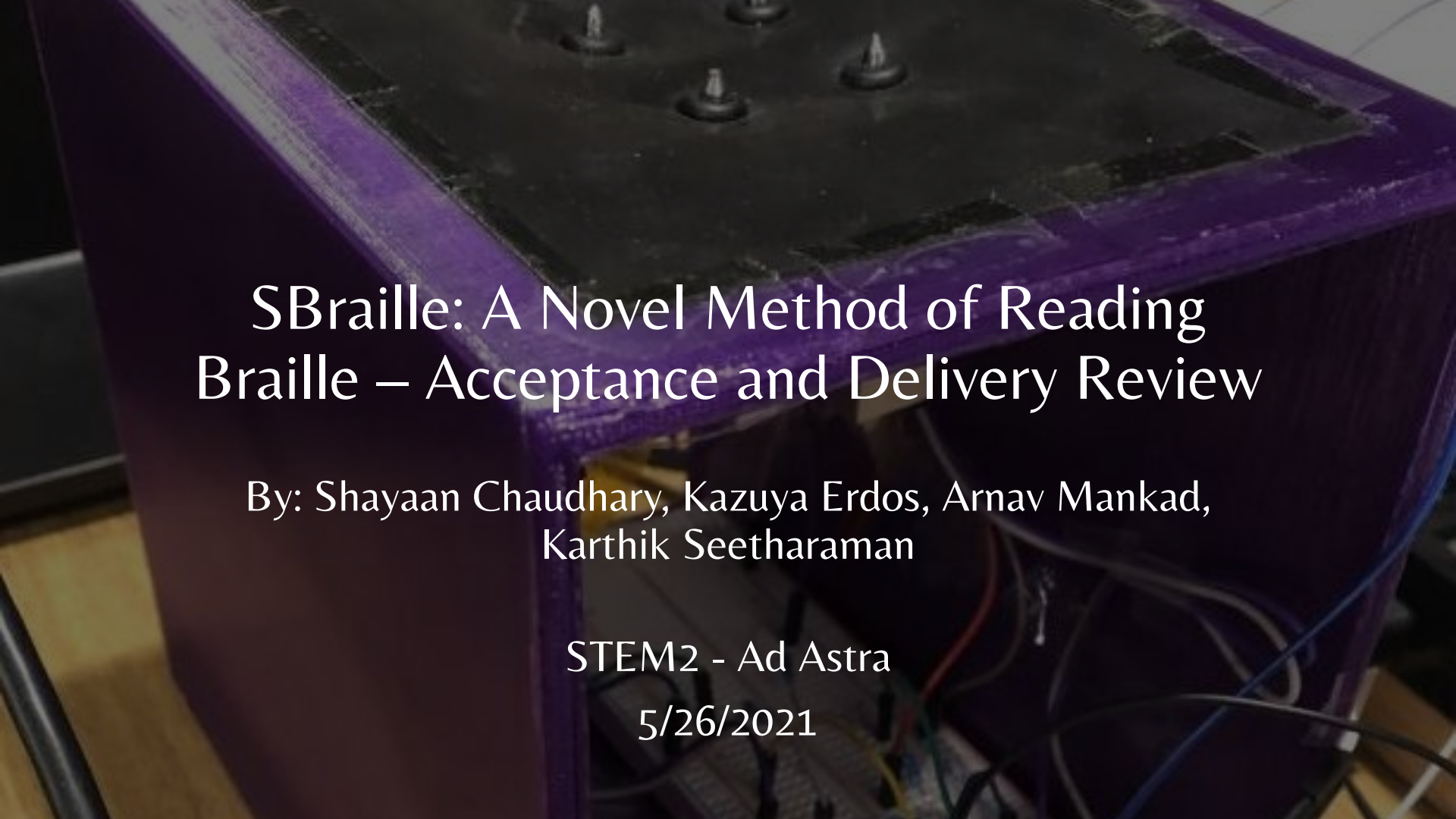


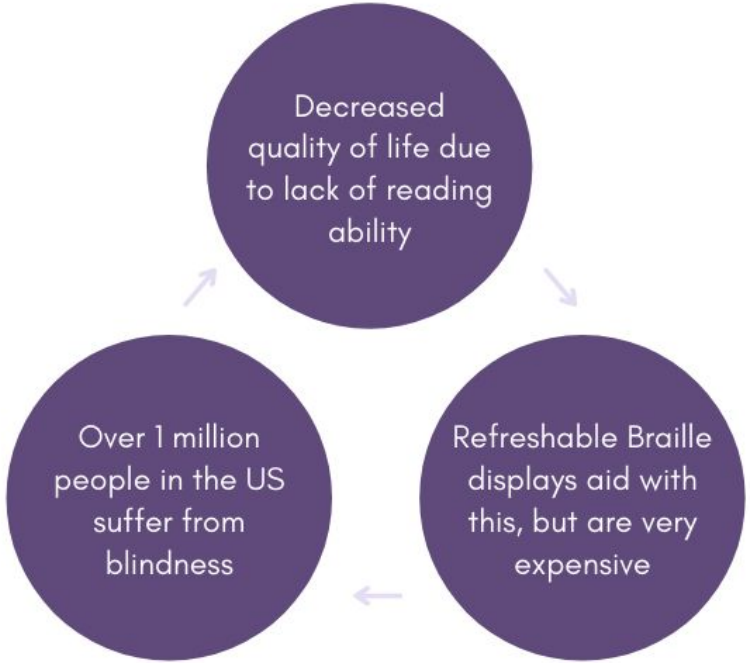
SBraille: A Novel Method of Reading Braille – Acceptance and Delivery Review

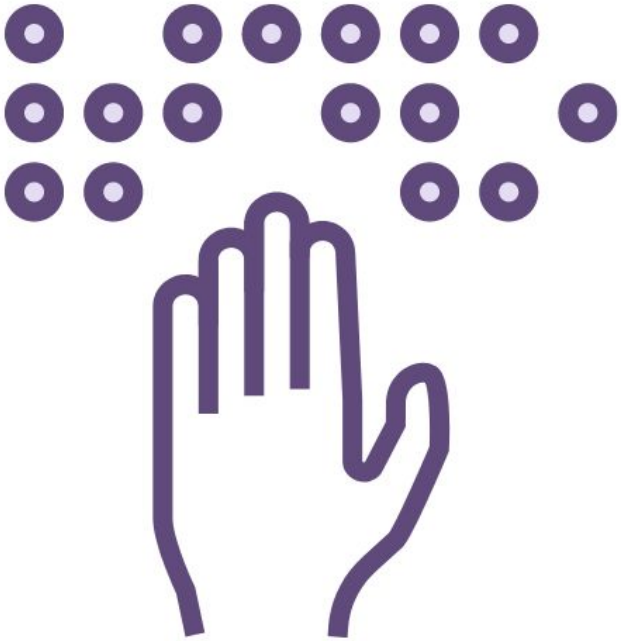
By: Shayaan Chaudhary, Kazuya Erdos, Arnav Mankad, Karthik Seetharaman

STEM2 - Ad Astra

5/26/2021

SBRAILLE





THE PROBLEM STATEMENT

General Requirements

Summary

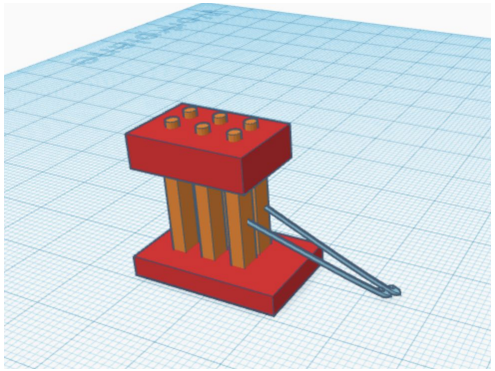
- Converts English to Braille
- Displays Braille Reliably
- Costs < $100
- Encapsulated in a moderately-sized frame

#	Requirement Type	Requirement	Level (1, 2, or 3)
1	Functional	A program to convert English to Grade 1 Braille	1
2	Functional	A method of displaying / producing braille (physical, vibration)	1
3	Functional	A link between the program and braille (hardware and software)	1
4	Functional	Vibration Design: Discernible vibrations (user can tell which nodes are vibrating)	1
5	Physical	Standard formatting: 2x3 grid with varying spacing	1
6	Cost	Unit cost: $100 / unit	1
7	Documentation	Documentation / manual written in a blind user-friendly form	1
8	User	User has a relatively keen sense of touch	1
9	User	User has access to compatible device (computer, tablet, etc.)	1
10	Physical	Device weights under 2 pounds	1
11	Physical	Device fits on a standard tabletop	1
12	Cost	Unit cost: $30 / unit	2
13	Functional	Device takes input from a keyboard	2
14	Functional	Device reads input from a text document	2
15	Functional	Device operates without an internet connection	2
16	Physical	Device fits inside a square foot	2
17	Physical	All electronics are contained within device	2
18	Functional	Device reads data from formatted sources, converting to input	3
19	Functional	Device reads and translates non-dictionary words	3
20	Functional	Device takes digital input text wirelessly	3
21	Functional	Device scans input text from non-digital sources (paper, wall signs, etc.)	3
22	Functional	Device translates text into Grade 2 Braille	3

PDR Minor Designs

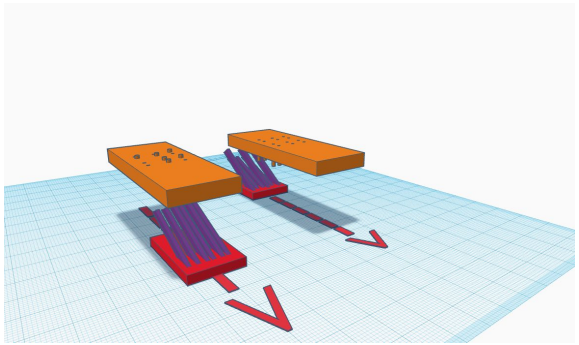
Solenoid-Based Design

- Grid of six moving pins attached to solenoids
- Pins move up and down to create Braille characters



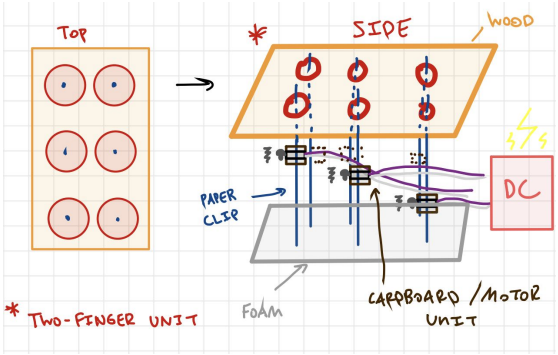
Slider-Based Design

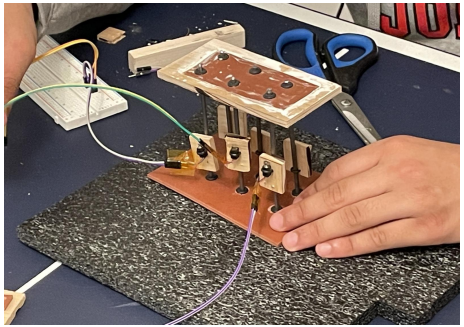
- Series of six-pin grids and a motor moving underneath
- The motor pushes pins up as it goes across to create Braille characters

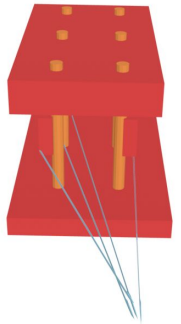


PDR Major Design

- Grid of six rods that vibrate in different patterns to represent Braille characters
- Each vibrating motor wired to breadboard in order to vibrate separately
- Vibrations isolated with rubber sheet and grommets
- Pins suspended in foam sheet at bottom

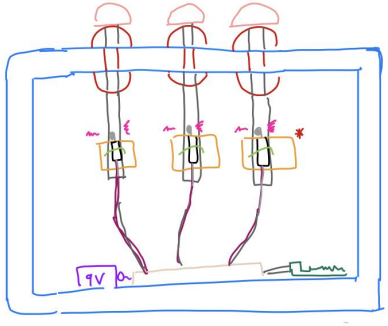


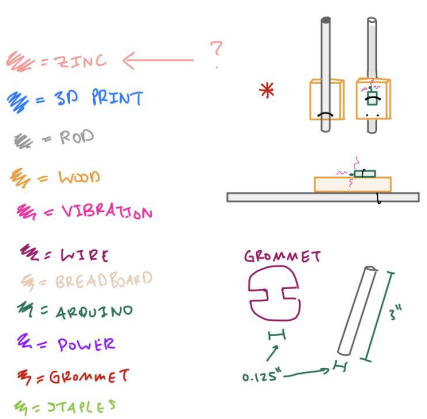


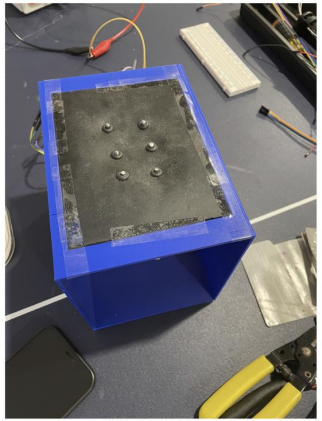


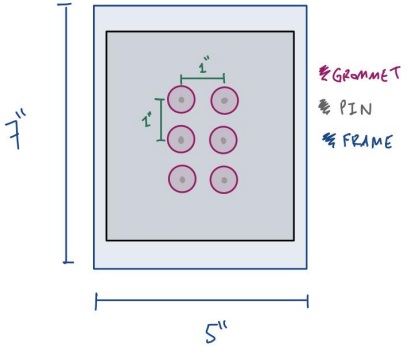
CDR Design

- Grid of six vibrating rods hanging freely from top
- Enclosed in 3D-printed box
- Rubber sheets and grommets used for vibration isolation
- Motors connected to Arduino to vibrate independently







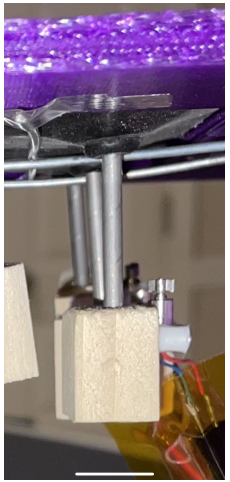


CDR Software

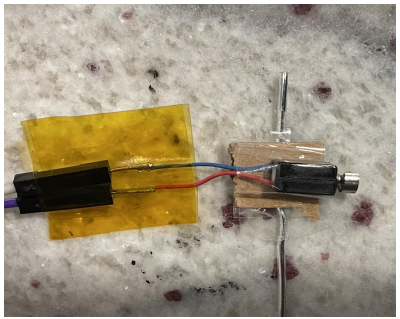
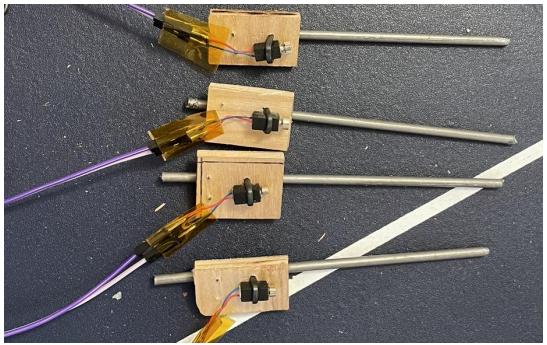
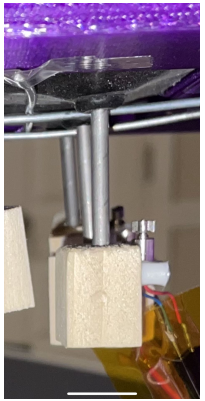
- CDR design featured hardcoded words into the Arduino using GPIO
- Words could not be inputted – entire program had to be changed to read different words

```
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(BOTTOM_LEFT_PIN, OUTPUT);
  pinMode(MIDDLE_LEFT_PIN, OUTPUT);
  pinMode(TOP_LEFT_PIN, OUTPUT);
  pinMode(BOTTOM_RIGHT_PIN, OUTPUT);
  pinMode(MIDDLE_RIGHT_PIN, OUTPUT);
  pinMode(TOP_RIGHT_PIN, OUTPUT);

}
// the loop function runs over and over again forever
void loop() {
      for (int i = 0; i < s.length(); i++) {
        BinaryToOutput(CharToBinary(s.charAt(i)));
      }
      delay(10000);
}

  digitalWrite(BOTTOM_LEFT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(MIDDLE_LEFT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(TOP_LEFT_PIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(BOTTOM_RIGHT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(MIDDLE_RIGHT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(TOP_RIGHT_PIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(3000);    // wait 3 seconds

  digitalWrite(BOTTOM_LEFT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(MIDDLE_LEFT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(TOP_LEFT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(BOTTOM_RIGHT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(MIDDLE_RIGHT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(TOP_RIGHT_PIN, LOW);   // turn the LED on (HIGH is the voltage level)
  delay(3000);
```

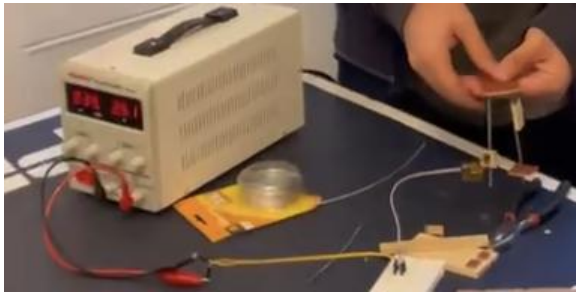
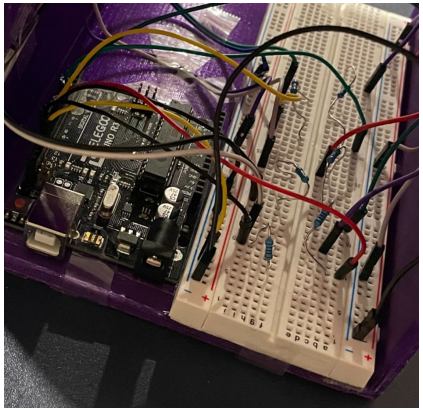
Device Progression (Outside)

Walls	Rubber Sheet	Rods
Walls of device thickened from 1/16 in. to 3/16 in.	Rubber sheets decreased in thickness to make grommets sit flush within the sheet	Rods with motors attached reduced in size by a factor of three
		

Device Progression (Motors)

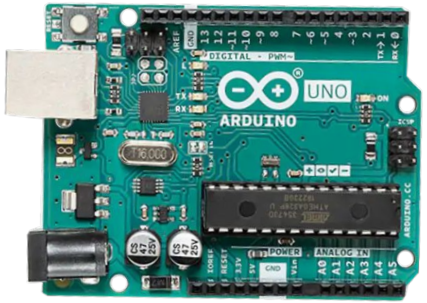
Prototype 1	Prototype 2	Current Prototype
Motors were attached to paper clips using cardboard sheets causing uneven vibrations and motors to fall off frequently	Motors were attached to flat wood planks which were attached to steel rods using staples	Motors attached to wood blocks by cable wire clips in which rods are directly suspended
		

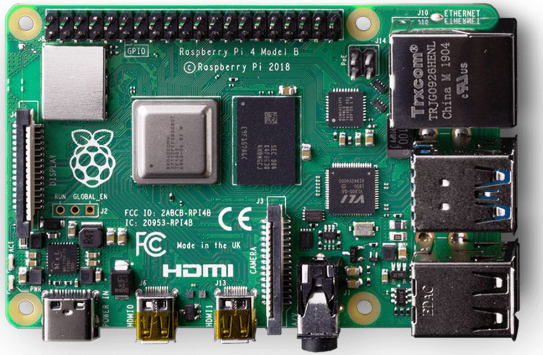
Device Progression (Wiring)

Prototype 1	Prototype 2	Current Prototype
Motors were connected to a DC power supply through a breadboard to simply turn on and off	Motors were routed to the 3.3 V output pins on an arduino to switch individual motors on/off	Motors are wired to individual GPIO pins on a Raspberry Pi and 100 ohm resistance was added
		

Software Progression (Pt. 1)

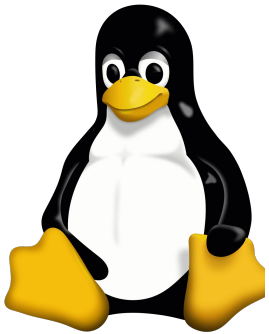
- Switched from Arduino UNO R3 to Raspberry Pi 3
  - Runs Python for open-source Braille translators
  - More storage and power than Arduino
  - Allows for text input rather than hard-coded words





Software Progression (Pt. 2)

- Open-source Braille translator ported onto the Pi through GitHub
- Code written in Linux and Python to perform translation functions
- Function implemented to stop the device from outputting vibration at any given time



Current Software

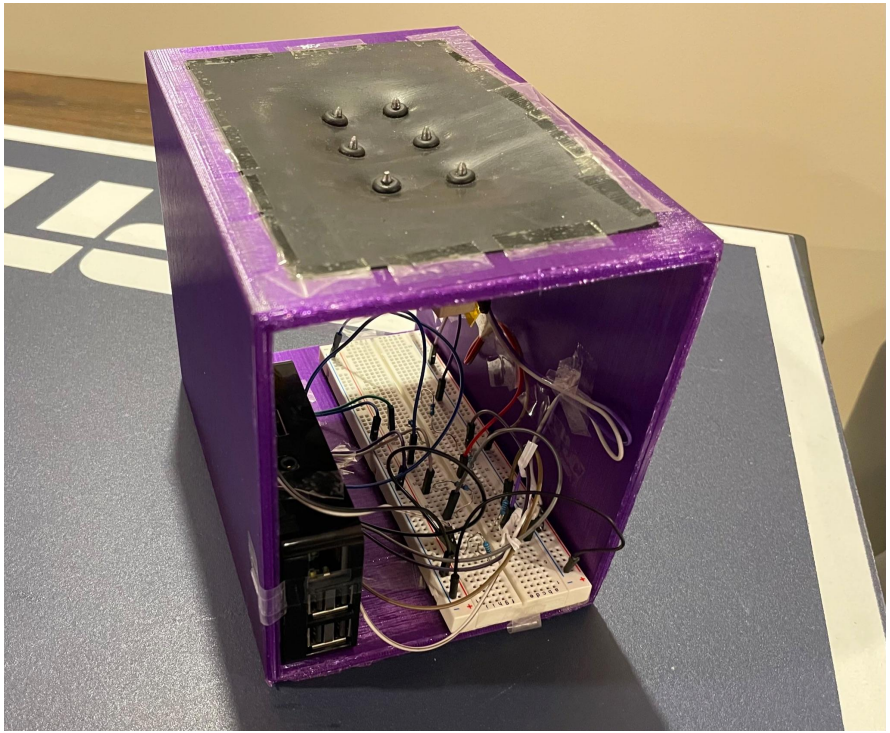
- Current design uses LibLouis, an open-source Grade 2 Braille translator to translate input
- Linux and Python scripts are used to convert inputted English text into Braille characters that can be felt through vibrations
  - Input is translated to Braille ASCII through Linux scripts and written to file
  - Text in file is read to Python program and is converted into vibrations on the device

```
#!/bin/bash
> BrailleOutputs.txt
sh BrailleRead.sh | tee -a /home/pi/python/BrailleOutputs.txt
echo finished
```

```
import pexpect
import RPi.GPIO as GPIO
from pathlib import Path
import time
s = input() #input text

proc = pexpect.spawn('sh braillewrite.sh') #run Linux script
proc.expect('')
proc.sendline(s) #send inputted text

proc.expect('finished') #confirm script is finished

txt = Path('brailleOutputs.txt').read_text() #read output from script
txt = txt[:-1] #cut off last character of txt
```

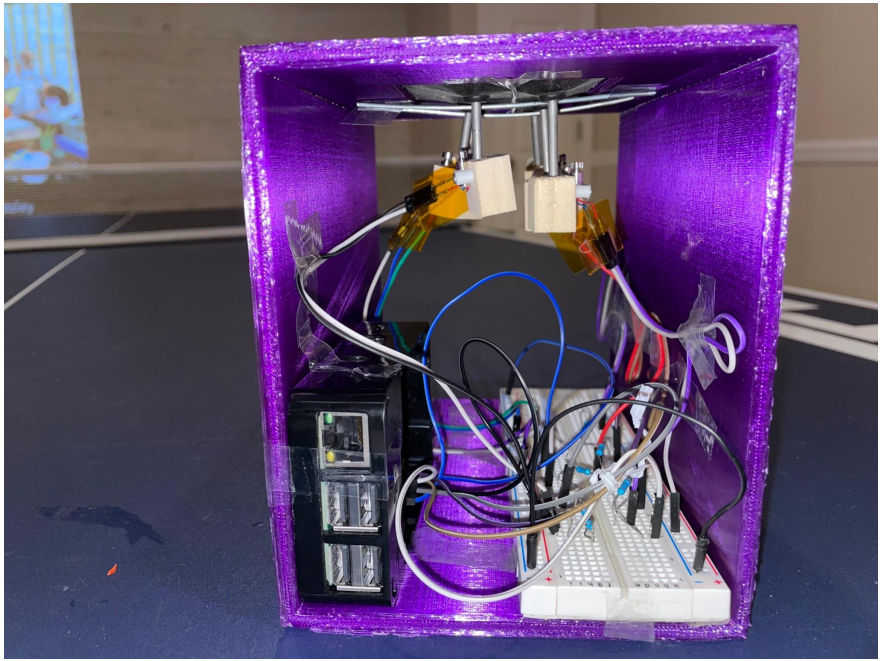
Current Device (Outside)

- Current design is housed in a 3D printed frame of dimensions 5”x7”x6”
- Rubber sheet on top to dampen vibrations
- Rounded pins on top
  - Surrounded by rubber grommet to isolate vibration
- Fully-functional MVP



Current Device (Wiring)

- Raspberry Pi added inside frame of device
  - Wires plugged into breadboard and Raspberry Pi GPIO
- Microcontroller -> sideways onto wall
  - Compactness/shortened port distance
- 100 ohm resistance between GPIO pins/motors
  - Limits current spikes or motor burnout
- Motor wires soldered to longer breadboard wires



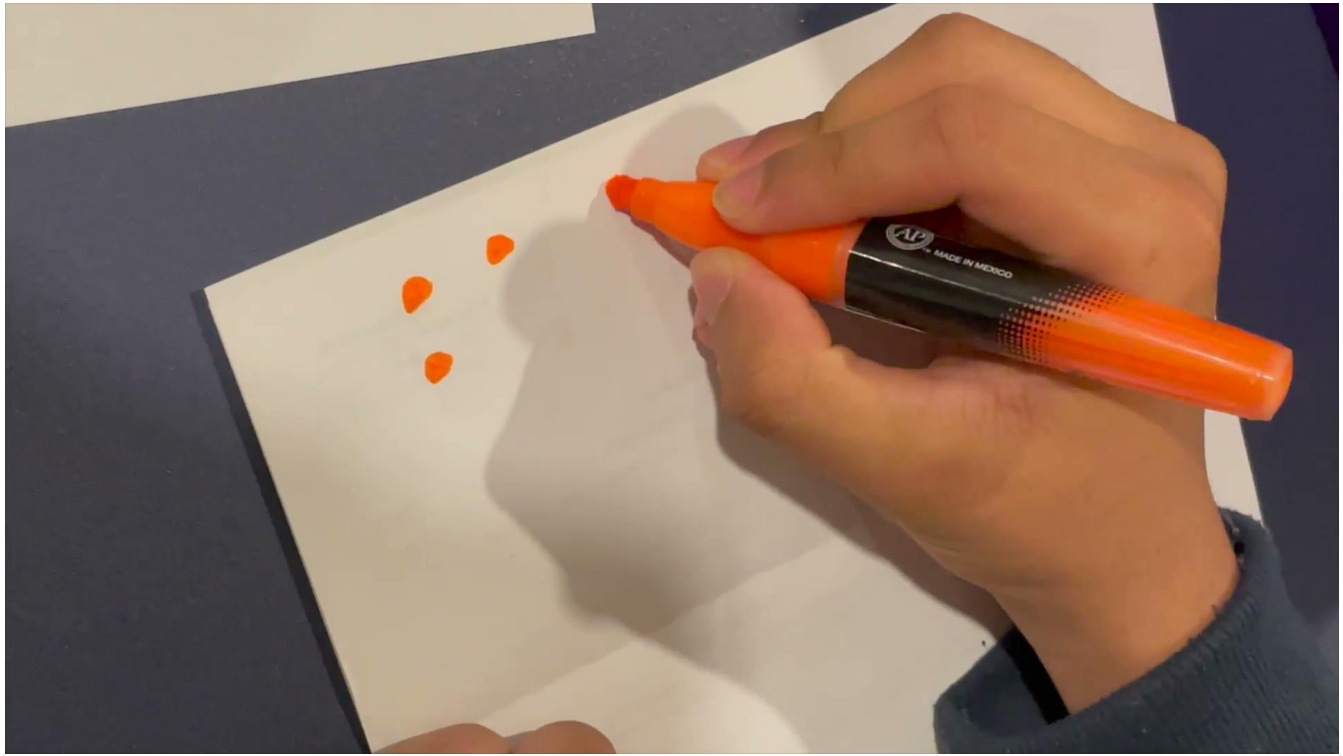
Requirements Addressed by SBraille

Summary

- Most level 1 criteria addressed
- Device does not read non-digital text or take in wireless input
- Device does not offer the ability to read non-dictionary text such as the presence of scroll bars, images, etc

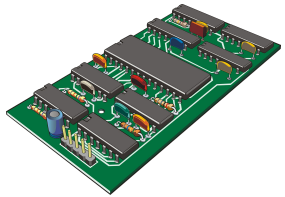
#	Requirement Type	Requirement	Level (1, 2, or 3)	Vibrating Module
1	Functional	A program to convert English to Grade 1 Braille	1	Yes
2	Functional	A method of displaying / producing braille (physical, vibration)	1	Yes
3	Functional	A link between the program and braille (hardware and software)	1	Yes
4	Functional	Vibration Design: Discernible vibrations (user can tell which nodes are vibrating)	1	Yes
5	Physical	Standard formatting: 2x3 grid with varying spacing	1	Yes
6	Cost	Unit cost: $100 / unit	1	Yes
7	Documentation	Documentation / manual written in a blind user-friendly form	1	Yes
8	User	User has a relatively keen sense of touch	1	Yes
9	User	User has access to compatible device (computer, tablet, etc.)	1	Maybe
10	Physical	Device weights under 2 pounds	1	Yes
11	Physical	Device fits on a standard tabletop	1	Yes
12	Cost	Unit cost: $30 / unit	2	No
13	Functional	Device takes input from a keyboard	2	Yes
14	Functional	Device reads input from a text document	2	Yes
15	Functional	Device operates without an internet connection	2	Yes
16	Physical	Device fits inside a square foot	2	Yes
17	Physical	All electronics are contained within device	2	Yes
18	Functional	Device reads data from formatted sources, converting to input	3	Yes
19	Functional	Device reads and translates non-dictionary words	3	No
20	Functional	Device takes digital input text wirelessly	3	No
21	Functional	Device scans input text from non-digital sources (paper, wall signs, etc.)	3	No
22	Functional	Device translates text into Grade 2 Braille	3	Yes

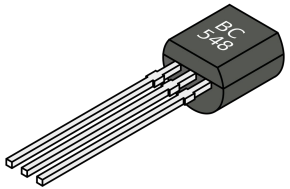
Demo

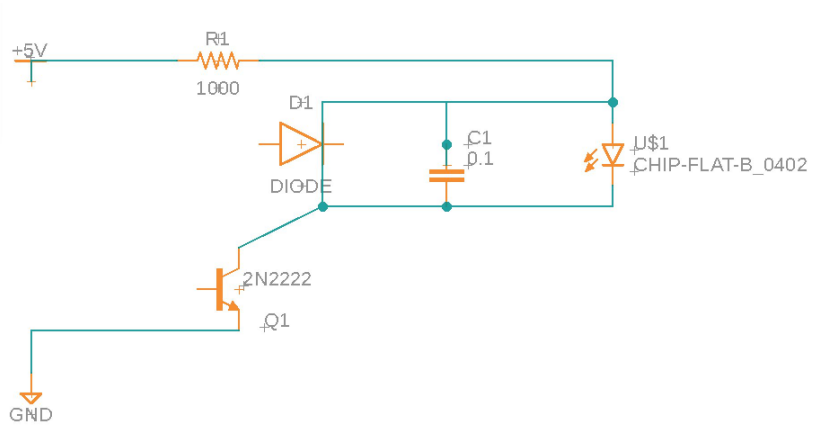


Future Work

- Remove manual input/incorporate screen reader
- Replace breadboard with PCB
  - Add transistors to device
- Add support for foreign languages
- Increase stability of pins and strength of vibrations
- Decrease size of device







References

Gutknecht, K. S. (1980). OPTACON - A TOOL FOR INDEPENDENCE. *American Education Journal, 16*(1), 8-13.

Refreshable Braille Displays | American Foundation for the Blind. (n.d.). Retrieved 4 April 2021, from https://www.afb.org/node/16207/refreshable-braille-displays

Rein, D. B., Wirth, K. E., Johnson, C. A., & Lee, P. P. (2007). Estimating quality-adjusted life year losses associated with visual field deficits using methodological approaches. *Ophthalmic Epidemiology*, *14*(4), 258–264. doi:10.1080/01658100701473267

The Braille Bookstore. (n.d.). Retrieved April 05, 2021, from http://www.braillebookstore.com/Braille-Bookstore

Thank You!
Any questions?