Sudan University Of Science and Technology
College of Engineering
School of Electrical and Nuclear Engineering

Text to Braille Converter
تحويل النصوص للغة برايل

A Project Submitted in Partial Fulfillment for the Requirements of the Degree B.Sc. (Honor) In Electrical Engineering

Prepared by

- **Mohamed Al Ameen Isam Al Deen Adm Mohamed**
- **Mohamed Yassin Ibrahim**
- **Moaaz Salah Abd Allah**
- **Mohammed El hassan Abu Alama**

Supervisor

Dr. Ebtihal Haidar

November, 2020

الآية

قال تعالى

بسم الله الرحمن الرحيم

(قَالُوا سُبْحَانَكَ لاَ عِلْمَ لَنَا إِلا مَا عَلمْتَنَا إِنكَ أَنَتَ الْعَلِيمُ الْحَكِيمُ)

صدق الله العظيم

سورة البقرة-الآية{32}

Dedication

Dedication to my mother…tfith warmth and faith…

Dedication to my father…tfith love and respect …

Dedication to my friends…tfhom we cherish their friendship

Dedication to my special people…tfho mean so much to me…

Dedication to all my teachers …In whom I believe so much

…

Acknowledgement

tfe extend our thanks to all who stood

tfith us to achieves this research which

It com because of grace of God

And reconcile.

tfe would like to give special thanks

My Supervisor

Dr. Ibtihal Haidar

For his great help and support.

And

Our teachers that gave us information and all staff in Sudan

University for science and technology.

Finally yet importantly we dedicate this project for everyone that

helped us to be at the place that we are today.

المستخلص

يعاني العديد من الأطفال من مشكلة فقدان البصر، والذي ينتج عن تشوه في مكونات العين الداخلية من شبكية أو تلف في الأعصاب وقد تصل الى تلف أجزاء في الدماغ وغالبا ما تكون وراثية او بسبب حادث، يمكن علاج بعض الحالات المذكورة عن طريق التدخل الجراحي او التصحيح الليزري أو زراعة شبكية العين والعدسة. من الوسائل المهمة لاستيعاب الأشكال والألوان هي البصر، حيث تعمل العين على نقل نمط معين لشكل او لون معين من الشبكية الى الدماغ وتخزين خصائص مميزة للتعرف عليه مرة أخرى. يقلل العمى من قدرة الطفل الى استيعاب الأشياء ولكن بعد ابتكار طريقة برايل اصبح بالإمكان استخدام رموز بارزة مكونة من ستة او ثمانية تميز الحروف التي تكون الكلمات او الحروف، وأصبح هذا الابتكار معتمدا في كافة انحاء العالم بالرغم من اختلاف اللغات فقد تم تكوين الرموز البارزة لكل اللغات تقريبا،ساعدت هذه اللغة الأطفال في تطوير المنظومة التعليمية للأطفال. في هذا المشروع تم ابتكار منظومة تستطيع تحويل النص الى لغة برايل عبر التحكم الالكتروميكانيكي "محرك يعمل بالتيار المستمر" ، ويتم التحكم عبر متحكم دقيق اردوينو لتحويل النص الى مجموعة حروف ومن ثم تحويل كل حرف الى ASCII والى BINARY ومنه الى خلية برايل. وقد تم عمل الدائرة الالكترونية وتشغيلها بنجاح.

Abstract

Many children suffer from the problem of vision loss, which results from a deformity in the internal components of the eye from the retina or nerve damage, and may reach parts of the brain, often being hereditary or due to an accident. Some of the mentioned cases can be treated by surgical intervention or laser correction. Or retina and lens implantation. One of the important means of understanding shapes and colors is vision, where the eye works to transmit a certain pattern of a certain shape or color from the retina to the brain and store distinct characteristics to recognize it again. Blindness reduces the child's ability to comprehend things, but after the invention of the Braille method, it became possible to use prominent symbols consisting of six or eight that distinguish the letters that make up words or letters, and this innovation became adopted all over the world despite the different languages, as the prominent symbols were formed for all languages Almost all of this language helped children in developing the educational system for children. In this project, a system was designed that to convert text into Braille through electromechanical control, "DC-Motor" and it is controlled via an Arduino to convert the text into a set of characters and then convert each letter into ASCII and BINARY and then into a braille cell. The electronic circuit has been working and running successfully.

Table of Contents

List of Figures

List of Tables

Abbreviations

DC	Direct Current
ASCII	American Standard Character
EBDR	Electronic Braille Document Reader
LGPL	Lesser General Public License
GPL	General Public License
I/O	input/output
USB	Universal Serial Bus
IDE	integrated development environment
LCD	liquid-crystal display

CHAPTER ONE

Literature Review

1.1 Introduction

The subject of blindness and education has included evolving approaches and public perceptions of how best to address the special needs of blind students. The practice of institutionalizing the blind in asylums has a history extending back over a thousand years, but it was not until the 18th century that authorities created schools for them where blind children, particularly those more privileged, were usually educated in such specialized settings. These institutions provided simple vocational and adaptive training, as well as grounding in academic subjects offered through alternative formats. Literature, for example, was being made available to blind students by way of embossed Roman letters. Louis Braille attended Haüy's school in 1819 and later taught there. He soon became determined to fashion a system of reading and writing that could bridge the critical gap in communication between the sighted and the blind. In his own words: "Access to communication in the widest sense is access to knowledge, and that is vitally important for us if we [the blind] are not to go on being despised or patronized by condescending sighted people.

1.2 Problem Statement

- Most of the blind academics uses braille and sound to each blind student.
- Difficulty of education.
- These days the development in education applications goes far, the existing techniques used to help blinds require development to match the existing technologies.

1.3 Objectives

To design and implement a circuit that convert a letter character to simple braille character using Actuators and Arduino.

1.4 Methodology

The system will be designed to convert text into Braille through electromechanical control, "DC Vibration Motor” and it is controlled via an Arduino developer board to convert the text into a set of characters and then convert each letter into ASCII then to BINARY and then into a braille cell.

1.5 Project Layout

This thesis consists of five chapters, in chapter two represent the literature review, while chapter three represent the methodology, in chapter four the results and discussion were included and in chapter five the conclusion and recommendation were included.

CHAPTER TWO

Background and Literature Review

2.1 Blindness and Visually Impairment

Blind and visually impaired individuals use a matrix of dots called Braille to read. Currently only a limited number of books get translated in to Braille which requires a human to read and type the entire book in Braille. The books which are translated are bulky due to the minimum size requirement of a Braille cell. The challenge the blind people face is having to source the desired books in Braille and then not being able to carry more than a few because of the large size. Refreshable Braille displays exist which have a line of Braille cells and can display text from a computer when connected to one. These allow the blind to be able to use computers and access the vast catalogue of literature online. The drawback of these is the phenomenal costs and size of one. There is a clear need for a device which can overcome these obstacles for the blind. This project looks into the feasibility of designing a device which would allow a blind user to access any digital text document by converting it to Braille and displaying it in a manner suitable for them to read. Braille is a tactile writing system used by people who are visually impaired. It is traditionally written with embossed paper. Braille users can read computer screens and other electronic supports using refreshable braille displays. They can write braille with the original slate and stylus or type it on a braille writer, such as a portable braille notetaker or computer that prints with a braille embosser.

Braille is named after its creator, Louis Braille, a Frenchman who lost his sight as a result of a childhood accident. In 1824, at the age of fifteen, he developed a code for the French alphabet as an improvement on night writing. He published his system, which subsequently included musical notation, in 1829.[1] The second revision, published in 1837, was the first small binary form of writing developed in the modern era. These characters have rectangular blocks called *cells* that have tiny bumps called *raised dots*.

The number and arrangement of these dots distinguish one character from another. Since the various braille alphabets originated as transcription codes for printed writing, the mappings (sets of character designations) vary from language to language, and even within one; in English Braille there are three levels of encoding: Grade 1 – a letter-by-letter transcription used for basic literacy; Grade 2 – an addition of abbreviations and contractions; and Grade 3 – various non-standardized personal stenography.

Braille cells are not the only thing to appear in braille text. There may be embossed illustrations and graphs, with the lines either solid or made of series of dots, arrows, bullets that are larger than braille dots, etc.

A full braille cell includes six raised dots arranged in two columns, each column having three dots.[2] The dot positions are identified by numbers from one to six.[2] There are 64 possible combinations, including no dots at all for a word space.[3] A cell can be used to represent a letter, digit, punctuation mark, or even a word.[2] Braille literacy is a social-justice issue.[4] Early braille education is crucial to literacy, education and employment among the blind. However, in the face of changes in education policy and screen reader software, braille usage has declined in recent decades, despite the fact that technologies such as braille displays have also made braille more accessible and practical.

2.2 Types of Braille

2.2.1 Six Dot Braille

There are many variations of Braille in existence which could be used for the Electronic Braille Document Reader (EBDR).

The most commonly used Braille system is the original six-dot Braille, which consists of a cell of six raised dots arranged in two columns of three dots. The dot positions are numbered top to bottom 1 to 3 on the first column (left), and 4 to 6 on the second column (right).

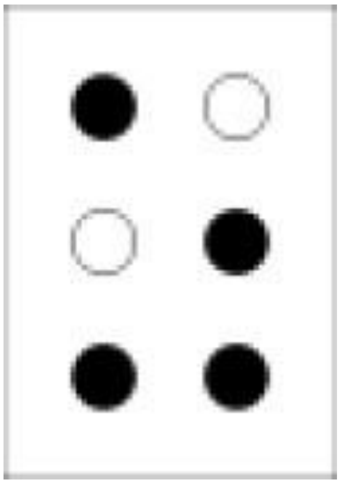

Figure 2.1 six dots character representation in braille

Any Braille character can be described using the positions, e.g. The letter 'S' ⠎ can be described as 2-3-4.

The six-dot Braille has a total of 63 combinations, but some of the combinations feel too similar to be used e.g. ⠊ and ⠔ so are omitted. The punctuations are represented by their own set of patterns. But numbers use the same patterns as the alphabets 'a' to 'j'. They are recognized by the context they are in and the symbol placed before it e.g. before a number a Braille pattern 3-4-5-6 ⠼ is placed.[2][8]

2.2.2 Eight Dot Braille

As the restriction of the six-dot Braille became evident it was extended to eight dot Braille which gave 256 combinations. The eight-dot Braille has two extra dots at the bottom of the cell; each eight-dot cell consists of two columns of four dots. The two extra dots positions are numbered 6(left) and 7(right). The extra combinations allow all special characters to have a unique pattern. The main advantage of the eight-dot Braille is that all details of the character can be represented in a single cell e.g. case, number or punctuation [3]. The eight dot Braille is also popular in technical areas such as mathematics and sciences. It has also gained popularity in refreshable Braille displays as the extra two dots can represent extra information such as cursor position and various text attributes.[9]

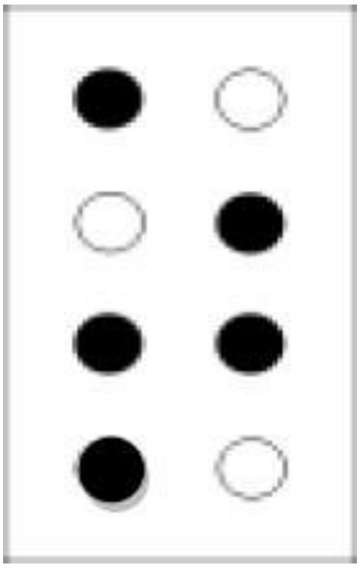

Figure 2.2 8 dots presentation of braille

2.2.3 Grade 1

Grade 1 Braille, which is sometimes also called uncontracted Braille, is the exact substitution of each letters to its corresponding Braille patterns of the alphabet. It is usually used for teaching beginners and labelling because it takes more space and slow to read. Figure 3 shows a quote written in grade 1 Braille.

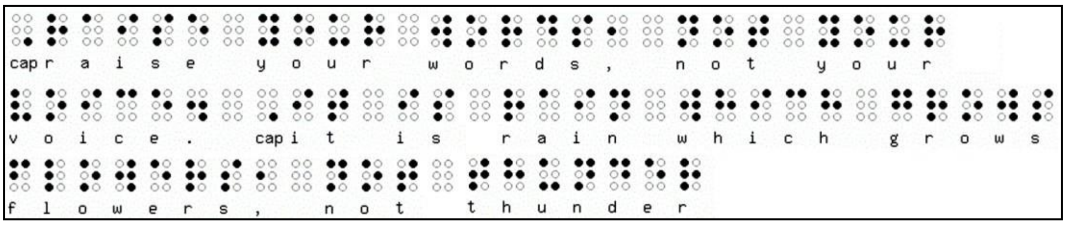

Figure 2.3 Quote in Grade 1 Braille

2.2.4 Grade 2

Grade 2 Braille, also known as contracted Braille is when words are written in shorthand. The grade 2 Braille uses the same Braille characters as the grade 1 but with some extra combinations for commonly used words and common sounds. Another way the grade 2 Braille differs from grade 1 is when writing; many words can be shortened to just a few characters e.g. Braille can be written as Brl . Some things in grade 2 Braille can mean different things depending on the context therefore this type of Braille is used by experienced Braille users.

Most publications use the grade 2 Braille because it's quicker to read and write and also takes up less space. The picture in Figure 4 shows a quote written in grade 2 Braille.[10][11] The quote was translated using online Braille converters, which can do the job but aren't 100% accurate as only humans are able to understand the context and are able to apply the rules accordingly.

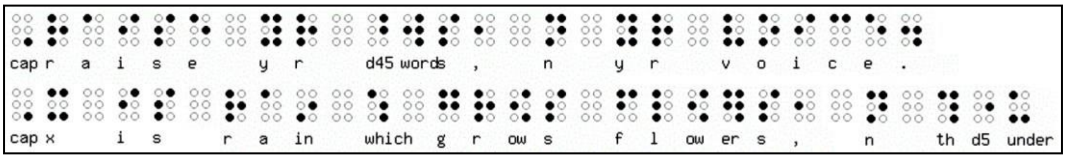

Figure 2.4 Quote in Grade 2 Braille

2.3 Braille in Different Languages

The popularity of Braille code to represent text for blind and visually impaired has encouraged others to use the Braille code to represent scripts used by other languages like Arabic and Chinese. Many languages which use the Latin alphabet like Turkish and Spanish found a relatively simple way of adapting the Braille code and including a few more patterns for letters like ş and ğ in Turkish which doesn't exist in the English or French alphabet. Other languages with completely different scripts like Arabic, Chinese and Hindi had a harder time trying to come up with a standard. At the time of India's independence there were eleven Braille scripts in use in different parts of the country for the many different languages spoken in India. The government urged UNESCO to help come up with a unified Braille code for the country. They understood the different languages in India, Pakistan and Sri Lanka were based on phonetics with different scripts representing the sounds produced by different constants and vowels.[12][13] A standard system was devised called the Bharati Braille. It assigns cells to characters based on the phonetic sounds. Similar sounds to the English characters are given the same Braille pattern. This also allows Indian languages to be transliterated to English and then encoded into Grade 1 Braille. The Bharati Braille uses the same patterns for all the languages, which causes a problem in multilingual texts that it's hard to spot the language change.

The difference can only be spotted from the context. But the advantage it has is that multilingual user’s don’t have to learn new codes for all the languages. The success of the Bharati Braille caused Bangladesh, Nepal, and Sri Lanka to also adopt it as a standard for Blind communication.[13][14]. The Arabic Braille is a bit more complicated though letter assignments generally correspond to English, Greek and Russian Braille. Arabic has its rules governing the pronunciation of words and are expressed using symbols surrounding letters. Even with the complexity, a standard Braille format in Arabic exists and the Holy Book Qur’an has also been made available in Braille for blind Muslims[11]. To understand the Arabic Braille one must first have thorough knowledge of the Arabic language so further information on Arabic Braille can be found online[10][14]. Chinese and similar scripts like Japanese are also complicated due to the way they are written. Chinese for instance does not have an alphabet as such but uses characters which are based on drawings of the real object. There are over 100,000 characters in the Chinese writing system [12]. So Chinese Braille is written using the sound of the language rather than the characters. Each syllable uses three Braille cells; one for the initial, one for the final and one for the tone [9]. The UNESCO report on world Braille usage explains the adaptation and s

2.2.1 Arabic Braille

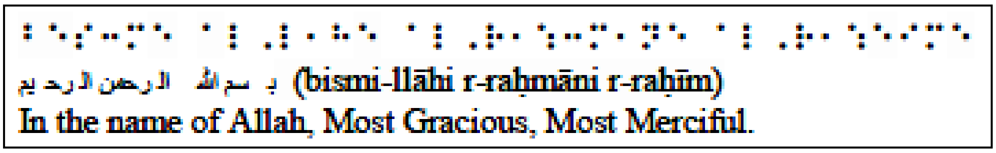

Figure 2.5 Example of Arabic Braille

2.2.2 Chinese Braille

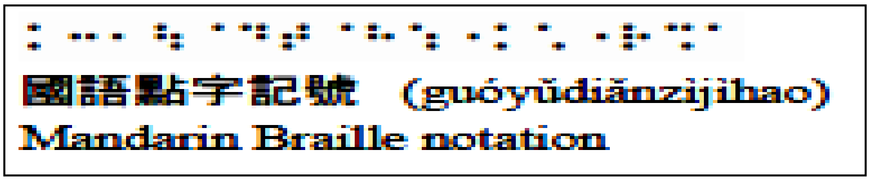

Figure 2.6 Example of Chinese Braille

2.3 Braille Technology

Through history as technology progressed people adapted it to aid the disabled. The new age brought electronics to the world in the shape of computers, phones, TV's and much more. Much of the technology was developed with just the sighted people in mind. As the technologies evolved and became more popular people began thinking of ways to extend this technological advances for the blind. Following are some examples of how technological advances; specifically in electronics has helped the blind.

2.3.1 Refreshable Braille Displays

In the 90s personal computers became very popular making their way into many households. As much of the computers use is based on visual interaction, they weren't accessible to the blind. Refreshable Braille Displays were developed to give the blind access to computers. A refreshable Braille display substitutes a computer screen with a device with usually two lines of refreshable Braille cells which actuate to form the text outputted from the computer. Piezo actuators are used to actuate the Braille dots in most displays available today. They also include some navigation keys and many have a Braille keyboard included. These displays are a great benefit to the blind and visually impaired as they provide a way to access the vast amount of literature available through the World Wide Web. By using Braille displays blind and visually impaired individuals are able to work in a society where few businesses operate without computers. Unfortunately these displays are very big, bulky and expensive. The piezo actuators used to actuate the dots tend to be big which also prevent the displays from having more than two lines to display. Currently each cell on the displays roughly costs $100; this means that an average sized

Braille display with 30-40 cells costs a few thousand dollars. The cost of these devices is a major limiting factor and prevents people from buying them. There is currently a lot of on-going research into improving these displays and bringing the cost down.[7][6]

2.3.2 Braille Embossers/Note takers

Braille embossers are like typewriters for Braille. A user feeds embossing paper in one end and uses the six keys; one for each dot of the cell and one button to move over to the next cell to emboss Braille patterns onto the paper. Note takers allow the user to take notes in electronic format. Note takers have a Braille keyboard and a line of refreshable Braille cells usually of about 8 – 20 cells, which allow for the text to be read back.

2.3.3 Speech Synthesizers

Speech Synthesizers read text out loud. These are used very commonly by blind to read as listening to an audio is much faster than reading Braille. Speech synthesizers are commonly used in conjunction with refreshable Braille displays by the blind when operating a computer. Many books come in audio book formats which have been recorded by a human reading the book. Speech synthesizer's use artificially produced human speech to read the text, usually in monotone and lack the human touch. The monotone and artificial nature of the voice makes it hard to understand at times and often not very pleasing to the ear, though advances in speech synthesizers are eliminating those issues.

2.3.4 Dictaphones

A Dictaphone is a device which can record speech for later playback. They are used by many professionals to take notes where writing down on paper or typing isn't suitable at the time. They are also used by blind people to take notes as an alternative to Braille note taking which can be cumbersome at times. Audio files can be stored for listening to later or to type up when suitable. As useful as they are to the blind and visually impaired community, they can't replace Braille completely. When listening to playback it isn't as easy to jump from one desired point to another as it is on Braille embossed paper. Another drawback being it's not always suitable to talk into a Dictaphone when in a meeting.

2.4 English Illustration in Braille

The English Braille alphabet has letters that correspond directly to the 26 letters of the English print alphabet, but also ligatures that are equivalent to digraphs and sequences in print.[10]

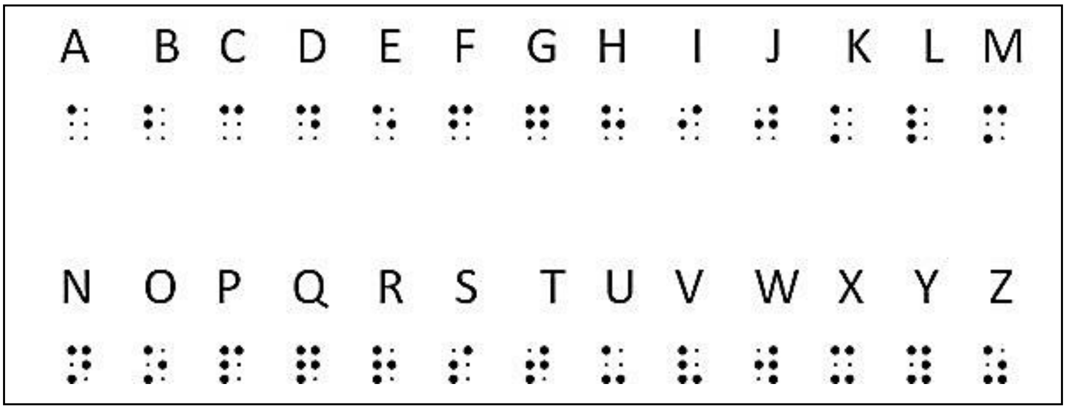

Figure 2.7: English litters vs. Braille litters

2.5 Arabic Illustration in Braille

أرقامه	حرف برايل	الحرف العادي	أرقامه	حرف برايل	الحرف العادي
1،2	⠃	ب	1	⠁	ا
1،4،5،6	⠹	ث	2،3،4،5	⠞	ت
1،5،6	⠱	ح	2،4،5	⠚	ج
1،4،5	⠙	د	1،3،4،6	⠭	خ
1,2,3,5	⠗	ر	2,3,4,6	⠮	ذ
2,3,4	⠎	س	1,3,5,6	⠵	ز
1,2,3,4,6	⠯	ص	1,4,6	⠩	ش
2,3,4,5,6	⠾	ط	1,2,4,6	⠫	ض
1,2,3,5,6	⠷	ع	1,2,3,4,5,6	⠿	ظ
1,2,4	⠋	ف	1,2,6	⠣	غ
1,3	⠅	ك	1,2,3,4,5	⠟	ق
1,3,4	⠍	م	1,2,3	⠇	ل
1،2،5	⠓	ه	1،3،4،5	⠝	ن
2،4	⠊	ي	2،4،5،6	⠺	و
1،6	⠡	ة	1،3،5	⠕	ى
			1،2،3،6	⠧	لا

Figure 2.8: Arabic litters vs. Braille litters

2.6 Six Type Braille Cell

The six-type braille cell consist of six dots into three rows and two column, these dots are related to binary system

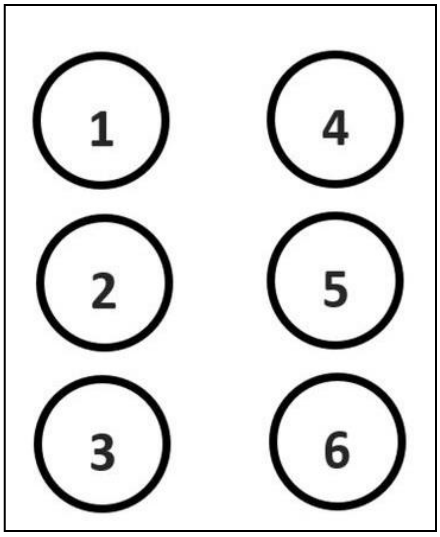

Figure 2.9: Braille Cell

CHAPTER THREE

Methodology

3.1 Block Diagram

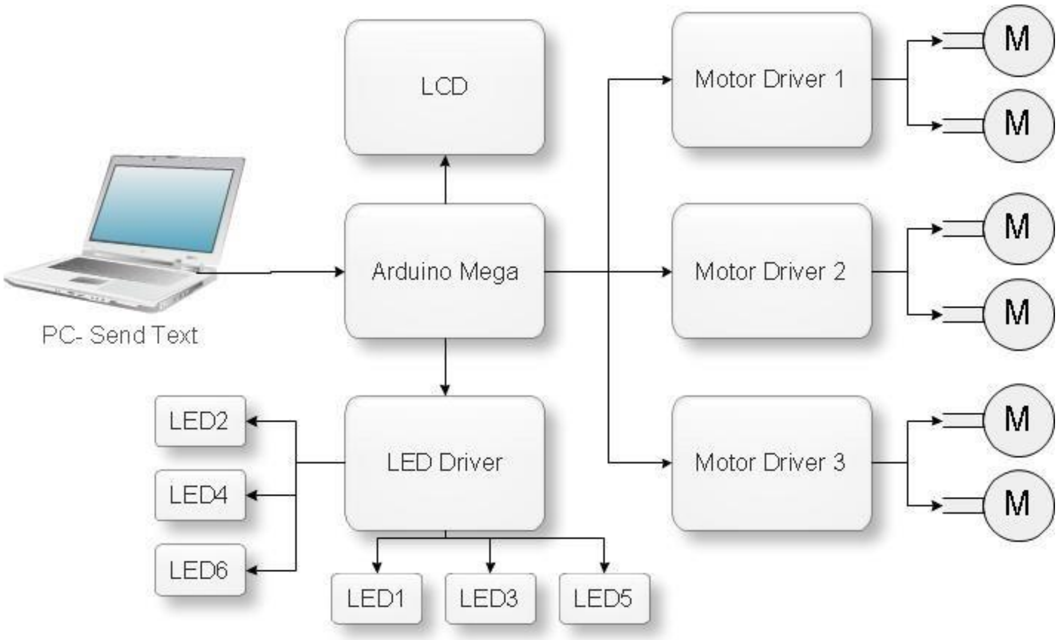

Figure 3.1: Text to Braille Converter Block Diagram

3.2 Flowchart

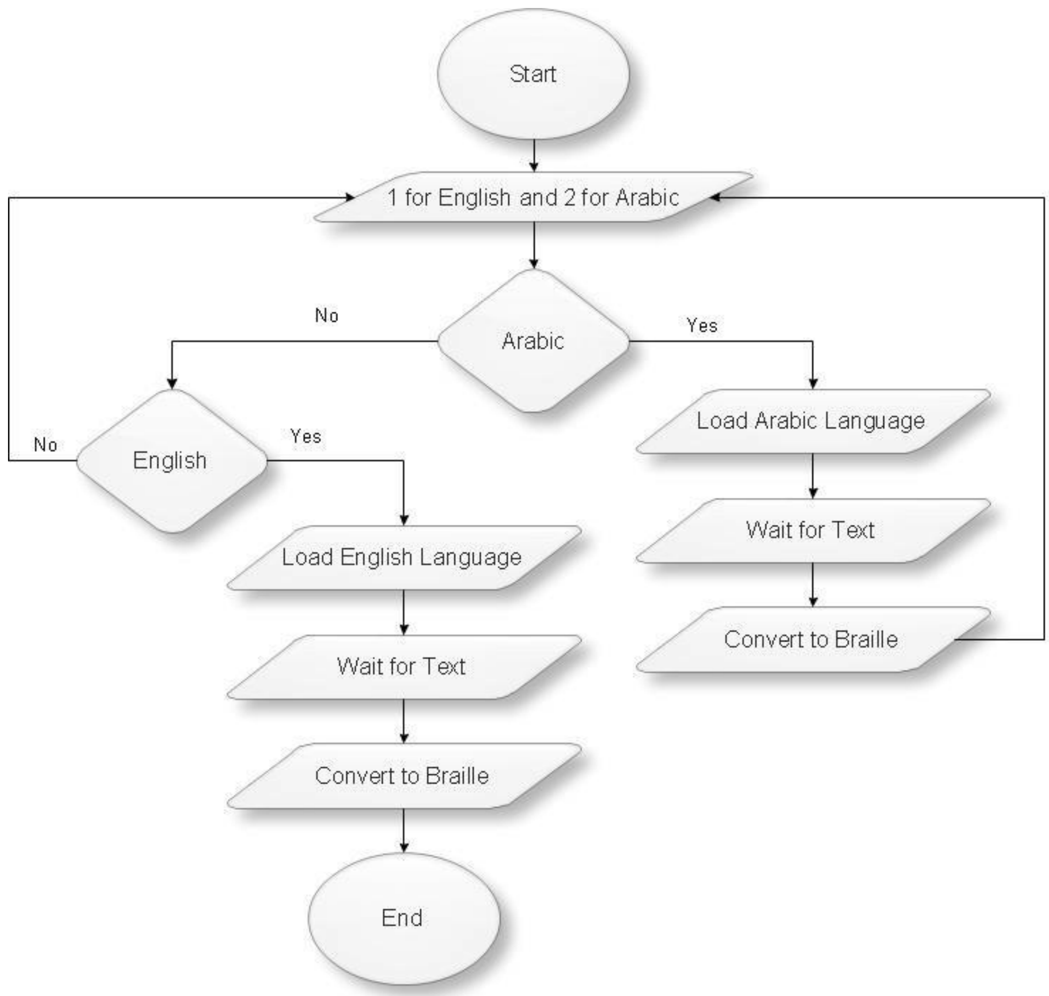

Figure 3.2: Text to Braille Converter Computer Model

3.3 Requirements and Components

3.3.1 Arduino Development Board

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

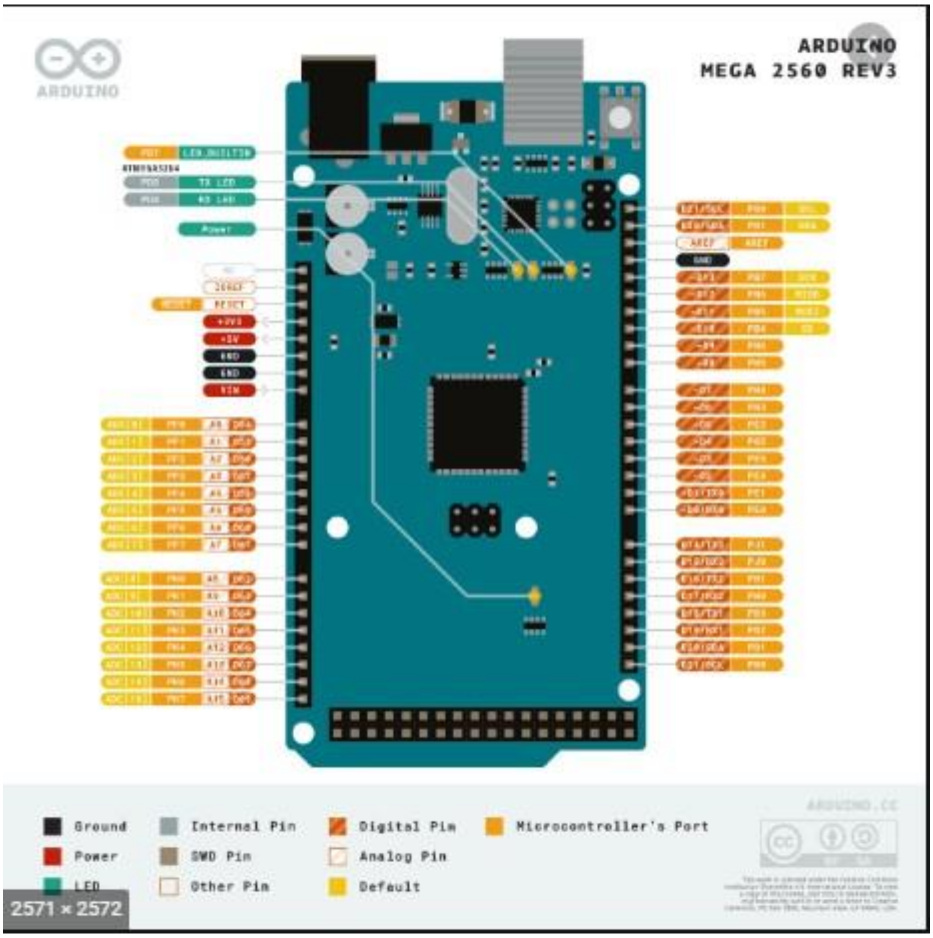

Figure 3.3: Arduino Mega Sample Board

Arduino-compatible R3 MEGA board made in China with no Arduino logo, but with identical markings, including "Made in Italy" text

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2. Never the less, an official Bill of Materials of Arduino boards has never been released by Arduino staff.

3.3.2 Potentiometer

A potentiometer measuring instrument is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Figure 3.4: shows a typical single-turn potentiometer

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

3.3.3 Colored Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. The current through a resistor is in direct proportion to the voltage across the resistor's terminals. Thus, the ratio of the voltage applied across a resistor's terminals to the intensity of current through the circuit is called resistance. This relation is represented by Ohm's law:[16]

$$I = \frac{V}{R} \tag{3.1}$$

Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel-chrome). Resistors are also implemented within integrated circuits, particularly analog devices, and can also be integrated into hybrid and printed circuits.

Figure 3.5: shows a resistor 4 Band

3.3.4 L298N Motor Driver Module

is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L298N Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.

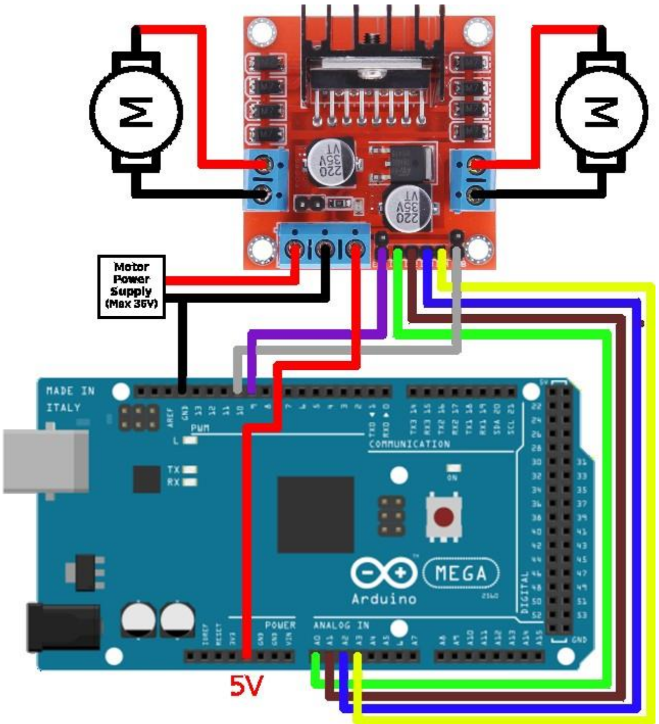

Figure 3.6: interfacing L298 to Arduino Mega

3.3.5 Liquid Crystal Display

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.

Figure 3.7: Liquid Crystal Display

3.3.7 Interfacing LCD to Arduino Mega

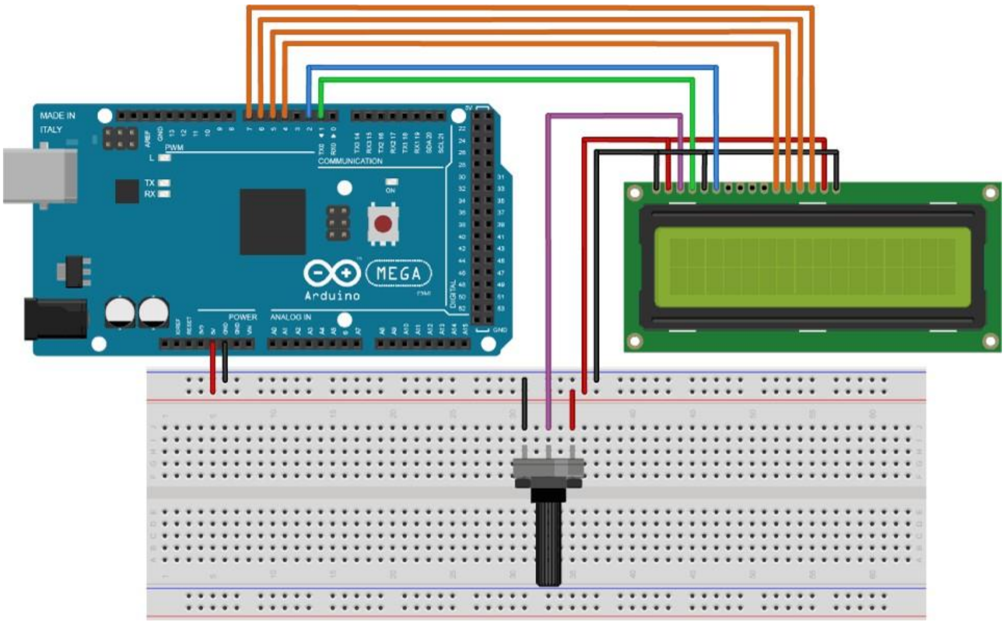

Figure 3.8: Wiring Liquid Crystal Display

3.3.8 Interfacing Led to Arduino mega

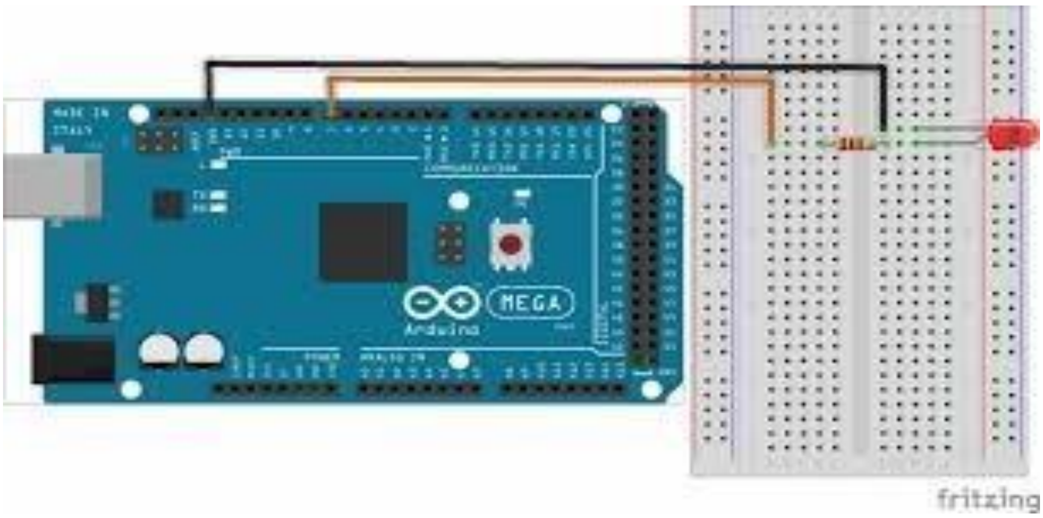

Figure 3.9: Wiring LED

3.3.9 Coin-type Vibration Motor

Vibration motor is a compact size coreless DC motor used to inform the users of receiving the signal by vibrating, no sound. Vibration motors are widely used in a variety of applications including cell phones, handsets, pagers, and so on. The main features of vibration motor is the magnet coreless DC motor are permanent, which means it will always have its magnetic properties (unlike an electromagnet, which only behaves like a magnet when an electric current runs through it); another main feature is the size of the motor itself is small, and thus light weight. Moreover, the noise and the power consumption that the motor produce while using are low. Based on those features, the performance of the motor is highly reliable.

The vibration motors are configured in two basic varieties: coin (flat) and cylinder (bar).

coin-type vibration motor is comprised of a weight, a ring magnet, rotor with commutation points attached in the front and coils assembled on the back, and power supplied brushes attached to the ring magnet. The commutation points, which are the yellow part on the bottom pic, are in contact with the end of the brushes. It will energize the electrical coils in the rotor. Energizing the coils produce a magnetic field and it is strong enough to interact with the ring magnet integrated into the stator, causing rotation. A force is generated due to the magnetic field. This force causes the weight to displace. The repeated displacement of the weight produces a varying force which is felt as vibration. The commutation points are used in changing the polarity pairs, so that as the rotator moves, the coils are constantly reversing the polarity.

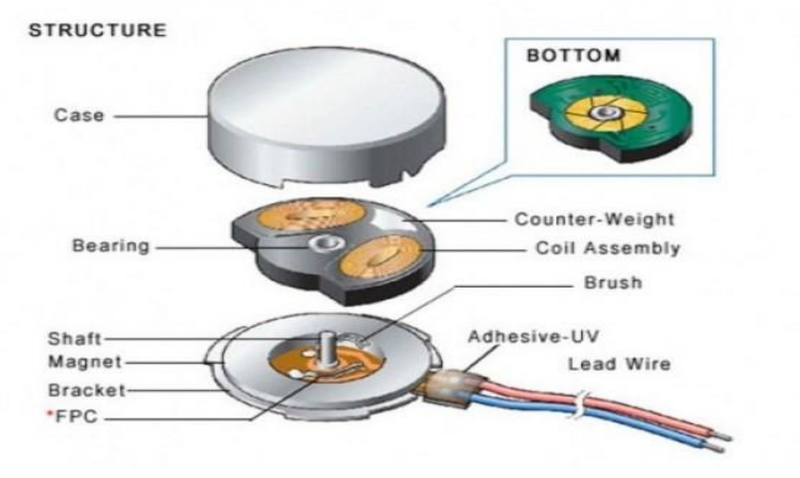

Figure 3.10: Coin-type Vibration Motor

3.4 Programming Environment

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

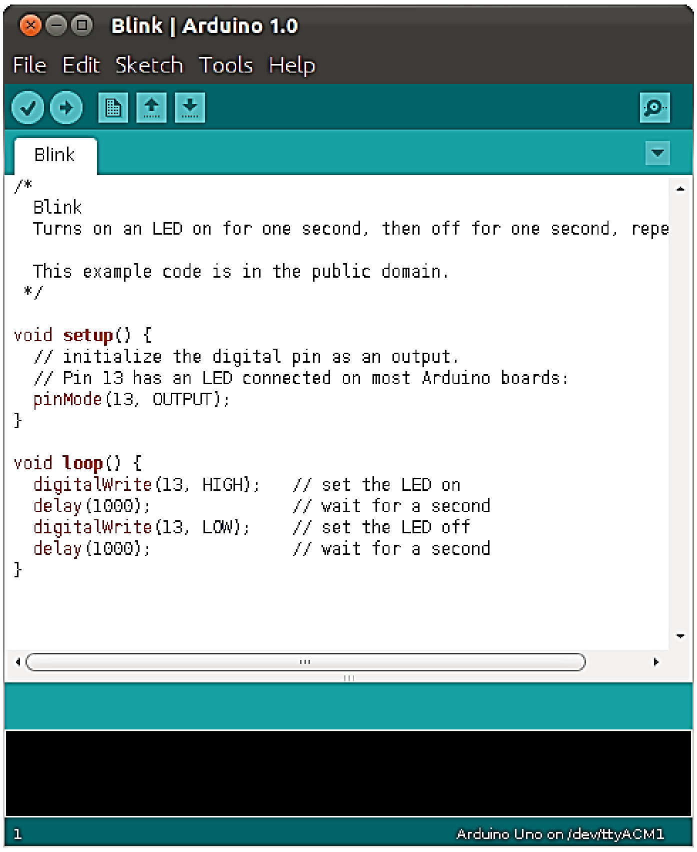

Figure 3.11: Arduino IDE Sketch.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.[3]7

3.4.1 Proteus Design Suite

The Proteus Design Suite is an Electronic Design Automation (EDA) tool including schematic capture, simulation and PCB Layout modules. It is developed in Yorkshire, England by Lab center Electronics Ltd with offices in North America and several overseas sales channels. The software runs on the Windows operating system and is available in English, French, Spanish and Chinese languages.

Figure 3.12: Proteus splash screen

CHAPTER FOUR

Simulation Results and Discussion

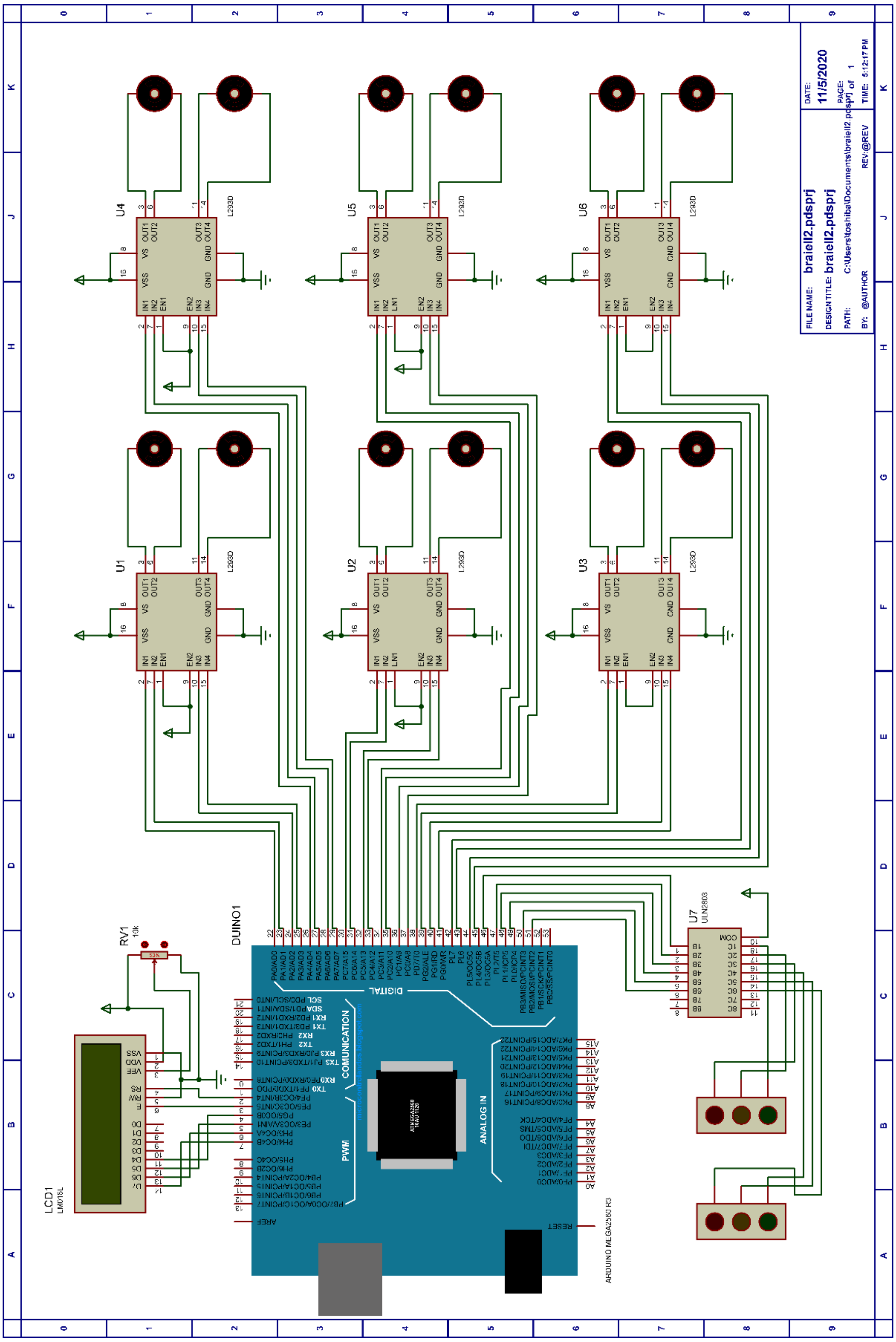

Figure 4.1 Circuit Diagram

Table 4.1: Converting of Characters to Binary

No.	Character input	Ascii Code	Binary
1	A	41	01000000
2	B	42	01000010
3	C	43	01000011
4	D	44	01000100
5	E	45	01000101
6	F	46	01000110
7	G	47	01000111

Table 4.2: Converting of Binary to Braille

No.	Character input	Binary	Braille
1	A	01000000	1
2	B	01000010	1,2
3	C	01000011	1,4
4	D	01000100	1,4,5
5	E	01000101	1,5
6	F	01000110	1,2,4
7	G	01000111	1,2,4,5

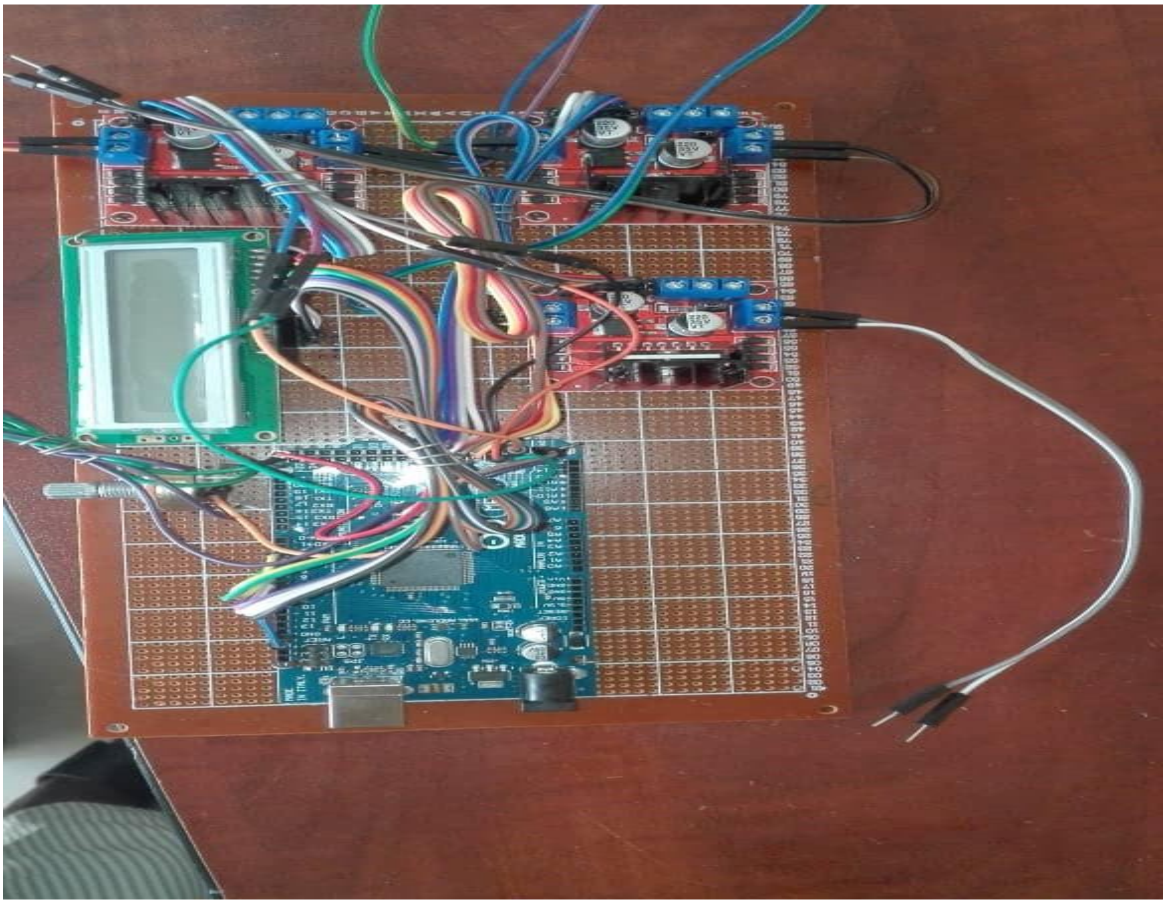

Figure 4.2 Circuit Layout

CHAPTER FIVE

Conclusion and Recommendation

5.1 Conclusion

In this project a design of hardware circuit based on Arduino was done to convert characters into braille, the circuit design with a low cost tools and components in order to reduce the overall cost on disable blind peoples, the problem behind the project is text messages in mobile phones or tablets that blind users cannot read and Most of the blind academics uses braille and sound to each blind student. In this project the programming of the Arduino has three stages, the first to capture the character or letter, then the convert it to ascii code, then to binary code, this is the basic algorithm used to convert any character into braille. The circuit was tested and reach the project goals successfully.

5.2 Recommendation

After finishing the project some recommendation can improve the work

1- To use an array of vibrators to represent more than a letter or character at once, in order to display a word.
2- Add double touch to play sound of the letter.
3- Add multi-languages to the programming.
4- Add voice recognition.

References

1. Dias MB, Teves EA. Towards technology with a global heart through compassionate engineering. Proceedings of Raising Awareness for the Societal and Environmental Role of Engineering and (Re)Training Engineers for Participatory Design (Engineering4Society); 2015. p. 96–100.

2. Köhlmann W, Lucke U. Alternative concepts for accessible virtual classrooms for blind users. Proceedings of 15th International Conference on Advanced Learning Technologies; 2015. p. 413–17.

3. Mikułowski D, Brzostek-Pawłowska J. Problems encountered in technical education of the blind, and related aids: Virtual cubarythms and 3D drawings. Proceedings of IEEE Global Engineering Education Conference (EDUCON); 2014. p. 995–8.

4. Bauwens B, Evenepoel F, Engelen JJ. Standardization as a prerequisite for accessibility of electronic text information for persons who cannot use printed material. IEEE Transactions on Rehabilitation Engineering,.1995 Mar; 5(1):84–9.

5. Dasgupta T, Basu A. Automatic transliteration of indian language text to braille for the visually challenged in India. Information Technology in Developing Counteries. International Federation for Information Processing Working Group 9.4 and Centre for Electronic Governance Indian Institute of Management. 2009 Oct; 19(3):1–10.

6. Wang C,Wang X, Qian Y, Lin S. Accurate Braille-Chinese translation towards efficient chinese input method for blind people. Proceedings of 5th International Conference on Pervasive Computing and Applications (ICPCA); 2010 Dec. p. 82–7.

7. Onur K. Braille-2. Automatic interpretation system. Proceedings of 23th Signal Processing and Communications Applications Conference (SIU); 2015 May. p.1562–5.

8. Nabiyev V, Akgül Ö, Braille T. Translation system. Proceedings of IEEE 18th Signal Processing and Communications Applications Conference (SIU); 2010 Apr. p. 856–9.

9. Hossain SA, Biswas LA, Hossain MI. Analysis of Bangla2-Braille machine translator. Proceedings of 17th International Conference on Computer and Information Technology (ICCIT); 2014 Dec. p. 300–4.

10. Parvathi K, Das JK, Samal BM. Odia Braille: Text transcription via image processing. Proceedings of International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE); 2015 Feb. p. 138–43.

11. Shreekanth T, Udayashankara V. A two stage Braille character segmentation approach for embossed double sided Hindi Devanagari Braille documents. Proceedings of International Conference on Contemporary Computing and Informatics (IC3I); 2014 Nov. p. 533–8.

12. Tang J. Using ontology and RFID to develop a new Chinese Braille learning platform for blind students. Expert Systems with Applications. 2013 Jun; 40(8):2817–27.

13. Araki M, Shibahara K, Mizukami Y. Spoken dialogue system for learning Braille. Proceedings of 35th IEEE Annual Computer Software and Applications Conference; 2011 Jul. p.152–6.

14. Velázquez R, Preza E, Hernández H. Making eBooks accessible to blind Braille readers. Proceedings of IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE); 2008 Oct. p. 25–9.

Appendix – Code

```
// Declare a char array with one index for every possible ASCII byte /
character
byte myBrailleDots[255];
int firstOutputPin = 2; // pin corresponding to least significant bit
int buzzerPin = 8; // Buzzer on pin  8
byte matrixPoints = 0; // byte that will store the point matrix configuration
                       // for a specific ASCII character
byte inByte;
byte mask = 1; //our bitmask

void setup() {
  // Temporarily assign 99 to every possible ASCII  byte/character
  // All the characters in the input string will decode to "99" by default
  for (int i = 0; i < 256; i = i + 1) {
    myBrailleDots[i] = 99;
  }

  // Now, only for the ASCII characters with a corresponding Braille
character.
  // assign the corresponding Braille Dot configuration
  myBrailleDots[32] = 0; // blank is 000000
  myBrailleDots[33] = 14; // exlamation mark is 001110
  myBrailleDots[34] = 7; // double quote is 000111
  myBrailleDots[34] = 2; // single quote is 000010
  myBrailleDots[40] = 15; // left parenthesis is 001111
  myBrailleDots[41] = 15; // right parenthesis is 001111
  myBrailleDots[44] = 8; // comma is 001000
  myBrailleDots[46] = 13; // period is  001101
```

```
myBrailleDots[48] = 28; // 0 is  011100
myBrailleDots[49] = 32; // 1 is  100000
myBrailleDots[50] = 40; // 2 is  101000
myBrailleDots[51] = 48; // 3 is  110000
myBrailleDots[52] = 52; // 4 is  110100
myBrailleDots[53] = 36; // 5 is  100100
myBrailleDots[54] = 56; // 6 is  111000
myBrailleDots[55] = 60; // 7 is  111100
myBrailleDots[56] = 44; // 8 is  101100
myBrailleDots[57] = 24; // 9 is 011000
myBrailleDots[58] = 12; // colon is 001100
myBrailleDots[59] = 10; // semicolon is 001010
myBrailleDots[63] = 11; // question mark is 001011
myBrailleDots[65] = 32; // A is 100000
myBrailleDots[66] = 40; // B is 101000
myBrailleDots[67] = 48; // C is 110000
myBrailleDots[68] = 52; // D is 110100
myBrailleDots[69] = 36; // E is 100100
myBrailleDots[70] = 56; // F is 111000
myBrailleDots[71] = 60; // G is 111100
myBrailleDots[72] = 44; // H is 101100
myBrailleDots[73] = 24; // I is 011000
myBrailleDots[74] = 28; // J is 011100
myBrailleDots[75] = 34; // K is 100010
myBrailleDots[76] = 42; // L is 101010
myBrailleDots[77] = 50; // M is 110010
myBrailleDots[78] = 54; // N is 110110
myBrailleDots[79] = 38; // O is 100110
myBrailleDots[80] = 58; // P is 111010
myBrailleDots[81] = 62; // Q is 111110
```

```
myBrailleDots[82] = 46; // R is 101110
myBrailleDots[83] = 26; // S is 011010
myBrailleDots[84] = 30; // T is 011110
myBrailleDots[85] = 35; // U is 100011
myBrailleDots[86] = 43; // V is 101011
myBrailleDots[87] = 29; // W is 011101
myBrailleDots[88] = 51; // X is 110011
myBrailleDots[89] = 55; // Y is 110111
myBrailleDots[90] = 39; // Z is 100111
myBrailleDots[97] = 32; // A is 100000
myBrailleDots[98] = 40; // B is 101000
myBrailleDots[99] = 48; // C is 110000
myBrailleDots[100] = 52; // D is 110100
myBrailleDots[101] = 36; // E is 100100
myBrailleDots[102] = 56; // F is 111000
myBrailleDots[103] = 60; // G is 111100
myBrailleDots[104] = 44; // H is 101100
myBrailleDots[105] = 24; // I is 011000
myBrailleDots[106] = 28; // J is 011100
myBrailleDots[107] = 34; // K is 100010
myBrailleDots[108] = 42; // L is 101010
myBrailleDots[109] = 50; // M is 110010
myBrailleDots[110] = 54; // N is 110110
myBrailleDots[111] = 38; // O is 100110
myBrailleDots[112] = 58; // P is 111010
myBrailleDots[113] = 62; // Q is 111110
myBrailleDots[114] = 46; // R is 101110
myBrailleDots[115] = 26; // S is 011010
myBrailleDots[116] = 30; // T is 011110
myBrailleDots[117] = 35; // U is  100011
```

```
myBrailleDots[118] = 43; // V is 101011
myBrailleDots[119] = 29; // W is 011101
myBrailleDots[120] = 51; // X is 110011
myBrailleDots[121] = 55; // Y is 110111
myBrailleDots[122] = 39; // Z is 100111
pinMode(buzzerPin, OUTPUT);
pinMode(2, OUTPUT);
pinMode(3,  OUTPUT);
pinMode(4,  OUTPUT);
pinMode(5,  OUTPUT);
pinMode(6,  OUTPUT);
pinMode(7,  OUTPUT);
Serial.begin(9600);
Serial.println("ASCII - Braille Arduino Converter");
Serial.println("LED test - begin");
digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
delay(3000);
digitalWrite(2,LOW);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
Serial.println("LED test -  end");
```

```
  Serial.println("Type some character: it will be transmitted to Arduino and displayed on a Braille 2 x 3 matrix");
}


void loop() {
  // Braille print data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    inByte = Serial.read();
    // say what you got:
    Serial.print("Received (inByte): ");
    Serial.println(inByte);
    // Translate inByte in matrix points
    Serial.print("Matrix points variable (myBrailleDots[inByte]): ");
    Serial.println(myBrailleDots[inByte]);

    // Braille print only admissible characters
    // the unadmissible ones decode to 99
    if (myBrailleDots[inByte] == 99)   // if unadmissible
    {
      Serial.println("Not a translatable character");
      digitalWrite(buzzerPin,HIGH); // buzz
      delay(250);
      digitalWrite(buzzerPin,LOW); // stop buzzing
    }
    else{
          int thisPin = 2;
          for (mask = 000001; mask<64; mask <<= 1) {
             Serial.print("thisPin = ");
```

```
            Serial.println(thisPin);
            if (myBrailleDots[inByte] & mask){ // if bitwise AND  resolves
to true
              Serial.print("AND successful, put pin on!");
              Serial.println(mask);
              digitalWrite(thisPin,HIGH);
            }
            else{ //if bitwise and resolves to false
              Serial.print("AND unsuccessful, put pin off!");
              Serial.println(mask); digitalWrite(thisPin,LOW);
            }
         thisPin = thisPin + 1;
         }
    }
    delay(3000); // allow 3 sec before passing to next  character
  }
}
```