orBED: Open Refreshable Braille Electronic Display for the Visually Impaired

Shouh Yann Mo
Pattarawadee Sanguanchom

Bachelor of Engineering in Software Engineering
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2020

Thesis – Academic Year 2020
Bachelor of Engineering in Software Engineering
Department of Computer Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang

Title: orBED: Open Refreshable Braille Electronic Display for the Visually Impaired

Authors

1. Shouh Yann Mo Student ID: 59090029
2. Pattarawadee Sanguanchom Student ID: 60090026

Approved for submission

…………………………………………

(Dr. Pipat Sookavatana)
Advisor

Date ……../……../……..

Acknowledgement

We would like to express our gratitude to Dr. Pipat Sookavatana, as our project advisor for his thorough guidance, constructive critiques and keeping track of our project's progress.

We would also thank our friends and family for sharing useful knowledge and guiding us to cope with the nature of the project development.

Abstract

orBED: Open Refreshable Braille Electronic Display for the Visually Impairedis a problem-driven project which aims to address the digitisation of the braille touch-reading system for the digital age through an open and affordable approach. The product will be constructed from economical hardware components, which allows for the product to be available and accessible to a wider demographic.

The refreshable braille display is expensive due to the mechanism behind it and it is only manufactured and sold in the US. orBED aims to find an alternative way to produce a refreshable braille display with cheaper cost and components but with limited functions. This project serves its purpose as a proof-of-concept on how such a system could be developed and further expanded on.

orBED is divided into two major systems. The first part is the software part where the text is captured and translated to motor rotations. The second part is the hardware part where the program rotates the disks to display the right braille characters.

This project is done by Shouh Yann Mo and Pattarawadee Sanguanchom.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Brief Overview

***or*BED: Open Refreshable Braille Electronic Display for the Visually Impaired**, which will henceforth be concisely referred to as ***or*BED** within the report, is a problem-driven project which aims to address the digitisation of the braille touch-reading system for the digital age through an open and affordable approach. The product will be constructed from economical hardware components, which allows for the product to be available and accessible to a wider demographic.

1.2 Motivation

The project will be created to address an existing issue, that being the often costly prices of existing refreshable braille display solutions, costing as much as $15,000 and at least $3,500 depending on the number of characters displayed and the additional functionalities.[22] However, the current cheapest refreshable braille display is $599.[19]

Figure 1.1: The US-sold Refreshable Braille Display Info[16]

Nov 25, 2020	Cost	Size	Weight	Characters/Cells	Battery duration	Source
Refreshable Braille Display						
Actilino	$2,795.00	16.6 cm width, 10.9 cm depth , 2.9 cm height	0.42 kg	16	30 hrs	https://hims-inc.com/product/actilino/
Active Braille (Pre-Owned)	$6,495.00	31.7 x 12.3 x 2.8 cm	0.90 kg	40	20 hrs	https://hims-inc.com/product/active-braille-preowned/
ALVA BC680	$10,706.16	59.5 x 7.5 x 1.8 cm	1.19 kg	80	10 hrs	https://in.optelec.com/products/alva-bc680.html,
						http://exceed.lv/index.php/en/produkti/braila-produkti/braila-rindas/view/227:braille-display-alva-bc680
ALVA USB640	-	33.8 x 7.5 x 1.8 cm	0.55 kg	40	USB	https://in.optelec.com/products/alva-usb-640-comfort.html
Braille Star 80	$9,995.00	58.5 x 24.5 x 2.4 cm	2.55 kg	80	USB	https://hims-inc.com/product/braille-star-80/
Braille Wave	-	36 x 12 x 3.5 cm	1 kg	40	20 hrs	https://www.nattiq.com/en/node/44
Brailliant B 80	$7,985.00	56.5 x 7.8 x 1.8 cm	1.07 kg	80	USB	https://store.humanware.com/hus/brailliant-80.html
Easy Braille Bluetooth	$4,495.00	30.5 x 9.0 x 2.9 cm	0.74 kg	40	USB	https://hims-inc.com/product/easy-braille-bluetooth/#tab-id-1
Focus 14 Blue Braille Display	$1,295.00	Varies x 8.2 x 1.9 cm	0.283 kg	14	USB	https://store.freedomscientific.com/products/focus-14-blue-5th-generation
Orbit Reader 20	$599.00	16.764 x 10.922 x 3.048 cm	0.435 kg	20	USB	https://www.orbitresearch.com/product/orbit-reader-20/
Seika 80	-	147.32 x 22 x 6.35 cm ???	1 kg	80	USB	http://www.hosoda-jp.co.jp/m_seika.html
Vario Ultra 20	-	18.8 x 8.9 x 2.0 cm	0.32 kg	20	USB	https://www.visiobraille.de/index.php?article_id=21&clang=2

Statistics from 2009 collected by the National Federation of the Blind in the USA show that less than 10% of the 1.3 million blind people in the USA can read braille.[29] This dwindling in the rate of literacy could be attributed to how inaccessible braille has become in the digital era, due to lack of widespread accessible technology which allows for the digitisation of braille. We would like to explore into the possibilities of developing an interface that can bring braille into the digital era in an open, accessible, and affordable form.

1.3 Challenges

The project will be adequately challenging to implement, as it proposes unique challenges in hardware design through the introduction of cost-efficiency requirements in the selection of implementation methods, which in turn will affect how the firmware will be developed to match the selected representation of braille physically.

1.4 Basic Usage Overview

The product, in its complete form, will be a physical device with erected dots arranged into cells representing a braille cells on disks. The disks can rotate individually to form characters in the braille alphabet, which the user can scan over with their finger to read. The product will be able to be connected to a computer or mobile device program, which will send text information from the operating system to the braille display to be displayed as braille. The braille display will be powered and controlled by a Raspberry Pi 4 as its processing unit on which the firmware will be contained and Arduino Mega to control the motors.

1.5 Benefits of Refreshable Braille Displays

Individuals with vision impairment may benefit more from using refreshable braille displays over other methods of non-visual information representation such as speech synthesis in certain contexts, such as when required to proofread for spelling or typographical formatting of text. It also gives them access to the information directly, and remains functional in environments or situations where hearing may also be impaired or affected.[22] This means that it is a more accessible approach to receiving information for those who have both vision and hearing impairments.

1.6 Scope and Limitations

Due to the goal of the project being to produce a refreshable braille display which is affordable, the number of cells that can be displayed by the display may be significantly lower than other, more expensive solutions available on the market, due to cost limitations on the hardware components. This means that they reader will be able to display less characters per line comparatively. As this product will be a proof-of-concept prototype, the braille cells may also have to be larger than most conventional cell sizes seen on other existing solutions, as to accommodate for the hardware components required to rotate each disk for each half character, as we may be using components which may be significantly bulkier in order to save on cost. When printing 3D objects with parts that protrude out, that part's quality tend to drop even if it's designed well or not.

1.7 Thesis Structure

This thesis consists of seven chapters which are arranged as follows:

- Chapter 1 **Introduction** - refers to the motivation, objectives, scope of work, and thesis structure of this thesis.

- Chapter 2 **Literature Review** – shows some projects or researches done by academics which relate to this project and are beneficial as references to support the development of this project.
- Chapter 3 **Background knowledge** - explains the knowledge and technology necessary for the reader to understand the thesis.
- Chapter 4 **System Architecture and Design** – explains and illustrates technical aspects of ***or*BED**.
- Chapter 5 **System Development** - contains the details about the project's development environment and phases.
- Chapter 6 **Results** - reports the project outcome in the current status.
- Chapter 7 **Conclusion** - is the chapter that talks about what have been done and the future of the project.

1.8 Terminologies Used In This Project

The terminologies and jargon with its explanations is in Table 1.1.

Table 1.1: Terminologies Used in ***or*BED**

Term	**Definition**
GUI	Graphical User Interface.
Stepper Motor	A brushless DC electric motor that divides a full rotation into a number of equal steps.
Steps	Total amount of increments to make a full revolution of a circle.
Disk	The object that is octagonal shape with bumps on it and holes at the side. It is used for the construction of a braille character, requiring two disks to make one character. Each face of the disk correlates to a pattern covered in half of the braille character. In this project, the character would be 6-dotted cell instead of 8-dotted cell.
Car	A vehicle to carry bipolar NEMA 17(Big stepper motor) back and forth to control which disk to rotate.

Chapter 2

Literature Review

2.1 Literature Survey

Note: This chapter of the report contains excerpts paraphrased or taken directly from the source paper. Due to the concise nature of the wording in the source paper, paraphrasing was deemed unnecessary in conveying the key points of some sections, and thus the section was taken from the sources directly. This is not an attempt to plagiarise, but rather, to preserve the readability and ease of comprehension from the source. As such, it is crucial to note that any factual excerpts in this chapter is in no way our own work, and is fully credited to be the works of the authors of the source paper.

2.1.1 Overview

This literature review section will address the paper "Refreshing Refreshable Braille Displays".[25]

2.1.2 Summary

Introduction and Background

For most visually impaired computer users, the preferred method of receiving information from electronic text is through speech synthesis, as it is often inexpensive when compared to braille, as it generally requires no additional hardware. The method, however, has been shown to provide a reading experience which is intrinsically different from reading text for those who prefer to learn visually. As such, active reading modes, such as braille, could offer a more advantageous experience of receiving information through active reception of text when compared to speech. It can also present as a better medium for representing more technical information, which speech poorly conveys. The unpopularity of braille as a reading medium in the current day can be attributed to costly equipment required to display electronic text as braille, decline in braille education to the newly-blind, and increasing costs in production of braille books causing a decline in production.

It has been shown that reading braille by passing the finger over the surface of the braille text improves ease of comprehension of the text. This is possibly due the fact that the friction between the finger and the text provides continuous stimulation which aids in the rendering of the braille image, or that the proprioceptive (perception of body movement and position) cues provided by the movement of the hands aid in the

understanding of the spatial orientation or structure of the characters. The paper tests the hypothesis that both factors are involved in reading braille.
There are two possible display types:

- static: display does not move
- sliding: display moves

and two possible proprioception conditions:

- proprioceptive: finger moves
- non-proprioceptive: finger does not move

All combinations of which form the conditions for reading braille.

Methods

The study contained used two display types (static and sliding), two proprioception conditions (proprioceptive and non-proprioceptive), and three levels of display presentation speed (baseline, fast, and slow). The participants were limited to their dominant reading finger.

The participants consisted of five experienced braille readers between 19 and 48 years of age, with a mean age of 36, from the University of Michigan and from the local chapter of the National Federation of the Blind. No participants reported to have known issues relating to the sensitivity of their fingertips. Due to technical difficulties, only data for four participants is reported in the paper.

A custom apparatus was built featuring a motorised platform that could easily be configured to one of the four conditions:

- CLOCK: static and proprioceptive
- LINE: sliding and proprioceptive
- TAPE: static and non-proprioceptive
- MOUSE: sliding and non-proprioceptive

Three attachments could be mounted onto the apparatus: A 32-cell refreshable braille display, A single-cell braille display surrounded by pin-less cells on each side, and a finger fixture resting platform.

Each iteration lasted approximately 1 hour and 45 minutes. Participants read and signed a consent form in braille, and had demographic information collected at the start of the session. The experiment consisted of 18 iterations. In each iteration, participants read, in sequence, five random letters out loud as they appeared under a single finger. Each iteration was initiated with a sequence of two equals signs (all pins up) and a space (all pins down). The separators between the cells within each trial was held constant so that the initiating sequence gave an indication of the speed at which the subsequent random letters would appear. The participants were told to read the letters aloud as accurately and quickly as possible. The iterations covered all experimental conditions.

Results

Participants made errors when naming letters more often under the static conditions (CLOCK and MOUSE) compared to sliding conditions (LINE and TAPE), which were intensified at faster reading speed conditions. Participants produced significantly fewer errors in sliding conditions, and error rates increased significantly with increasing display speed. However, proprioceptive cues on participants' error rates was not found to be significant. Additionally, significant interaction was found between sliding and speed.

Conclusion

The results of the experiments indicate that sliding action positively affects accurate comprehension of braille letters, but that passive proprioceptive action does not affect reading accuracy. This suggests that single-cell displays that do not incorporate sliding contact are likely to be less effective for braille reading than displays which do incorporate sliding motion.

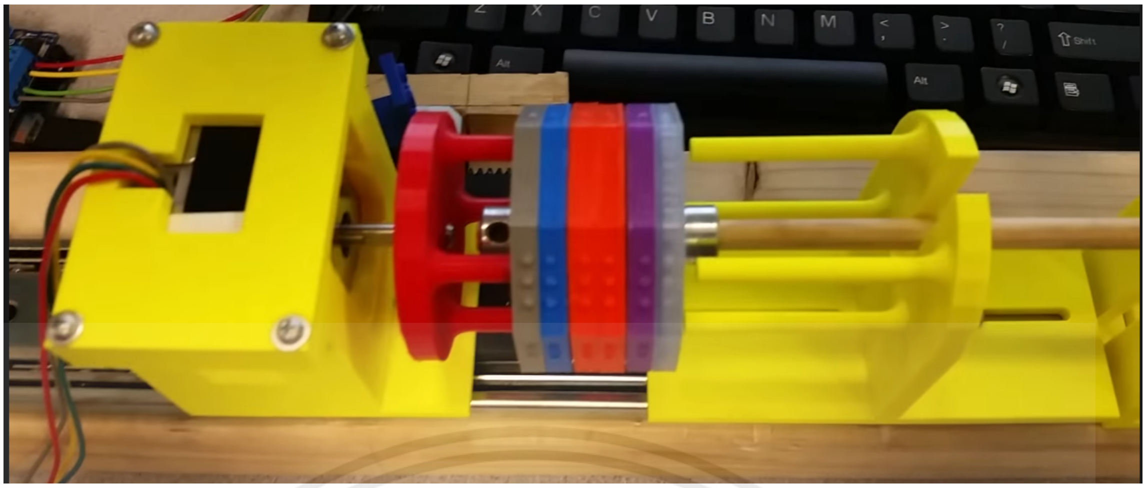
2.1.3 Relevance to Project

This paper gives insights into the effectiveness of different possible implementations of the mechanisms behind refreshable braille display responsible for representing braille characters for tactile reading. The results sheds light on the effectiveness of each implementation design based different methods of reading braille text, which we can take into consideration when designing our own implementation for the project's hardware module, specifically with regards to the pin actuation mechanism.

2.2 Related Work

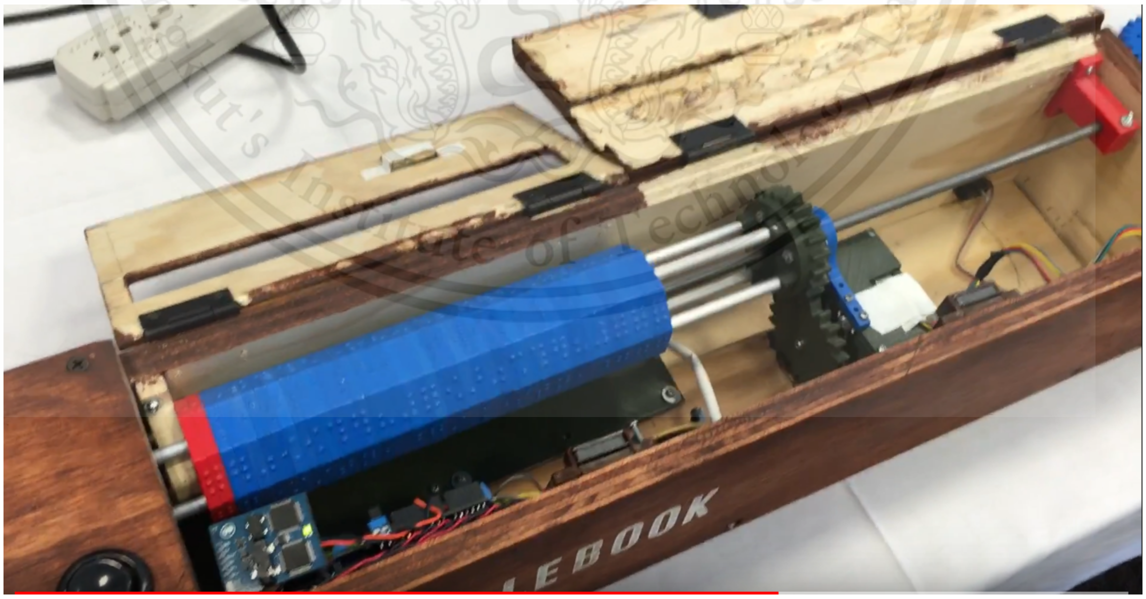
There are two videos that helped us decide to do this project. The concept itself is interesting.[8] It uses one big stepper motor and two gears to rotate each disks to its designated position for its prototype. It also uses a push in and lock mechanism on each disk to select which disk to rotate. See Figure 2.1 on the next page.

Figure 2.1: Inspired By [23][24]



(a) The Prototype

(b) Final Product



(c) Final Product's inner workings

Chapter 3

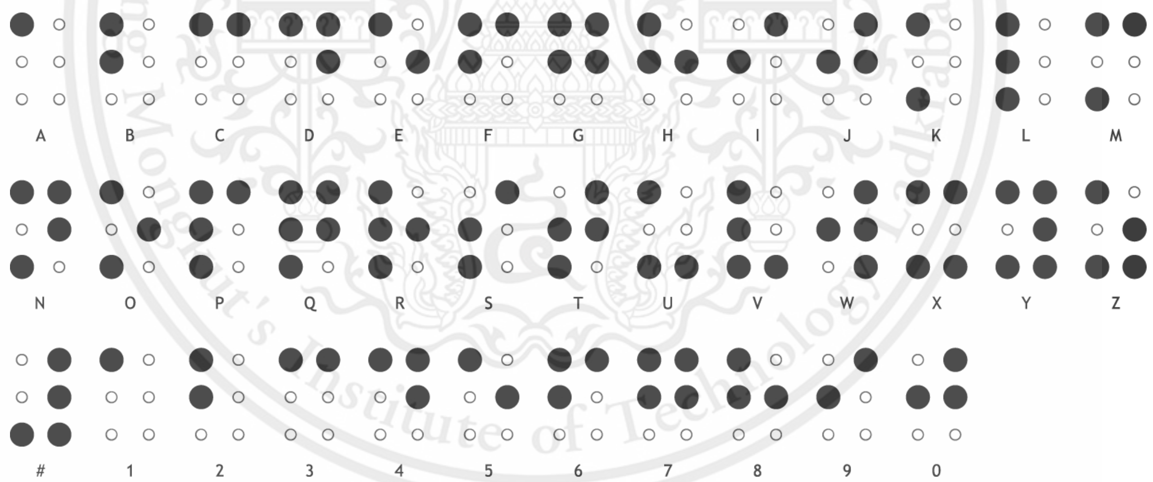
Background Knowledge

3.1 The Braille System

3.1.1 Overview

Braille is a text-representation system for the visually impaired to retrieve information by tactile means, as well as produce them for other visually impaired people to read. It uses a set of dots which are arranged to form unique combinations representing characters, symbols, or punctuation marks. Figure 3.1 shows the the Braille Alphabet as used internationally.

Figure 3.1: The International Braille Alphabet[28]

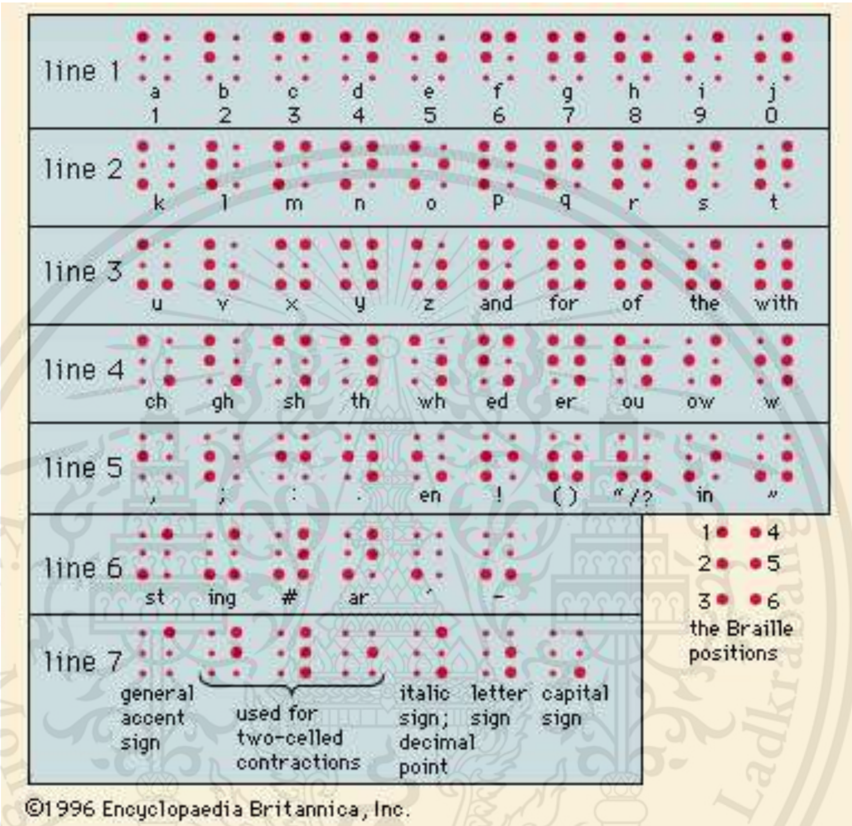


Braille can be read by touching one's fingers against the embossed braille text from cell to cell, typically moving from left to right along each line.

A line of Braille is typically about 40 characters long, and a page of Braille is about 25 lines long. The paper used for printed braille is thicker to accommodate for the embossing of the dots. This results in books for braille readers being significantly thicker than standard-sized printed books.

Modern braille standards exist in many different formats and configurations, which are dependent on variation in sizing, spacing, and languages. Contracted/abbreviated variations of words are also commonly used in order to save on the amount of cells needed to represent a concept, thus saving on space. Figure 3.2 includes common abbreviations used in English Braille. Those are classified under Grade II Braille. Otherwise, Grade I is without the contractions, where every word is spelled out letter-by-letter.[34]

Figure 3.2: English Braille[5]



3.1.2 History

Charles Barbier, who served under Napoleon's army during the early 1800s, developed a unique system for soldiers to communicate. The concept of a tactile medium for communications arose as a result of soldiers being spotted during night time due to illumination from lamps which was required to read text visually in dark environments. This unique system came to be known as "night writing", and used cells consisting of dots arranged in a 6 by 2 configuration. The system allowed soldiers to be able to communicate with each other in complete darkness, as the tactile mode of reading did not require any illumination in order to be read, as the text did not require any visual comprehension. However, due to the large size of the cells, the average human fingertip could not feel all the dots at once, thus making the text format counterintuitive in that regard.

Later, Louis Braille, who was blind from his early life, expanded on Charles Barbier's night writing system and aimed to create an efficient communication system for fellow visually impaired people. After the development and refinement of the system, the form factor of the cells became a 3 by 2 arrangement as can still be seen being used today, and became one of the official ways to communicate for blind people.[30]

3.1.3 The Braille Alphabet

The basic braille alphabet, including numbers, punctuation, and special symbols, are constructed from a combination of up to six dots in a vertical three-by-two arrangement. This yields a total of 64 possible combinations for the dots to be arranged in. The Unicode block for 6 dot braille is U+2800 - U+283F.[28] The dots are numbered by row first, then by column. For 8-dot braille cells, the bottom two dots are numbered last, making them dots 7 and 8. See Figure 3.3 on page 10.

Braille Patterns

There are 3*2 block version and 4*2 block version of braille characters. For the Unicode, from U+2800 to U+283F is for 6-dot characters and from U+2840 to U+28FF is for 8-dot characters.

Figure 3.3: Braille Dot Numbering[21]

For the braille fonts, some use unpunched dots as small dots and some Linux fonts uses squares instead of circles.[1]

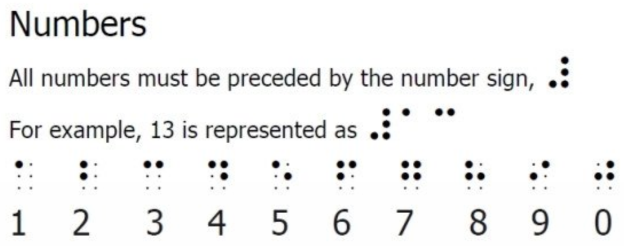
The International Standard Braille Alphabet

The letters a-z are common and standard for most braille country tables, as seen in Figure 3.1.[28]

The International Standard Braille Numbers

Numbers are represented in Braille using the same characters used to represent the alphabet characters a to j and a special number symbol consisting of dots 3, 4, 5 and 6 must precede number strings. Therefore if you want to use braille number you have to start the first cell with dots 3, 4, 5 and 6, as seen in Figure 3.4.[28]

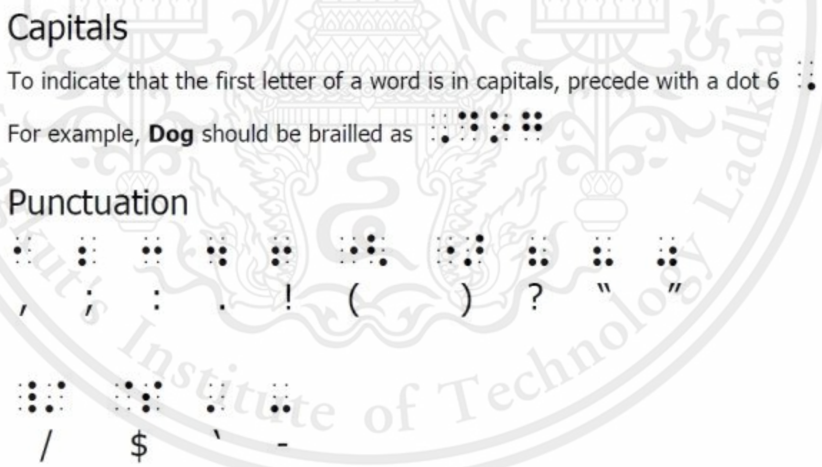
Figure 3.4: Braille Numbers



Braille Punctuation

Punctuation marks are commonly used in almost all languages, and include symbols such as period, question mark, exclamation point, comma, semicolon, colon, dash, hyphen, parentheses, brackets, braces, quotation marks, as seen in Figure3.5.[28] Capital Letters are made by putting a capital sign which is dot 6 in front of the letter you want to capitalize, as seen in Figure3.5.

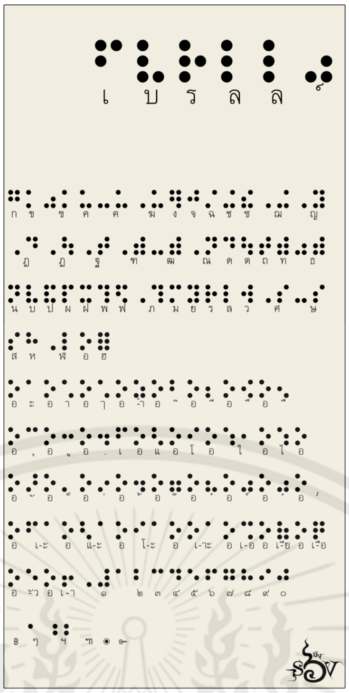
Figure 3.5: Braille Punctuation[6]



Braille in other languages

The Thai braille uses one 3*2 cell and two 3*2 cells to represent their characters.

Figure 3.6: Thai Braille[27]



3.1.4 Braille Cell Dimensions

Different braille standards exist in different countries, and they can vary in character spacing or dot spacing within each character cell. Most countries use a standard produced by the Perkins mechanical braille writer. The standard was established from extensive testing by braille readers to optimise the legibility of the braille text.

Other standards also exist for more specific applications of braille text, such as on pharmaceutical labels, so that the visually impaired can read medical instructions on the packaging of medication. An example of a standard that was designed specifically for that usage is the Marburg Medium braille standard, which has now been officially adopted by the European Union.[7]

3.2 Refreshable Braille Displays

3.2.1 Overview

A refreshable braille display is an electro-mechanical device for displaying braille characters which has a series of refreshable, modifiable braille cells on its surface. Most displays contain a single line including between fourteen to eighty braille cells. Instead of small holes in a piece of paper, each braille dot in these cells is represented by round-tipped pins raised through holes in the plate surface. This allows people who are blind to read information in braille by running their fingers over the refreshable braille cells and then changing the display to the next, usually with a press of a button, to show the next set of characters. Working similarly to a computer monitor, braille displays must be given information in order to function. This information can come from a computer running screen reading software, a mobile devices such as a smart phone or from text the user enters on the keyboard of the device. Many individuals who are blind attach a braille display to a computer running a screen reader in order to interact with the computer.[32]

3.2.2 Why use Refreshable Braille Display

There are many benefits of refreshable braille display. The first being that hardcopy or paper braille are large form factors by nature, causing them to take up more physical storage space. This poses difficulties when required to be moved around. Refreshable braille displays work on a principle that is dependent on receiving information from electronic files which means a large amount of information can be stored on a small electronic device called a storage card. For example, a hardcopy version of the World Book Encyclopedia was produced in braille in the 1970s. This book was well over one hundred volumes and took up an entire wall of a library. In an electronic version, the same document could be transferred to a storage card which would easily fit physically in a small place. When connected to a refreshable braille display, the entire text of the encyclopedia is available. This compact form factor allows for larger braille files to be shared across many electronic devices between many people, allowing text written for braille to be distributed with ease. [32]

3.3 Implementation Methods

3.3.1 Methods of Integration

Screen reader software take text from UI elements and convert them into formats which can be consumed by the visually impaired, such as through speech synthesis or through conversion to braille code. It could be taken another step further by using pictures taken by the camera that contains text and convert it to braille. Currently there are products that have both text-to-speech and braille display such as ScreenReader by Dolphin. It also is able to scan text from papers to be read or display as braille. The downside is that it is expensive.[26]

3.3.2 Methods of Pin Actuation

The reason why most refreshable braille readers are expensive is due to the actuators. They use actuators called Piezoelectric actuators, which allows them to expand when a voltage is driven through them.[20] One of the alternatives is to use cam-follower assembly that uses a similar principle of mechanical pens.[17] There is also an interesting alternative of using microfluids and pumps to replace the pins by pumping in the fluid to make bumps on the surface.[12]

3.4 Embedded Systems

Being an integral part of modern computing, embedded systems can be found in many electronic devices and machinery.[14] Embedded system is a hardware and software system that uses microprocessor or microcontroller to accomplish dedicated functions within a larger mechanical or electrical system.[31] It can be found everywhere but is usually hidden from sight. Such examples are a television, laptop, and smartphones.

3.4.1 Microprocessor and Microcontroller

Microprocessor is a small chip that is capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it.[15] The common components of microprocessors consists of control unit, I/O(input/output) Units, ALU, and registers.

Microcontroller is a small and low-cost microcomputer, which is designed to perform the dedicated tasks of embedded systems. The general microcontroller is composed of the processor, the memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters), etc.[15]

The most common standard power supply voltages that is used in the microcontrollers are 3.3 volts and 5 volts DC though other voltages are also used for external hardware that uses more than 5 volts.[13] This electric supply both powers the microcontroller unit and acts as binary logic level indicator.

3.4.2 DC Motors and Motor Driver Boards

Direct Current Motor

DC Motor or Direct Current Motor is an electromechanical device which use the interaction of magnetic fields and conductors to convert the electrical energy into rotary mechanical energy.[9] It is most commonly used for producing continuous movement and the speed of rotation can be controlled easily, making them ideal for use in applications were speed control, servo type control, and/or positioning is required.[9] A DC motor consists of two parts, a “Stator” which is the stationary part and a “Rotor” which is the rotating part.[9] There are three types of DC Motor in use:

1. Brushed Motor: It produces a magnetic field in a wound rotor (the part that rotates) by passing an electrical current through a commutator(The rotor of a DC machine that consists of current carrying conductors connected together at one end to electrically isolated copper segments) and carbon brush assembly, hence the term “Brushed”.[9] The stators (the stationary part) magnetic field is produced by using either a wound stator field winding or by permanent magnets. Generally brushed DC motors are cheap, small and easily controlled. This includes series and shunt DC motors, see Figure 3.7 and Figure 3.8.

Figure 3.7: A General-Purpose DC Brush Motor Diagram[11]

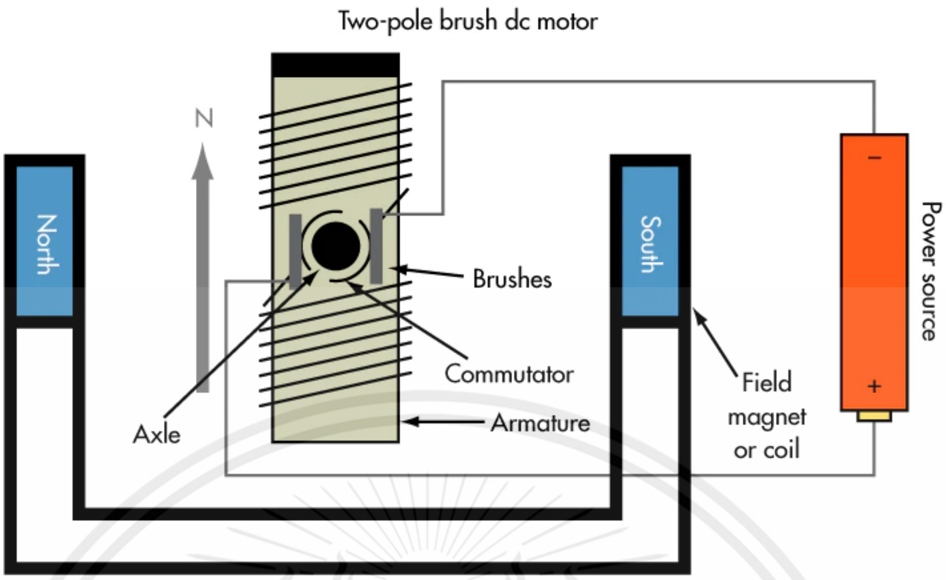
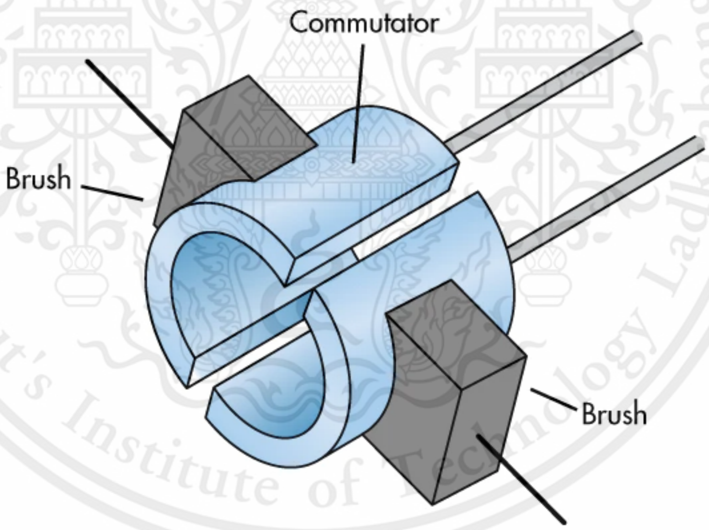
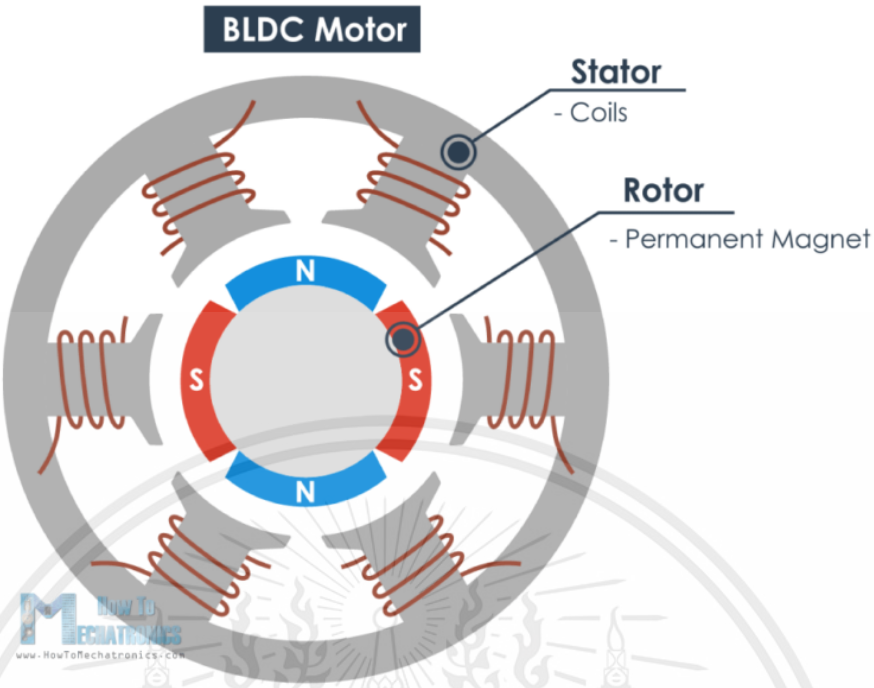


Figure 3.8: Inner Part of the Brush DC Motor[11]



2. Brushless Motor: It produces a magnetic field in the rotor by using permanent magnets attached to it and commutation is achieved electronically. They are generally smaller but more expensive than conventional brushed type DC motors but they have better torque/speed characteristics, are more efficient and have a longer operating life than equivalent brushed types.[9] See Figure 3.9.

Figure 3.9: Brushless DC Motor Diagram[10]



3. Servo Motor: A brushed DC motor with some form of positional feedback control connected to the rotor shaft. They are connected to and controlled by a PWM type controller and are mainly used in positional control systems and radio controlled models.[9]

The DC Stepper Motor is a type of synchronous brushless motor in that it does not have an rotor with a commutator and carbon brushes but has a rotor made up of many, some types have hundreds of permanent magnetic teeth and a stator with individual windings.[9]

The stepper motor does not rotate in a continuous fashion like a conventional DC motor but moves in discrete "Steps" or "Increments", with the angle of each rotational movement or step dependant upon the number of stator poles and rotor teeth the stepper motor has.[9]

Motor Drive Board

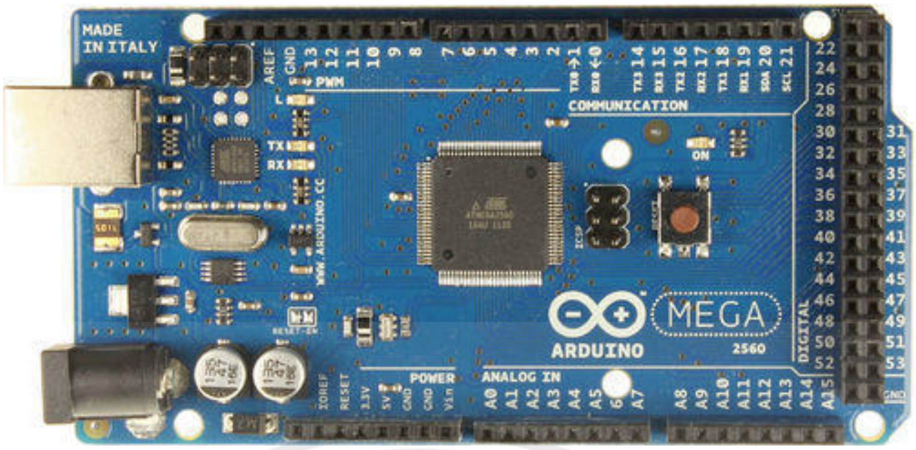
A motor driver board is just a convenient way of creating what is known as a "H Bridge". This is an arrangement of 4 (or more, but 4 primary) transistors.

A motor driver board gives you greater control over a motor than a single transistor alone. It allows you to control the direction, which you can't do with a single transistor, and it also gives you access to braking modes where the motor becomes harder to turn thus assisting in slowing down the load.[18]

Microcontroller in this Project

The microcontroller in this project is from the Arduino family. Arduino is a microcontroller which provides an inexpensive, beginner-friendly yet powerful open source platform for hardware interfacing and hardware system design. An example would be Arduino Mega as you can see in Figure [3].

Figure 3.10: Arduino Mega[3]



3.4.3 Wire Communication

Communication in this case means data flowing between each component is an important part of embedded systems routines.

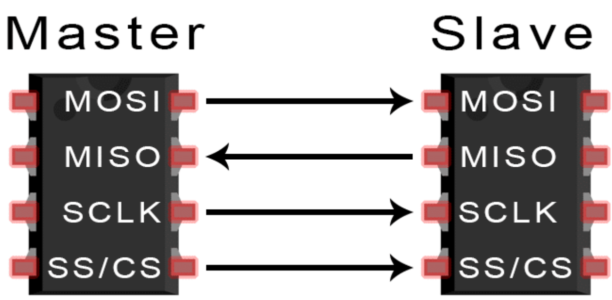
Communication in embedded systems relies on two factors: logical value of data (0 or 1) and clock.[33] Clock refers to an alternating 0 and 1 signal in a uniform constant frequency which drives the flip-flops in a processor or similar devices to sign the time of propagation through the next sequence of execution. A clock cycle (single 0-1 sequence) is the smallest time unit of a processor time.

Serial Peripheral Interface

Serial peripheral interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals.[2] SPI which is shown in Figure 3.11, connects one master to a slave or more by using four communication lines to operate:[4]

- MOSI (Master Output/Slave Input): Line for the master to send data to the slave
- MISO (Master Input/Slave Output): Line for the slave to send data to the master
- SCLK (Clock): Line for the clock signal
- SS/CS (Slave Select/Chip Select): Line for the master to select which slave to send data to.

Figure 3.11: SPI Wiring [4]



GPIO/Analog Signal

Some data are not bound to real-time constraints nor holding a complex data type.[33] An example of such data is one or combined logic or voltage level(s) transmitted through GPIO pin(s). These values are pulled into a variable in the code. The code may be also able to modify the levels in the hardware. This routine is normally done in an asynchronous manner.

3.5 Embedded Systems Development

Embedded systems development integrates knowledge and skills in both hardware and software development. Most of the time, embedded system software is written in low level programming, i.e. more machine-focused and relies heavily on real-time systems. Interaction between software, hardware and user inputs plays crucial role in the usability of an embedded system product.[33]

For the design of this project, to comply with the previously mentioned design principles, the materials and parts required to build were chosen to be easily sourced and not too expensive, while being sufficient to enable the design to deliver its capabilities as close as possible.

Chapter 4

System Architecture and Design

4.1 Requirements

4.1.1 User Requirements

***or*BED**'s user requirements are given in Table 4.1:

Table 4.1: User Requirements

Spec. #	**Description**	**FURPS+**
U1	User will be able to upload the text file into the program for the program to translate text to motor steps.	Functional
U2	User will be able to navigate through the text by pressing either the *Next* or *Previous* button to advance to the next text element.	Functional
U3	User will be able to know what word(s) is on display on the ***or*BED** device via the Arduino Serial Monitor.	Functional

4.1.2 Functional System Requirements

***or*BED**'s functional requirements are given in Table 4.2:

Table 4.2: Functional System Requirements

Spec. #	**Description**	**FURPS+**
F1	***or*BED**'s firmware will be able to retrieve braille-formatted text information from the text document to be represented on the display.	Functional
F2	***or*BED**'s firmware will be able to use braille-formatted text information to determine which disk and steps to rotate for the display to correctly represent the braille information.	Functional

4.1.3 Non-functional Requirements

***or*BED**'s non-functional requirements are given in Table 4.3:

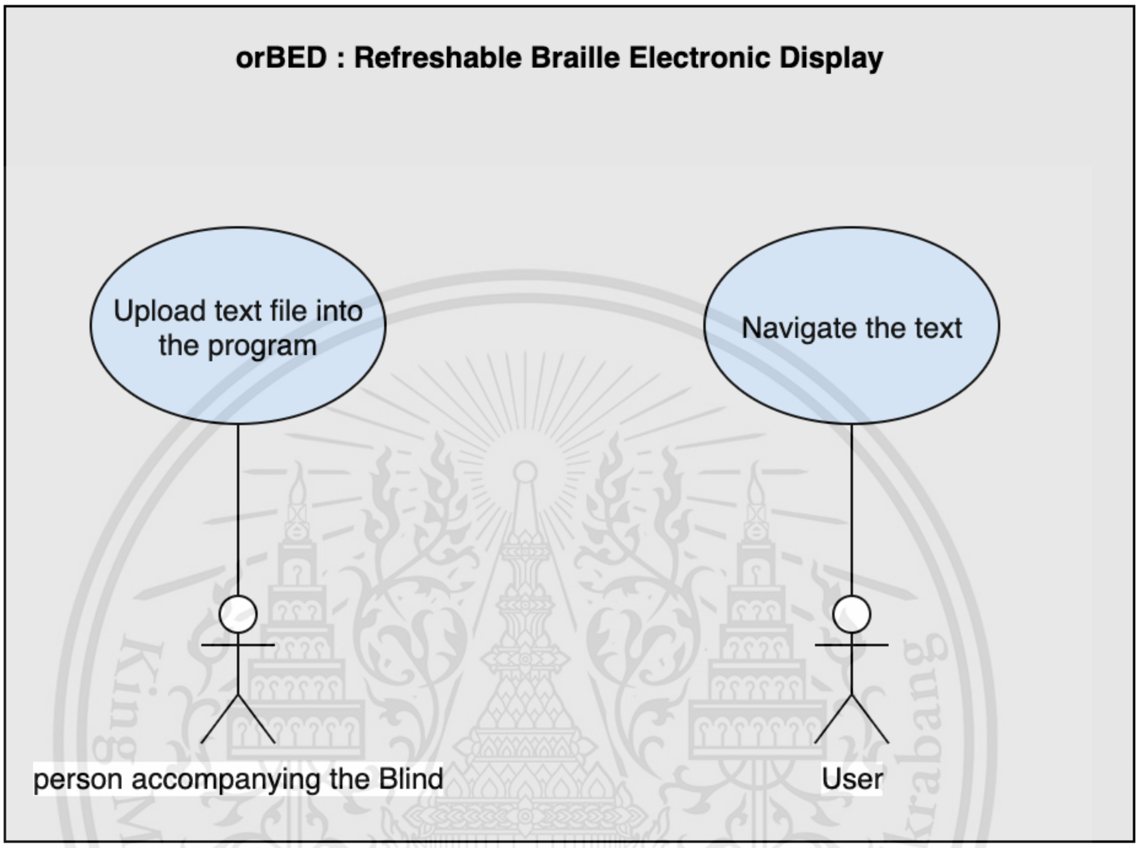
Table 4.3: Non-Functional Requirements

Spec. #	Description	FURPS+
N1	***or*BED**'s firmware will be implemented on a laptop.	Implementation
N2	***or*BED**'s firmware will be implemented using C++.	Implementation
N3	***or*BED**'s braille display will consist of 12 braille cells.	Imp., Usability
N4	The dimensions of ***or*BED**'s braille cells will follow the English Giant Dot braille cell dimensions as close as possible to achieve a balance between legibility and feasibility in implementation.	Imp., Usability

4.2 Use Cases

***or*BED** is compromised of three major use cases currently, which are described below.

Figure 4.1: Use Case



4.2.1 Use Case 1: User Uploads the Text File

[**Summary**] User uploads a .txt file to the firmware and the firmware will translate characters in a word to motor steps. Until the user press *Next* button, the firmware will wait for the user to start moving to the next word.

Table 4.4: Use Case 1

Use Case Section	**Description**
Use Name	Upload Text File
Scope	Firmware (In-device)
Level	User-goal level
Primary Actor	The one accompanying the blind person
Preconditions	The text file doesn't have a word more than five characters and only contains alphabets and no capital letters. The text must be in English.
Success Guarantee	The text is uploaded and loaded waiting for the user to interact.

Main Success Scenario	Actor	System Response
	1 The user uploads the text file into the firmware.	**2** The firmware store the text and splits them into an array of words. **3** The firmware waits for the user input via buttons.

4.2.2 Use Case 2: User Navigates the Text

[**Summary**] In order for the User to read the next word(s) or the previous word(s), the user have to press the *Next* button or *Previous* button on the ***or*BED** device. The firmware signals the program to send text from the next/previous text element to the firmware. The firmware loads in the text from the new text element. While the firmware loads the text element, the text element would be displayed to show what is being loaded into the firmware. The motor will then turn the disks into the loaded word arrangement.

Table 4.5: Use Case 3

Use Case Section	**Description**	
Use Name	Next Text Element	
Scope	Firmware and Software (Out of device)	
Level	User-goal level	
Primary Actor	The blind person	
Preconditions	The program has a text element that can be selected next and the disks are not rotating. The blind person must be able to read English braille	
Success Guarantee	The disk displays the text corresponding to the loaded text element.	
Main Success Scenario	**Actor**	**System Response**
	1 The user presses the *Next / Previous* button.	**2** The firmware signals to the program to request for text from new text element. **3** The firmware loads in the beginning of the corresponding text. **4** The disks display the next text and reset after user press either *Next / Previous* button.
Extensions	**3a** No text element to be loaded and nothing is done upon.	
		1) Nothing happens.
	3b No text element to be loaded and User press *Previous* button.	
		1) Nothing happens.

4.3 Design Overview

Figure 4.2: Current Situation

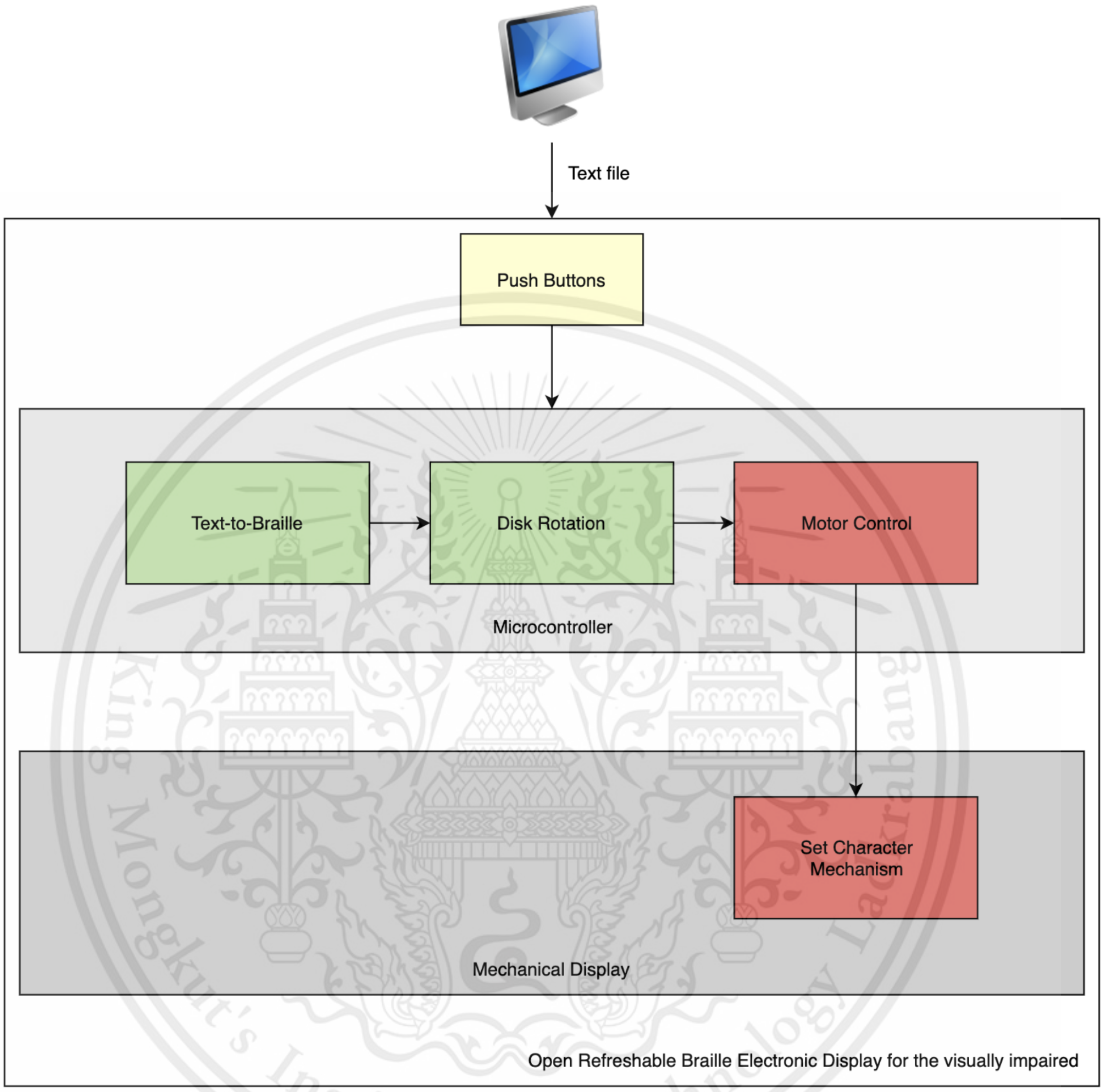
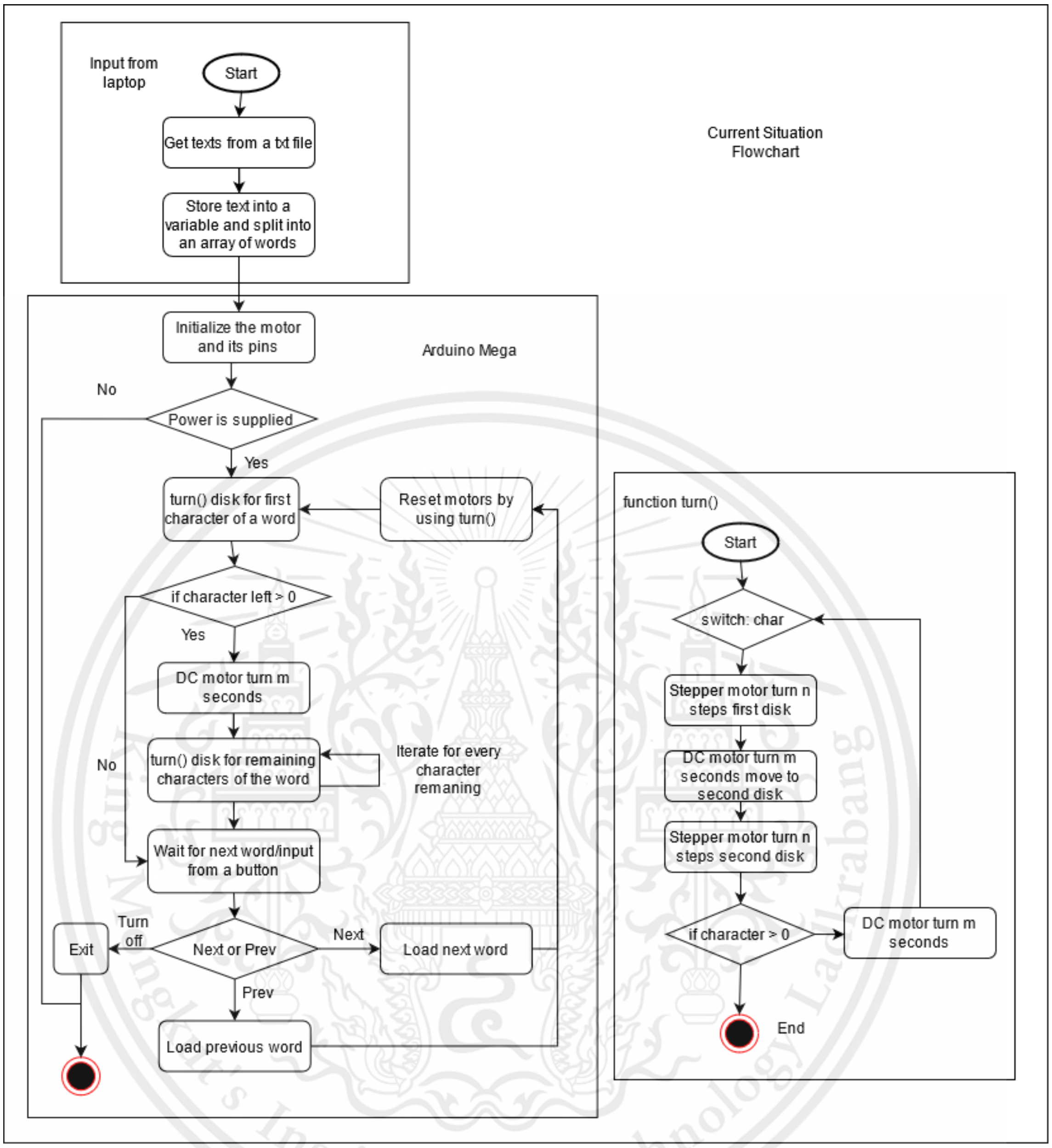


Figure 4.3: Current Situation(More details)



The general flow of how the project goes as the following: the user uploads the text from the computer via text files or image files that contain text. Then the program will extract the text from the input to translate to motor steps according to the characters. It will command the disks to spin according to the word being loaded. The program will wait for the interaction by the user by pressing one of the buttons. The hardware will only display at most a 2 character word as this is a proof of concept project.

Chapter 5

System Development

5.1 Development Tools

5.1.1 Hardware

Prototype 1

- Arduino Mega: additional pin slots and controlling the motors
- 12 Volts 28BYJ-48 stepper motors with X113647 Stepper Motor Driver Boards and ULN2003A IC Chip: turning the disks into right positions
- MicroSD Card Adapter: read text file or image with text from
- 2-legged buttons: notifying the program whether to proceed to the next word or previous word

Prototype 2

- Arduino Mega: additional pin slots and controlling the motors
- 5-12V DC motor: running the Car that carries NEMA 17 to adjust which disk to turn
- 17HS8401 NEMA 17, 1.8A Bipolar Stepper Motor: a big stepper motor turning the disks into right positions
- GT2 Timing Pulley 36 Teeth 5mm Bore: A part that connects the NEMA 17 and the turning gear on the main shaft
- GT2 Closed Loop Timing Belt Rubber: the belt track that synchronizes the timing pulley
- Snappy Wheel: the wheel for the Car
- MicroSD Card Adapter: read text file or image with text from
- 2-legged buttons: notifying the program whether to proceed to the next word or previous word
- 20cm cable: for connecting the stepper motors and the breadboard as the Car would be moving

- Steel wire: one 3 4mm diameter, for placing the disk and the disk turner in

5.1.2 Software

IDE

- Arduino IDE

Programming Languages

- C++ language: microcontroller coding and translating text to motor steps

Libraries and Frameworks

- Arduino Framework
- Accelstepper library: controlling more than one stepper motor
- Serial library: communication between MicroSD Card and Arduino Mega

Miscellaneous

- tinkercad.com: modeling 3D shapes
- Flashprint: change the format of the model to be able to 3D print
- Fritzing: wiring and schematic design

5.2 Hardware Mock-up Illustrations

5.2.1 Prototype 1

This section explains the prototype's structure which is inspired by Figure 2.1

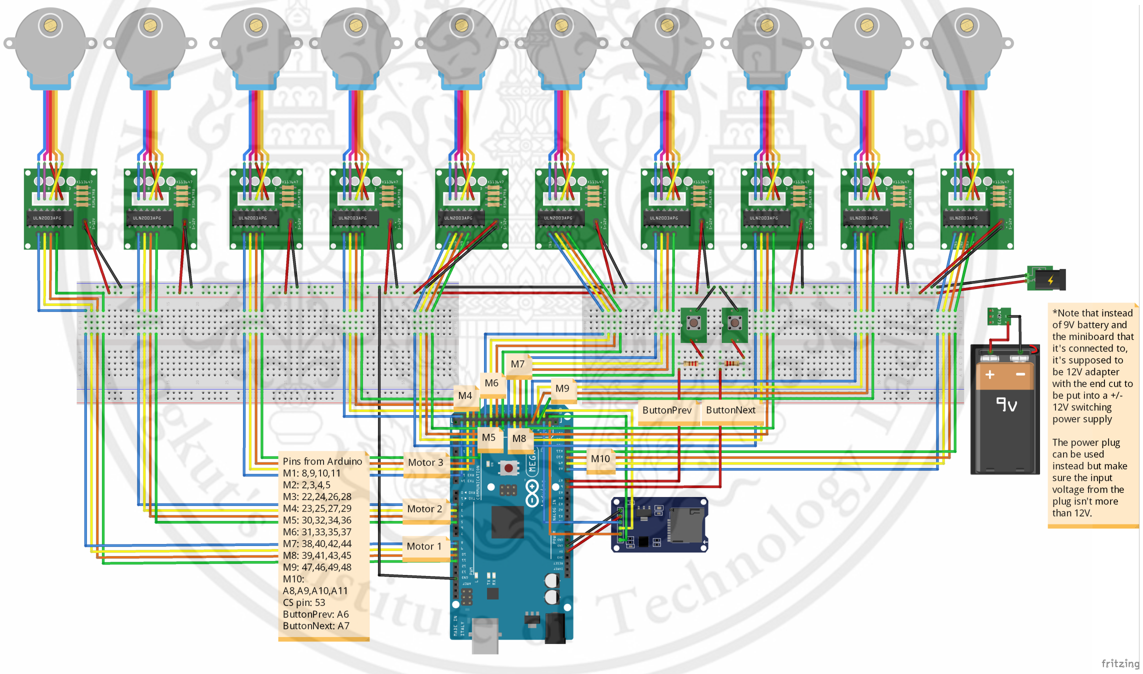
Features

- Display a five character word via text file or picture that contains English text.
- Navigate through the text with *Next* and *Previous* buttons.

Wiring

There are 10 12V 28BYJ-48 stepper motors, an Arduino Mega, and 12V input. Each stepper motors uses up 4 pins from Arduino Mega and they require 12V power input. Arduino receive the text via connection from laptop via USB port and Micro SD card.

Figure 5.1: Hardware Connection



5.2.2 Prototype 2 (Current)

This section explains the prototype's structure which is updated to be more practical

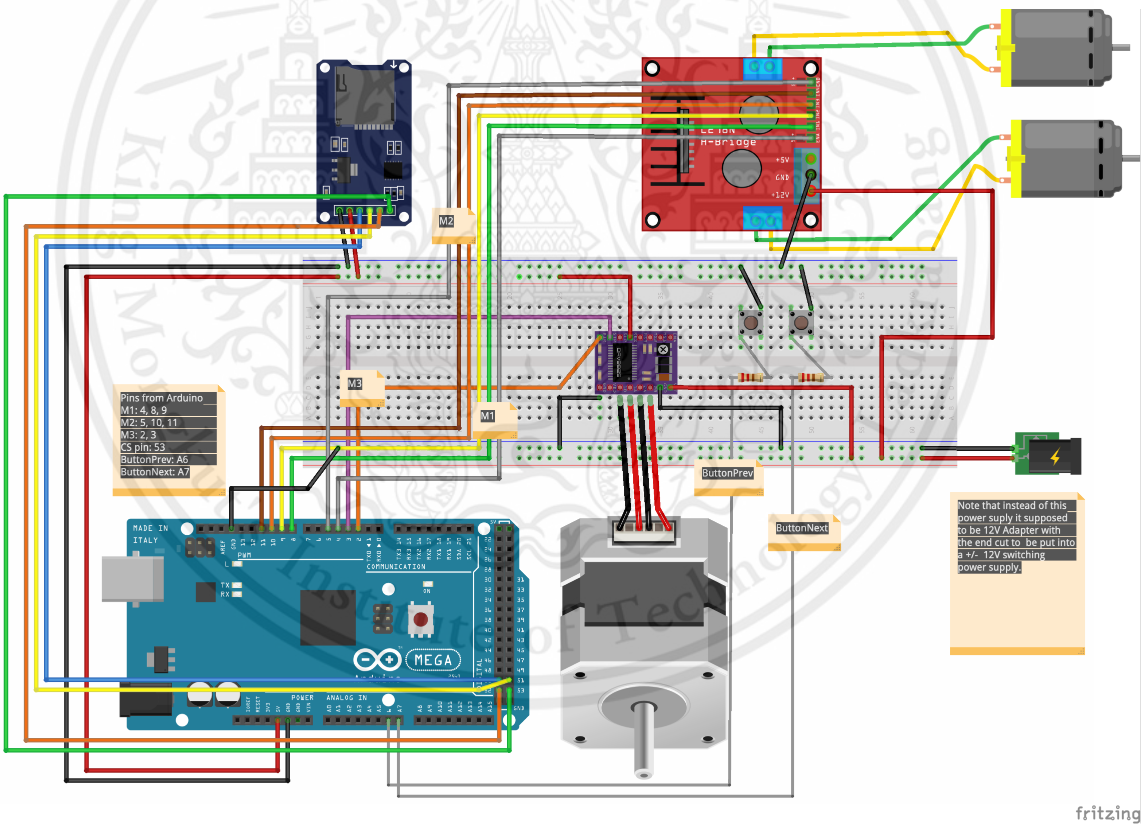
Features

- Display a five character word via text file or picture that contains English text.
- Navigate through the text with *Next* and *Previous* buttons.

Wiring

There are two DC motors capable of rotating forwards and backwards to control the Car in order to select which disk to rotate. There will be a single bipolar NEMA 17 controlling the disks rotation. The former requires 6 pin slots including the 12V voltage and ground and latter requires 4 pin slots. The Arduino Mega and the breadboard may stick on the Car or on the ground.

Figure 5.2: Hardware Connection



3D Printed Parts

Each character has two Disks(left and right) that combines together to become a single character. Each Disk's dimensions are 45*45*4mm excluding the braille character bumps which is around 2*3*2mm. It also has a center hole of 5mm and eight square holes which has the size of 6.8mm.

The Disk turner has similar properties as a Disk but it included an outer ring and holes that connect to the center. The larger hole in the middle is for the gear that will be tied through the side holes to make them a single piece.

The rods that connect with the Disk turner which are used to insert and turn the Disks has the dimension of 5.5*5.5*(30mm or 90mm) excluding the extrusion which is 5mm in length and 2.7mm in diameters.

The picture reference is at Fig.5.5.

Car

This object has two wheels which are used to transport a NEMA 17 back and forth for controlling which Disk to turn. Two DC motors will be attached to a platform and the wheels. Two to three plates would be put perpendicularly to the platform in order to move together with the Disk turner.

Models

Figure 5.3: The First Iteration Model

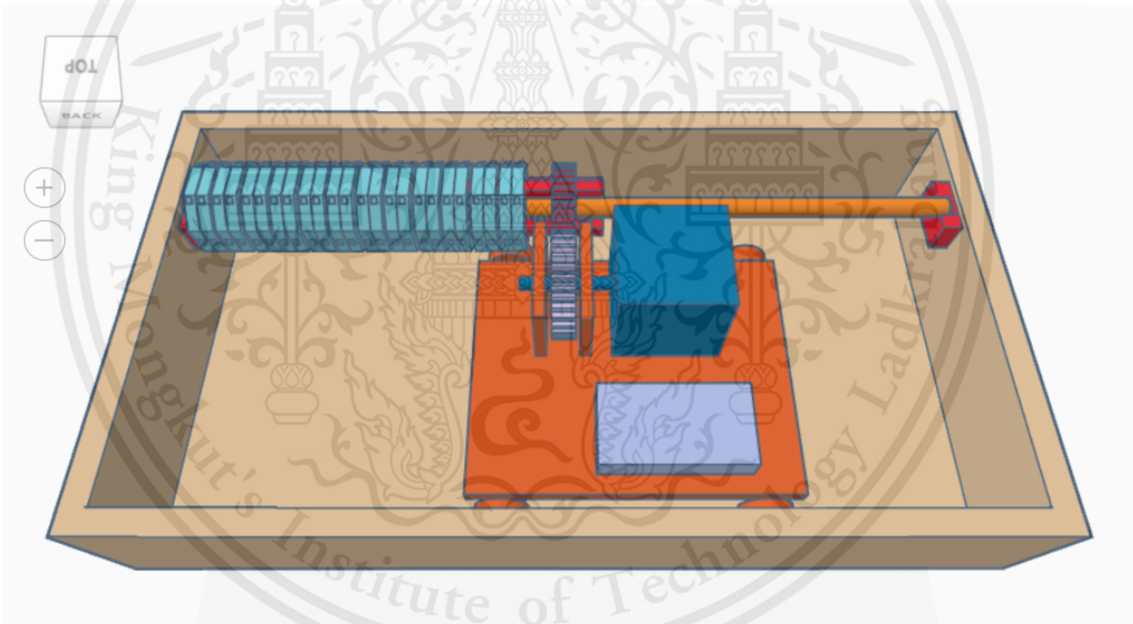


Figure 5.4: The Current Iteration Model

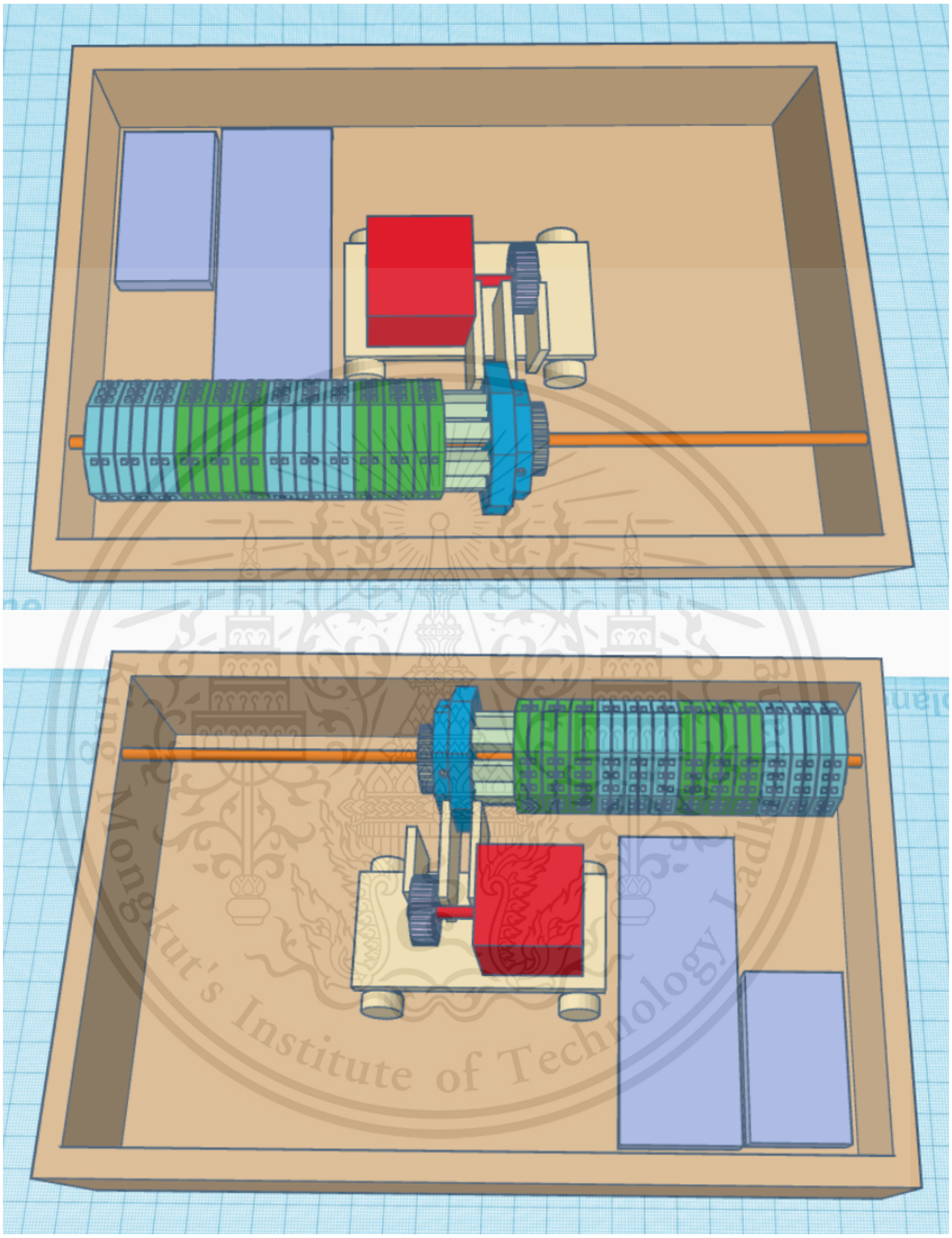
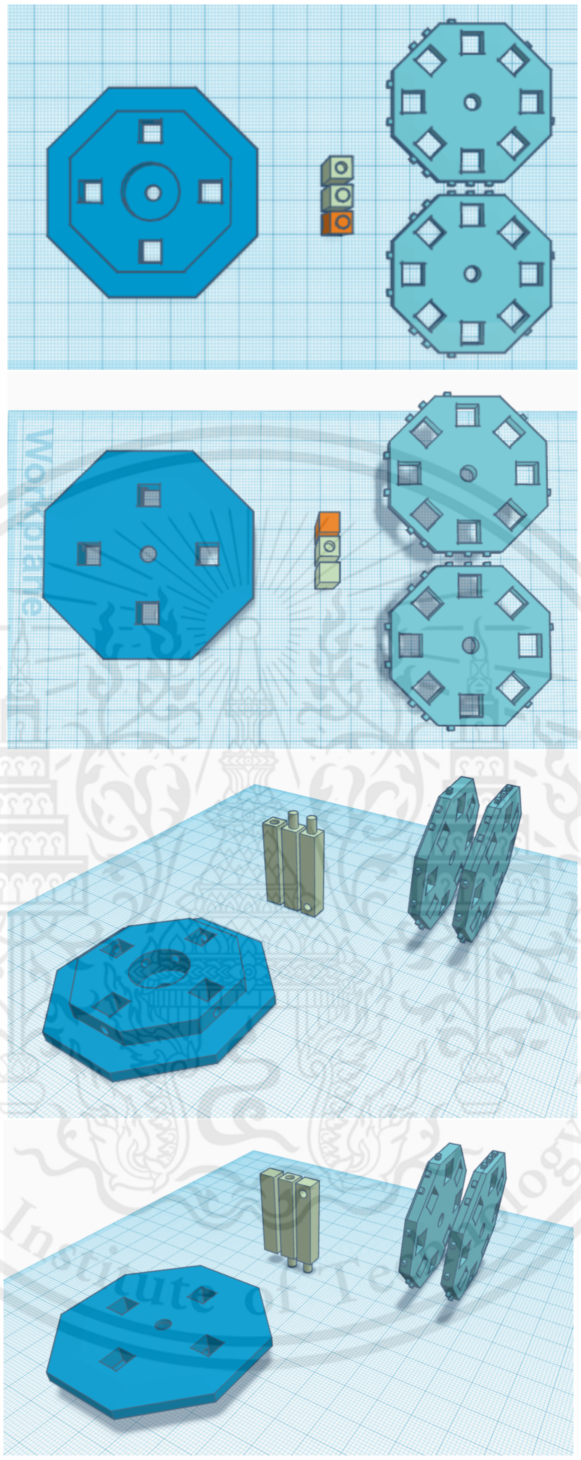


Figure 5.5: Left: Disk turner, Middle: rods, Right: Disks



Chapter 6

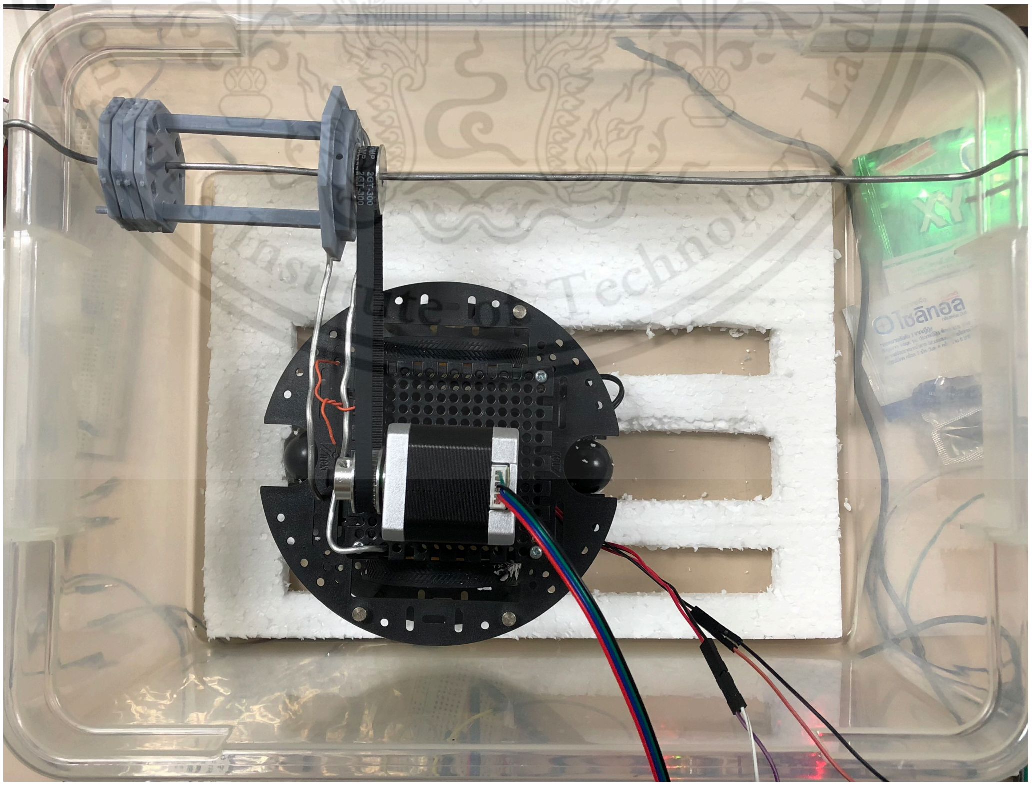
Results

We are unable to test with a blind person who is capable of reading International Standard Braille Alphabet which is in English due to COVID-19 concerns.

The disks are able to turn accurately depending on the pattern that is being ordered but we encountered the problem of the rods unable to remove itself from the disks individually sometimes which makes the disks cling to the rods whenever the car moves. There is also the problem of when the current disk turns after the previous disk have detached from the rod, the turning rod turns the disk that is previously detached slightly due to the turning disk is in contact with the current disk.

The car is able to move at a correct distance along with the disk turner from disk to disk.

Figure 6.1: Constructed Model



The figure 6.1 is our model that we made. The user is at the side where the disk are.

The cost of the project when considering only the hardware we need(excluding the 3D prints):

Figure 6.2: Hardware costs

Item	Quantity	Price(฿)
Arduino MEGA	1	420
NEMA 17 Stepper Motor	1	425
Smart car	1	265
Pulley Gear	2	105
Mini Dual +/- 12V Switching	1	290
Mini Push Button Switch	2	25
SD Card Module	1	65
H Bridge Dual Motor Controller Module	1	55
Breadboard	1	195
Female to Male wires	1	60
Male to Male wires	1	60
Total Cost		2,095 ฿

The cost of 3d printing from on our experience with a 3D printer shop, the cost of the print uses the following formula: total cost of the 3D prints (probably the type of printer and type of the filament to use) * hours to print + transporting fee. In our case, we selected SLA type 3D printer so the 3D prints with 20 disks should cost around 1500 baht times and the number of hours to print is estimated as 45 hours.

Chapter 7

Conclusion

The progress of the project is greatly hampered by COVID-19 on our work of the project, however, we are done with some parts of the project that concerns the basic functions.

The project goals has theoretically been achieved which are: cost, customization, and replicability. Excluding the 3D prints, other parts of the hardware are rather cost-wise acceptable. It is imperatively better if the user/customer to buy a 3D printer themselves than ordering a shop to print for them but if it is not possible, then they will have to order the shop to print for them. The customization here meant that the project's 3D prints can be changed to a person's preference by it's size and shape and it is possible to add more modules into the project. Replicability in this project meant that the project should be relatively easy to produce as the source of the hardware are not hard to find and reasonably cheap.

The project at its current state could be used as a teaching tool to the blind people. It cannot be carried around like a refreshable Braille display that is sold in the market due to its weight and size. This project is a worked on the base that this is a proof-of-concept concept that we can make a relatively cheaper refreshable Braille display than what is sold in the market.

The project can be expanded further into:

- Customizing and ironing more details on the project itself per person or group in different areas and languages that support Braille.
- Add more modules such as navigation on the monitor and speech-to-text.
- Make a mobile app for sending text or picture that contains text to the project's program.

References

[1] URL: https://www.fileformat.info/info/unicode/char/2895/fontsupport.htm.

[2] URL: https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all.

[3] *Arduino Mega 2560 with USB Cable*. URL: https://www.indiamart.com/proddetail/arduino-mega-2560-with-usb-cable-14768163197.html.

[4] *Basics of the SPI Communication Protocol*. 2018. URL: https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/.

[5] *Braille*. URL: https://www.britannica.com/topic/Braille-writing-system.

[6] *Braille*. 2012. URL: https://teachvisimpstudents.wordpress.com/blind-students/braille/.

[7] *Braille Cell Dimensions*. 2008. URL: http://www.tiresias.org/research/reports/braille_cell.htm.

[8] *Braillebook*. URL: http://www.ecs.umass.edu/ece/sdp/sdp16/team15/sdp16.my-free.website/index.html.

[9] *DC Motors and Stepper Motors used as Actuators*. 2019. URL: https://www.electronics-tutorials.ws/io/io_7.html.

[10] Dejan. *How Brushless Motor and ESC Work*. 2019. URL: https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/.

[11] Mat Dirjish. *What's The Difference Between Brush DC And Brushless DC Motors?* 2012. URL: https://www.electronicdesign.com/technologies/electromechanical/article/21796048/whats-the-difference-between-brush-dc-and-brushless-dc-motors.

[12] Wei Gu et al. "Computerized microfluidic cell culture using elastomeric channels and Braille displays". In: *Proceedings of the National Academy of Sciences of the United States of America* 101 (Nov. 2004), pp. 15861–6. DOI: 10.1073/pnas.0404353101.

[13] *Learn the Fundamentals of Mechatronics!* URL: https://www.newbiehack.com/MicrocontrollersAlternativePowerSources.aspx.

[14] Qing Li and C. Yao. *Real-Time Concepts for Embedded Systems*. 2003.

[15] *Microprocessor - Overview*. URL: https://www.tutorialspoint.com/microprocessor/microprocessor_overview.htm.

[16] Shouh Yann Mo. *Refreshable Display Costs and Functionalities*. URL: https://docs.google.com/spreadsheets/d/1xZzt9subBmNKrhm2yo7oE1e_EaxBAsp6YtJkKRErnMY/edit?usp=sharing.

[17] Tinashe Harry Mubvuma and Lynne Molter. *Alternative Actuators for Braille Devices*.

[18] Majenko MarkD Mythics Winter Greenonline. *Why use a motor driver board?* 2016. URL: https://arduino.stackexchange.com/questions/32105/why-use-a-motor-driver-board.

[19] *Orbit Reader 20 – Braille Display, Book Reader and Note-taker. Includes an SD Card, Charger and a USB cable*. URL: https://www.orbitresearch.com/product/orbit-reader-20/.

[20] *Piezo Actuators: Types and Applications*. 2016. URL: https://www.americanpiezo.com/piezo-theory/actuators.html.

[21] Elizabeth J. Pyatt. *Braille Script*. URL: http://www.personal.psu.edu/ejp10/symbolcodes/bylanguage/braille.html.

[22] *Refreshable Braille Displays*. 2020. URL: https://www.afb.org/node/16207/refreshable-braille-displays.

[23] Rich. *Braille Display using Arduino*. 2015. URL: https://www.youtube.com/watch?v=e5dcvkylgz4.

[24] Rich. *braillebook*. 2016. URL: https://www.youtube.com/watch?v=BM0kxXhXlFY.

[25] Alexander Russomanno et al. "Refreshing Refreshable Braille Displays". In: *IEEE transactions on haptics* 8 (Apr. 2015). DOI: 10.1109/TOH.2015.2423492.

[26] *ScreenReader: Dolphin Computer Access*. 2020. URL: https://yourdolphin.com/en-gb/products/individuals/screen-reader.

[27] *SOV Braille*. URL: https://www.f0nt.com/release/sov-braille/.

[28] *The Braille Alphabet*. URL: https://www.pharmabraille.com/pharmaceutical-braille/the-braille-alphabet/.

[29] *The Braille Literacy Crisis in America*. 2009. URL: https://www.nfb.org/images/nfb/documents/pdf/braille_literacy_report_web.pdf.

[30] *The History of Braille [Your Braille Resource]*. 2019. URL: https://brailleworks.com/braille-resources/history-of-braille/.

[31] *What is an Embedded System? Definition and FAQs*. URL: https://www.omnisci.com/technical-glossary/embedded-systems.

[32] *What is Refreshable Braille Anyway?* 2019. URL: https://wcblind.org/2018/02/what-is-refreshable-braille-anyway/.

[33] Damian Satya Wibowo, Dan Suwanseree, and Sitinut Waisara. "Universal IoT hardware platform for sensor data collection". PhD thesis. 2019.

[34] Tracy V. Wilson. *How Braille Works*. 2007. URL: https://people.howstuffworks.com/braille.htm.