal pins which perform vertical up and down motion to represent the desired Braille character.

Braille is a form of written language for blind people, in which characters are represented by patterns of raised dots that are felt with the fingertips. These characters are displayed in either a six-dot configuration or eight dot configuration represented in a 3-by-2 or 4-by-2 matrix. Each character has a unique set of dots raised above the surface. The pins are used in place of dots in case of RBDs.

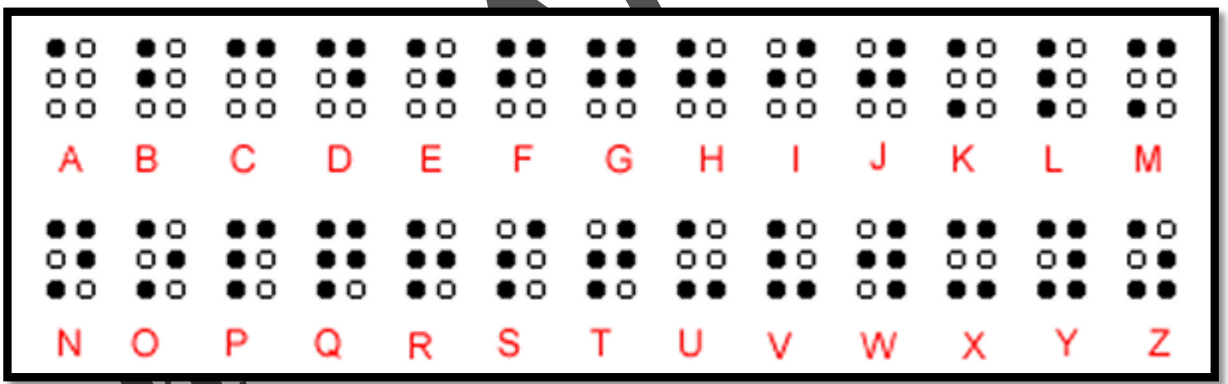

Fig. 1.1: Characters in braille Script

RBD is a device similar to any other display device, except that the output of this device is in braille. There are a lot of books already available in braille. But the accessibility to digital information has become the need of daily life. In such a scenario, depending on the braille books (paper based) and magazines does not suffice.

Currently available RBDs costs a very high amount due the expensive materials and technologies involved in its fabrication. This project aims to fabricate a cheaper prototype of

RBD by using low cost material called Shape Memory Alloy. The hysteresis property of this material is used to achieve the basic functioning of the device.

1.2 Motivation

Today we are in the information age. We have a large number of online resources for study and research. We access this information through visual display devices such as computer monitors, tablet devices and mobiles phones. This is not the case with the visually impaired. They have access to these resources only through speech auditory outputs and tactile display devices [1].

The problem with speech auditory books is that the tone is monotonous and after a brief period it becomes irritating especially if it is an interesting novel [2]. Another problem with these books is their availability, that is, not all books are converted to audio format. Same is the case with printed braille books, they are not available for all books according to the demand of a user.

This is where RBDs come into picture. Like eBooks are available for normal users, RBDs can prove to be the same for the visually impaired users.

1.3 Problem Formulation

The problem of already available RBDs is their very high costs [3]. These RBDs are based on piezoelectric crystal based actuation of the pins, which cost a minimum Rs.5000-6000 per braille character. With increasing dependence on synthetic speech, Braille literacy among persons with blindness is decreasing. Braille literacy plays a significant role in uplifting education, employment and income and this trend is thus considered significantly damaging [4].

Therefore a need arises to develop an RBD device which is as efficient as the ones available in the market but is affordable for the common man as well. The need for making learning interesting for visually impaired users allowed to add a voice input feature which allows users to learn braille themselves.

1.4 Implemented Solution

Refreshable Braille display is a device for displaying braille characters by means of round-tipped pins raised through holes on a flat surface. It consists of electro-mechanical actuators to control these pins. The challenge involved in creating such a device is developing an actuator, and placing multiple of these together and matching the standard braille display requirements. [1]

Currently available Braille display devices use bimorph piezoelectric actuators to control the braille pins. These actuators are expensive and fragile [5 6]. Various attempts have been made to reduce the cost [7]. The proposed alternatives include actuators based on electromagnetism, electrostatics, pneumatics, thermo-pneumatics and smart materials such as Shape Memory Alloys (SMA), Electro active Polymers (EAP), Electro rheological (ER) Fluids and Magneto rheological (MR) Fluids. But none of these smart materials are employed in commercially available Braille display devices successfully. [1]

Among the above mentioned alternatives, smart materials are preferred because they match the size and force constraints required in a typical braille device as per the reviews [8]. Moreover, the prior attempts show the applicability of this alternative.

A Voice based Refreshable Braille Display (VRBD) is a RBD which displays braille characters when given a voice input. The device works in the following way:

Given a voice input to a VRBD:

1. the code in the processor recognizes the text character and generates six corresponding signals for six individual pins
2. desired pins are pulled down to display the Braille form of the input character

1.5 Scope of the project

Developing an RBD, with voice based input can have various applications varying from a self-learning tool for a blind folk to a complete touch screen reading device. Keeping mind the project tenure and the available resources, we restrict to simply demonstrate this technology by displaying a single character in Braille with an audio input through a smartphone application.

CHAPTER 2
LITERATURE REVIEW

According to data from the 1994 National Health Interview Survey on Disability approximately 527,000 persons in the U.S. use some type of vision device. The following tables provide a breakdown of technology use by age of person and type of device.

Table 2.1: Number and Percent of persons using Vision Devices

	AT device	**All Ages**	**Ages 44 and under**	**Ages 45-64**	**Ages 65 and over**
Total number of people	Any vision device	527,000	123,000	135,000	268,000
Percentages of people using each device out of all those who use any vision device	Telescopic lenses	30%	32%	37%	26%
	Braille	11%	23%	17%	3%
	Readers	13%	12%	10%	15%
	White cane	25%	28%	36%	18%
	Computer equipment	6%	16%	6%	3%
	Other	53%	42%	56%	56%

As shown in Table 2.1 the number of visually impaired users using Braille script to read is less as compared to that of the other devices. Amongst these, users under age of 44 use it the most while for other age groups, other devices are preferred [2].Commercially available Braille displays use bimorph piezoelectric bending actuators to control individual Braille dots [5 6]. Attempts at reducing cost of these devices have brought it down to current levels in the range of 2500-4000 USD (65-100 USD per Braille character), which are yet significantly high [7].

Hence, recent attempts at developing affordable Braille displays have concentrated on developing new actuators, which are more cost-effective, as opposed to piezoelectric actuators. Notable actuation alternatives proposed and developed for this purpose include actuators based on electromagnetics, electrostatics, pneumatics, thermo-pneumatics and smart materials.

Smart materials used for electromechanical actuation include Shape Memory Alloys (SMAs), electro-active polymers (EAPs), electro-rheological (ER) fluids and magneto-rheological (MR) fluids. No commercial Braille display has been reported, which successfully exploits any of the above mentioned alternatives.

SMAs, notably NiTi and derived ternary alloys, exhibit interesting thermo-mechanical properties due to their unique microstructural behaviour. Martensite twinning/detwinning and martensitic phase transformations with changes in temperature and stress lead to characteristic temperature and history dependent stress–strain relations [9]. This property is usually exploited in the development of SMA based actuators, especially noted for commercial use in aerospace and automation applications.

SMA based actuation, in general, offers certain distinct advantages compared to other actuation alternatives. When compared to other smart materials, SMAs offer higher actuation stress as well as strain, which is preferred for a Braille display due to definite requirements of stroke and resisting force [10]. Compared to conventional actuation alternatives, SMAs offer a higher power density, allowing development of miniature actuators suited for the size and spacing requirements of braille [11].

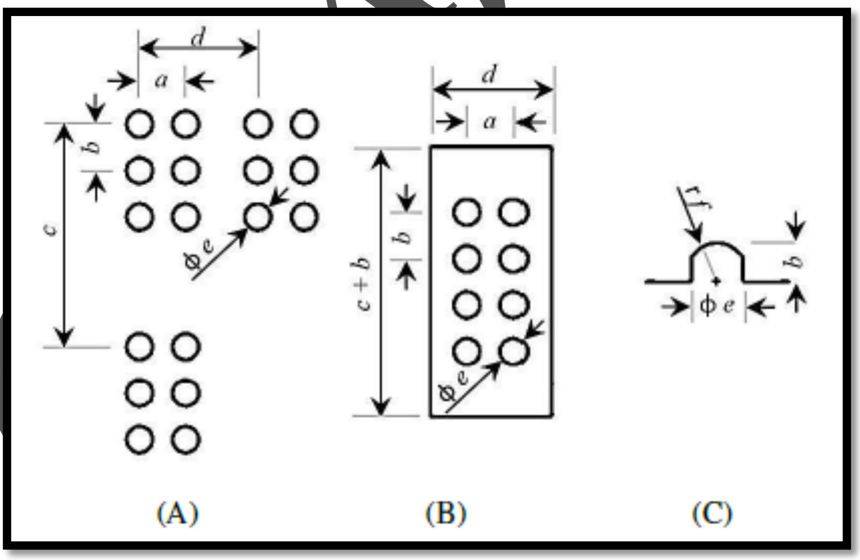

Fig. 2. 1: DIMENSIONS OF **(A)** 6-DOT PAPER BASED BRAILLE, **(B)** 8-DOT BRAILLE CELL, AND **(C)** BRAILLE DOT PROFILE

Fig. 2.3 illustrates the different dimensions for size and spacing of Braille on paper and on a Braille cell. Dimensional requirements of Braille dot and character are derived from various

Standards practiced internationally. Prescribed sizes and spacing according to different standards are given in [12]. It is however, impossible to conform to all standards, with a single choice for each dimension. A single value for each dimension conforming to the acceptable range has been chosen such that each fulfils most international standards. Table 2.2 gives the acceptable ranges and chosen values for each dimension. Profile of a Braille dot has been constructed with a dome shaped top for smooth tactile perception, as recommended in [13], and for its similarity to paper-based braille.

Table 2. 2: Range and Values for Various Dimensions

Dimension	Acceptable Range(mm)	Chosen Value(mm)
A	2.3-2.6	2.5
B	2.3-2.6	2.5
C	10-13	12
D	6.0-7.0	6.4
ϕ e	1.3-1.6	1.4
r f	0.7-0.8	0.75

CHAPTER 3
SYSTEM ANALYSIS

3.1 Functional Requirements

According to the functions of the system it is divided into three subsystems as listed below. The functional requirements of each subsystem are mentioned along with it.

3.1.1 Input subsystem

- Acts as an interface for connecting input devices
- Fetches related data from the input device
- Interpret the necessary information from them
- Initialize the processing subsystem
- Input system can be PCs, cell-phones etc.

3.1.2 Processing subsystem

- It is the core software involved in the device.
- It uses data from input subsystem.
- Analyse and collect the necessary data for processing.
- After processing it provides necessary signals to the output subsystem.

3.1.3 Output subsystem

- Through this part the system communicates with the outside world.
- It should be able to produce the required braille script.
- It should perceive and interpret the signals from the processing subsystem and should produce the desired braille output.

3.2 Non-Functional Requirements

The non-functional requirements of the system are the following

- Safety of hardware.
- Proper cooling mechanism.
- Durable and maintainable hardware.
- Portability.
- Scalable to any text format

3.3 Specific Requirements

Hardware requirements of the system are as follows:

- Minimum 5 Hz refresh rate.
- Communication with devices over USB.
- Micro-controller with minimum 12Hz speed for efficient USB communication.
- At least 40 single usage I/O lines to drive SMAs using I/O lines.
- At least one USB or UART port.

3.4 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

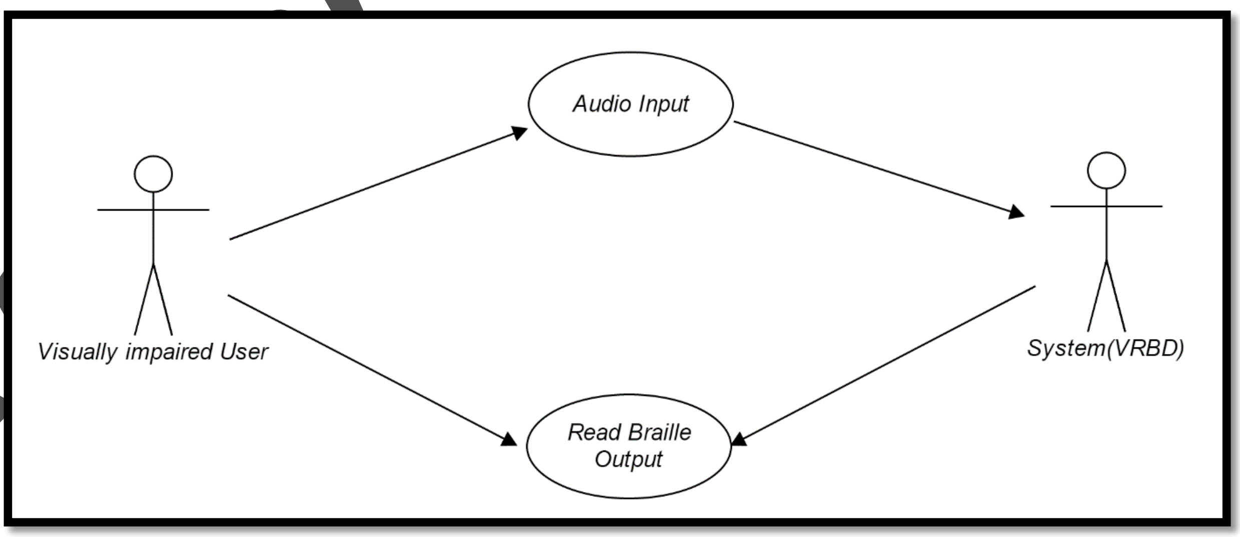

Fig. 3. 1: Use Case Diagram

3.4.1 Use case Descriptors

1) Audio Input

Table 3.1: Use Case Description Table 1

Use Case ID:	UC-1		
Use Case Name:	Audio Input		
Created By:	Project Team	**Last Updated By:**	Project Team
Date Created:	20-10-2014	**Last Revision Date:**	21-4-2015
Actors:	User		
Description:	This use case takes audio input from user		
Trigger:	The user clicks on speak and speaks an alphabet		
Normal Flow:	1.The user start the system 2.Provides input by speaking an alphabet 3.System recognizes the audio 4.System gives user an audio confirmation of the letter recognized		
Frequency of Use:	High		
Special Requirements:	Smartphone with an audio recognition application		

2) Read Generated Braille

Table 3.2: Use Case Description Table 2

Use Case ID:	UC-2		
Use Case Name:	Read Generated braille		
Created By:	Project Team	**Last Updated By:**	Project Team
Date Created:	20-10-2014	**Last Revision Date:**	21-4-2015
Actors:	User		
Description:	This use case displays braille character on the device		
Trigger:	The user gives an audio input		
Normal Flow:	1.The gives an audio input 2.System recognizes the character 3.Displays the corresponding braille character 4.User reads by touching the screen		
Frequency of Use:	High		
Special Requirements:	User must have knowledge about alphabets		

CHAPTER 4

ANALYSIS MODELLING

4.1 Functional Modelling

A function model or functional model in systems engineering and software engineering is a structured representation of the functions (activities, actions, processes, operations) within the modelled system or subject area.

4.1.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects.

DFD Level 0

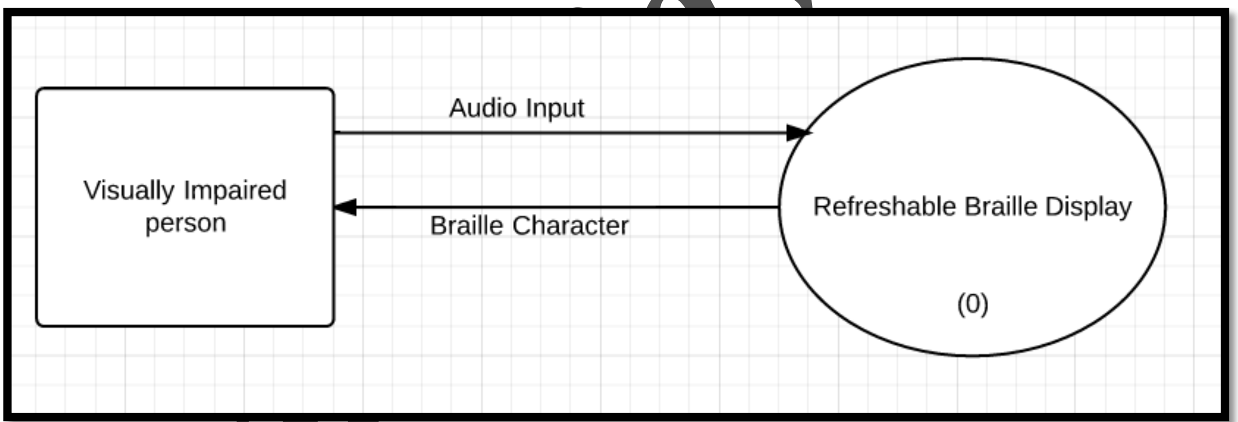

Fig. 4.1: DFD- 0 Context Diagram

Table 4.1: Data Dictionary for DFD-0

Data	Description	Data Type
Audio Input	The user gives audio input of the character to be displayed	Audio
Braille Character	The system generates the Braille character of the input	Braille character

DFD Level 1

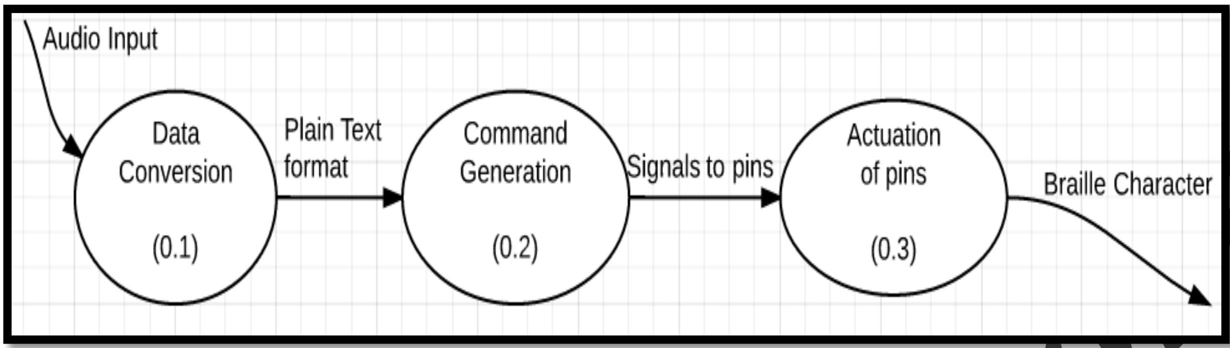

Fig. 4.2: DFD Level 1

Table 4.2: Data Dictionary for DFD level 1

Data	Description	Data Type
Plain Text	The system generates the Plain text of recognized audio	String
Signals to pins	Corresponding signals are sent to pin individually for actuation and retraction	Voltage

DFD Level 2

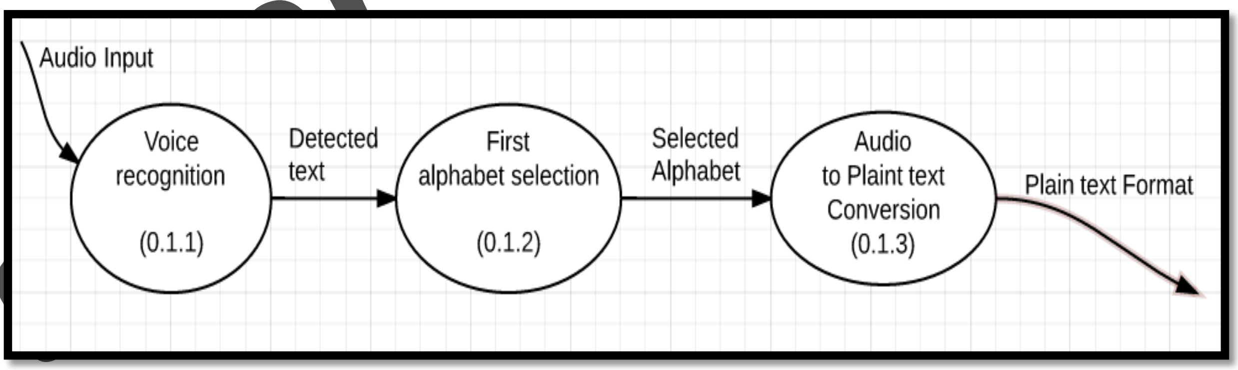

Fig. 4.3: DFD-1: Level 2

Table 4.3: Data Dictionary for DFD level 2 in Fig. 4.3

Data	Description	Data Type
Text	The system recognized audio is converted to text	String
Alphabet	Only first alphabet of detected text is selected for display	String

DFD Level 2

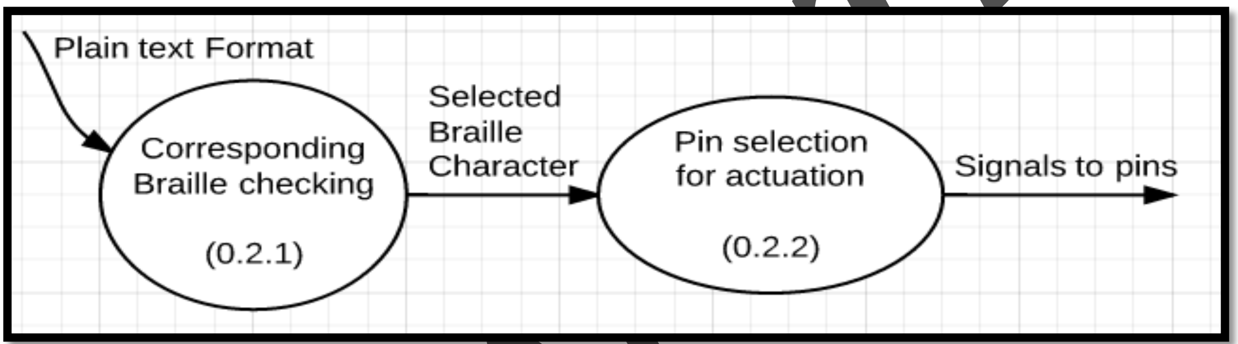

Fig. 4.4: DFD-1: Level 2

Table 4.4: Data Dictionary for DFD level 2 in Fig. 4.4

Data	Description	Data Type
Braille Character	Corresponding Braille character of input audio character is sent to processor for displaying	String

4.2 Time Line Chart:

ID		Task Name	Duration	Start	Finish	Predecessors
1		BE Project	178 days	Fri 08-08-14	Thu 09-04-15	
2		**Documentation**	160 days	Fri 08-08-14	Mon 16-03-15	
3		**Problem Statement Formulation**	**26 days**	**Fri 08-08-14**	**Fri 12-09-14**	
4		Understanding the Project Domain	10 days	Fri 08-08-14	Thu 21-08-14	
5		Presenting the topic	1 day	Fri 22-08-14	Fri 22-08-14	4
6		Finalising the Project Topic	5 days	Mon 25-08-14	Fri 29-08-14	5
7		Understanding the Base Paper	10 days	Mon 01-09-14	Fri 12-09-14	6
8		**Requirements Gathering**	**12 days**	**Mon 15-09-14**	**Sun 28-09-14**	
9		Study and analyse the scope of the project	4 days	Mon 15-09-14	Thu 18-09-14	7
10		Study the type of videos and the nature needed	3 days	Fri 19-09-14	Tue 23-09-14	9
11		Formulate the Problem Statement	5 days	Wed 24-09-14	Sun 28-09-14	10
12		**System Analysis**	**7 days**	**Fri 19-09-14**	**Sat 27-09-14**	
13		Gather the functional requirments	3 days	Fri 19-09-14	Tue 23-09-14	9
14		Gather the non-functional requirements	3 days	Fri 19-09-14	Tue 23-09-14	9
15		Gather the software requirements	4 days	Wed 24-09-14	Sat 27-09-14	14
16		**Modelling**	**3 days**	**Sun 28-09-14**	**Tue 30-09-14**	
17		Model the Architecture of the System	3 days	Sun 28-09-14	Tue 30-09-14	15
18		Model the Circuit Diagram	2 days	Mon 29-09-14	Tue 30-09-14	15
19		**Implementation Finalisations**	**48 days**	**Sun 28-09-14**	**Mon 01-12-14**	
20		Find all the available hardware devices required	11 days	Sun 28-09-14	Fri 10-10-14	15
21		Select the based ones based on there costs and other factors	4 days	Sun 12-10-14	Wed 15-10-14	20
22		Finalizing the SMA wire to be used	11 days	Wed 15-10-14	Wed 29-10-14	20
23		Make the final flow chart and project flow	1 day	Thu 30-10-14	Thu 30-10-14	22
24		Study working in the Latex environment and make the proj	22 days	Fri 31-10-14	Mon 01-12-14	23
25		**Implementation**	**53 days**	**Thu 01-01-15**	**Mon 16-03-15**	
26		Purchase the required hardware	3 days	Thu 01-01-15	Mon 05-01-15	24
27		Test the actuation based on SMA wire	18 days	Wed 07-01-15	Fri 30-01-15	26
28		Coding the microcontroller	13 days	Wed 14-01-15	Fri 30-01-15	26
29		Assembling the hardware with microcontroller	12 days	Tue 03-02-15	Wed 18-02-15	28
30		Testing the prototype	12 days	Thu 19-02-15	Fri 06-03-15	29
31		Scale the prototype to display more characters	6 days	Mon 09-03-15	Mon 16-03-15	30
32		**Test the Project**	7 days	Tue 17-03-15	Wed 25-03-15	25
33		**Find the performance of the project and gauge it**	5 days	Thu 26-03-15	Wed 01-04-15	32
34		**Present and submit the Project**	6 days	Thu 02-04-15	Thu 09-04-15	33

Fig. 4.5: Timeline Chart for BE project (August, 2014 to April, 2015)

4.3 Gantt Chart

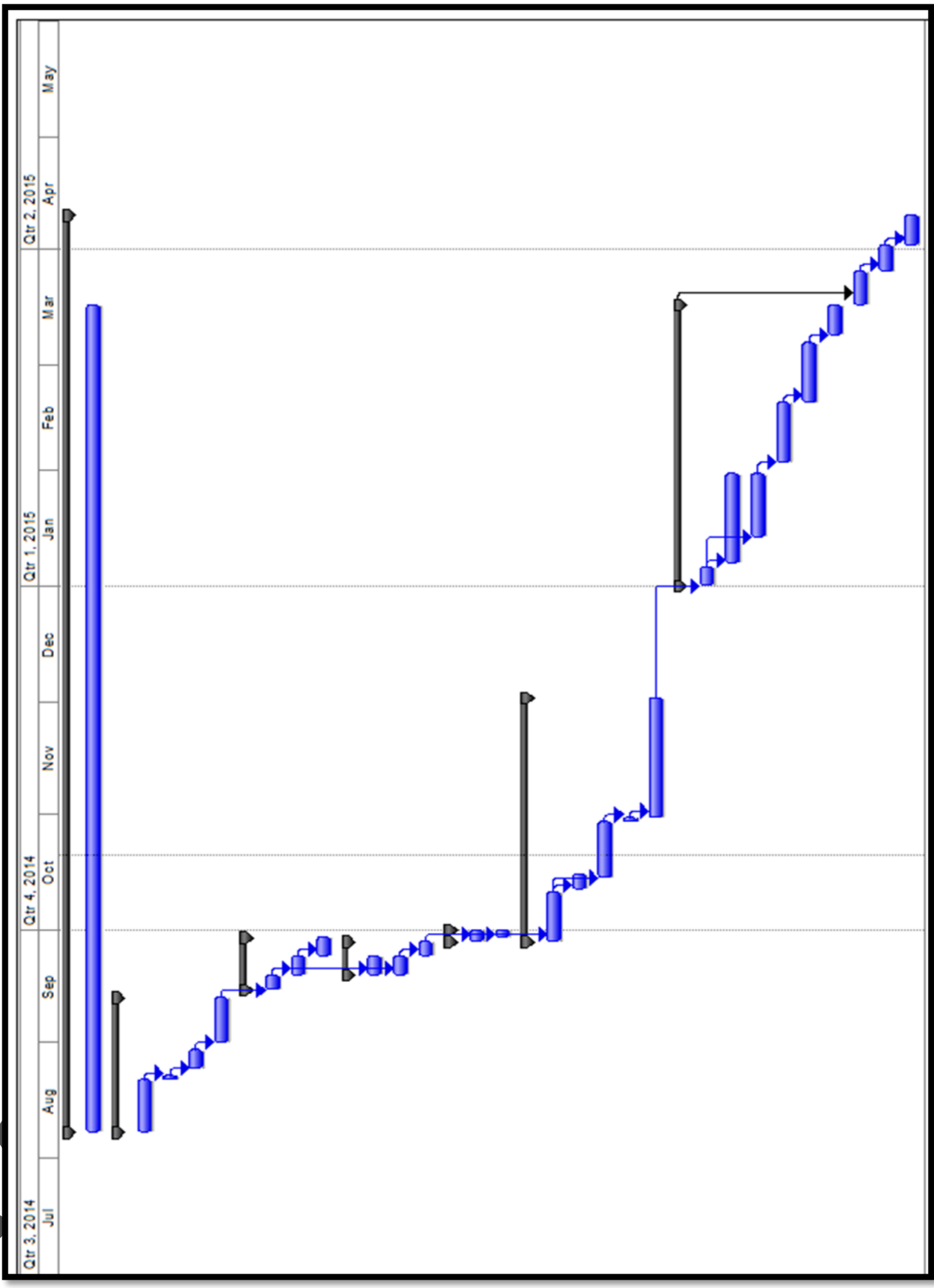

Fig. 4.6: Gantt chart for BE project (August, 2014 to April, 2015)

CHAPTER 5

DESIGN

5.1 Architecture

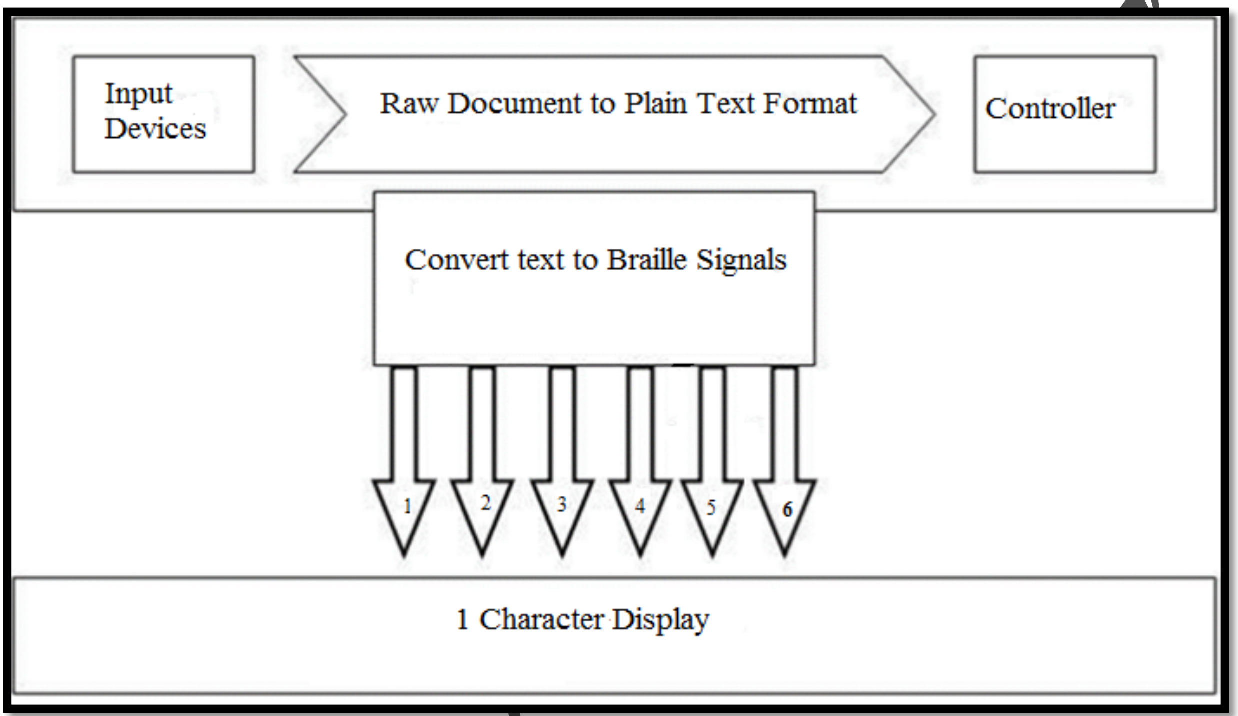

Fig. 5.1: Architecture of Proposed System

As the diagram suggests, the data to be displayed on the tactile display will either be uploaded from PCs, smart-phones etc. or taken from the storage in Refreshable Braille Display System.

Text data will be converted to simple characters stream plain format before upload. The Refreshable Braille Display System will generate commands and load them onto the SMAs in parallel.

5.2 Circuit Design

The Circuit Design gives a brief description of the microcontroller used and the various other circuit components.

5.2.1 Micro-controller

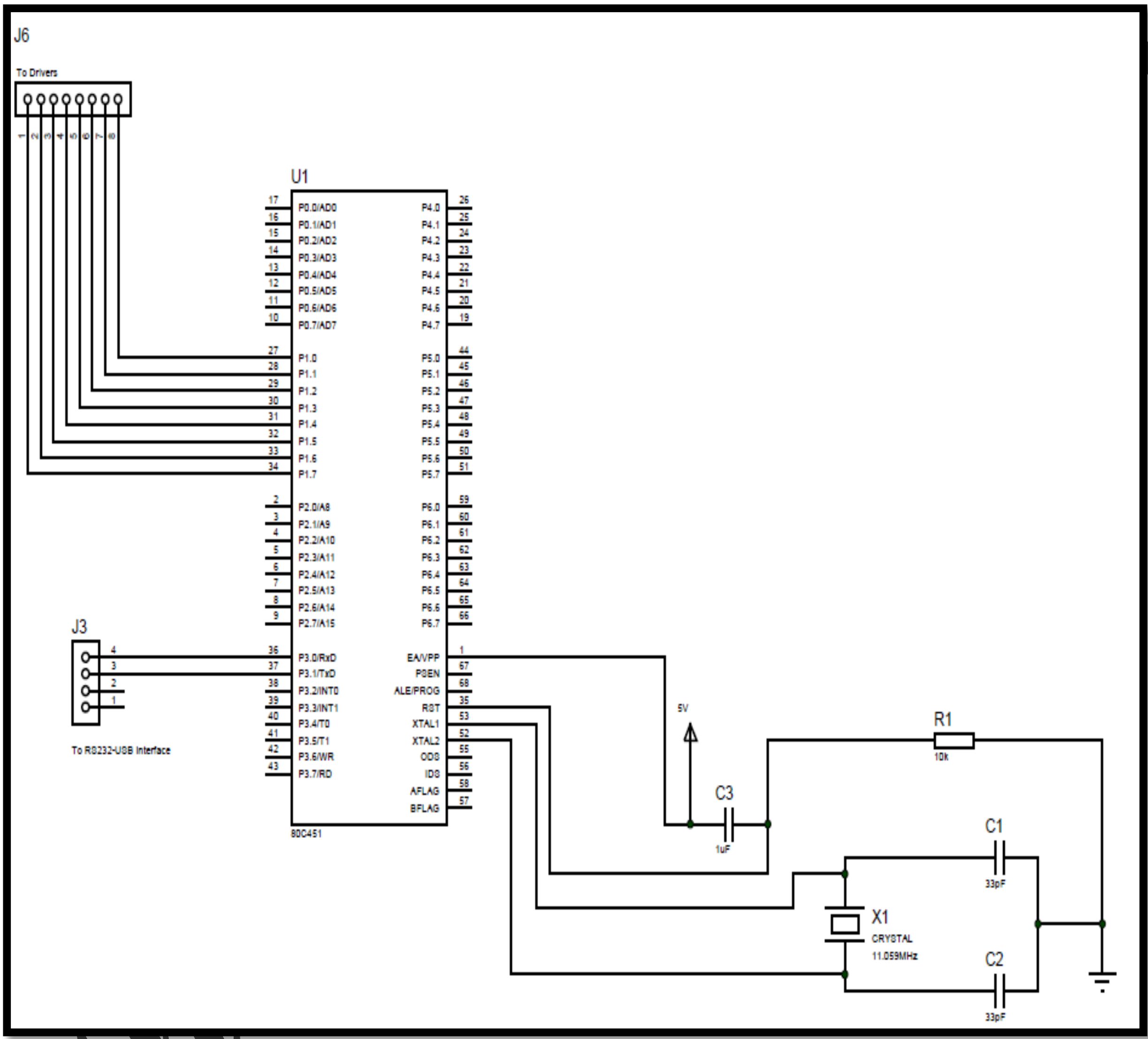

Fig. 5.2: Typical 1 character configuration with 8051 family controller

Here, an 8051 descendant micro-controller is used. Its port 27-34 are used to drive the SMA pins. Pin 36 and 37 are RXD and TXD respectively. These pins are used to communicate with host device. A USB to UART circuit is connected to these pins. On the other end of this circuit the host connects via USB. Every 8051 microcontroller requires an external crystal connected across XLAT1 and XLAT2. In this case an 11.0592 MHz crystal is used for best time outputs

5.2.2 Drivers for SMA

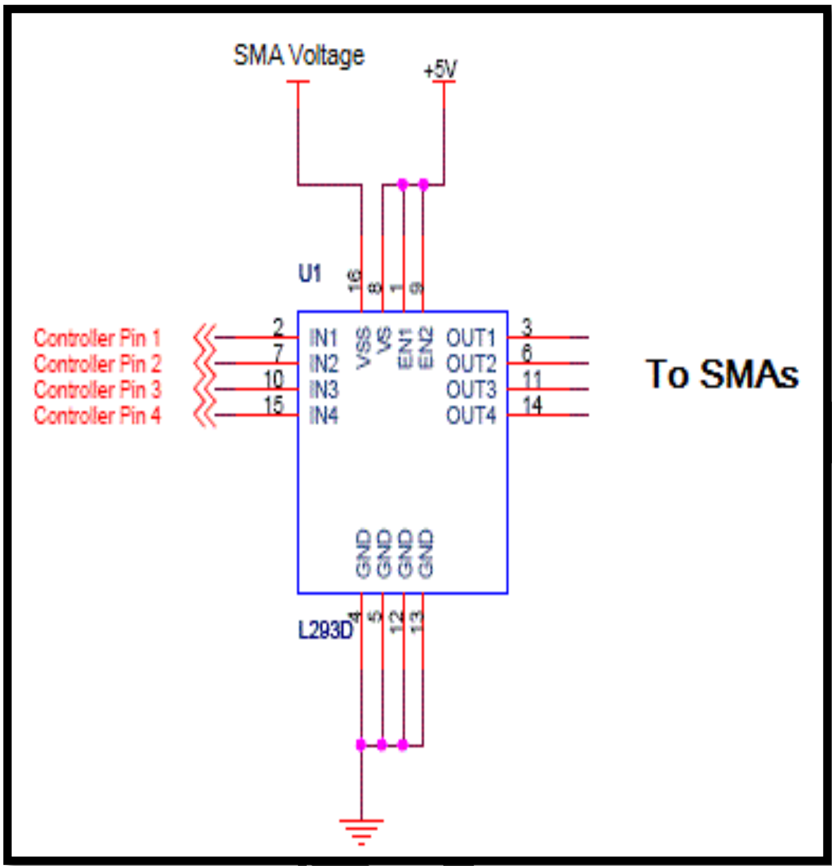

Fig. 5.3: Drivers for SMA actuation

SMA wires require electric current to be applied to them for actuation. This required voltage and current can be different from that the microcontroller is able to source. Hence drivers are used to supply voltage and current to the SMA via the Microcontroller. These drivers are usually used in motors for similar purposes.ULN2003 and L293D ICs can be used as drivers.

5.2.3 RS232 to USB connector:

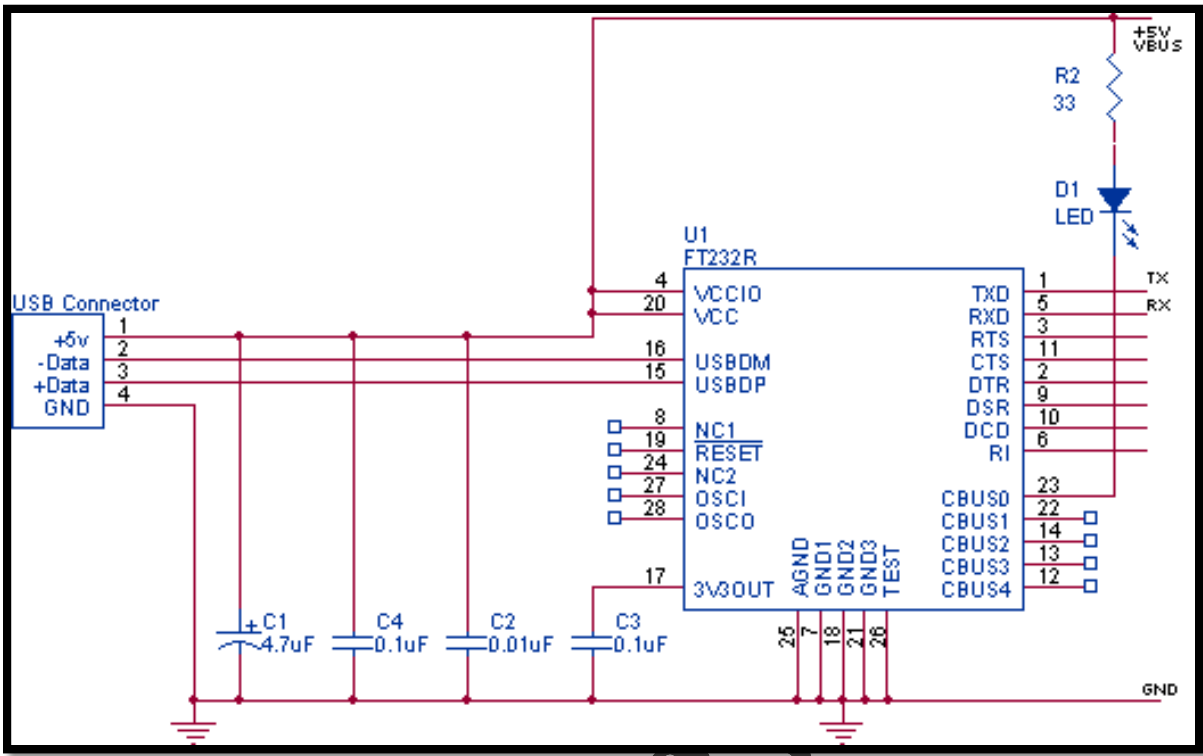

Fig. 5.4: RS232 to USB interface

Most of 8051 family microcontrollers support UART. With UART and RS232 outdated, there is a need to create an interface from USB to UART. Fig. 5.4 shows how MAX232 IC is used to interface UART to RS232. This circuit is used to program many programmable ICs in ISP mode. FT232R IC can be used to further interface RS232 to USB. This allows communication between hosts and the device.

5.2.4 Actuator Configuration

Nickel-Titanium (NiTi) SMA wire is chosen as the actuator element, for its advantages discussed in [14]. Fig. 5.5 gives a schematic of the actuator configuration. Different from a regular spring-biased SMA wire actuator, the proposed actuator uses the SMA wire element in an inverse-U configuration. With such a configuration, all the terminals are placed on a single plane, simplifying electrical connections. Further, it allows the wire to have removable connections to the structure, allowing easy assembly and disassembly.

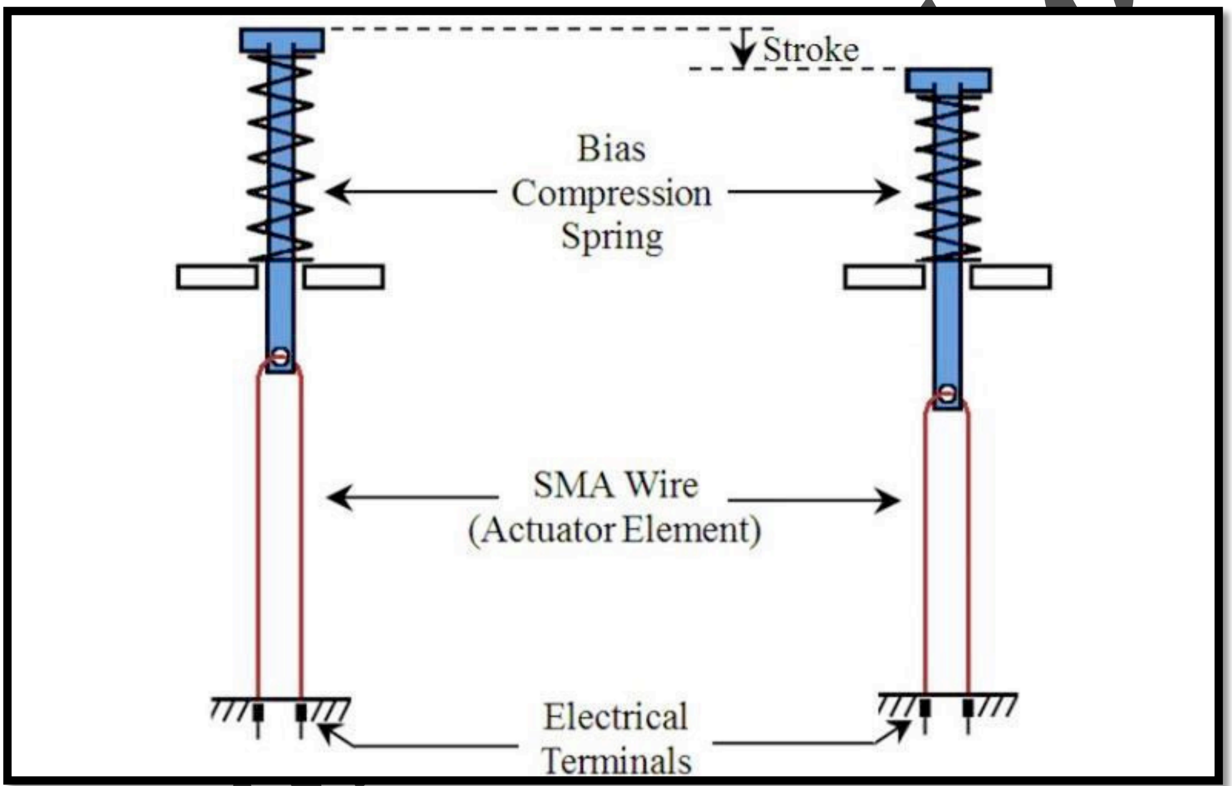

Figure 5.5: Actuator Configuration

First order calculations were performed to determine the natural length *Ln* of the SMA wire, stiffness K_s of the bias compression spring and the stress σ_A and strain ε_A in the actuated SMA wire.

Following design equations are used in [1]:

$$\boldsymbol{K_s x = 2A_w \sigma_M} \tag{1}$$

$$\boldsymbol{K_s(x + \Delta x) = 2A_w \sigma_A} \tag{2}$$

$$\boldsymbol{L_n(\varepsilon_M - \varepsilon_A) = \Delta x} \tag{3}$$

$$\boldsymbol{K_s(x + \Delta x) \geq F_R} \tag{4}$$

where,

- σ_M and ε_M are the stress and strain, respectively, in unactuated SMA wire
- A_w is the cross-section area of the SMA wire
- Δx is the Braille dot stroke
- x is the compression of the spring in the default state
- F_R is the minimum desired resisting force
- ε_M has been purposefully kept below 2.5%, as recommended in [15 16] for an average actuator life greater than 106
- Δx has been chosen as 0.7 mm for a stroke of 0.6 mm, to compensate effects of functional fatigue instability
- Further, K_s has been constrained by practical considerations of spring material, size and manufacture limitations

CHAPTER 6

IMPLEMENTATION

6.1 Methodology

The application used the property of shape memory alloy for lifting of the Braille pins. As shown in Fig. 6.1 the wire has two states namely Austenite and Martensite. Martensite is the default state of the wire. When wire is attached to a load (Braille pins and spring) it goes to De-twinned Martensite state having high stress and strain values. When electricity is applied on SMA wire, it gets heated and due to which the wire goes to Austenite state. This state is used for pulling the braille pins down. When the electric supply is turned off, the wire starts cooling and hence it goes back to the default Martensite state. Hence the pin is lifted up again by the spring connected to it.

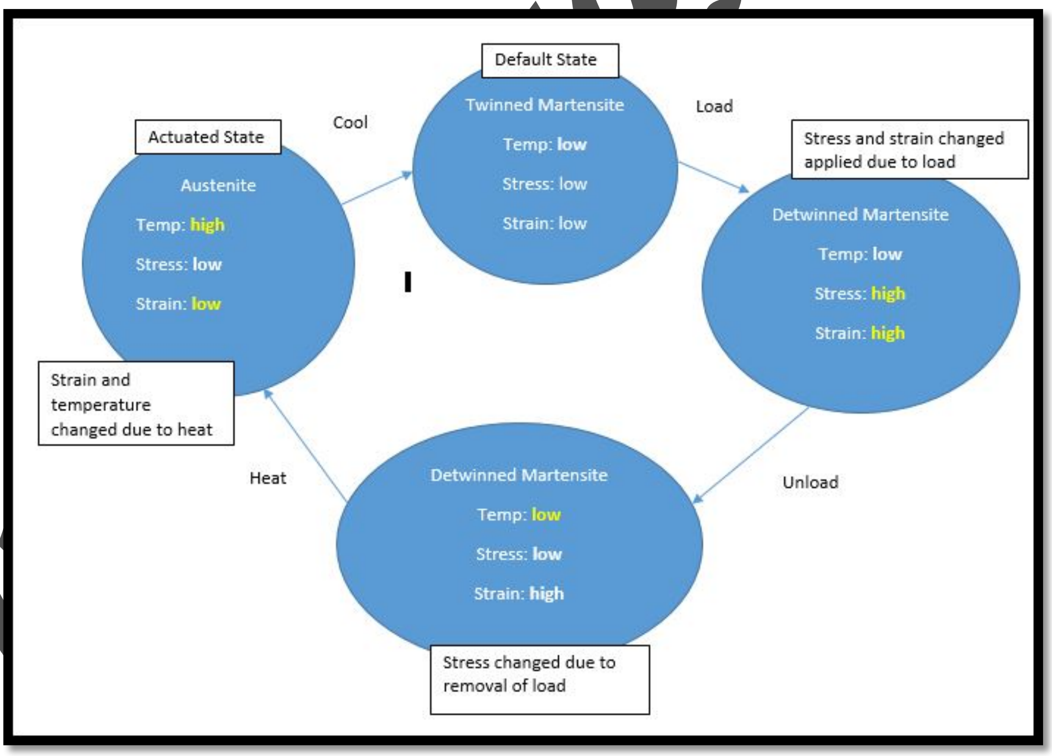

Fig. 6.1: SMA wire working

6.2 Calculations

Important parameters such as spring stiffness ($\boldsymbol{K_s}$) were calculated using the equations (1), (2), (3) and (4) and using trial and error method.

The finalized spring configuration is,

- Diameter of the spring: 0.3 mm
- Outer diameter of the spring: 1.4 mm
- Free length of spring: 5 mm
- Number of active coils: 6

The surface configuration is,

- Height: 40 mm
- Width: 6.4 mm
- Hole Diameter: 1.4 mm
- Distance between consecutive holes: 2.5 mm

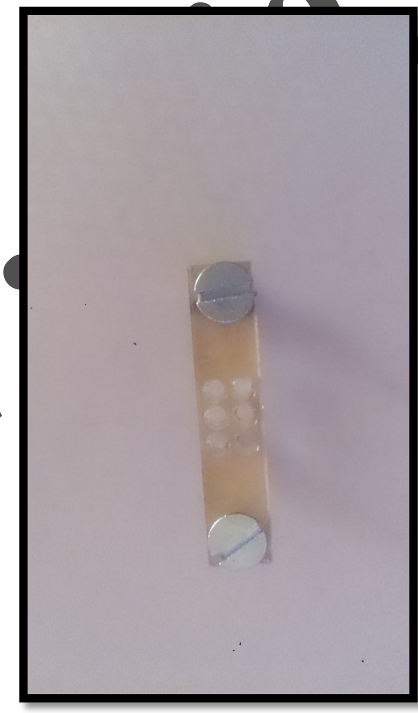

Fig. 6.2: Surface Configuration

The finalized pin configuration is,

- Height: 11.75 mm
- Diameter: 1.3 mm

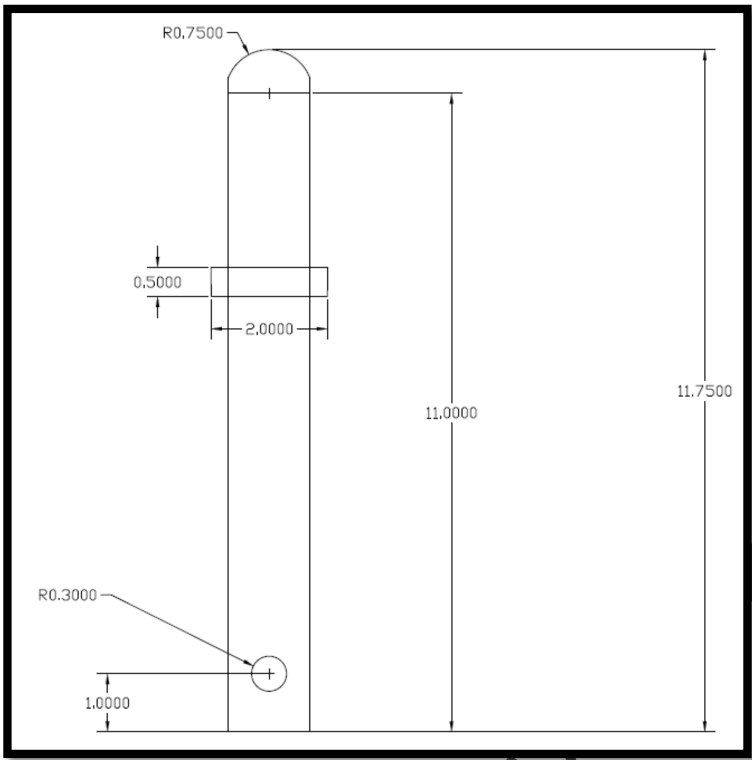

Fig. 6.3: Pin configuration

6.3 Pseudo Code

The following pseudo code is used to control the braille pins. The code is deployed into an Arduino chip which either start or stop electric flow to the pins.

```
Set  Baud Rate to 9600
Set P0 as output port
P0 = 0xFF
// Pin Layout :
// Bit 4 Bit 5
// Bit 3 Bit 6
// Bit 2 Bit 7
// Bit 1 (unused)
// Bit 0 (unused)
char c;
while(true){
c = recieveUART();
```

```
if(c=='a'){
}
else if(c=='b'){
}
else if(c=='c'){
}
else if(c=='d'){
}
else if(c=='e'){
}
else if(c=='f'){
}
//.
//.
//.
//.
//Till 'z'
else if(c=='z'){
}
}
P0 = 0x08
P0 = 0x18
P0 = 0x0C
P0 = 0x0E
P0 = 0x0A
P0 = 0x1C
P0 = 0x2B
```

6.4 Snapshots

Fig. 6.4 shows interface for the android voice recognizer which takes voice input and feed it to the Arduino chip:

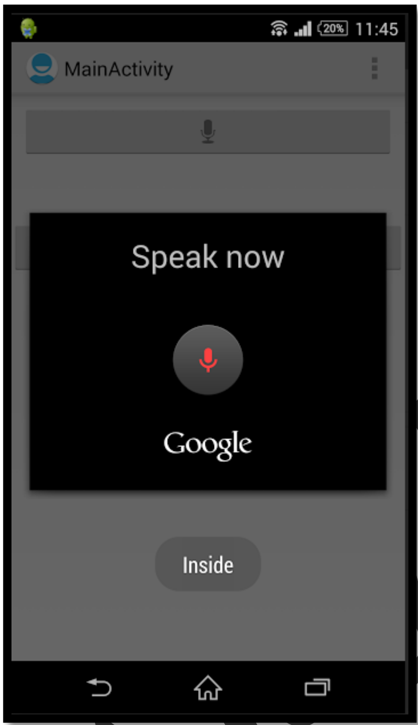

Fig. 6.4: Voice recognizer user interface

Fig. 6.5 shows the interface when an audio input is given:

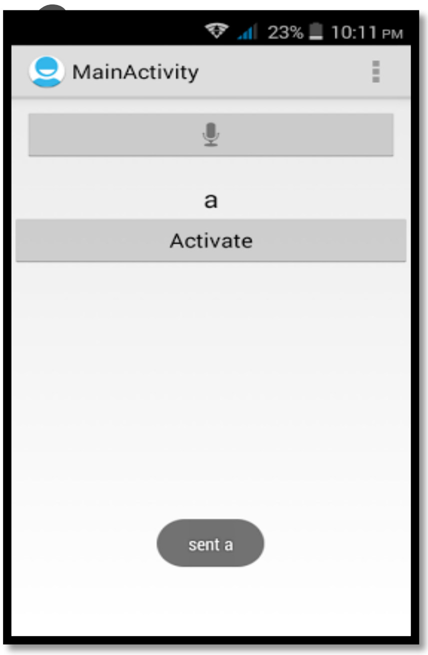

Fig. 6.5: Voice recognizer input feedback

Fig. 6.6 shows the arduino uno processor used fitted into a plastic case:

Fig. 6.6: Arduino Uno processor

Fig. 6.7 shows the mobile phone connected to Arduino Uno:

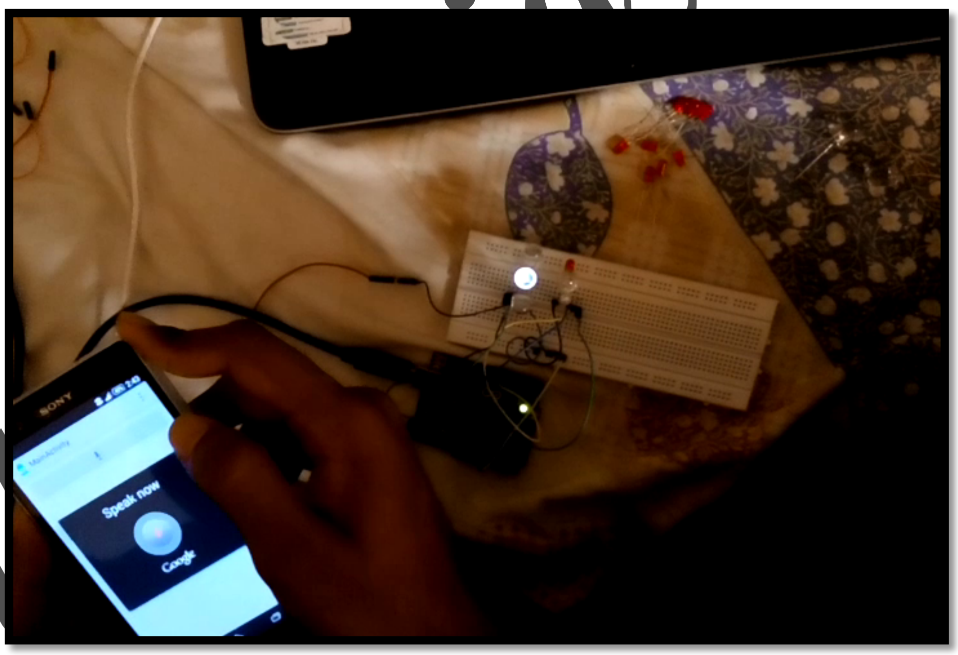

Fig. 6.7: Smartphone-Arduino configuration

Fig. 6.8 shows arduino connected to braille pins via two L293D IC boards:

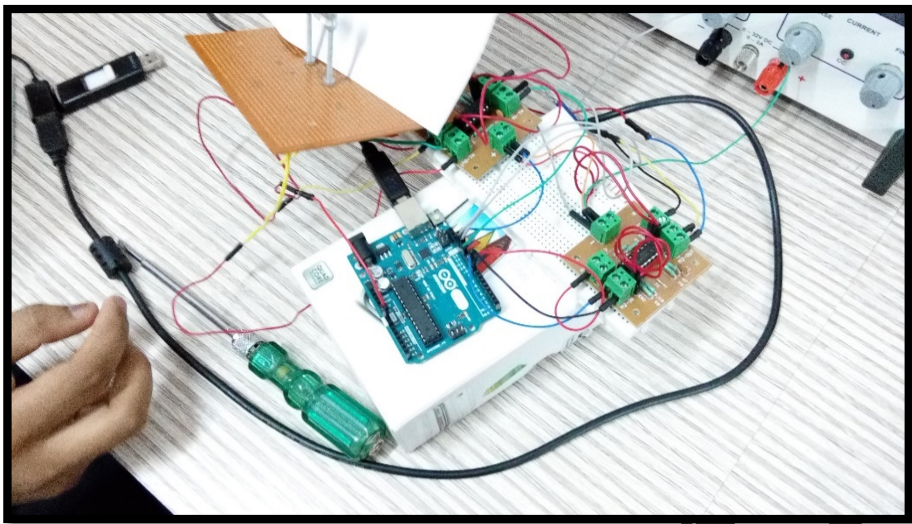

Fig. 6.8: L293D connection with Arduino and Braille pins

Fig. 6.9 shows the respective braille output is as shown in the diagram below:

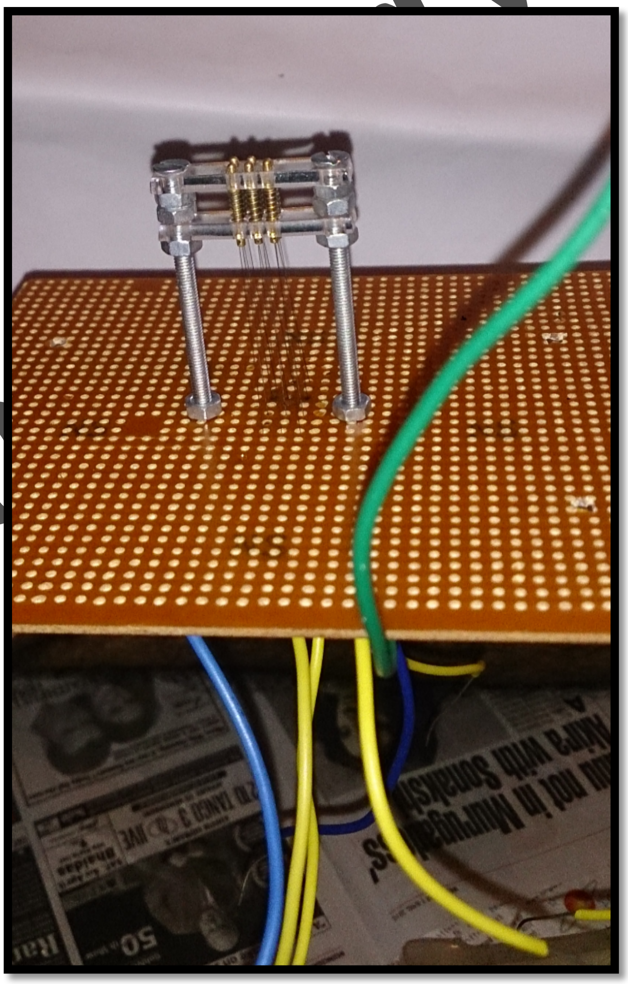

Fig. 6.9: Braille Display Device- Output

CHAPTER 7

TESTING

7.1 Test cases

The test cases performed on the hardware and software application are as follows:

1) British English needed by Voice Recognition application.

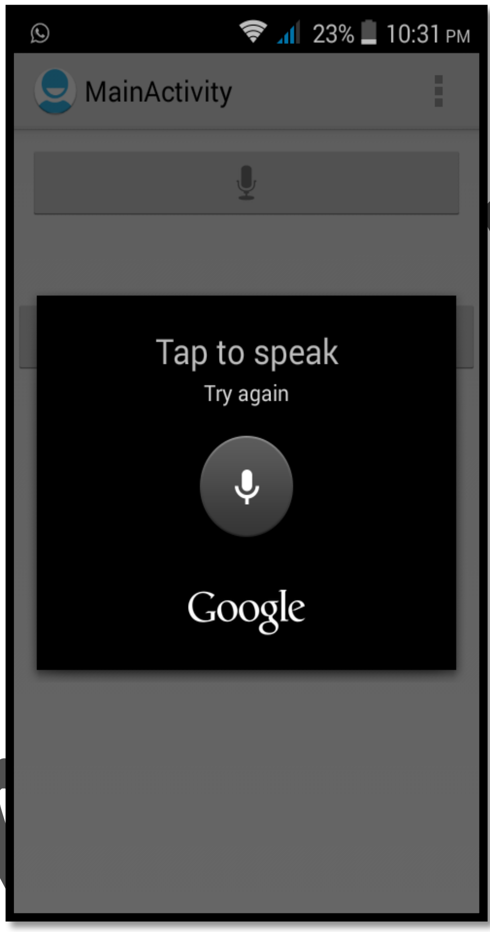

Fig. 7.1: Testing invalid input

As shown in Fig. 7.1, if an invalid voice input is given the application asks the user to speak again.

2) If a wrong character is detected by the application

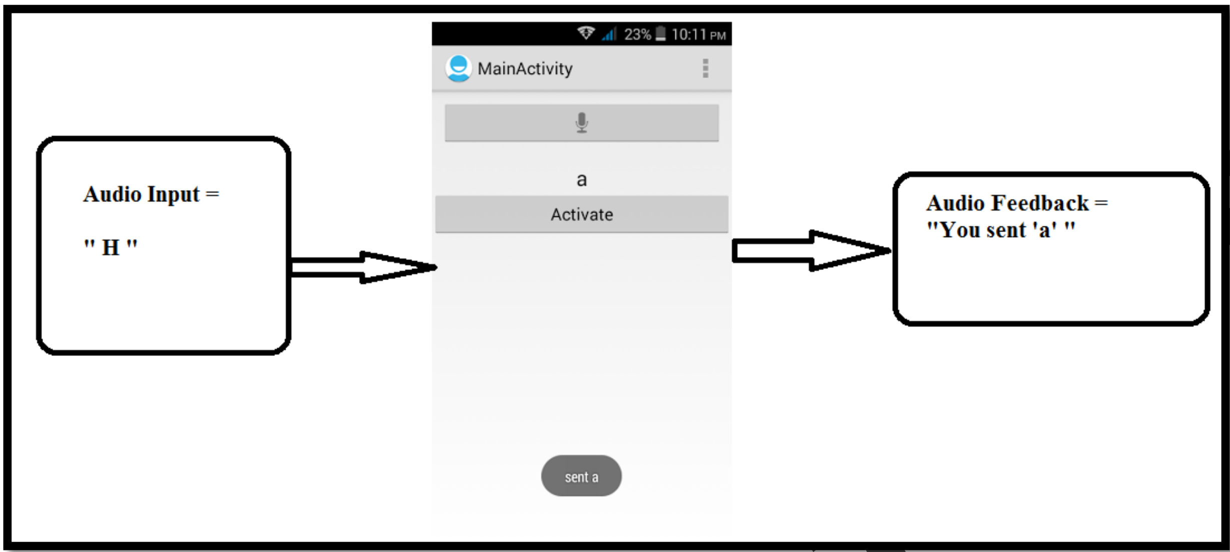

Fig. 7.1: Audio Feedback Mechanism

As shown in Fig. 7.2, an audio feedback mechanism is used for verifying if a correct alphabet is recognized by the voice recognizer

3) Power Supply should be 5V, 1.7A

Table 7.1: Test Case 3

Test case	Result
Power Supply > 5V	SMA wire overheating
Power Supply < 5V and >3V	Efficient working (Best Case)
Power Supply <3V	Lesser Contraction
Current > 1A and < 1.5A	4-5 pins contract
Current > 1.5A and < 1.7A	5-6 pins contract
Current < 1A	1-3 pins contract

4) Spring should have 6 number of turns

Table 7.2: Test Case 4

Test case	Result
Number of turns > 6	SMA contracts but does not regain
Number of turns < 6	SMA doesn’t contract
Number of turns = 6	SMA contracts and regains original length

CHAPTER 8
RESULTS AND DISCUSSIONS

Functional prototype is developed and results prove that the designed actuator and packing help meet the desired level of functionality. Device specifications, as observed, are tabulated in Table 8-1.

Table 8.1: Performance and Specifications

Parameter	Performance
Braille Dot Stroke	0.7 mm
Inter-Dot Spacing	2.5 mm
Dot Base Diameter	1.4 mm
Dot Profile	Dome(0.75 mm radius)
Operating Temperature Range	0-60 ºC
Max. Power Requirement (per cell)	5V,1.7A

Result Analysis:

- Refresh time for a pin is 3 to 5 seconds.
- Accuracy of characters displayed in our system as compared to original Braille Script is 60 to 70%.
- Use of the Google voice recognizer makes the character recognition accuracy very high.

CHAPTER 9
CONCLUSIONS & FUTURE SCOPE

In the challenge of developing affordable refreshable Braille displays, developing a suitable low-cost alternative to piezoelectric bending actuators is the key. Although prior efforts have identified multiple such alternatives, none has translated into successful products. SMA based actuation, usually used in micro and miniature actuation applications, is among the proposed alternatives. It carries certain advantages making it suitable for Braille displays. However, certain challenges in actuator design and packing, actuation frequency, power requirement etc. have limited prior attempts. After these challenges were successfully tackled, an audio input service for the displaying of characters as per the audio was added.

The Voice based Refreshable braille Displays (VRBDs) can be viewed as the future of learning tools for the visually impaired students. The Braille literacy is around 30% of the entire blind population around the world. This system provides an interesting way of learning the braille language which indirectly promotes Braille literacy.

The implemented system, VRBD is very affordable when compared to already available RBDs in market which cost about five times of that of our SMA based RBD. Also there is the added functionality of audio input for braille character generation so that the visually impaired user does not have to depend on any other individual for learning. This makes it a self-learning Braille script tool for the visually impaired who are willing to learn braille themselves.

In future, developments in SMA based actuation, for example, improvements in materials and processes and evolution of joining and attachment methods may help in further reduction of display cost and improvements in performance. Also the dependency on the smartphone application can be eliminated if an efficient standalone voice recognizing technology is developed hence reducing the costs even more.

BIBLIOGRAPHY

[1] Anshul Singhal et al. "Application of shape memory alloy based actuation for refreshable display of braille" *IEEE-2013*

[2] Ryles et al. "The Impact of Braille Reading Skills on Employment, Income, Education and Reading Habits" *Journal of Visual Impairment and Blindness, May–June 2003*

[3] Tetzlaff et al. "Electromechanical Braille Cell" *United States Patent -1981*

[4] Strobel et al. "Industry Profile on Visual Impairment" *Rehabilitation Engineering Research Centre on Technology Transfer, NY, USA. 2004*

[5] Blazie D et al. "Refreshable Braille Now and in the Years Ahead" *The Braille Monitor 43 (1), January 2010*

[6] Vidal-Verdú et al. "Graphical Tactile Displays for Visually-Impaired People" *IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 119–130, 2007*

[7] B. G. Xu et al. "Pattern Recognition Of Motor Imagery EEG Using Wavelet Transform" *J. Biomedical Science and Engineering, pp. 64-67, 2008*

[8] Huber, J. E., et al. "The Selection of Mechanical Actuators Based on Performance Indices" *In Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 453(1965), pp. 2185–2205, October 1997*

[9] Parris S. Wellman et al. "Mechanical Design and Control of a High- Bandwidth Shape Memory Alloy Tactile Display" *International Symposium on Experimental Robotics, Barcelona, Spain. June 1997*

[10] Damin J. Siler et al. "Variable stiffness mechanisms with SMA actuators" *Industrial and Commercial Applications of Smart Structures Technologies, San Diego, CA, February 25, 1996*

[11] Strobel, W. A., et al. "Industry Profile on Visual Impairment" *Rehabilitation Engineering Research Centre on Technology Transfer, NY, USA- 2004*

[12] Ivan Poupyrev et al. "Actuation and Tangible User Interfaces" *TEI'07, Baton Rouge, LA, USA, pp. 15-17, Feb 2007*

[13] Taylor, P. M., et al. "A Sixty-Four Element Tactile Display Using Shape Memory Alloy Wires", *pp. 163-168, 1998*

[14] Fernando Vidal-Verdú et al. "Graphical tactile displays for visually impaired people" *Neural Systems and Rehabilitation Engineering, IEEE Transactions, pp. 119-130, March 2007*

[15] W. Huang "On the selection of shape memory alloys for actuators" *School of Mechanical and Production Engineering, Singapore, April 2001*

[16] Gill, J. M., et al. "Braille Cell Dimensions". *RNIB Digital Accessibility Team, London, 2008.*

[17] Runyan et al. "EAP Braille display needs and requirements", *WW-EAP Newsletter 11(1), pp. 9–10, 2009*