Brailliant

E-Book & Learning Tool for the Visually Impaired

Team A0 - Yujun Lee, Ziyu Li, Samay Sahoo

The Problem & Solution

- 10% literacy in braille
- Most blind children attend public schools: few teachers know braille
- Most aids are audio based
  - Braille provides complete command of written language
- Current refreshable braille readers are expensive: $2000-$6000

Need a cheap and accessible way to

help blind students learn and read braille!

- A small form factor braille reader
  - Mechanical innovation to drastically reduce price
  - No proprietary parts, all purchasable or 3D-printable
  - Avoiding previous solution with solenoids -> lower power consumption
  - Open source and DIYable
  - Any text or learning guide can be inputted from a web app

Use Case and Design Requirements

10-letter word: fabricated — **10 letters**

Grade 1 (uncontracted): ⠋⠁⠃⠗⠊⠉⠁⠞⠑⠙ — **10 cells**

Grade 2 (contracted): ⠋⠁⠃⠗⠊⠉⠁⠞⠫ — **9 cells**

Battery life **6 hr** in a school day	**14400 mAh** battery (0.6W/motor * 20*6hr)
Readable physical braille	CAD: ensure **1mm** pin extrusion, **6-8mm** spacing of pins/holes **10 cells**: length enough to display any word
Beginner reading speed (12wpm)	**0.5s/cell** actuation time
Portability in form factor	Dimensions: **< 12" x 8"**
Accurate braille display	Motor code: Recognize stalling and initiate reset

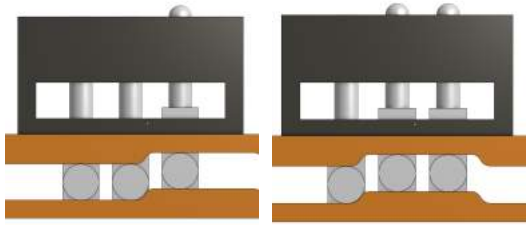
One cell:

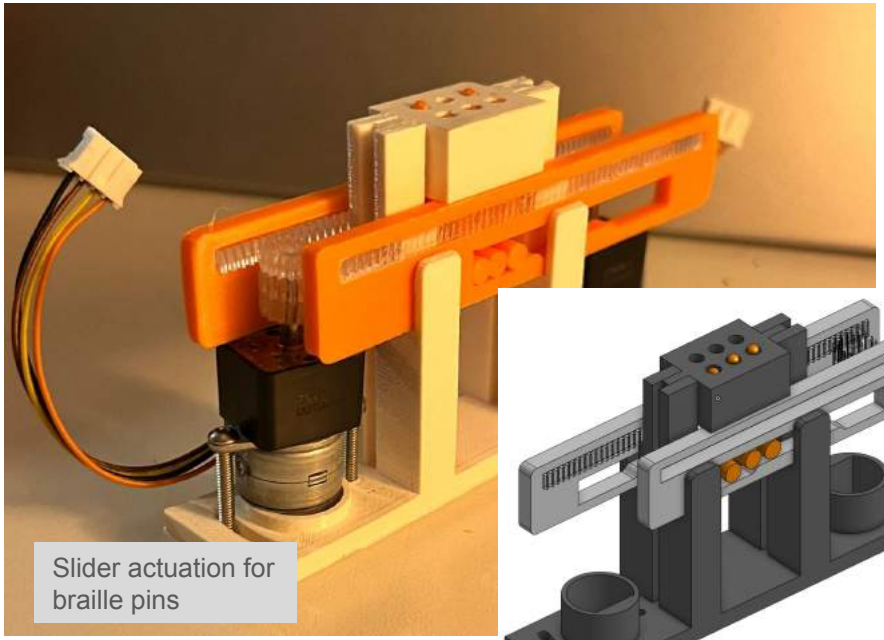
Ethical Concerns

- Public Health, Safety, and Welfare:
  - Ensured higher accuracy for translation (95% -> 100%) + Motor jam detector
  - No visually impaired users should be harmed by being reliant on our product
- Cultural:
  - Support for English translation only could generate cultural barriers between the user and product
  - Open source braille translation code that could be easily altered with varying dictionary implementation
- Economic & social:
  - High prices of previous designs made products accessible only to the privileged
    - Solution: ensured low design cost with slider implementation (two motors per cell)
  - Previous design of inputting a txt file could create barriers for people unfamiliar with tech
    - Change: simple web app design that directly takes input as user types on the prompt
    - Change: clear visualization of words inputted into the braille pad

Solution Approach - Hardware

An all new & custom actuator design & fabrication solution

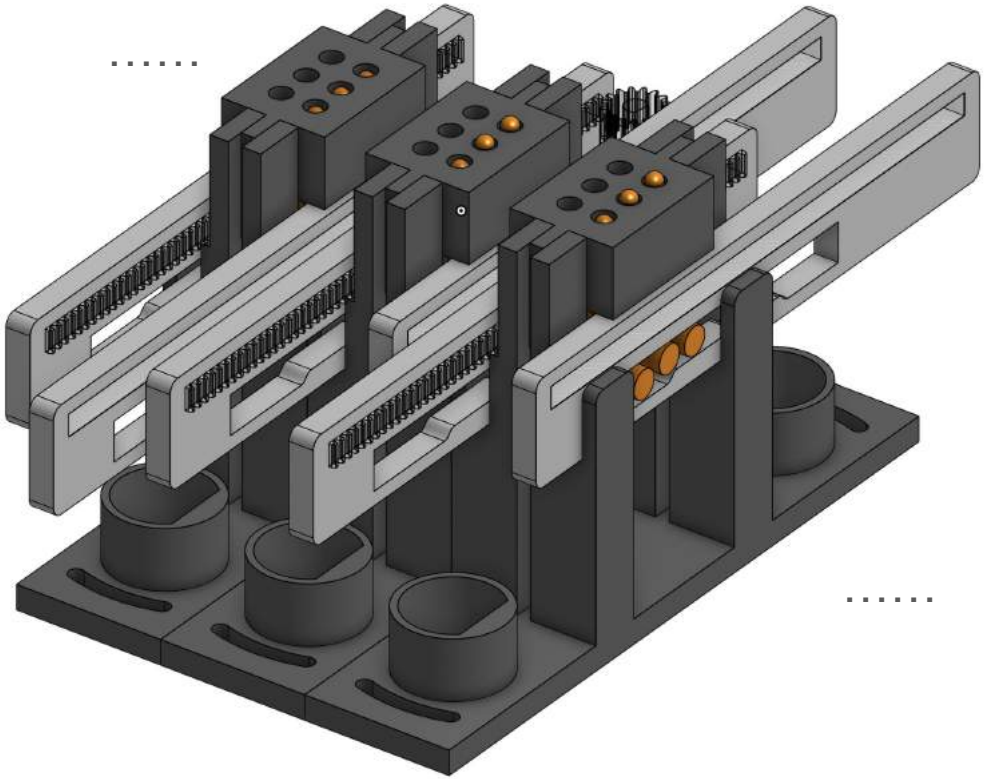




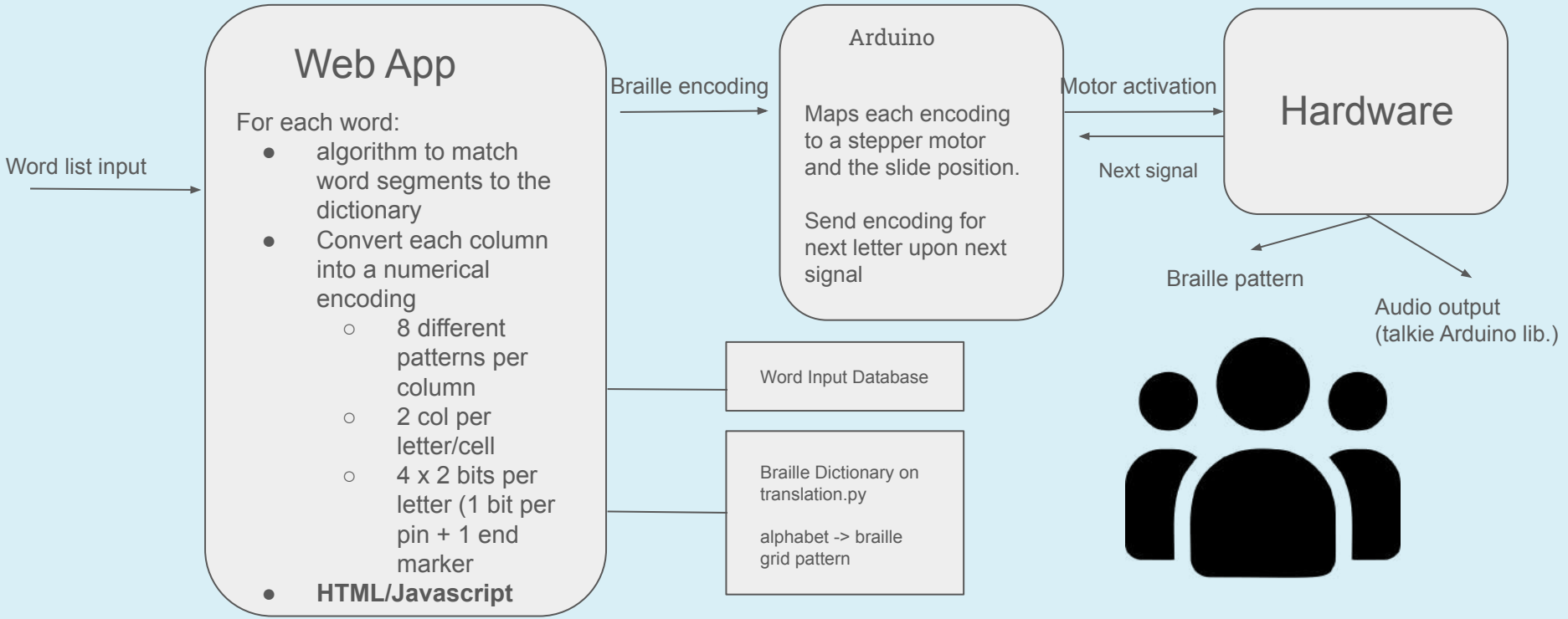
Slider actuation for braille pins

Gear and rack config for motor

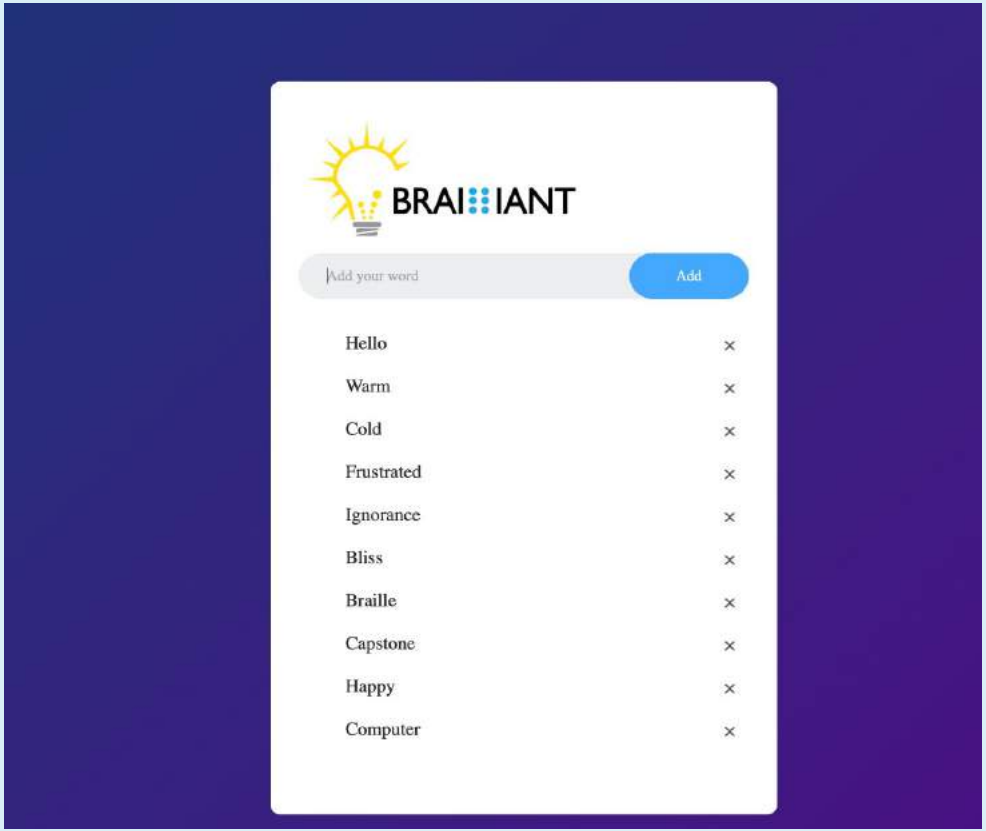
Complete Solution - Scalable Array



Software approach

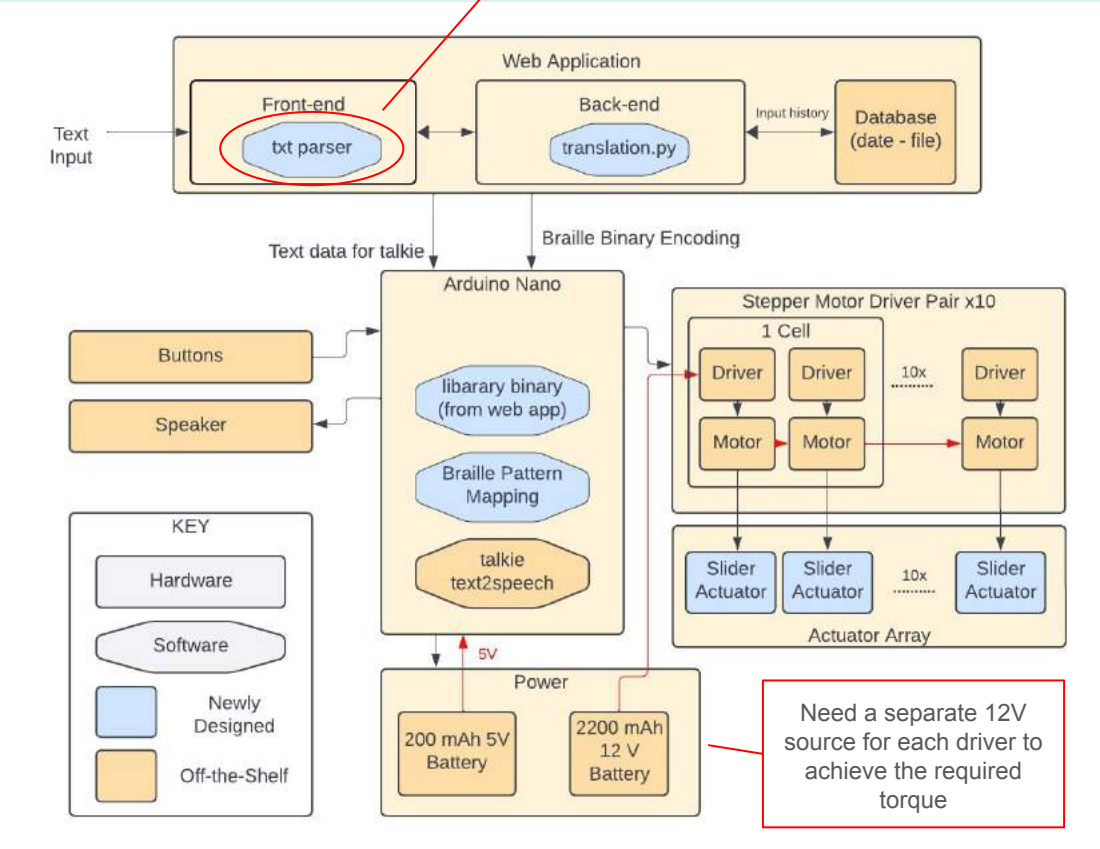


Software approach



- Data sent to braille pad with translated encoding when added/removed
- Prompt to add words
- Words easily removed with x button
- Data saved to web database (refreshing does not refresh data)

Full Block Diagram



Verification Metrics

Refresh rate of each cell (pair of motors) at learning reading speed	Enter 10 words and measure time as device actuates one cell, one character at a time
Power consumption (battery life > 6hr)	Measure the max run time of device from full charge under normal use
Text to Braille Algorithm Accuracy (100%)	Test on 100 word samples and compare device pin outputs to online braille translator (slide 4)
Physical braille pin pattern display accuracy (slider accuracy)	Test physical grid on more than 50 braille pattern inputs to reach >80% accuracy
Cost cap at $600	Cost analysis by summing all the parts needed
Web App test for usability and simplicity	User testing on 5 students (feedback on difficulty to use on a scale of 0-10)

Results (Hardware)

Test Design:

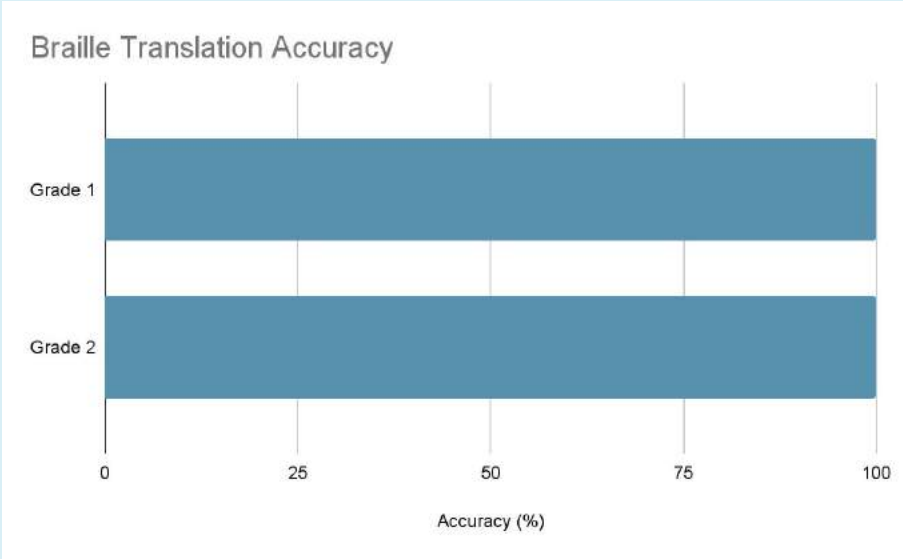
- Goal: 0.5s per cell
- Test on 50 words
    - Encoding outputs for inputs sent from translation.py to Arduino
- Record video to observe accurate number of position changes per cell + responses to any jams

Goal	**Result**
0.5s/cell	0.73s
Stall testing	Recovery within 3s
1hr stress test (accuracy testing)	Success; but motor temperature concerns

Results (Software)

Test Design:

- Goal: use case requirement of 100% braille translation accuracy
- Test on 100 words
  - Randomly generated
  - 50 non contractible words (Grade 1)
  - 50 contractible words (Grade 2)
- Pattern display function made for braille encodings



Output displayed on test function

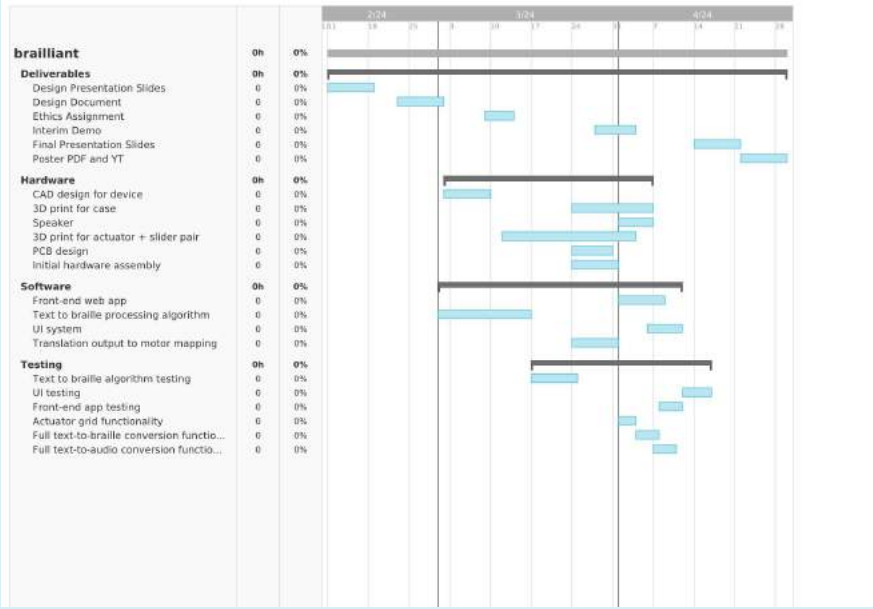
```
['hello']
 1  0   1  0   1  0   1  0   1  0  x
 1  1   0  1   1  0   1  0   0  1  x
 0  0   0  0   1  0   1  0   1  0  x
```

Reference solution from BrailleTranslator.org

Trade-Offs

- Braille cell sizing is 2x the standard spacing to reduce pin sliding error
- Torque over speed to prioritize actuation functionality
- Require additional 12V power source to achieve above torque

Current Schedule



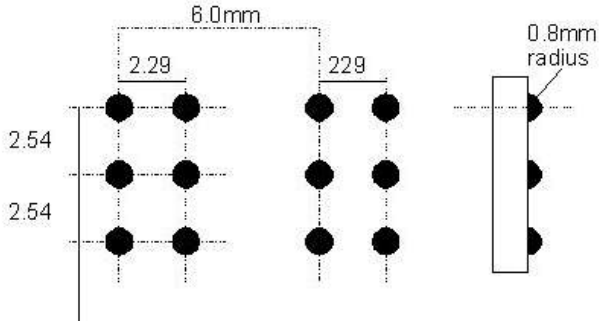
Actuator Verification - Requirement

Size:

1. <= 2x the American braille standard.
2. Maintaining all correct proportions

Refresh Rate:

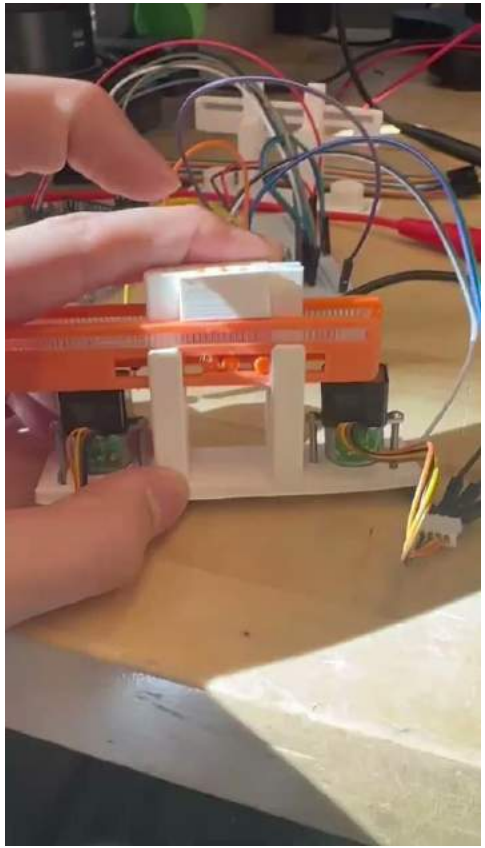
500ms slider side to side movement time



Refresh rate: **0.5s/cell = 5s** *max.*

for 10-cell word => **12 wpm**

Actuator Verification - Result



- Size: 2x the standard braille size, on par with our requirement.
- Refresh Rate: ~4s side to side slider movement speed, way exceeding requirement, need to perform further optimization and verification.

Actual dimensions fabricated