Dot.

⠠⠙⠕⠞⠲

DIVERSIFYING OPTICAL TECHNOLOGY (DOT.)

OWNER’S MANUAL

Table of Contents

1.0: Product Overview....................3
2.0: Product Elements....................3
2.1: Components....................4
2.2: Assembly/Recreation Instructions....................4
2.2.1: OCR, Text-to-Braille Conversion, Display on Web Server....................4
2.2.2: PCB Assembly....................5
2.2.3: Connecting and Testing the Arduino Nano....................6
3.0: Device Usage....................6
3.1: Initial Setup....................6
3.1.1: OCR and Text-to-Braille....................6
3.1.2: Arduino, PCB, and LED....................6
4.0: Safety Warnings....................7

1.0: Product Overview

Problem Statement: Visually impaired people (VIPs) encounter significant challenges in accessing written information. Learning Braille from a young age aids with literacy, enhancing future life implications for VIPs. However, there is a lack of interactive educational devices for younger VIPs.

Engineering Goals: The goal is to design an assistive device for VIPs that uses optical character recognition (OCR) to take pictures of text in the environment. The device translates text from the image into Braille configurations and provides a tactile medium for the VIP to read the translated Braille.

Braille Literacy
In daily interactions, sight is critical, as it is the most powerful method of accessing information in our environment. According to the World Health Organization (WHO), around 40 million people in the world are blind, while another 250 million have some form of visual impairment (The International Agency for the Prevention of Blindness, 2022). For Visually Impaired People (VIPs), these obstacles require modern inventive modifications so that those with disabilities can navigate and thrive in a society that often overlooks their needs.

Through Braille, blind people can access written information without relying on sighted readers. While fewer than 10 percent of 1.3 million people who are legally blind are Braille readers, access to Braille enhances knowledge accessibility, and autonomy (National Federation of the Blind, 2009). Adolescents comprise a large number of the visually impaired community. Despite the well-documented importance of Braille literacy, the Braille literacy of students with visual impairments stands at around 10% (Braille Works, 2023). Braille literacy has major implications on future life outcomes for a VIP, as literacy affects an individuals health, lifespan, wellness, and socio-economic status.

This device associated with this manual is a tactile refreshable Braille display, which uses Optical Character Recognition (OCR) to extract text characters from images to convert them into Braille configurations using an electromechanical refreshable Braille technology.

2.0: Product Elements

Product Dimensions: 6 x 3.3 x 2.5 in

2.1: Components

Part	Item	Amount
Braille Display Module	PLA Filament	N/A
Braille Cells	SLA Filament	N/A
PCB Board	Raspberry Pi	1
	Camera (webcam, Raspberry Pi Cam, etc.)	1
	PCB boards	16
	Pitch Pin Headers	30
Braille Module Mechanism	Adafruit Micro-Lipo Charger for LiPoly Battery with USB Type C Jack	1
	Lithium Ion Polymer Battery - 3.7v 1200mAh	1
	Neodymium Disc Magnets	100
	Ferrite Core	90
Solenoid Winding	Drill Chucks	2
	4mm shaft gearbox DC motor with an encoder	1

2.2: Assembly/Recreation Instructions

Access code at https://github.com/kg250/braille-camera

2.2.1: OCR, Text-to-Braille Conversion, Display on Web Server

Raspberry Pi setup

1. Install Raspbian bookworm
2. git clone braille-camera repository
3. Create and start python virtual environment

```
mkdir ~/venvs
python3 -m venv ~/venvs/braille-camera
source  ~/venvs/braille-camera/bin/activate
pip install -r requirements.txt
```

Demo

1. Connect to at2024, Password: *****
2. Go to http://jimbo.local:5000/

Development with VS Code

1. Install Remote-SSH Extension in VS Code
2. Connect to at2024
3. Open Terminal
4. Run ssh-keygen
5. Run `ssh-copy-id jimbo@jimbo.local` with password: jumbo
6. From VS Code, View/Command Palette "Add New SSH Host":
   ```
   ssh jimbo@jimbo.local
   ```
7. Open Folder.
8. Source code is in ~/braille-camera/src
9. Restarting the server after making modifications
   1. Launch Terminal/New Terminal in VS Code
   2. sudo systemctl status braille-camera # check if service is running
   3. sudo systemctl restart braille-camera # restart the service
10. Shutdown Raspberry PI using `sudo shutdown -h now`

2.2.2: PCB Assembly

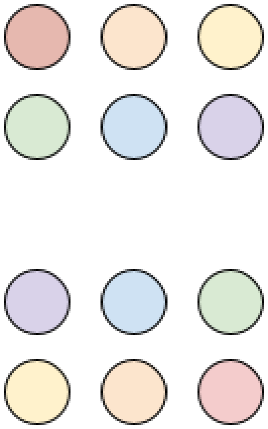
Following the locations of each part on this spreadsheet, solder the following parts to the PCB board in this order:

1. Diode arrays
2. Decoders
3. Voltage regulator
4. Transistor array
5. Capacitors and resistors
6. Arduino nano headers

In addition, make sure to use solder flux when soldering the decoders to avoid any short-circuiting, as they are in very close proximity. In addition, make sure that decoders are positioned in the correct direction (circle on top of the decoder should match the circle on the PCB or white line on the side of the white box where it is placed). In addition, make a circuit at the "Jump to power from Arduino" - solder a pin header that has a connected top to this portion of the PCB.

2.2.3: Connecting and Testing the Arduino Nano

1. Download Arduino IDE.
2. Access Testing code at https://github.com/kg250/braille-camera/tree/master/arduino_testing
3. Upload this code to the Arduino Nano using a micro-USB cable.
4. To test if soldering and code works properly, connect a LED to the appropriate pin headers, according to the diagram below (use the same color for each wire on the LED):



5. If the soldering was done correctly and the code correctly uploaded to the Arduino Nano, the LED light will flash on and off periodically, about every 1.5 seconds.

3.0: Device Usage

3.1: Initial Setup

3.1.1: OCR and Text-to-Braille

1. Connect to at2024, Password: *****
2. Go to http://jimbo.local:5000/
3. View the camera feed, the text being pulled from the feed by Tesseract, and the text converted to Braille Unicode characters.
4. To shutdown Raspberry PI, enter `sudo shutdown -h now` in terminal

3.1.2: Arduino, PCB, and LED

1. Connect Arduino to power source.
2. Run code and verify LED action.

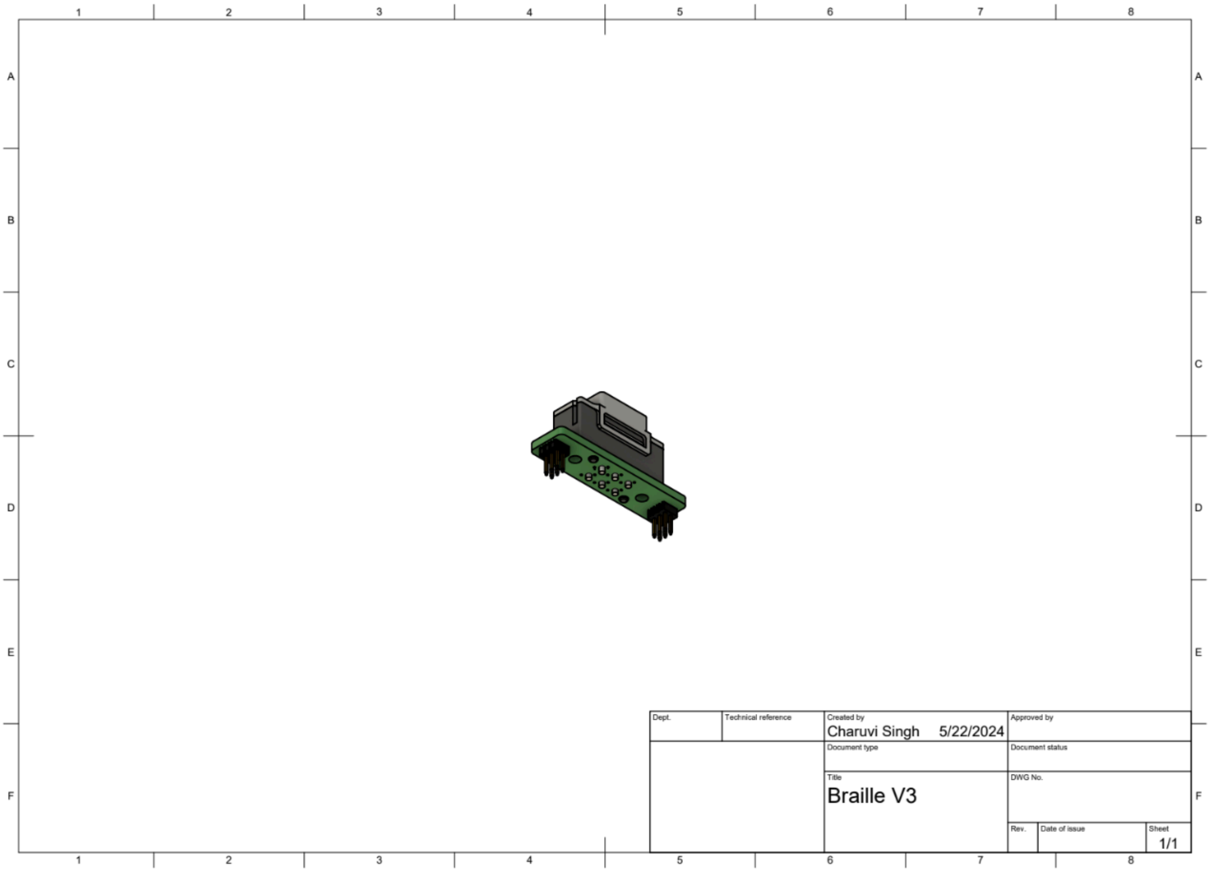
4.0: Safety Warnings

- Do not immerse device in water
- This device is a prototype; don’t rely on this device in high risk environments

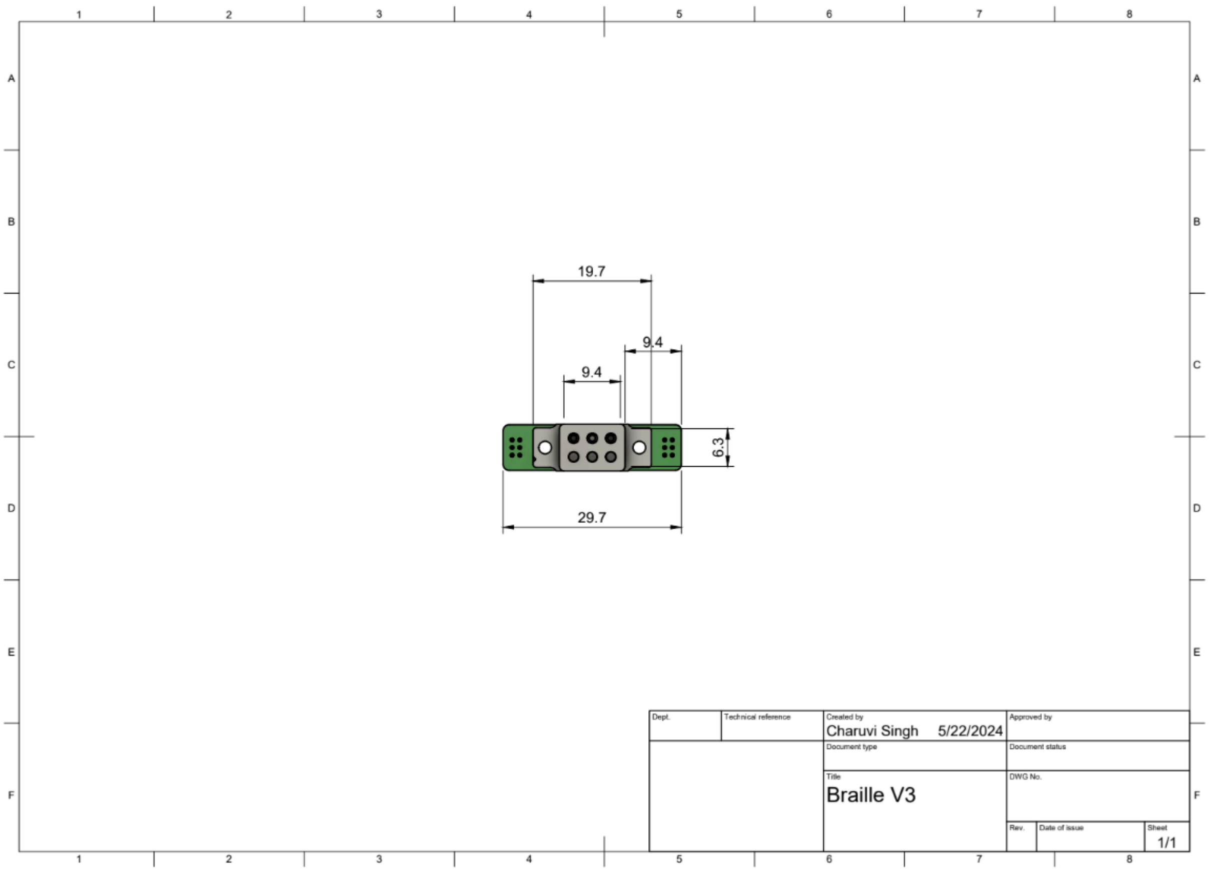
DOT. is not liable for any misuse or accidents resulting from failure to follow these instructions.

5.0: Appendix

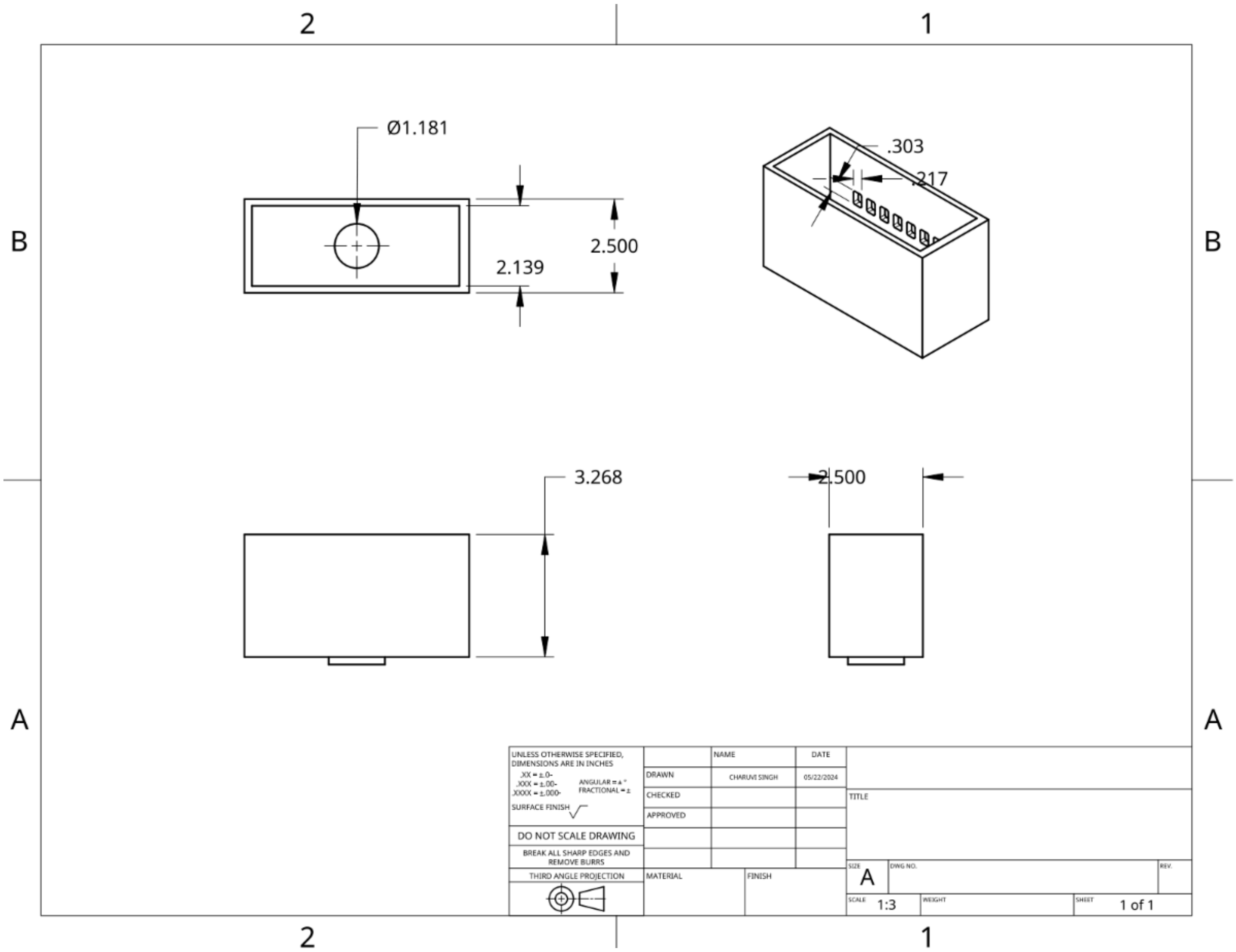
5.1: CAD Files for Braille Cell



Side View of Braille Cell



Top View of Braille Cell



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES
.XX = ±.0-
.XXX = ±.00-
.XXXX = ±.000-
ANGULAR = ± °
FRACTIONAL = ±
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION

	NAME	DATE
DRAWN	CHARUVI SINGH	05/22/2024
CHECKED		
APPROVED		

MATERIAL
FINISH
TITLE
SIZE A
DWG NO.
REV.
SCALE 1:3
WEIGHT
SHEET 1 of 1

CAD for Braille Module containing Braille Cells and PCB