Brailliant

E-Book & Learning Tool for the Visually Impaired

Team A0 - Ziyu Li, Samay Sahoo, Yujun Lee

The Problem

- Low literacy in braille for: 10%
- Most blind children attend public schools where few teachers know braille
- Most aids are audio based:
  - Various limits to audio aid
  - 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!

Our Solution

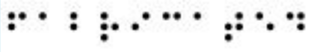
- A **small form factor braille reader** made from cheaper components
  - Mechanical innovation to drastically reduce price
  - No proprietary parts, all purchasable or 3D-printable
  - Avoiding previous solutions with solenoids -> lower power consumption

- Open source and DIYable at home
  - Text/learning guide can be inputted from a web app

…but cheap and easily manufacturable

Design Requirements

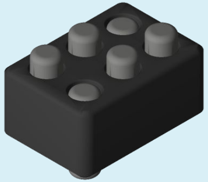
10-letter word: fabricated **10 letters**

Grade 1 (uncontracted):  **10 cells**

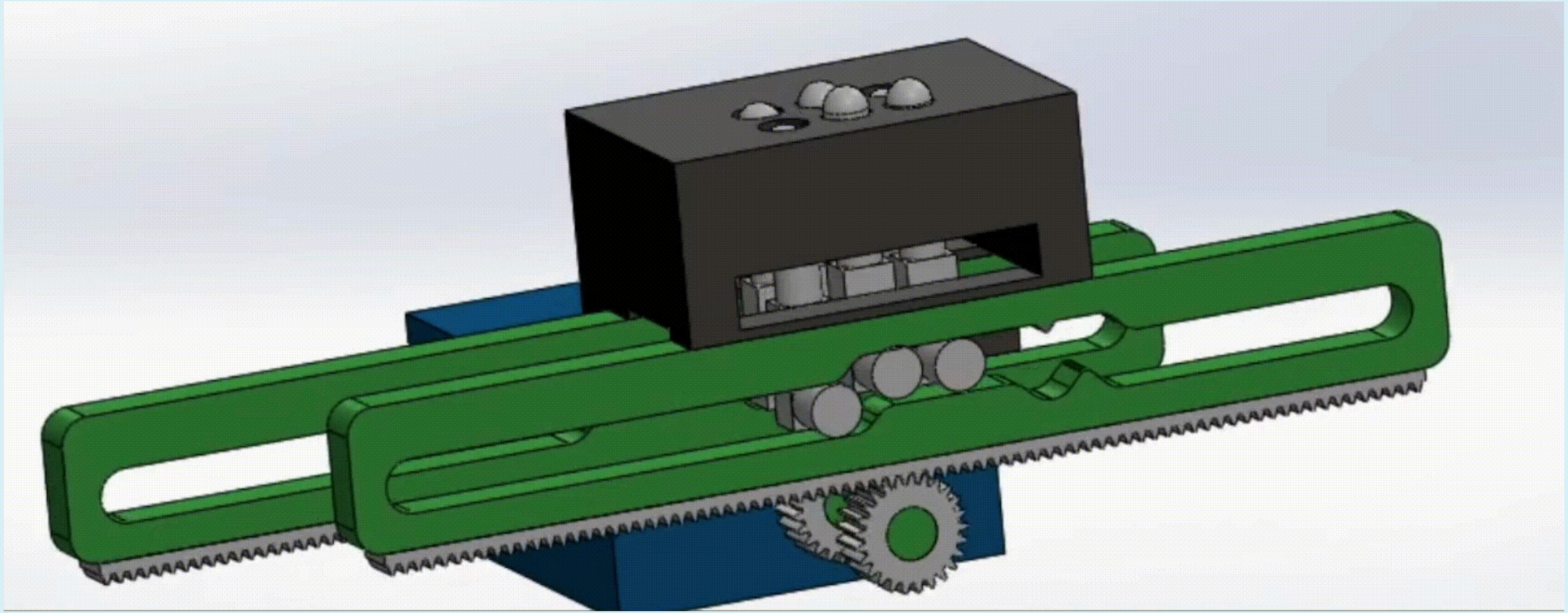
Grade 2 (contracted):  **9 cells**

- Portable form factor: **< 12” x 8”**
- Braille on physical buttons
- Resolution: **10 braille cells = 60 pins**
- Cost: **≤ $600**
- Refresh rate: **0.5s/cell = 5s** *max.* for 10-cell word => **12 wpm**
- Battery life **1 hr:** (0.6 W/motor) = 12W = **2400 mAh** battery
- **>80% accuracy:** text to braille encoding
    - Error handling for unrecognizable characters
- **>80% accuracy:** pin patterns on device

One cell:

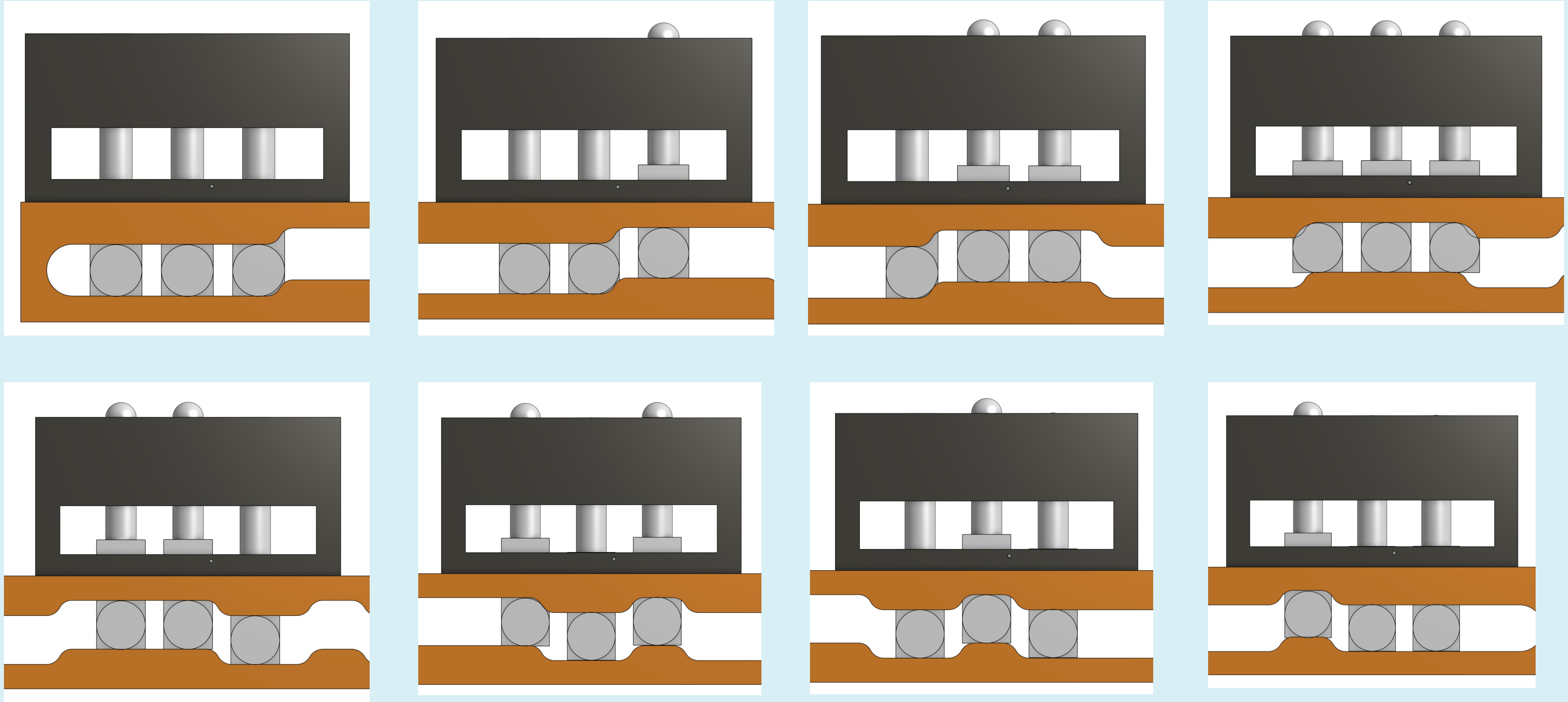


Hardware Solution – 2 Sliders



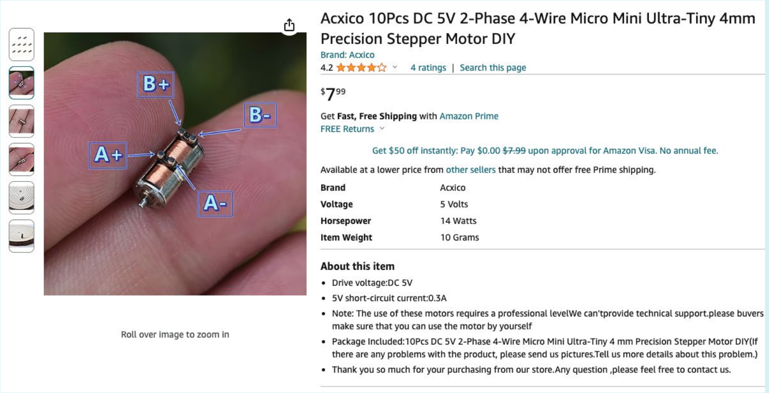
Actuate Patterns Instead of Individual Dots!

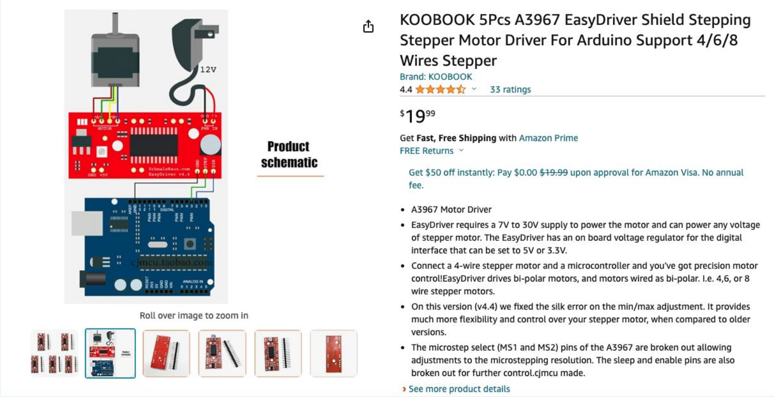
Hardware solution - Patterns Attainable



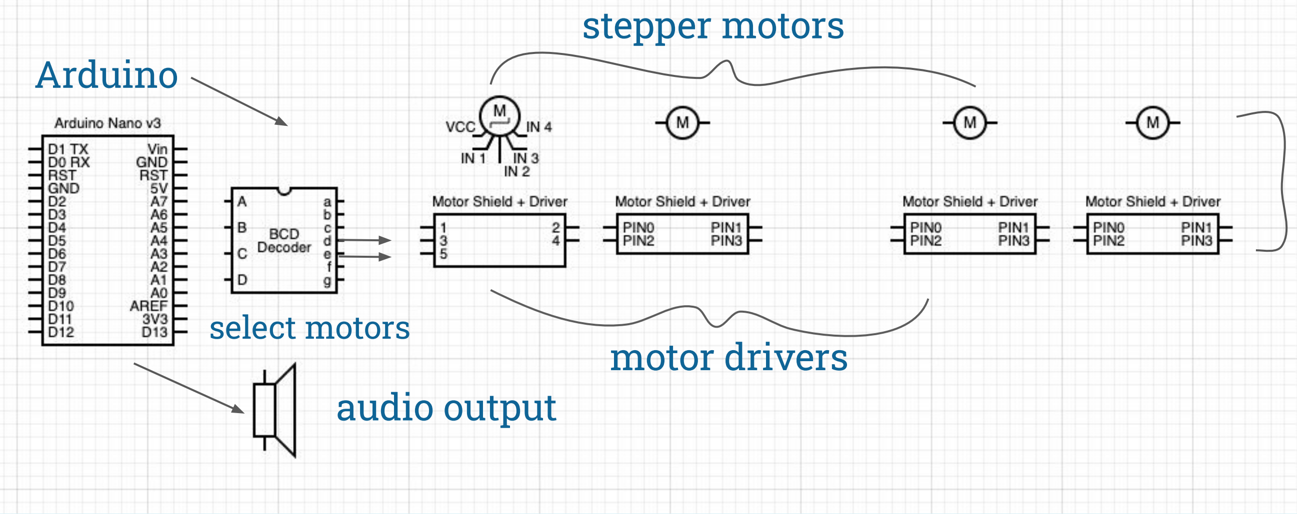
Implementation Plan

- Micro stepper motors for each slider
- Arduino microcontroller to drive motors
  - Need a motor controller per motor
  - Motor shield to interface with Arduino
- 3D printed sliders with pin pattern combinations
- Decoder to driver motors sequentially
  - Due to limited Arduino DC pins and power
- Audio text-speech voice
  - Output to a speaker after each word
- Rechargeable battery (> 2400 mAh)





System Specs / Block Diagram

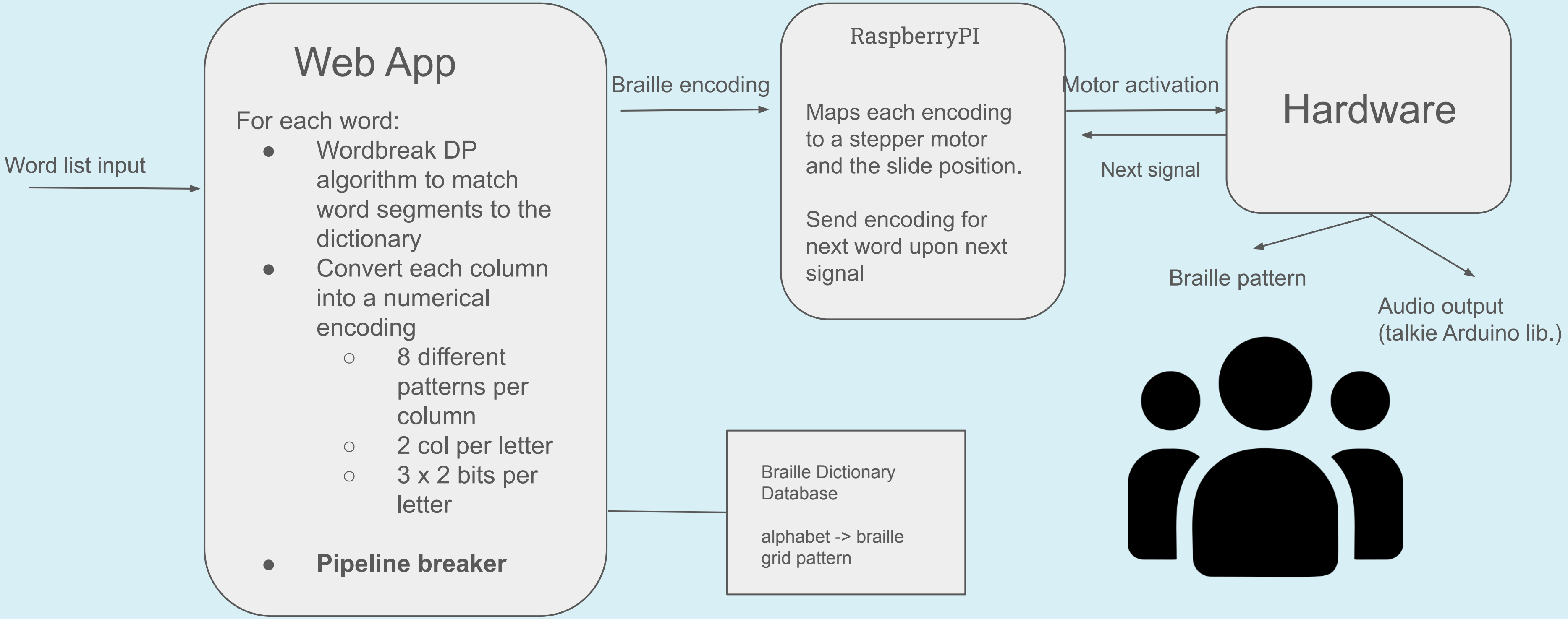


2 motors + drivers per cell

10x

● ● ●

Software approach



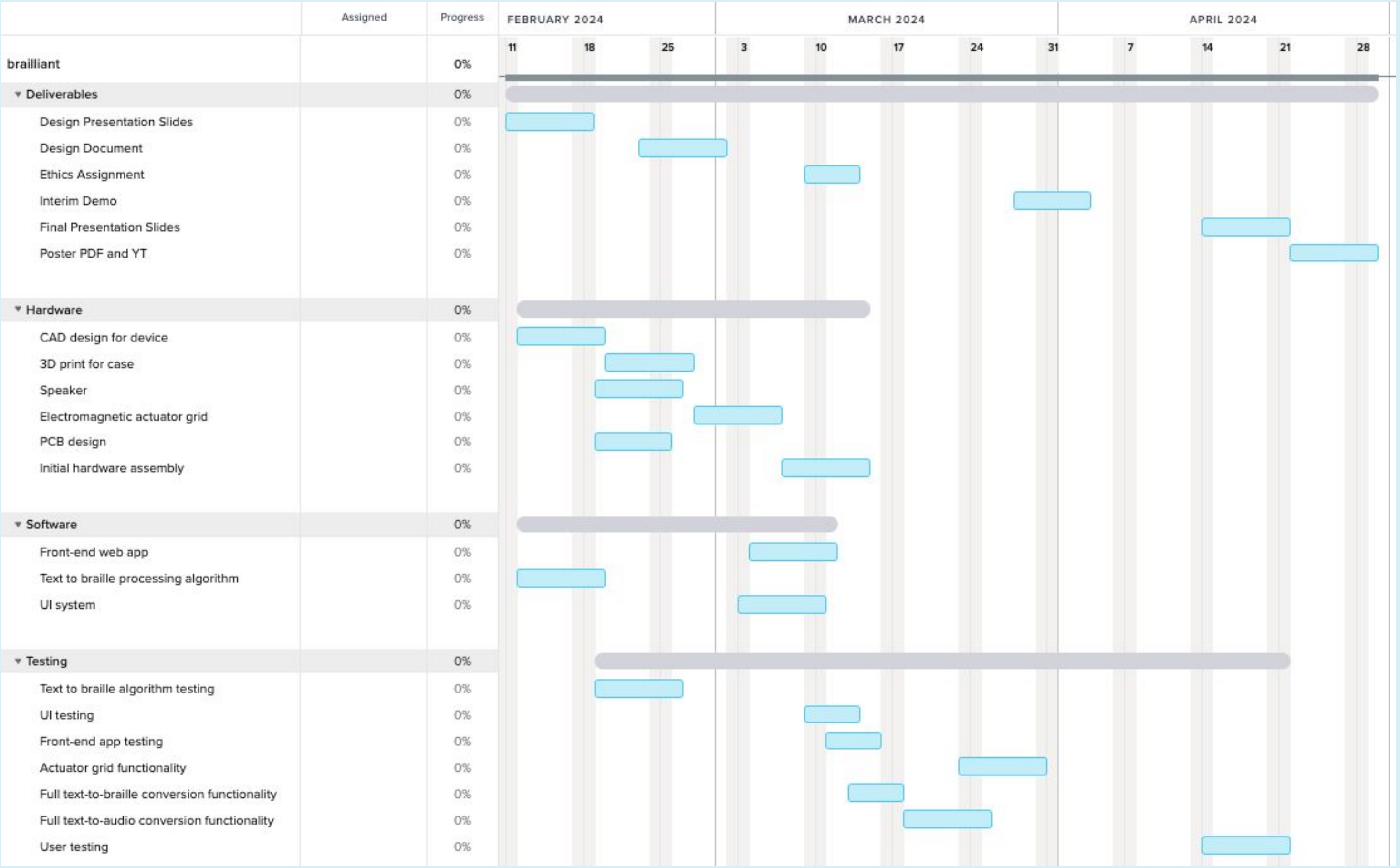
Testing/Verification Metrics

Refresh rate of each cell (2x motor) at learning reading speed	Enter a 10-cell braille word and measure time as device controls motors sequentially
Power consumption (battery life > 1hr)	Measure the max run time of device from full charge under normal use
Text to Braille Algorithm Accuracy	Test on 50 text 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
Resolution to effectively display readable braille characters	User testing on 10 visually impaired students; Office of Disability Resources/School for Blind Children

Division of Labour

Software (Yujun Lee)	• Front-end (web app): UI for text data input from user + filtering on unidentified words • Text to grid parser for Raspberry Pi • Text to braille using word-break algorithm
Hardware (Samay Sahoo, Ziyu Li)	• Speaker implementation - Samay • Step motor grid - Ziyu • PCB design (power management, microcontroller, I/O buttons) - Ziyu • CAD design + 3D printing for slider and pins - Ziyu • Physical product design - Samay
Data Collection (All)	• Sequence of braille-translatable words • Braille patterns for actuator grid testing

Schedule


Assigned
Progress
FEBRUARY 2024
MARCH 2024
APRIL 2024
11
18
25
3
10
17
24
31
7
14
21
28
brailliant
0%
Deliverables
0%
Design Presentation Slides
0%
Design Document
0%
Ethics Assignment
0%
Interim Demo
0%
Final Presentation Slides
0%
Poster PDF and YT
0%
Hardware
0%
CAD design for device
0%
3D print for case
0%
Speaker
0%
Electromagnetic actuator grid
0%
PCB design
0%
Initial hardware assembly
0%
Software
0%
Front-end web app
0%
Text to braille processing algorithm
0%
UI system
0%
Testing
0%
Text to braille algorithm testing
0%
UI testing
0%
Front-end app testing
0%
Actuator grid functionality
0%
Full text-to-braille conversion functionality
0%
Full text-to-audio conversion functionality
0%
User testing
0%