Text to Braille translator

(Dotykowy translator tekstu do Braille'a)

Agata Pokorska

Praca inżynierska

Promotor: dr Marek Materzok

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

5 luty 2026

Abstract

In this paper, I introduce my idea for a refreshable Braille translator (display) device and an application that together enable reader to translate text from a photo into tactile Braille output. I also review currently available solutions, highlighting their pros and cons, and compare them with my own idea. In the end I present a detailed analysis of how it works and costs involved.

W tej pracy przedstawiam opis urządzenia i aplikacji, które w połączeniu umożliwiają zamianę dowolnego tekstu pozyskanego z obrazu na wyczuwalną dotykiem reprezentację kolejnych znaków w alfebecie Braille'a. Zapoznaję czytelnika z dostępnymi obecnie rozwiązaniami, ich zaletami i wadami. Porównuję z nimi swój pomysł i prezentuję szczegółową analizę rozwiązania i kosztów z nim związanych.

Contents

1 Introduction

1.1 Subject of the work

The subject of my work is the *translator*, a device responsible for translating a received text into Braille and displaying its tactile representation, as well as the *TextToBraille* application, which enables interactive use of the translator.

Designed for Android smartphones, the app takes a photo of a specific text, which is then sent to the translator. Based on this photo, a Braille representation of the text is generated by bumping up specific pins. By pin, I mean one of the six points, either convex, or flat. The translator and the app allow to read any text captured with a smartphone camera.

1.2 Idea behind this project

1.2.1 What is Braille

Braille is a variant of the alphabet that allows blind people to read text using their sense of touch. A single Braille character consists of six pins, arranged in three rows of two columns. Each pin is either flat, or convex. There are sixty-four possible pin arrangement combinations. They represent the letters a-z, punctuation marks, numerals, capital letters, parentheses, and so on.

1.2.2 Available solutions and problems related to them

The most common solution for blind people remains audiobooks, or other solutions based on *text-to-speech* technology; however this does not seem to be an optimal approach. People absorb information in many different ways. Some prefer visually, others auditory, and still others need to move, or feel an object for the appropriate neural connections to be established during the learning process. Study [1] suggests that blind people have much higher tactile sensitivity than sighted people. Therefore, it is important to ensure that blind people also have diverse opportunities to pick up information.

In addition to audiobooks, there are also books printed in Braille using a dedicated mechanism [2], which was created in 1892. However, the cost of such books is outrageous (up to four times higher than their traditionally printed equivalents).

A catalog of popular books printed in Braille can be found at [3].

The problem with both solutions is that not all interesting titles are available as audiobooks/books printed in Braille (like textbooks). I should also note that purchasing individual audiobooks is related to a high cost. There are apps offering subscription models (like Audible), but these have, firstly, a very limited offer, and secondly, the resource is hosted on the app's server, so even after purchase, it never truly belongs to the buyer.

An alternative to these solutions are electronic Braille displays, which also have their drawback: they are extremely expensive.

I highlight these as following factors in this regard:

- A module for displaying a single Braille character is not a typical electronic component.
- Due to the complexity of the structure, the cost of displaying a single character is high.
- Demand for this kind of tech products is relatively low.

Despite that, electronic Braille displays have a significant advantage – they enable to display **any** text, which we already have, in a tactile form without additional costs.

1.2.3 My motivation

I also want to mention my own experience that remains a driving force behind this project. While in the library in my hometown, I asked about collection for blind people. Unfortunately only one book [4] was in the library's offer. I noticed that many book's pins were simply worn out as I was unable to read it. This copy was four years old and yet, it was so broken to noticeably make reading harder. Considering traditionally printed book's degradation pace, it does not seem right about Braille printed books.

My goal was to implement a simplier, budget-friendly yet working tactile Braille display with usage of common, cheap electronic components.

2 Review and analysis of available solutions

Solutions are devided into three categories, based on technology that was used: electromagnetic, piezoelectric, and other (which differ in their approach from classic devices from the first two categories).

2.1 Electromagnetic-based solutions

An example device using electromagnetism looks like this: [5]. In simple words, electromagnet attracts other magnet, or remain idle depending on electric charge presence. In the mentioned example, magnet was placed inside a rotating module, which raises/lower one of the six Braille character pins. It works similar to E-ink display [6] (like Kindle).

Advantages and drawbacks of using electromagnetism

The above solution looks quite promising. Among the advantages, I can mention that it is open-source and that the production cost is very low, reportedly around 1 USD. However, I see a few drawbacks.

Here is a quote directly from the logs of the project mentioned above [7]: *"Having access to a printer directly made all the difference when iterating for 20-30 μm tolerances. At these sizes, you can't really trust the CAD software anymore, and every assembly has to be empirically tested by printing it out."* Author is refering here to difficulties in developing a single working module, because of its exceptionally small size. He claimes that every component has to be assembled and tested for functionality. Another disadvantage of this approach is rapid heating of the electromagnet itself [8]. It requires another heat dissipation parts, which is complicating already extensive construction. Every friction inside construction is causing slow abrassion of the structure. With components this small, any slight difference in structure, dust, or other types of contamination may cause it to malfunction.

2.2 Piezoelectric based solutions

Piezoelectricity is a property of certain materials (i.e quartz) that generates electrical energy when a force is applied to the material. Braille displays are using the exact opposite principle – that is by applying appropriate current it causes material to deform in more, or less controlled manner. There are available piezoelectric actuators [9] like [10], which price has gone up by 80 USD over the last two years. It makes a single character display cost significantly high. In order to display a single one, we need six of these actuators. In video [11], author disassembles the popular commercial Braille Lite 2000 display [12].

Advantages and drawbacks of piezoelectric usage

Main drawback of piezoelectric usage in refreshable Braille displays is definitely its cost. Relatively cheaper option is [13] display, which is sold for 800 USD. Another disadvantage is the need for propoer maintenance and cleaning to prevent dirt from getting into the precision mechanism.

Advantages are definitely its neat design and relatively small size. Also, there are plenty of options on the market.

2.3 Other ideas worth mentioning

Very compact solution is BrailleRing [14], which also enables to read whole lines of text, but it is done in a completely different way. BrailleRing device, as its name suggest, is ring-shaped and has twenty single-character modules. The module looks and is rotating just like Rubic's cube but with three layers and two columns instead of classic 3x3. Six tactile pins are placed on one side of a cube. There are two pins on each cube's layer. By rotating a layer, the layer's pin combination changes to one of these: both pins are flat, only one is convex, or both are convex. The idea is very innovative not only because of its ring shape, but the fact that it minimizes number of needed piezoelectric actuators. It reduces from six for one character, to only nine actuators for the whole construction. Actuators are divided three-by-three, one actuator for each layer, and each layer needs to be rotated maximum of three times in order to change into other Braille character.

Another idea comes from the popular site with open-source 3D projects, Thingiverse [15]. This project uses motors to operate position of a long rack-and-pinion mechanism, and that position decides about one from eight possible combinations of pins from one column. The idea differs from the others by the fact that whole movement is based only on mechanical parts. Potential disadvantage, that I see is 3D printing is usually not precise enough, thus makes it harder in developing. Fully mechanical movement, means that it is susceptible for increased friction, and damage resulting from it.

3 Comparison of my solution with other ideas

NOTE: Servo - a common name for a servomechanism, in this case one that moves a T-bar by a degree, which determines state of a pin.

3.1 Translator

The device is made of just a few components:

- ESP32 WiFi+BT 4.2 WROOM-32,
- three 3.3V microservos and three 5V microservos (for change of pin's state),
- six 40mm x 6mm cylinders 3D printed (these are the actual pins),
- double-sided PCB 5cm x 7cm (all electronics is managed here),
- three logic voltage level converters 2CH 3.3V/5V (ESP works at 3.3V but some servos need more power),
- USB to USB-C wire (needed for charging the battery),
- Li-ion battery 3.6V; 5000mAh; 25A,
- battery box,
- capacitor $1000\mu F$,
- battery charging controller,
- step-up converter set manually at 5V.

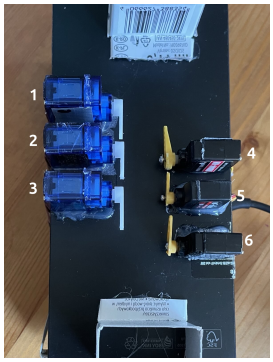

Figure 3.1: Six servos with its corresponding pin numbers

User to app communication flow

- Text is now being chosen. For best results, high-contrast is recommended (like book).
- After clicking the "Photo" button, the camera app is being launched.
- AI model [16] extracts the text from uploaded photo.
- Text is displayed in app's dedicated area.
- After clicking "Send" button, the text is now sent to the translator.
- The text can be overriden by repeating the steps above.

App to translator communication flow

- Bluetooth channel connecting ESP32 with an active app is created.
- The recognised text is sent one-by-one from the app to an ESP.
- ESP causes certain servos to move, making the output tactile.
- Servos' state is held for a second, then all pins go down.
- After a character is displayed, a 300 ms pause is made.

ESP32 software is implemented in C++. Channel of communication between the app and translator is done using Bluetooth 4.2. Implementation details are available in my repository dedicated for developing this project [17].

3.2 Tools used to make software for the translator

- VS Code – a very convenient code editor, which I chose especially because of cross-platform support and PlatformIO plugin availability.
- PlatformIO – IDE for compiling and uploading ESP32 source code.
- C++ – due to its compatibility with Arduino interface for programming microcontrollers.
- `Arduino.h` – well-supported library for programming ESP.
- `BluetoothSerial.h` – used for creating connection between the app and ESP.
- `ESP32Servo.h` – it gives simple interface for controlling servos by ESP32.

3.3 Tools used to create the app

- Android Studio – best known to me IDE for creating Android apps. It also has a built-in Android emulator.
- Kotlin – recommended language made for implementing Android apps, supported by Android Studio.
- Google ML Kit Text recognition v2 – used for text recognition, it is free and can work offline.

3.4 Pros and cons of my solution

Considering the electronic parts and tools I used, as well as the servos construction photo above, I will now present pros:

- Translator is fully mobile when powered by the battery.
- Design simplicity – one servo per pin.
- Usage of pre-assembled, commercially available cheap electronic parts.
- Relatively low cost of production.
- 3D printed cyllinder is light, so servo uses low power while boosting pin up.
- All pins are syncronised simultaneously.
- Due to its large size, it is dust and dirt resistant.

Here are the crucial cons:

- Translator displays one character at a time.
- Character is displayed statically, which as shown in [18] is not an optimal way to read Braille.
- Translator's device is quite large as for one character display.

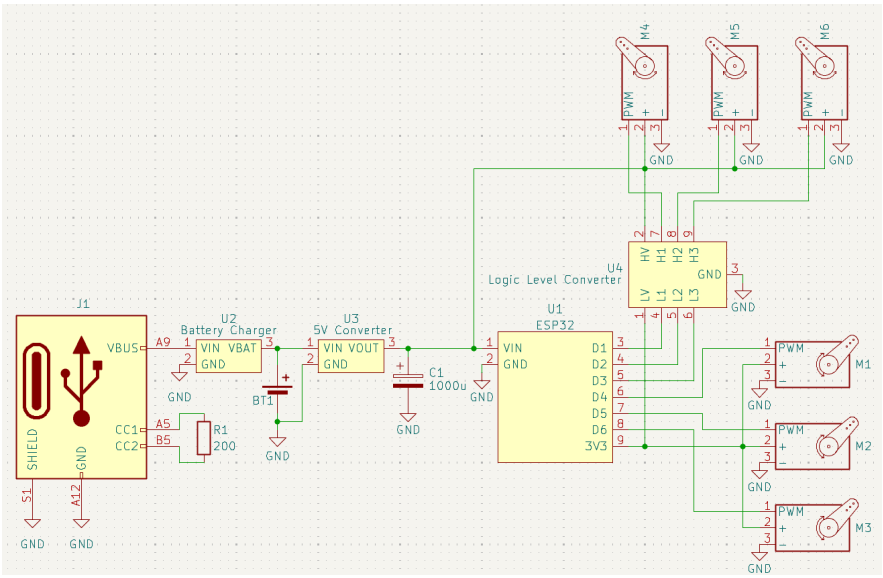

Figure 3.2: Simplified electronic circuit scheme

4 User manual

4.1 How to use the translator

First, it is required to connect traslator to power source either by connecting USB, or inserting the battery into the box. After that translator is ready to connect to the active TextToBraille app.

To ensure proper work of the device, do not use too much force while touching tactile output and do not disconnect it from the power source.

4.2 How to install the app

App can be installed by downloading *texttobraille.apk* from my repository [17] and by clicking install. To properly install the .apk file on your Android device, you need to *enable installing of apps from external sources.* You can find the manual how to enable it here [19].

4.3 How to use the app

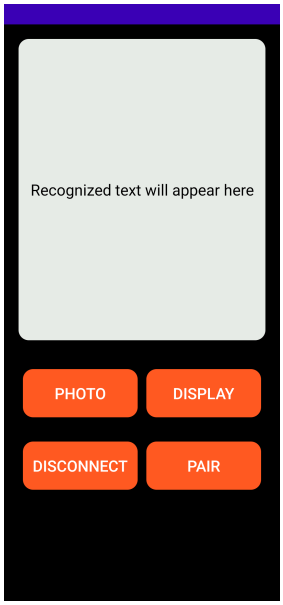

Figure 4.1: TextToBraille app interface

After opening the application *TextToBraille*, there are visible five elements of an interface, with functionalities as below:

Text field – Here the recognised text is shown.

Pair – Request a connection between app and the translator. Message about connection status will be shown as a popup.

Disconnect – End an active connection between both devices.

Photo – Launch a camera app to take photo of a text.

Display – Recognised text is sent and displayed by the translator. If no text is recognised, appropriate popup will be shown.

4.4 Use cases

4.4.1 Taking a photo while using the app

Actor: A blind person with an installed TextToBraille app on a working Android smartphone.

Use case: Person clicks on the *Photo* button, which launches a built-in camera app. Next they take a photo of a chosen text. They accept this photo by clicking *accept*. Text is being extracted from the photo, which will be displayed in the preview.

4.4.2 Sending text to the translator

Actor: A blind person who already has AI-recognised text in the app's text field.

Use case: Person clicks on the *Pair* button, which initiates connection with translator. Person clicks *Send* button and the tactile output is displayed one-by-one on the translator device.

4.4.3 Ending connection with the translator

Actor: A blind person with an active TextToBraille app and succesfully connected translator.

Use case: Person clicks on the *Disconnect* button, which causes an immidiate end of connection. Scanned text will not be displayed, until the connection is re-established.

5 Summary

5.1 Estimated prototype production cost

In the calculation below, I focus on price of a single component unit and multiply it by the used quantity. The given price is the lowest among those found on popular web sellers such as Botland, Allegro, Kamami, Nettigo, and TME. Delivery cost was not included. Most of these parts are already pre-assembled.

- ESP32 WiFi+BT 4.2 WROOM-32 – **22.49 PLN**.
- Three 3.3V microservos – **14.90 PLN x 3**.
- Three 5V microservos – **17.90 PLN x 3**.
- Six 40mm x 6mm cylinders– **0.70 PLN**.
- Double-sided PCB – **3 PLN**.
- Three logic voltage level converters – **2.50 PLN x 3**.
- USB-C wire– **3.47 PLN**.
- Li-ion battery – **13.23 PLN**.
- Battery box – **2.89 PLN**.
- Capacitor $1000\mu F$ – **0.48 PLN**.
- Battery charge controller – **2.28 PLN**.
- Step-up converted – **3.89 PLN**.

In summary: **158.33 PLN** so it's significantly less expensive that the previously described commercial Braille display for 800 USD.

5.2 Final effect

As a final result of my work, I have a fully working, mobile device that enables to translate any text into tactile output. I also have a dedicated TextToBraille app, which is fullfilling its purpose. Both translator and sole pin boosting mechanism were made the simpliest way possible, from cheap, easily accessible electronic components. All source code was made using only free of a charge developing platforms. Considering cost of parts and materials used for making the device, the final price is approximately 160 PLN.

5.3 Future improvement ideas

- Creating TextToBraille app equivalent for IOS.
- Device size reduction by advancing the construction.
- Changing Google AI model to an open-source one.
- Redesign device with levers to reduce the gaps between pins.

Bibliography

[1] Daniel Goldreich i Ingrid M. Kanics. "Tactile Acuity is Enhanced in Blindness". W: *Journal of Neuroscience* 23.8 (2003), s. 3439–3445. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.23-08-03439.2003. eprint: https://www.jneurosci.org/content/23/8/3439.full.pdf. URL: https://www.jneurosci.org/content/23/8/3439.

[2] Wikipedia contributors. *Perkins Brailler — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-January-2026]. 2025. URL: https://en.wikipedia.org/w/index.php?title=Perkins_Brailler&oldid=1321563538.

[3] *Catalog of bestselling braille-printed books*. http://www.braillebookstore.com/Adult-Bestsellers.

[4] *Czarna książka kolorów - Menena Cottin*. https://wydawnictwo-widnokrag.pl/ksiazki/czarna_ksiazka_kolorow/. 2012.

[5] *Refreshable Braille Display with electromagnets usage*. https://hackaday.io/project/191181-electromechanical-refreshable-braille-module. 2023.

[6] *Jak działa e-papier i co warto o nim wiedzieć*. https://forbot.pl/blog/e-papier-jak-dziala-i-co-warto-o-nim-wiedziec-id53934.

[7] *12th time is the charm - A project log for Electromechanical Refreshable Braille Module*. https://hackaday.io/project/191181-electromechanical-refreshable-braille-module/log/224233-12th-time-is-the-charm. 2023.

[8] *Why do electromagnets generate overheating during using - causes and solutions*. https://www.drsolenoid.com/news/why-do-electromagnets-generate-over-heating-during-using-causes-and-solutions/. 2024.

[9] Frédéric Giraud i Christophe Giraud-Audine. "Chapter One - Introduction". W: *Piezoelectric Actuators: Vector Control Method*. Red. Frédéric Giraud i Christophe Giraud-Audine. Butterworth-Heinemann, 2019, s. 1–42. ISBN: 978-0-12-814186-1. DOI: https://doi.org/10.1016/B978-0-12-814186-1.00005-3. URL: https://www.sciencedirect.com/science/article/pii/B9780128141861000053.

[10] *Piezoelectric bending transducer expensive solution*. https://piezo.com/products/piezoelectric-bending-transducer-s234-h5fr-1803xb.

[11] *Braillelite2000 device disassembly demonstration*. https://www.youtube.com/watch?v=guZ2rFT_FKQ. 2020.

[12] *BrailleLite2000*. https://www.eyeway.org.in/?q=braille-lite-2000.

[13] *Cheap commercial refreshable braille display*. https://www.orbitresearch.com/product/orbit-reader-20/.

[14] Michael Treml i in. W: *Current Directions in Biomedical Engineering* 6.2 (2020), s. 20202004. DOI: doi:10.1515/cdbme-2020-2004. URL: https://doi.org/10.1515/cdbme-2020-2004.

[15] *Fully mechanical concept for refreshable braille display module*. https://www.thingiverse.com/thing:90144. 2013.

[16] *Google ML Kit Text recognition v2*. https://developers.google.com/ml-kit/vision/text-recognition/v2.

[17] *My TextToBraille project repository*. https://github.com/gagata7/TextToBraille. 2025.

[18] Joerg Fricke i Helmut Baehring. "Displaying laterally moving tactile information". W: *Computers for Handicapped Persons*. Red. Wolfgang L. Zagler, Geoffrey Busby i Roland R. Wagner. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, s. 461–468. ISBN: 978-3-540-48989-4.

[19] *Guide to Installing APK Files on Android Devices*. https://www.lifewire.com/install-apk-on-android-4177185. 2025.