Braille Reader

GROUP MEMBERS:

Munib Ahmed (I15-0811)

Muhammad Montaha (I15-0476)

Aziz Hassan (I15-0423)

SUPERVISORS:

Dr. Ataul Aziz Ikram

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), ISLAMABAD

Developer's Submission

This report is being submitted to the Department of Electrical Engineering of the National University of Computer and Emerging Sciences in partial fulfillment of the requirements for the degree of BS in Electrical Engineering.

Developer's Declaration

"We take full responsibility of the project work conducted during the Final Year Project (FYP) titled "*Braille Reader*". We solemnly declare that the project work presented in the FYP report is done solely by us with no significant help from any other person; however, small help wherever taken is duly acknowledged. We have also written the complete FYP report by ourselves. Moreover, we have not presented this FYP (or substantially similar project work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

We understand that the management of Department of Electrical Engineering of National University of Computer and Emerging Sciences has a zero-tolerance policy towards plagiarism. Therefore, we as an author of the above-mentioned FYP report solemnly declare that no portion of our report has been plagiarized and any material used in the report from other sources is properly referenced. Moreover, the report does not contain any literal citing of more than 70 words (total) even by giving a reference unless we have obtained the written permission of the publisher to do so. Furthermore, the work presented in the report is our own work and we have positively cited the related work of the other projects by clearly differentiating our work from their relevant work.

We further understand that if we are found guilty of any form of plagiarism in our FYP report even after our graduation, the University reserves the right to withdraw our BS degree. Moreover, the University will also have the right to publish our names on its website that keeps a record of the students who committed plagiarism in their FYP reports."

_____	_____	_____
Munib Ahmed	Muhammad Montaha	Aziz Hassan
BS(EE) i15-0811	BS(EE) i15-0476	BS(EE) i15-0423

_____

Certified by Supervisor

_____

Verified by Plagiarism Cell Officer

Dated: _____

Abstract

Education not only plays an important role in the progress of the country but also helps in mind development of the individual. As in our modern world, most of books are available in soft-document. This education is bounded to people of clear vision. Our society also includes visually impaired people who do not have access to the education. This education is mostly in the soft document form. Due to disability they feel difficulty in reading the soft document. For this purpose, visually impaired people use braille to read the document.

Our approach for this project is to convert the English characters into the braille code and the braille code will be understood with the sense of touch. It will be also able to generate the audio signal of that character.

We hope to provide assistance to visually impaired people with this product since there is little to no support for the disabled community and they have to heavily rely on their family for the most basic of tasks.

Acknowledgments

First and foremost, I am grateful to God for the good health and wellbeing that was necessary to complete this book.

I have to thank my project supervisor, Dr. Ataul Aziz Ikram. Without his dedicated assistance and advice in every step of the way, this project wouldn't have been accomplished. I would like to thank you very much for your support and understanding over the past year.

Getting through my Final Year Project required more than just academic support, and I have many, many people to thank for listening and tolerating me over the course of this program. I cannot begin to express my gratitude and appreciation for their friendship and support.

Most importantly, none of this would have been possible without my family. To my parents – this dissertation stands as a testament to your unconditional love, support and encouragement.

Table of Contents

Table of Illustration

1) Chapter

Introduction and Background

1.1 Introduction

In this section, we have explained previous approaches adopted by others and the problems in their design. This section gives the brief description of the previous approaches and the problem statement of our project.

1.2 Introduction & Background

In 1821, Universal Braille System was introduced Louis Braille. Braille System is the most adopted system for visually impaired people. Braille is not a language but a system converts characters into a combination of 6 dots. Thus, each character of English is mapped to a specific pattern. Using braille alphabets, a blind person can sense/read character by touching it. In order to understand Braille characters, one must have the knowledge of braille system.

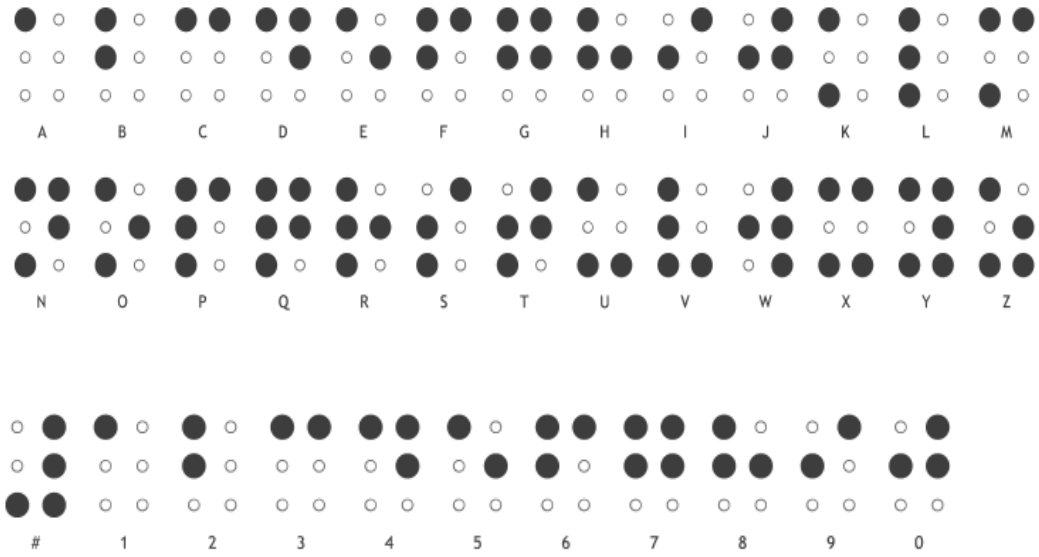

Figure 1.1: The Braille System Numbers and Alphabets

1.3 Previous Approaches

Over the course we have studied six-nine publications which overall deal with the problem in a comparatively similar manner.

In Paper [1] they discussed some common methodology used by blind people to understand the document i.e. Braille System. Later, they explained some common devices which are already being used to help visually imparted people i.e. Screen Reader, Finger Reader, Braille Note taker, Braille Printer, Screen Magnifier and E-book reader.

In this research they have used touch display (Android or smart phone), which can act as landmark for basic navigation on smart phone screen. Blind Reader converts touched words into audible signal, thus it helps the reader to understand the word they are being touched. A visually impaired (blind) person touches the word which is then voiced by the system. Thus, it helps blind (visually imparted) people to understand the soft document easily.

The system takes the whole document as a collection of words. Then each word is distinguished that user is touching. A document containing text is opened in this display in an organized format i.e. pdf format, can be mapped spatially. From the selected document, word which are selected in document is converted in the audible signal, with the help of a text-to-speech synthesizer. It provides a basic level model that gives the sense of reading a document with the help of touching the screen and by hearing.

For this algorithm, document is first converted into the require format then user can touch and convert the signal to audible format. Pictorial information and graphics are filtered out, before reading the document. This algorithm also uses the Auto-scroll, by the help of motion sensor. As the device is tilted the document is scrolled up and down, accordingly. (Shahed Anzarus Sabab)

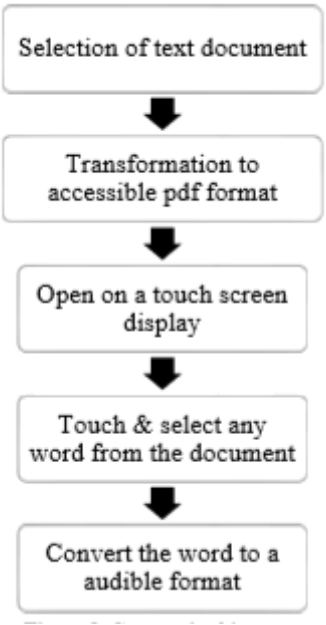

Figure 1.2: Text to Audio Flowchart

In Paper [2] its starts by exploring importance of education in visually imparted people. Then they explain methods that are already being used for teaching visually imparted students i.e. wooden board with embossed dots using Braille System. They also explain the basics for Braille Systems. They gave a brief description of the components they are using, for example touchscreen keyboard, Microcontroller, Power Supply ranging from 1.9 to 3.6 volts, wireless module and tactile display (known as Braille Cell Display, actuating Mechanism for Braille Cell required to be constructed for producing tactile feedback to student).

A keyboard is provided to a tutor; keyboard is connected to a microcontroller which not only process the data entered by the keyboard but also sends the braille System code to the tactile device with the help of the wireless module, so that the student is able to read the Braille code. Thus, one tutor can teach more than one student with the help of this device, without any braille System skill. This device use wireless module so tutor can teach all the visually imparted students, at once.

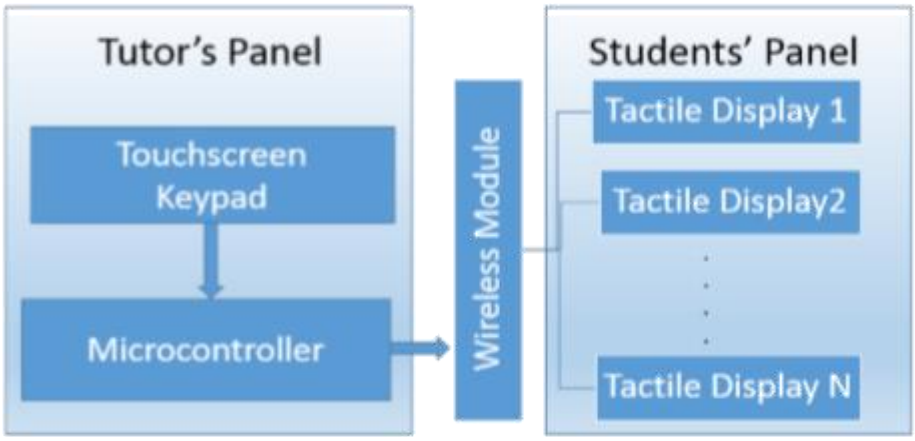

Figure 1.3: Tutor Student Panel Block Diagram

Once the device is operating, microcontroller waits for tutor to enter the data from the keyboard. As the data is entered microcontroller waits for the confirmation of the data, as tutor press the button microcontroller confirms the data, and alphabet is provided to all the students, through wireless module, where Braille System Code is generated corresponding to the alphabet, which was received. Thus, specific Braille code is embossed on braille cell. This process is repeated again and again. (Shreya Gandhi)

In paper [3], braille system is just like a TactoBook which is a novel device that allow the visually impaired people to read any text document using a portable electronic refreshable Braille touch display.

Now day's regular print books are commonly found in electronic format and the current trend is of a digitized system. EBook portable readers already exist which consist of some hardware device that's comfortable to hold and can easily be carried by the user. There is no system perfectly equivalent for the visually impaired (blind).

Blind people can read the text or document by touching the hands to the pattern of raised dots. Each braille cell is made of six dots which are arranged in 2 columns with three dots each. With the help of dots 64 combinations can be represented. The TactoBook system is a computer based system that converts the text file into the

Braille code. The resulting Braille version is converted into the new file. There is file upload function the file can be upload in pdf, txt, docx format. User defined text can also be converted in Braille language.

This paper uses the mechanism of piezoelectric linear motor (TULA-35). It consists of a rod, slider and ceramic transducer. Upon polarizing transducer, vibrations are generated in rod. These vibrations/movement are transformed into the linear motion of slider. (Ramiro Velázquez)

In paper [4] it discussed about braille books which are imperfect and their implementation is time taking process in which each of the letter has to be translated into the braille system alphabet, six contact points which involving stapling, cutting and gluing the labels on a paper. Braille books also become uneasy to hold and carry when the book is once read it becomes less interesting and it does not last very long.

This paper introduces tacto book system that is efficient device which translate eBooks to braille language that help the blind people to easily access eBooks. The translated eBook are stored USB and then read with the combination of dots. This device can and cannot depend on the computer as the data is stored in the USB so it makes a portable device.

In this paper braille system consist of 10 braille cell. Braille system follows the piezoelectric ultrasonic actuation approach also named as TULA piezo-motors. TULA piezo-motors has a shaft, slider, and piezoelectric ceramic disk. Ceramic disk consist of the rigid part and some sort of transducer. Whenever, the signal is applied

vibrations are generated by the transducer it creates up and down movement. This movement is then transferred to shaft, thus the braille pin moves.

When implementing the 10-cell Braille terminal, about 60 of the TULA piezo-motors are required to make the device. Piezo motors will have to be placed independently so that their motion does not affect the other motors and piezo motors must be within

the braille cell in order to access convenient power. To obtain the best performance of this piezo-motors, voltage and frequency characteristics are very important. (Ramiro Vel´azquez)

1.4 Problem Statement

Every year, a large number of visually impaired people cannot get proper education because of the disabilities. Quality books are usually not available in the braille system, but this System is only source for them to access the documents. Technical and informative documents are usually available in soft copies which they can't read. Due to which they can't access soft document, thus, visually imparted people lack in field of technology. In order to keep them up to date with modern technology, they must access soft documents. That's why we build a low cost, portable and reliable device for visually impaired people.

This design aims to produce a product that will perform vibrations according to Braille System alphabets. Thus, it will help visually imparted people to understand soft-document without the help of anyone.

2) Chapter

Design & Implementation

Our objective was to design a portable device for the visually impaired people that will not only provide the braille system but also provide the audio signal to that will help the blink one to access the soft-documents.

2.1 Solution

A braille converter is a simple yet very help for the visually impaired one. Our proposed solution is to construct a device that will read the soft- documents and convert it into braille code and audio signal.

Our device will also provide different features. Some of the features are as follows:

- Maintaining multiple files and formats: Our device will be capable of storing different documents. So user can choose between the documents saved in the device and device is capable of reading from different format for example text, word and PDF files.
- Smart reading features: User (visually impaired person) can move forward and backward in the document, which will help the user to read the document from where they want to read.
- Deletion of the file: User can also delete the useless document, which are read.
- Device is also capable of toggling between audio signal and the braille code.

For this purpose, we have used Raspberry Pi, which is capable of dealing with multiple processes at the same time. All the programming for this project is done on python language.

Audio signal is generated using build in raspberry pi module names as eSpeak. This module is capable of converting any text into audio.

For the document selection, we have used dirpath in walk module, which allow as to move and select documents from the folder. Different format files are also handles using the respective instructions.

To generate the braille code, we have used a set of linear actuator (linear motor) which moves up and down, and generated the respective pattern. These linear actuators are controlled by raspberry pi GPIO pins, for this purpose we have used another module available, known as GPIO module and the time duration is controlled by help of time module.

For the power we have used an external battery which is interfaced with the linear motors using a Motor Driver IC (ULN2003A).

Thus, all of the programming is done in python using raspberry pi and the components are interfaced according to the requirement of the project.

2.2 Block Diagram

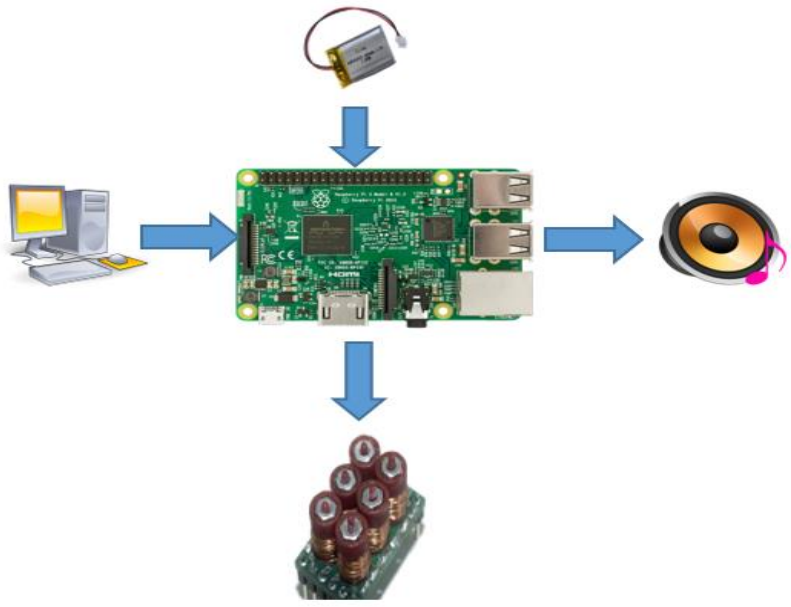

Figure 2.1: Block diagram

2.3 Modules tested and implemented

The following topics explain the equipment simulated, used and implemented in the final project and in the block diagram.

2.3.1 Raspberry Pi

The Raspberry Pi acts as central processing unit for the entire Braille Reader device. It receives any file and stored in the raspberry pi, processes it and able to generate braille character to corresponding English alphabet. It also has the feature of text to speech through which we can hear the audio signal through the audio jack.

2.3.2 Linear Actuator

Linear actuator is basically electromagnetic solenoids it has a copper coil with armature in center. When coil is energized, armature is pulled into center of coil that would make the linear actuator able to push (from the other side) or pull (from one side).

This solenoid is very tiny, having 20mm body and an armature with a return shaft, which means when provided with ~5V DC, the shaft moves and when the voltage is removed it comes back to the original position.

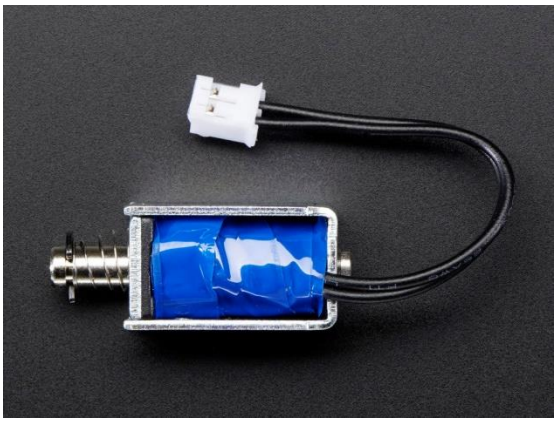

Figure 2.2: Linear Actuator

2.3.3 Battery

A 6V battery is connected to the raspberry pi which gives power to raspberry pi to make it a portable device rather giving it from an external power supply, also it connected to the linear actuator which operates on 5v.

2.3.4 Motor Driver IC (ULN 2003A)

A Darlington pair IC is attached with linear actuator to make it work because when the voltage signal comes from the raspberry pi the current is very low so a Darlington pair IC is attached that amplifies the current up to the linear actuator specification.

2.4 Flowchart

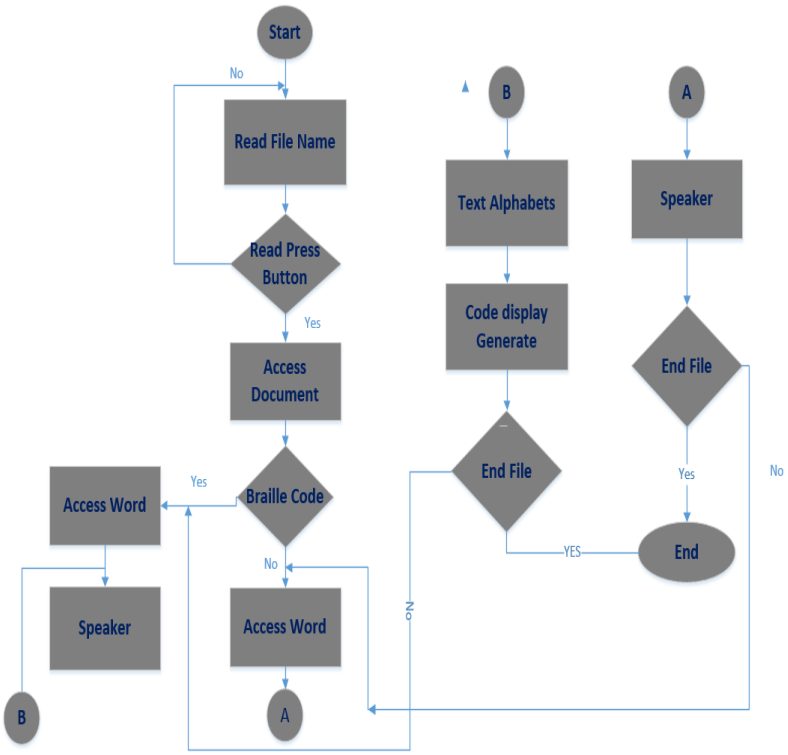

Figure 2.3: Flow Chart

2.5 Description of Flowchart Block

- **File Selection:** This is first step of the entire process, in which we select the document we want to read. It is achieved using dirpath in walk module.

  Once the file is stored in the document, it is accessed using these modules. Raspberry pi reads the names of the documents available, user selects the document by pressing the select button. Thus, the document is selected.

- **Access the Document:** Document is then accessed i-e opened to read the data from the document to perform certain operations

- **Braille code generation:** Once the data from the document is received, its braille code is generated using time and GPIO module.

- **Audio Signal generation:** For the audio signal espeak module is used. By pressing the button, the user may toggle between the braille code and the audio signal of the text in the available document.

- **Output the signal:** After the processing, audio signal then passed through the speakers and the braille code is passed to the linear actuators, which moves up and down with the help of motor driver IC. Thus, the output is achieved.

- **End the document:** Once the document ends the device waits for the input so that the next document is accessed as required.

3) Chapter

Conclusions & Results

After the implement and compilation of the above mentioned modules the required output is achieved, which satisfy the main objective of this project.

3.1 Results

Whenever, the braille code signal is generated by the microprocessor, motor driver current control IC acting as the relay for the circuit, amplifies the signal and passes the amplified signal to the linear motor(Actuator) which moves up and down according to the specific pattern given to it. In case of the audio signal it is generated by microprocessor and passed to the speakers.

3.1.1 Implementation on Designing software

Keeping the main idea of this project in regard, the device was designed according to the design of the common mobile. It is a portable device with a primary component Braille cell and a speaker. It also has some buttons which add smart features to this device.

Figure 3.1: Isometric View of 3D Model

3.1.2 Conclusion

After the implement and compilation of the above mentioned modules the required output is achieved, which satisfy the main objective of this project. Raspberry Pi is programmed according to the algorithm required for the certain results. Use of python and raspberry pi gives the precision accuracy of more than 85%. Raspberry pi is used because of its reliability and high performance.

3.2 Future Prospects

- **IoT Integration:** IoT can be added to allow the device to have remote document access. We can also use Wi-Fi modules for this purpose.
- **Research purposes:** It can be used in research purposes as tool for embedded engineers to study the interfacing of the different components of a device.
- **Power Efficient:** It can be made power efficient by designing some power efficient Linear Actuators.
- **Size Reduction:** Its size can be reduced by using compact smaller linear motors (Actuators) and efficient Power supply.

3.3 Budget

The total expenses that occurred during the course of our project are mentioned below.

Particulars	Price (PKR)	Quantity	Amount (PKR)
Raspberry pi	4500	1	4500
Lithium ion Battery	300	7	2100
Speaker	500	1	500
Female to male header cables	200	2	400
USB cable	150	1	150
Linear (Actuator) Motor	200	6	1200
Charging Adopter	600	1	600
Button Designing	200	5	1000
Body Designing	3500	1	400
TOTAL			**14450**

4) Chapter

References and Citations

[1] S. A. Sabab and M. H. Ashmafee, "Blind Reader: An intelligent assistant for blind," 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp.229-234.

[2] S. Gandhi, B. Thakker and S. Jha, "Braille cell actuator based teaching system for visually impaired students," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1381-1385.

[3] A. Russomanno, S. O'Modhrain, R. B. Gillespie and M. W. M. Rodger, "Refreshing Refreshable Braille Displays," in IEEE Transactions on Haptics, vol. 8, no. 3, pp. 287-297, 1 July-Sept. 2015.

[4] *H. Xiaoli, L. Tao, H. Bing, C. Qiang, X. Qiang and H. Qiang, "Electronic Reader for the Blind Based on MCU," 2010 International Conference on Electrical and Control Engineering, Wuhan, 2010, pp. 888-890.*

[5] *R. Velazquez, E. Preza and H. Hernandez, "Making eBooks accessible to blind Braille readers," 2008 IEEE International Workshop on Haptic Audio visual Environments and Games, Ottawa, Ont., 2008, pp. 25-29.*

[6] R. Velázquez, H. Hernández and E. Preza, "A portable eBook reader for the blind," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, 2010, pp. 2107-2110.

5) Chapter

Appendix-I Program Code

Python® Code

```
import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(40,GPIO.OUT)#21

GPIO.setup(38,GPIO.OUT)#20

GPIO.setup(36,GPIO.OUT)#16

GPIO.setup(33,GPIO.OUT)#13

GPIO.setup(35,GPIO.OUT)#19

GPIO.setup(37,GPIO.OUT)#26

GPIO.setup(7,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(8,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(10,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(15,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

GPIO.output(40,GPIO.LOW)

GPIO.output(38,GPIO.LOW)

GPIO.output(36,GPIO.LOW)

GPIO.output(33,GPIO.LOW)

GPIO.output(35,GPIO.LOW)

GPIO.output(37,GPIO.LOW)
```

```
def start_system(channel):
    global start
global read_counter
 global inter
  if ((start==0)&(read_counter==0)):
        start=1
    if ((read_counter==0)&(inter==1)):
        read_counter=1
def audio_stuff(channel):
    global audio_counter
    global audio
  print('in 2')
    audio_counter=audio_counter+1
    if (audio_counter==2):
        audio=1
 if (audio_counter==4):
        audio=0
 audio_counter=(audio_counter)%4


time.sleep(0.25)
def inputchange3(channel):
```

```
 global allow
    global j
    global text
 print('in 3')
    if allow==1:
        if((j+10)<(len(text))):
            j=j+10
def inputchange4(channel):
    global allow
    global j
    global text
 print('in 4')
    if allow==1:
        if((j-10)>1):
            j=j-10
def inputchange5(channel):
    global allow
    global j


  global text
```

```
 print('in 5')
    j=len(text)
GPIO.add_event_detect(7 ,GPIO.RISING, callback=start_system, bouncetime=3)
GPIO.add_event_detect(8 ,GPIO.RISING, callback=audio_stuff, bouncetime=3)
GPIO.add_event_detect(10 ,GPIO.RISING, callback=inputchange3, bouncetime=3)
GPIO.add_event_detect(11 ,GPIO.RISING, callback=inputchange4, bouncetime=3)
GPIO.add_event_detect(15 ,GPIO.RISING, callback=inputchange5, bouncetime=3)
while(1):
 from os import walk
 import os
from subprocess import check_output
from gpiozero import LED
 import time
    def braill(alphabet):
        code=[0,0,0,0,0,0]
        print ('inside function : ' + alphabet)
if((ord(alphabet)>64) &(ord(alphabet)<91)):
            GPIO.output(37,GPIO.HIGH)

  time.sleep(0.5)
            GPIO.output(37,GPIO.LOW)
```

```
if((ord(alphabet)>47)&(ord(alphabet)<58)):
        GPIO.output(37,GPIO.HIGH)
        GPIO.output(36,GPIO.HIGH)
        GPIO.output(33,GPIO.HIGH)
        GPIO.output(35,GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(37,GPIO.LOW)
        GPIO.output(36,GPIO.LOW)
        GPIO.output(35,GPIO.LOW)
        GPIO.output(33,GPIO.LOW)
    if ((alphabet=='a') | (alphabet=='A') | (alphabet=='1')):
        code=[0, 0, 0, 0, 0, 0]
    if ((alphabet=='b') | (alphabet=='B') | (alphabet=='2')):
        code=[1, 1, 0, 0, 0, 0]
    if ((alphabet=='c') | (alphabet=='C') | (alphabet=='3')):
        code=[1, 0, 0, 1, 0, 0]
    if ((alphabet=='d') | (alphabet=='D') | (alphabet=='4')):
        code=[1, 0, 0, 1, 1, 0]
    if ((alphabet=='e') | (alphabet=='E') | (alphabet=='5')):
        code=[1, 0, 0, 0, 1, 0]
```

```
if ((alphabet=='f') | (alphabet=='F') | (alphabet=='6')):
    code=[1, 1, 0, 1, 0, 0]
if ((alphabet=='g') | (alphabet=='G') | (alphabet=='7')):
    code=[1, 1, 0, 1, 1, 0]
if ((alphabet=='h') | (alphabet=='H') | (alphabet=='8')):
    code=[1, 1, 0, 0, 1, 0]
if ((alphabet=='i') | (alphabet=='I') | (alphabet=='9')):
    code=[0, 1, 0, 1, 0, 0]
if ((alphabet=='j') | (alphabet=='J') | (alphabet=='0')):
    code=[0, 1, 0, 1, 1, 0]
if ((alphabet=='k') | (alphabet=='K')):
    code=[1, 0, 1, 0, 0, 0]
if ((alphabet=='l') | (alphabet=='L')):
    code=[1, 1, 1, 0, 0, 0]
if ((alphabet=='m') | (alphabet=='M')):
    code=[1, 0, 1, 1, 0, 0]
if ((alphabet=='n') | (alphabet=='N')):
    code=[1, 0, 1, 1, 1, 0]
if ((alphabet=='o') | (alphabet=='O')):
    code=[1, 0, 1, 0, 1, 0]
```

```
if ((alphabet=='p') | (alphabet=='P')):
        code=[1, 1, 1, 1, 0, 0]
    if ((alphabet=='q') | (alphabet=='Q')):
        code=[1, 1, 1, 1, 1, 0]
    if ((alphabet=='r') | (alphabet=='R')):
        code=[1, 1, 1, 0, 1, 0]
    if ((alphabet=='s') | (alphabet=='S')):
        code=[0, 1, 1, 1, 0, 0]
    if ((alphabet=='t') | (alphabet=='T')):
        code=[0, 1, 1, 1, 1, 0]
    if ((alphabet=='u') | (alphabet=='U')):
        code=[1, 0, 1, 0, 0, 1]
    if ((alphabet=='v') | (alphabet=='V')):
        code=[1, 1, 1, 0, 0, 1]
    if ((alphabet=='w') | (alphabet=='W')):
        code=[0, 1, 0, 1, 1, 1]
    if ((alphabet=='x') | (alphabet=='X')):
        code=[1, 0, 1, 1, 0, 1]
    if ((alphabet=='y') | (alphabet=='Y')):
        code=[1, 0, 1, 1, 1, 1]
```

```
if ((alphabet=='z') | (alphabet=='Z')):
code=[1, 0, 1, 0, 1, 1
    if (alphabet=='.'):
        code=[0, 1, 0, 0, 1, 1]
    if (alphabet==','):
        code=[0, 1, 0, 0, 0, 0]
    if (alphabet==':'):
        code=[0, 1, 0, 0, 1, 0]
    if (alphabet==';'):
        code=[0, 1, 1, 0, 0, 0]
    if (alphabet=='?'):
        code=[0, 1, 1, 0, 0, 1]
    if (alphabet=='!'):
        code=[0, 1, 1, 0, 1, 0]
    if ((alphabet=='(') | (alphabet==')')):
        code=[0, 1, 1, 0, 1, 1]
    if (alphabet==""):
        code=[0, 1, 1, 0, 0, 1]
    if (alphabet=='+'):
        code=[0, 1, 1, 0, 1, 0]
```

```
if (alphabet=='-'):
        code=[0, 1, 0, 0, 1, 0]
    if (alphabet=='*'):
        code=[0, 0, 1, 0, 1, 0]
    if (alphabet=='@'):
        code=[0, 0, 1, 1, 1, 0]
    if (alphabet=='<'):
        code=[1, 1, 0, 0, 0, 1]
    if (alphabet=='>'):
        code=[0, 0, 1, 1, 1, 0]
    if (alphabet=='/'):
        code=[0, 0, 1, 1, 0, 0]
    if (alphabet=='='):
        code=[0, 1, 1, 1, 1, 0]
    if (alphabet=='#'):
        code=[0, 0, 1, 1, 1, 1]
    if (alphabet=='_'):
        code=[0, 0, 1, 0, 1, 1]
    if(code[0]==1):
        GPIO.output(40,GPIO.HIGH)
```

```
if(code[1]==1):
        GPIO.output(38,GPIO.HIGH)
    if(code[2]==1):
        GPIO.output(36,GPIO.HIGH)
    if(code[3]==1):
        GPIO.output(33,GPIO.HIGH)
    if(code[4]==1):
        GPIO.output(35,GPIO.HIGH)
    if(code[5]==1):
        GPIO.output(37,GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(40,GPIO.LOW)
    GPIO.output(38,GPIO.LOW)
    GPIO.output(36,GPIO.LOW)
    GPIO.output(33,GPIO.LOW)
    GPIO.output(35,GPIO.LOW)
    GPIO.output(37,GPIO.LOW)
  for dirpath, dirnames, filenames in walk('/home/pi/Documents'):
    print(filenames)
  d=os.getcwd()
```

```
print(d)
    d1=os.path.join(d,"Documents")
    print(d1)
    start=0
    #Start variable initialized to zero
    allow=0
    #allow variable initialized to zero. Permission variable
    read_counter=0
  #read_counter variable initialized to zero. Saves number of      #words    in    the
document
    inter=0
    #index variable initialized to zero
    audio_counter=0
    #audio_counter variable initialized to zero
    audio=0
    #audio variable initialized to zero
    while (1):
        print("start_up stage, start = 0")
        if(start >0):
            break
```

```
counter=-1
    speak = check_output(['espeak','following readable files are available in the folder'])
    inter=1
    for x in filenames:
        counter=counter+1
        speak = check_output(['espeak',x])
        speak = check_output(['espeak','if you want to read this press the button'])
        time.sleep(1)
        if(read_counter==1):
            break
 i=0;
    fname=os.path.join(d1,filenames[i])
    f=open(fname,"r")
    text= f.read()
    text =text.split(' ')
    print (len(text))
    j=0
    while j<len(text):
        print(text[j])
        allow=1
```

```
 if audio==1:
            speak = check_output(['espeak',text[j]])
        if audio==0:
            for my_character in text[j]:
                co=braill(my_character)
        j=j+1
    speak = check_output(['espeak','the file ends '])
GPIO.output(40,GPIO.LOW)
GPIO.output(38,GPIO.LOW)
GPIO.output(36,GPIO.LOW)
GPIO.output(33,GPIO.LOW)
GPIO.output(35,GPIO.LOW)
GPIO.output(37,GPIO.LOW)
f.close
GPIO.cleanup()
```

6) Chapter

Appendix-II User Manual

6.1 Introduction

This manual explains the information regarding this project and its steps of operation. This Braille Reader device is made in our own country, Pakistan. It will provide all necessary information and steps for customer to use the product without damaging. Before using the device

- Read instructions and steps carefully.
- Keep the instructions and steps for reference.
- Read all the warnings carefully. So that, user may not harm the product.
- Follow the instructions in order to avoid any inconvenience.

In case of any confusion or inconvenience, do contact the product distributors.

6.2 Operating Instructions

To operate the braille reader device, the user (visually impaired person) has to go for following steps:

- Start the device by pressing the start-up button.
- Once the device is start, press the select button to read the names of the file. Press the select button on the name of the file. File will be selected.
- Toggle button is to toggle between audio and the braille system code. For the audio signal you need earphones and in the case of braille code you need to place your finger on the Braille cell.
- User may move forward and backward in the text by using the respective button available. This will help user to read the document from where they want to read, without any issue.
- Once the file is ended. It will wait for the user to press the select button to read the new file. User can end the file himself by pressing the end button.

6.3 Safety Requirements

We have already done all the engineering research we could do to minimize any chances of any problem that could occurring. We make sure you feel comfortable with the product. As we know there is always a chance of problem to occur, It is important to make sure that you use the product the same way as described in the guide. The safety features one has to take with this product are as follows:

- Don't over charge the battery. This may cause the battery to overheat and thus, may cause battery to expire.
- Place the finger properly otherwise the braille cell may got damaged.
- Don't try to press the braille cell.
- its fragile device. So, make sure you don't throw it from height or place it under heavy object or pressure.
- Don't place it in the water or any liquid thing. It may cause the device to operate in abnormal behavior or even shutdown.
- Be careful while giving to children under 8 years.

6.4 Maintenance Instructions

Don't open the device. As this device does not have any such component which requires regular maintenance so the user will probably not have any periodic maintenance issues.

But, in case of damage or if the device is not operating normally, contact the distributor of the device for the qualified servicing.

6.5 Cleaning Instructions

Only the outer surface of the product needs to be cleaned, periodically. Cleaning may be done using a clean cloth just in case to remove the dust. Use clean cotton cloth for such purpose and rub the outer body of the device to clean it. Follow these instructions:

- Clean the outer body with cotton cloth but make sure you the cloth is not wet.
- Do not rub the outer body with any hard materials.
- You may keep it in the casing to make sure the dust does not enter the inter body.

6.6 Warranty

Thank you for your interest in our device and service we are offering. This warranty applies only upon the Braille Reader Device purchased from us or our distributors. We warrant to the customers that the Products are free of defects that can cause their better performance for period of three to five years from the Date of the Delivery. Under normal usage during Warranty period, this Warranty covers any uncertainty in material or working of the product.

We agree to return the cash or repair or replace all products, at no charge, that proves defective or guilt at our side because of inappropriate material or improper workmanship, under the normal use and periodic maintenance provided that:

- Defect is received under the above mentioned warranty period.
- The Products have not been changed or subjected to incorrect maintenance.
- We are not responsible for any accidental damage.
- In case of the replacement of products, our user has benefit of the applicable warranty.