# Braille Labels
## OPENSCAD TEMPLATE GUIDE

## Overview

The open source software OpenSCAD is used to create custom Braille labels. There is an OpenSCAD file (.scad) included on the GitHub page which is already set up to create Braille label STL's based off of the inputted characters. There is no modeling, programming, or prior CAD knowledge necessary when using this OpenSCAD template. The steps to create custom label STL's can be found below.

**Note:** There is also a video included on the GitHub page that walks you through the same process.

**Note**: Do not resize the Braille as it may render it unreadable.

## Creating Braille Labels from Template:

### Step 1: Downloading OpenSCAD

- The download link for OpenSCAD can be found here: https://openscad.org/downloads.html
- The software is free and available on Windows, MacOS, and Linux

### Step 2: Open the OpenSCAD Braille Template File

- Once OpenSCAD is downloaded, locate the "Braille_Template_Module_MMC.scad" file on the GitHub page and double click it to open the file in OpenSCAD.

### Step 3: Finding Available Characters, numbers, signs, and contractions

- On the left side of the screen, there is a text pane where the code that creates the template is located.
- When you scroll down you will come across a large section that has a lot of "else if" statements like shown in the below figure.

```
196     } else if (char ==
        "CH") {
197         letter([
198             1,0,
199             0,0,
200             0,1
201         ]);
```

- Each character, number, sign, or contraction is shown in quotations and below that a series of 1's and 0's are used to control the braille dots. A 1 adds a braille dot in that part of the cell. A 0 has no dot.

- Scrolling through the text on the left side you will be able to see all of the available characters that are available in this document. There is potential to add characters. This is described later in this document.

## Step 4: Entering Text

- In the text pane on the left side of the screen, you will see all the programming used to build the tool. At the bottom of the text field (line 662-665), you will find an area to edit the braille shown below:

```
chars =
["cap", "TH", "space", "P", "R",
    "IN", "T", "S", "space", "B",
     "R", "L", "!"];
char_count = 13;
```

- All characters must be enclosed in quotation marks. Letters and contractions must be capitalized. Commas are used before any new cell added. Note that there should be no comma on the last character before the square bracket is closed.
- Enter the number of characters you're using after "char_count = " . As shown in the example above there is 13 characters in the desired input and therefore the user has input the character input to 13. Further examples can be seen below:

### Examples

This prints braille!

```
chars =
["cap", "TH", "space", "P", "R", "IN", "T", "S",
"space", "B", "R", "L", "!"];
char_count = 13;
```

Mill St.

```
chars =
["cap", "M", "I", "L", "L", "space", "cap", "S", "T", "."];
char_count = 10;
```

3×4 = 12

```
chars =
["#", "n3", "dot4", "CH", "n4", "space", "dot46", "K",
"space", "#", "n1", "n2"];
char_count = 12;
```

## Step 5: Creating the 3D Printable Braille Model

- Once step 4 is complete and you are happy with the text you have entered, click on the "**Preview**" button at the top of the text pane to view your model. This looks like a cube with two arrows going to the right.

- If everything looks correct and the model loads without error, click the "**Render**" button to prepare your model. If you're not ready, make the necessary edits and then hit "Preview" again.

- After rendering the model, you're ready to export the file. Click on the "**Export as STL**" button and save the file. This file can now be used to 3D print. Please see 3D Printing Guide document for tips on printing Braille.


## Adding New Characters to the Current Template:

If you cannot find the letter, number, or contraction you are looking for in the template and want to create your own, please follow this method:

- Copy one of the already defined characters starting with the line that says "module" until the next semicolon and paste it after the code for an existing character. For example, you would copy a section like the one below:

```
39 module braille_char(
       char) {
40     if (char == "A") {
41         letter([
42             1,0,
43             0,0,
44             0,0
45         ]);
```

- Edit the character name in the quotes, and alter the braille dots by using 1's and 0's. 1's will be the location where the Braille dots are and 0's will be empty space.