Design Document

Optical Text-to-Braille Translation Device
Date: May 22, 2024
DOT. - Abigail Figueroa, Charuvi Singh, Derek Desrosiers, Kruthi Gundu, Samhitha Bodangi

Minor Cosmetic & Functional Changes

From the CDR, we made a few changes to our prototype. We narrowed our designs down from three to one, through our testing. Additionally, we attempted to resize the Braille cells after running into issues with the resolution of the SLA printer. After testing, we decided the design performed at a practical and accessible level for the project's time constraint.

Current Design

The current device consists of two developed components. First, is the foundational aspect of the braille display, which couldn't be fully developed to the extent we wished because of consistent misprinting errors of the braille cell. However, our testing and design studies show that the concepts function as intended and that we have all the necessary parts. The only improvement needed is a more precise 3D printer that can handle printing the Braille cell in higher resolution. Second is the text-to-braille configuration conversion aspect, in which we were able to create and refine a computer model for. The Raspberry Pi was used to take camera images and perform Optical Character Recognition using Python's Tesseract library. After converting images into text, the Raspberry Pi uses the pybraille library to convert tests into Unicode braille, which is displayed on the live web server. In the future, both components will be combined to automate the process and connect the OCR to the device itself.

Overview of Build Steps

Braille Pin Assembly

Set Up Winding Tool:

1. Add drill chucks to the solenoid winding tool for a firmer grasp on the ferrite rod.

Wind Solenoids:

2. Use the tool, drill, and drill chucks to rotate the iron ferrite rod and wrap copper coil around several times.

Prepare Cams:

3. Cut the cams from their 3D printed supports.
4. Place the rare-earth magnet inside each cam.
5. Support the cam on a small axle printed on top of the braille cell, ensuring it can rotate up and down.

Assemble Braille Pins:

6. Place the cam into the slot for the braille pin.
7. Place the braille pin on top of the cam.

Test Solenoids:

8. Perform the "One-Dot" Test by connecting the pin to a 6V voltage source.
9. Verify that the pin raises and lowers successfully.

PCB Assembly

Gather Components:

1. Refer to the spreadsheet for files and parts needed (besides the PCB boards themselves).

Solder Components:

2. Follow the designations for each part's placement on the board.
3. Solder each part in its appropriate place using a magnifying glass and smaller soldering iron for precision.
4. Ensure decoders are facing the correct direction.

Use Solder Flux:

5. Apply solder flux to help solder sit in its proper pad, especially for parts in close proximity on the evaluation board.

Install Arduino Nano:

6. Use an Arduino Nano to run the necessary code for switching the polarity of the Braille cell pins and to regulate voltage throughout the PCB board.

OCR and Text Conversion System

Set Up Raspberry Pi:

1. Connect a web camera to the Raspberry Pi.
2. Install necessary dependencies.
3. Create a virtual environment in Python.

Initialize Web Server:

4. Initialize a web server on the Raspberry Pi to display the visual conversion between text and braille.

Perform OCR:

5. Use the Pytesseract library to perform OCR.
6. Take a snapshot of the camera image and run the Braille conversion program.
7. Use the Pybraille library to convert text into Unicode Braille characters.
8. Output the Braille Unicode characters on the web server for digital text translation into tactile Braille.

Strengths and Weaknesses of Final Design

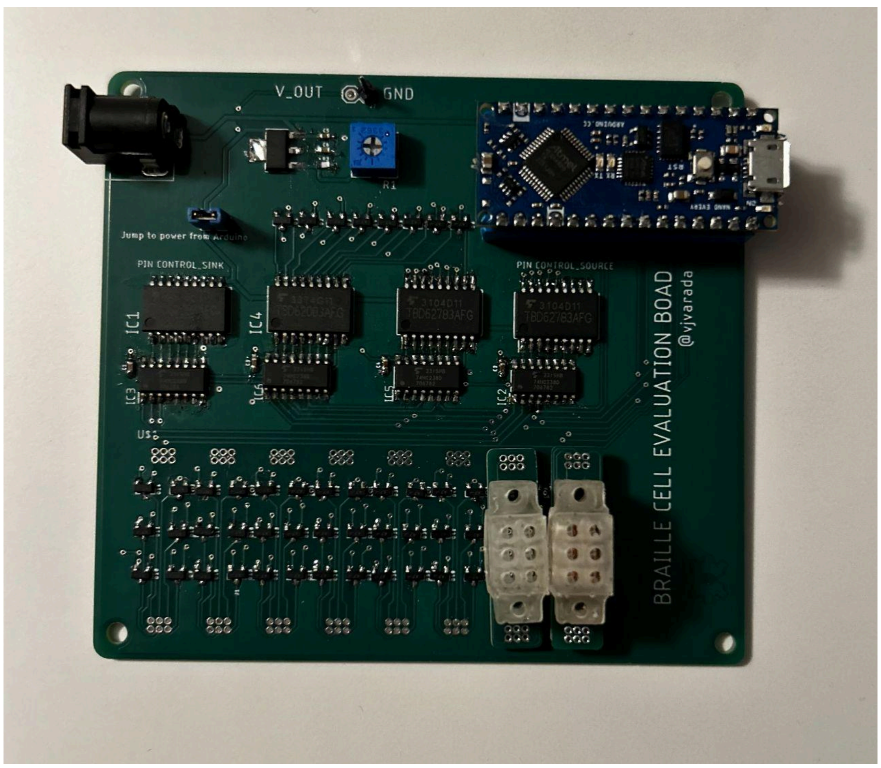

Figure 1: Final Prototype

A few strengths of our final design is that it fulfills the primary criteria, including being cost effective, portable, durable, and functional. In addition, the design is subtle and elegant due to its small size, light weight, and minimal look.

The only level one requirements not met are those to do with the audio component of the device, which would repeat the text in the text-to-speech format and provide auditory instructions to the user for positioning the camera in the ideal location. In hindsight, perhaps these requirements would have been better off being labeled as level two, as they are not critical to the functionality of the device, but the reason for the failure to meet was that we chose to devote most of our time to ensuring that key aspects of the device perform the way they need to.