

# Dotykowy translator tekstu do Braille'a

(Text to tactile Braille output translator)

Agata Pokorska

Praca inżynierska

**Promotor:** dr Marek Materzok

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Instytut Informatyki

5 luty 2026



## **Streszczenie**

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 alfabecie 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.

---

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.



# Spis treści

<b>1. Wprowadzenie</b>	<b>7</b>
1.1. Przedmiot pracy . . . . .	7
1.2. Skąd taki pomysł . . . . .	7
<b>2. Przegląd i analiza dostępnych rozwiązań</b>	<b>9</b>
2.1. Rozwiązanie wykorzystujące zjawisko elektromagnetyzmu . . . . .	9
2.2. Rozwiązania wykorzystujące zjawisko piezoelektryczności . . . . .	10
2.3. Inne dostępne rozwiązania warte uwagi . . . . .	11
<b>3. Porównanie zastosowanych w projekcie rozwiązań</b>	<b>13</b>
3.1. Translator . . . . .	13
3.2. Aplikacja . . . . .	14
3.3. Zalety i wady mojego rozwiązania . . . . .	15
<b>4. Instrukcja dla użytkowników</b>	<b>17</b>
4.1. Do czego służy aplikacja . . . . .	17
4.2. Jak zainstalować aplikację . . . . .	17
4.3. Jak używać aplikacji . . . . .	17
4.4. Jak używać translatora . . . . .	17
<b>5. Użyte narzędzia</b>	<b>19</b>
5.1. Narzędzia użyte do stworzenia oprogramowania translatora . . . . .	19
5.2. Narzędzia użyte do stworzenia aplikacji . . . . .	19
<b>6. Przypadki użycia</b>	<b>21</b>
6.1. Wykonanie zdjęcia za pomocą aplikacji . . . . .	21
6.2. Przesłanie tekstu do translatora . . . . .	21
6.3. Zakończenie połączenia z translatorem . . . . .	21
<b>7. Dalsze kroki rozwoju projektu</b>	<b>23</b>

7.1. Moje wnioski . . . . .	23
7.2. Szacowany koszt wytworzenia prototypu . . . . .	23
7.3. Potencjalne usprawnienia do projektu . . . . .	24
<b>Bibliografia</b>	<b>25</b>

# 1. Wprowadzenie

**Definicja 1** *Aplikacja: oprogramowanie umożliwiające użytkownikowi interaktywne korzystanie z translatora.*

**Definicja 2** *Translator: urządzenie odpowiadające za translację do Braille’a otrzymanego znaku tekstu, oraz wyświetlenie jego Braille’owskiej reprezentacji.*

**Definicja 3** *Pin: jeden z sześciu punktów znaku Braille’a. Może być w stanie płaskim (schowanym) lub wypukłym (wysuniętym).*

## 1.1. Przedmiot pracy

Przedmiotem mojej pracy są *translator*, oraz aplikacja *TextToBraille* umożliwiająca korzystanie z urządzenia.

Aplikacja zaprojektowana bazowo na smartfony z systemem Android, służy do wykonania zdjęcia konkretnego tekstu, który następnie jest przesyłany do translatora i na jego podstawie, generowane jest odwzorowanie wyczuwalne dotykiem znaków tegoż tekstu w alfabecie Braille’a.

Translator wraz z aplikacją dają osobom niewidomym możliwość odczytu *dowolnego* tekstu, który da się uchwycić aparatem smartfona.

## 1.2. Skąd taki pomysł

Problem z dostępnymi obecnie na rynku wyświetlaczami Braille’a jest taki, że są *niezwykle drogie*. Jako istotne dla tego faktu czynniki wyróżniam:

- Moduł do wyświetlania pojedynczego znaku Braille’a nie jest typowym komponentem elektronicznym.
- Ze względu na złożoność konstrukcji koszt wyświetlenia pojedynczego znaku jest wysoki.
- Popyt na tego typu produkty jest stosunkowo niski.

Najbardziej powszechnym rozwiązaniem dla osób niewidomych pozostają audiobooki lub inne rozwiązania wykorzystujące technologię *text-to-speech*, co jednak okazuje się średnim kompromisem. W badaniu [1] sprawdzono, czy utrata wzroku powoduje znaczące polepszenie słuchu. W skrócie, wychodzi na to, że nie. Mózg osoby niewidomej jest do pewnego stopnia neuroplastyczny, co pozwala na "dostrajanie" się do specyficznych bodźców z otoczenia, jednak nie powoduje to ogólnego polepszenia słuchu. Z kolei przeciwnie można powiedzieć o zmyśle dotyku. W tym [2] badaniu zasugerowano, że osoby niewidome wykazują się zauważalnie wyższą czułością dotykową niż osoby widzące. Ludzie przyswajają informacje na *różne sposoby*. Niektórzy preferują wzrokowo, inni słuchowo, jeszcze inni muszą się poruszać czy poczuć dany obiekt, aby zostały stworzone odpowiednie połączenia neuronalne w procesie uczenia się. Warto więc zadbać, żeby osoby niewidome też miały zróżnicowane możliwości przyswajania informacji.

Chciałabym też wspomnieć o osobistym doświadczeniu, które dalej pozostaje motorem do rozwoju tego projektu. Będąc w bibliotece w swoim rodzinnym mieście, zapytałam o zbiór pozycji dostępnych dla osób niewidomych. Na całą bibliotekę, była tylko jedna książka [3] napisana Braillem. Przyglądając się książce dłużej i z bliska, zauważyłam że pędziesiąt znaków Braille'a było zwyczajnie wytartych od użytkowania. Ta książka miała dopiero cztery lata, a już była zużyta na tyle, żeby utrudnić czytanie. Uwzględniając do tego tempo degradacji książek drukowanych tradycyjnie, nie brzmi to sprawiedliwie w stosunku do osób niewidomych.

Podsumowując – sądzę, że pomimo ogólnodostępnych narzędzi korzystających z technologii *text-to-speech*, warto dalej skupiać się na rozwoju nowych i ulepszaniu dostępnych rozwiązań konwertujących tekst do Braille'a.



## 2. Przegląd i analiza dostępnych rozwiązań

Dostępne rozwiązania podzielę na trzy główne kategorie wg. zastosowanych przez nie technologii tj. elektromagnetyczne, piezoelektryczne oraz inne, czyli takie, które różnią się swoim podejściem od klasycznych urządzeń z dwóch pierwszych kategorii.

### 2.1. Rozwiązanie wykorzystujące zjawisko elektromagnetyzmu

Przykładowe urządzenie wykorzystujące elektromagnetyzm wygląda następująco [4]. Upraszczając, elektromagnes zmienia swoją *polaryzację* w zależności od obecności ładunku elektrycznego. Dzięki tej właściwości powoduje on odpowiednio przyciąganie lub odpychanie innego magnesu. We wspomnianym wcześniej przykładzie magnes został umieszczony w środku głowicy, która płynnym ruchem obrotowym (góra/dół) powoduje podbicie/spadek jednego z sześciu pinów znaku Braille’a. Mechanizm ten możemy kojarzyć również z popularnych czytników E-ink [5] (np. Kindle)

#### Zalety i wady zastosowania elektromagnetyzmu

Powyższe rozwiązanie wygląda dość obiecująco. Z zalet, mogę wymienić otwartoźródłowość oraz to, że szacowany koszt produkcji takiego komponentu jest niski. Widzę w nim jednak kilka wad.

Oto cytaty bezpośrednio z logów wymienionego wyżej projektu [6]: *"Having access to a printer directly made all the difference when iterating for 20-30  $\mu$ m 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."* Autor odnosi się tutaj do trudności w wytworzeniu pojedynczego modułu, ze względu na jego bardzo mały rozmiar i twierdzi, że każdy komponent musi zostać złożony a następnie sprawdzony pod kątem poprawności działania. Inny problem jaki dostrzegam, to szybkie nagrzewanie się elektromagnesu [7]. Wymaga to kolejnych części do odprowadzania ciepła, co jeszcze bardziej komplikuje już i tak gęsto zwartą konstrukcję. Kolejny problem jaki widzę, związany jest z tarciem wywoływanym przez ruch głowicy, co może powodować stopniowe zużywanie się układu. Przy takich rozmiarach komponentów, wszelkie drobne różnice w strukturze, kurz, czy innego rodzaju zanieczyszczenia mogą powodować zakłócenia w prawidłowej pracy.

## 2.2. Rozwiązania wykorzystujące zjawisko piezoelektryczności

Przechodząc do następnej kategorii rozwiązań, wykorzystujących piezoelektryki, należy wspomnieć na czym to polega. Jest to właściwość konkretnych materiałów (np. kwarcu), która sprawia, że w momencie oddziaływania siły na ten materiał, powstaje energia elektryczna. Czytniki Braille’a używają dokładnie *odwrotnej zasady*, czyli poprzez zastosowania odpowiedniego prądu, materiał zaczyna odkształcać się w mniej lub bardziej kontrolowany sposób. Na rynku dostępne są tzw. siłowniki piezoelektryczne [8] jak np. [9], który nawiasem mówiąc przez ostatnie 2 lata, podrożał o ponad 80 USD. Ewentualnie bardziej budżetowa opcja [10]. Nadal jednak koszt pojedynczych komponentów jest bardzo wysoki. Do wyświetlenia *pojedynczego* znaku Braille’a, potrzebujemy aż sześć tego rodzaju siłowników. W filmie [11] autor rozkłada na części popularny komercyjny wyświetlacz Braille Lite 2000 [12].

### Zalety i wady zastosowania piezoelektryków

Główną wadą związaną z użyciem tej technologii w wyświetlaczach jest ich cena. Najtańszy wyświetlacz jaki udało mi się znaleźć [13] kosztuje zawrotne 800 USD. Inną wadą jest potrzeba prawidłowej konserwacji takiego urządzenia, ze względu na ilość otworów, w które może dostać się brud, co może nawet uszkodzić wyświetlacz.

Zaletą jest ich zgrabny wygląd i stosunkowo niewielki rozmiar. Innym argumentem działającym na korzyść czytnika, jest też fakt prostej przewagi nad audiobookami. Nawet taki dość drogi czytnik, pozostanie naszą własnością na zawsze i mamy gwarancję, że będzie działał. Jeśli chodzi o audiobooki, kupowane pojedynczo są drogie i jeśli ktoś *naprawdę* bardzo to lubi, to będzie go to słono kosztowało przy kilkudziesięciu egzemplarzach. Może nawet więcej niż koszt komercyjnego czytnika. Jeśli zaś nie interesuje nas kupowanie audiobooków, można wtedy skorzystać z oferowanych modeli subskrybcyjnych jak np. EmpikGo, Legimi czy Audible. Łatwo jednak zauważyć, że w obu tych przypadkach audiobooki tak naprawdę nigdy nie należą do nas, tylko są trzymane w chmurze aplikacji. Zauważmy też, że mimo bogatej oferty audiobooków, dalej istnieje szansa, że nie znajdziemy tam tego, czego akurat chcemy np. podręczników do nauki. Może więc jednorazowy wydatek w wysokości 800 USD nie brzmi aż taki źle, biorąc pod uwagę, że daje to możliwość odczytania dowolnego tekstu.

## 2.3. Inne dostępne rozwiązania warto uwagi

Na koniec omówię pozostałe istniejące rozwiązania, mniej lub bardziej efektywne. Warto wspomnieć iż oprócz czytników istnieją też książki fabrycznie drukowane Braillem za pomocą specjalnie stworzonego do tego mechanizmu [14], który powstał już w roku 1892. Koszt produkcji takich książek jest jednak horrendalnie wysoki (nawet czterokrotnie wyższy w porównaniu do ich drukowanych tradycyjnie odpowiedników). Katalog popularnych książek drukowanych Braillem znajduje się w [15].

Jednym z bardziej kompaktowych rozwiązań jest BrailleRing [16], który również pozwala na odczytywanie linii tekstu, jednak robi to w sposób zupełnie inny. Urządzenie to przyjmuje kształt pierścienia a wewnątrz ma wbudowane dwadzieścia modułów. Moduł składa się z sześciu pinów i służy do wyświetlenia pojedynczego znaku Braille’a. Sześć pinów w module zostało podzielone na trzy poziome grupy, każda po dwa piny (dwa górne, dwa środkowe i dwa dolne). Para pinów ma cztery możliwe stany: oba opuszczone, jeden z nich podniesiony lub oba podniesione. Każda taka kombinacja pary pinów jest odwzorowana na jednej ze ścian czworokąta, który ma możliwość obrotu o 90 stopni. Poprzez wybór odpowiedniej grupy i rotację, możemy za pomocą jednego modułu przedstawić dowolny znak Braille’a. Pomysł jest o tyle innowacyjny, że właśnie dzięki ułożeniu modułów w pierścień, w jego górnej połowie moduły najpierw są obracane do reprezentacji odpowiednich znaków, a następnie użytkownik poruszając palcem po wnętrzu dolnej połowy pierścienia, sam nadaje tempo czytania podsuwając sobie kolejne znaki linii. Taka architektura znacznie ogranicza liczbę potrzebnych siłowników do dziewięciu. Jeden siłownik jest w stanie obrócić jedną ścianę czworokąta, z czego do zmiany reprezentacji są potrzebne maksymalnie trzy zmiany.

Jeszcze inny pomysł, wywodzi się z popularnego serwisu z otwartoźródłowymi projektami do druku 3D, Thingiverse [17]. Projekt ten wykorzystuje silniki do poruszania długą zębatkową przekładnią, która obsługuje trzy z sześciu pinów (jedną kolumnę) i pozycja tejże przekładni (czyli to jak bardzo jest przesunięta w lewo lub w prawo) odwzorowuje jedną z ośmiu kombinacji tych trzech pinów. Rozwiązanie różni się od pozostałych, gdyż działa w pełni mechanicznie. Potencjalne problemy jakie tu widzę, to fakt iż drukarka 3D ma swoje ograniczenia, jeśli chodzi o precyzję w tak małej skali oraz to, że w pełni mechaniczny moduł jest podatny na nadmierne tarcie części o siebie i wynikające z tego powodu uszkodzenia.



### 3. Porównanie zastosowanych w projekcie rozwiązań

**Definicja 4** *Serwo: potoczna nazwa na serwomechanizm, w tym przypadku wprawiający w ruch śmigło sterujące wysokością położenia pinu*

#### 3.1. Translator

Na urządzenie składa się zaledwie kilka komponentów:

- ESP32 WiFi+BT 4.2 WROOM-32 (mózg układu),
- trzy mikroserwa 3.3V oraz trzy mikroserwa 5V (to podbija piny),
- sześć walców o 20mm x 6mm wydrukowanych w 3D z plastiku (to są właśnie piny, one są bezpośrednio wyczuwalne palcami),
- uniwersalna płytką PCB 5cm x 7cm dwustronna (tu jest zamocowana cała elektronika),
- trzy konwertery poziomów logicznych napięć 2CH 3.3V/5V (gdyż ESP32 działa na 3.3V a serwa potrzebują wyższego napięcia),
- kabel USB do USB-C (potrzebny do zasilania układu),
- kartonowe pudełko po Samsungu Galaxy S20,
- zworki, klej na gorąco i mocna szara taśma.

Oprogramowanie translatora zostało zaimplementowane w języku C++. Cała komunikacja translatora z aplikacją (i odwrotnie) odbywa się za pomocą technologii Bluetooth 4.2.

Szczegóły implementacji znajdują się w moim repozytorium poświęconym rozwojowi tego projektu [18]

### **Przepływ komunikacji od aplikacji, do translatora**

Najpierw zostaje uruchomiony kanał komunikacji Bluetooth między ESP32 a podłączoną aplikacją na smartfonie. Następnie kolejne znaki tekstu z aplikacji zostają przesłane do mikrokontrolera. W kolejnym kroku każdy pojedynczy znak zostaje przechwycony, a następnie zostaje znaleziona jego reprezentacja za pomocą odpowiednich pinów. W ostatnim kroku serwa odpowiadające konkretnym pinom, obecnym w reprezentacji danego znaku - zmieniają pozycję ze 150 stopni (dół) na 170 stopni (górze) co powoduje podbicie pinu. Stan ten jest podtrzymywany przez sekundę a następnie wszystkie piny zostają sprowadzone do stanu pierwotnego (dół). Po każdym wyświetleniu znaku następuje krótka przerwa o długości 300 ms.

## **3.2. Aplikacja**

Najważniejszym zadaniem aplikacji jest wykorzystanie istniejącego, wbudowanego w smartfon aparatu w następujący sposób:

Najpierw wybieramy sobie tekst, który nas interesuje. Najlepiej taki kontrastujący z tłem (np. książka - czarny tekst na białym tle). Następnie wykonujemy za pomocą aplikacji zdjęcie tego tekstu. Potem używając dostępnego modelu AI do rozpoznawania tekstu z obrazu [19], ze zdjęcia zostaje przechwycony tekst. Tekst zostanie wyświetlony do podglądu w aplikacji. Dzięki czemu wiemy co zostało uchwycone i w jaki sposób. Na koniec ten właśnie tekst możemy nadpisać – powtarzając powyższe kroki na nowo, albo wysłać do translatora za pomocą dedykowanego przycisku.

### 3.3. Zalety i wady mojego rozwiązania

Po omówieniu mojego rozwiązania, jak i tych z poprzedniego rozdziału, przedstawię jego zalety i wady. Zalety wyglądają następująco:

- Prostota konstrukcji – na pojedynczy pin przypada jeden serwomechanizm.
- Użycie gotowych, dostępnych na rynku tanich części elektronicznych.
- Relatywnie niski koszt produkcji w porównaniu do pozostałych rozwiązań.
- Piny są lekkie, plastikowe, ważą mniej niż 1 gram zatem serwo praktycznie nie traci energii przy podtrzymywaniu stanu pinu.
- Wszystkie piny są synchronizowane jednocześnie.
- Dedykowana aplikacja, łącząca się z translatorem przy użyciu Bluetooth (tak jak pozostałe urządzenia).
- Ze względu na duże gabaryty translatora, jest on odporny na kurz i pył.

A teraz wady, które są jednak dość istotne w odniesieniu do pomysłu na projekt:

- Na ten moment urządzenie wyświetla pojedyncze znaki tekstu po kolei.
- Znak jest wyświetlany statycznie – co jak wykazano w [20] nie jest optymalnym sposobem wyświetlania tekstu.
- Translator jest dość dużych rozmiarów jak na jeden znak.





## 4. Instrukcja dla użytkowników

### 4.1. Do czego służy aplikacja

Zadaniem aplikacji jest komunikacja użytkownika z translatozem. Użytkownik poprzez interfejs aplikacji ma możliwość wykonania zdjęcia, oraz przesłania wyekstrahowanego tekstu do przetłumaczenia do translatora.

### 4.2. Jak zainstalować aplikację

Aplikację można zainstalować pobierając plik *texttobracille.apk* z mojego repozytorium [18] a następnie instalując go na urządzeniu. *Aby poprawnie zainstalować plik .apk na smartfonie z Androidem, należy wcześniej zezwolić na instalowanie aplikacji z zewnętrznych źródeł. Instrukcja jak wykonać to poprawnie znajduje się w [21].*

### 4.3. Jak używać aplikacji

Po otwarciu aplikacji *TextToBraille* naszym oczom ukazuje się pusty obszar w kolorze jasnego fioleto i cztery ciemnofioletowe guziki. Funkcje elementów interfejsu zostały opisane poniżej:

**Widok tekstowy** – wyświetla rozpoznany tekst ze zrobionego zdjęcia

**Connect** – inicjuje rozpoczęcie komunikacji po Bluetooth z translatozem

**Disconnect** – przerywa istniejące połączenie Bluetooth z translatozem

**Photo** – uruchamia systemową aplikację do robienia zdjęć

**Send** – przesyła niepusty tekst w widoku tekstowym do translatora

### 4.4. Jak używać translatora

Translator w pierwszej kolejności należy podłączyć do gniazda zasilania USB. Po podłączeniu translator jest w pełni gotowy, aby połączyć się ze smartfonem za pomocą aplikacji. *Aby translator działał poprawnie, nie należy stosować dużej siły podczas odczytywania pinów znaku Braille’a oraz odłączać go od źródła zasilania w trakcie działania*



## 5. Użyte narzędzia

### 5.1. Narzędzia użyte do stworzenia oprogramowania translatora

- VS Code – bardzo wygodny edytor kodu, który wybrałam ze względu na wieloplatformowość i sporo dostępnych pluginów (m.in PlatformIO).
- PlatformIO – zintegrowane środowisko do kompilacji, wgrywania kodu źródłowego na ESP32.
- C++ – użyty ze względu na szeroką kompatybilność z interfejsem Arduino do programowania mikrokontrolerów.
- `Arduino.h` – używana do programowania ESP32 w prosty sposób.
- `BluetoothSerial.h` – używana do postawienia kanału i dalszej komunikacji mikrokontrolera z aplikacją.
- `ESP32Servo.h` – daje prosty interfejs do kontroli pozycji serwa.
- GitHub Copilot – użyty do poprawy składni i generowania żmudnych odwzorowań znaków na piny.

### 5.2. Narzędzia użyte do stworzenia aplikacji

- Android Studio – najlepsze znane mi środowisko do wytwarzania aplikacji na Androida. Wybrałam też dlatego, że daje możliwość emulowania i podglądu interfejsu graficznego.
- Kotlin – powszechnie rekomendowany język programowania androidowych aplikacji, wspierany przez Android Studio.
- Google ML Kit Text recognition v2 – użyty do rozpoznawania tekstu ze zdjęcia. Główna zaletą jest to, że działa bez dostępu do internetu.
- ChatGPT – do nauki składni i poprawy kodu w Kotlinie.



## 6. Przypadki użycia

### 6.1. Wykonanie zdjęcia za pomocą aplikacji

Aktor: Osoba niewidoma posiadająca smartfona z Androidem i zainstalowaną aplikacją TextToBraille.

Przypadek użycia: Osoba naciska guzik *Photo* co otwiera systemową aplikację do robienia zdjęć. Następnie osoba wykonuje zdjęcie tekstu, który chce przeczytać. Zdjęcie zostaje zrobione. Osoba potwierdza chęć przetworzenia tego zdjęcia, poprzez kliknięcie *akceptuj*. Zdjęcie zostaje przetworzone na tekst, który zostaje wyświetlony w aplikacji.

### 6.2. Przesłanie tekstu do tłumacza

Aktor: Osoba niewidoma posiadająca już tekst w aplikacji TextToBraille.

Przypadek użycia: Osoba naciska guzik *Connect*, co inicjuje połączenie z tłumaczem. Następnie klika guzik *Send*, po czym tekst z aplikacji zostaje przesłany do tłumacza. Tłumacz natychmiast zaczyna wyświetlać litery tekstu w alfabecie Braille’a.

### 6.3. Zakończenie połączenia z tłumaczem

Aktor: Osoba niewidoma z uruchomioną aplikacją TextToBraille i połączonym po Bluetooth tłumaczem.

Przypadek użycia: Osoba naciska guzik *Disconnect*, a to powoduje zakończenie połączenia po stronie smartfona. Do chwili ponownego nawiązania połączenia, nie ma możliwości przesłania zeskanowanego tekstu.



## 7. Dalsze kroki rozwoju projektu

### 7.1. Moje wnioski

Stworzenie tego projektu było naprawdę trudne. Wielu ze wspomnianych wcześniej narzędzi wcześniej nie używałam, także było to poznawanie ich praktycznie od zera. Wymagało to ode mnie sporo czasu i nauczania się nowych umiejętności z różnych obszarów jak np:

- lutowanie i obsługa elementów elektronicznych,
- programowanie w języku Kotlin,
- rozeznanie w Android Studio,
- planowanie i organizacja.

Podsumowując nakład czasowy poświęcony na rozwój projektu, wygląda to tak: Na sam research dotyczący projektu, poświęciłam około 15h. Na składaniu translatora i poprawiania problemów z nim związanych około 42h. Na tworzeniu kodu aplikacji jak i dla mikrokontrolera około 18h. Na spisywanie pracy inżynierskiej w Latexu ok. 16h.

### 7.2. Szacowany koszt wytworzenia prototypu

- ESP32 WiFi+BT 4.2 WROOM-32 – **22.49 PLN**.
- Trzy mikroserwa 3.3V – **14.90 PLN x 3**.
- Trzy mikroserwa 5V – **17.90 PLN x 3**.
- Sześć walców o 20mm x 6mm – **0.70 PLN**.
- Uniwersalna płytki PCB – **3 PLN**.
- Trzy konwertery poziomów logicznych napięć – **2.50 PLN x 3**.
- Kabel USB-C – **3.47 PLN**.

Sumarycznie: **135.56 PLN** a więc zdecydowanie mniej niż 800 USD

### 7.3. Potencjalne usprawnienia do projektu

- Dodatkowa weryfikacja połączenia Bluetooth w kodzie mikrokontrolera.
- Dostosowanie interfejsu aplikacji pod niewidomych.
- Stworzenie odpowiednika aplikacji na IOS.
- Modyfikacja rozmiaru i działania konstrukcji.
- Otwartoźródłowy model do rozpoznawania tekstu zamiast od Google.



# Bibliografia

- [1] Carina J. Sabourin, Yaser Merrikhi, and Stephen G. Lomber. “Do blind people hear better?” In: *Trends in Cognitive Sciences* 26.11 (2022), pp. 999–1012. ISSN: 1364-6613. DOI: <https://doi.org/10.1016/j.tics.2022.08.016>. URL: <https://www.sciencedirect.com/science/article/pii/S1364661322002121>.
- [2] Daniel Goldreich and Ingrid M. Kanics. “Tactile Acuity is Enhanced in Blindness”. In: *Journal of Neuroscience* 23.8 (2003), pp. 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>.
- [3] *Czarna książka kolorów - Menena Cottin*. [https://wydawnictwo-widnokrag.pl/ksiazki/czarna\\_ksiazka\\_kolorow/](https://wydawnictwo-widnokrag.pl/ksiazki/czarna_ksiazka_kolorow/). 2012.
- [4] *Refreshable Braille Display with electromagnets usage*. <https://hackaday.io/project/191181-electromechanical-refreshable-braille-module>. 2023.
- [5] *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>.
- [6] *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.
- [7] *Why do electromagnets generate overheating during using - causes and solutions*. <https://www.dr-solenoid.com/news/why-do-electromagnets-generate-over-heating-during-using-causes-and-solutions/>. 2024.
- [8] Frédéric Giraud and Christophe Giraud-Audine. “Chapter One - Introduction”. In: *Piezoelectric Actuators: Vector Control Method*. Ed. by Frédéric Giraud and Christophe Giraud-Audine. Butterworth-Heinemann, 2019, pp. 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>.
- [9] *Piezoelectric bending transducer expensive solution*. <https://piezo.com/products/piezoelectric-bending-transducer-s234-h5fr-1803xb>.
- [10] *Piezoelectric bending transducer cheaper solution*. <https://www.thorlabs.com/item/PB4VB2W>.
- [11] *Braillelite2000 device disassembly demonstration*. [https://www.youtube.com/watch?v=guZ2rFT\\_FKQ](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] 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](https://en.wikipedia.org/w/index.php?title=Perkins_Brailler&oldid=1321563538).
- [15] *Catalog of bestselling braille-printed books*. <http://www.braillebookstore.com/Adult-Bestsellers>.
- [16] Michael Trembl et al. In: *Current Directions in Biomedical Engineering* 6.2 (2020), p. 20202004. DOI: doi:10.1515/cdbme-2020-2004. URL: <https://doi.org/10.1515/cdbme-2020-2004>.
- [17] *Fully mechanical concept for refreshable braille display module*. <https://www.thingiverse.com/thing:90144>. 2013.
- [18] *My TextToBraille project repository*. <https://github.com/gagata7/TextToBraille>. 2025.
- [19] *Google ML Kit Text recognition v2*. <https://developers.google.com/ml-kit/vision/text-recognition/v2>.
- [20] Joerg Fricke and Helmut Baehring. “Displaying laterally moving tactile information”. In: *Computers for Handicapped Persons*. Ed. by Wolfgang L. Zagler, Geoffrey Busby, and Roland R. Wagner. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 461–468. ISBN: 978-3-540-48989-4.
- [21] *Guide to Installing APK Files on Android Devices*. <https://www.lifewire.com/install-apk-on-android-4177185>. 2025.