



Custom Fonts for CircuitPython Displays

Created by Ruiz Brothers



Last updated on 2019-03-04 12:19:05 AM UTC

Overview



More Fonts

Are you looking to display new fonts on your PyPortal? You can use just about any font on your computer or downloaded from the internet. This guide will walk you through generating bitmap fonts using the [FontForge open-source \(https://adafru.it/DZk\)](https://adafru.it/DZk) project.

Why Bitmaps?

PyPortal uses the [CircuitPython Bitmap Font Library \(https://adafru.it/DZI\)](https://adafru.it/DZI) to render "live" text on the display. A bitmap font stores each character as an array of pixels. Bitmap fonts are simply groups of images. For each variant of the font, there is a complete set of images, with each set containing an image for each character.

Computers, on the other hand, use variable size 'TrueType' or 'Postscript' fonts, where there's a mathematical algorithm that defines each character, so it can be drawn at *any* size.

Font Forging

This is where FontForge comes into play. FontForge is an open-source font editor for Windows, Mac OS and GNU+Linux. It features tools for converting existing fonts into different font formats.



Getting Started with FontForge

Head on over to the [fontforge page \(https://adafru.it/DZm\)](https://adafru.it/DZm) and download the app for your platform. You can choose to donate, subscribe via email or simply click the "**Subscribe/Confirm and Download**" button (no need to enter an email). Follow along with the detailed installation guide to get setup with FontForge.

<https://adafru.it/DZn>

<https://adafru.it/DZn>

Where Do I Get Fonts?

Here's a list of some neat places to obtain some fresh fonts.

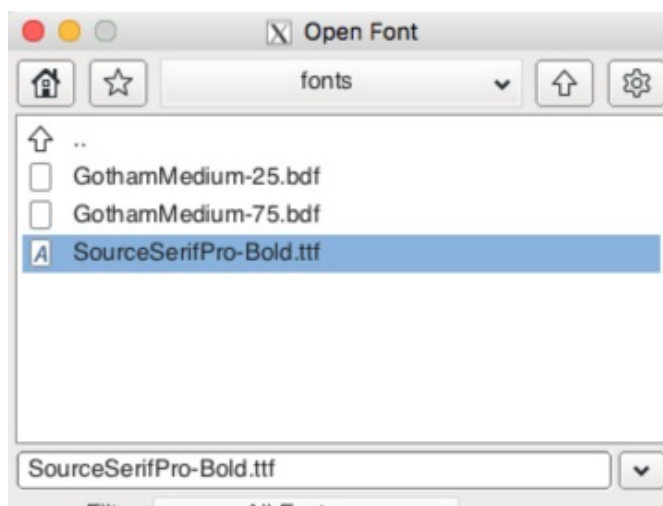
- [Font Squirrel \(https://adafru.it/DZo\)](https://adafru.it/DZo)
- [Google Fonts \(https://adafru.it/DZp\)](https://adafru.it/DZp)
- [Adobe Fonts \(https://adafru.it/DZq\)](https://adafru.it/DZq)
- [DaFont \(https://adafru.it/DZr\)](https://adafru.it/DZr)
- [Font Library \(https://adafru.it/DZs\)](https://adafru.it/DZs)

Use FontForge



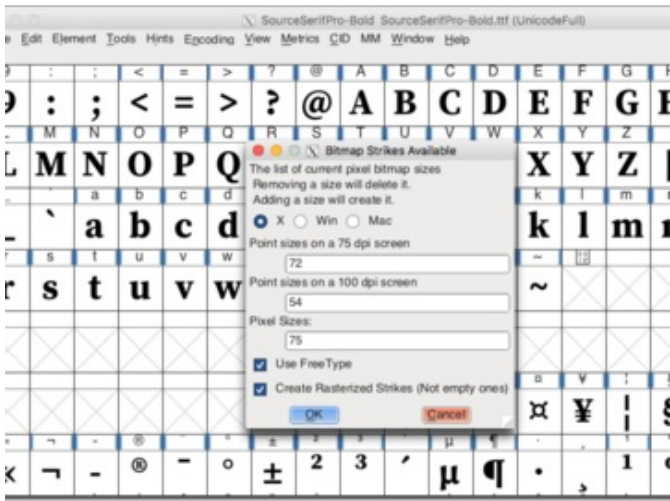
Demo Walkthrough

In this example, we're going to convert a **.TTF** (TrueType Format) into a **.BDF** (Bitmap Distribution Format). I'm using an open licensed font downloaded from Google Font, [Source Serif Pro](https://adafru.it/DZt) (<https://adafru.it/DZt>).



Open Font

Use the **file** menu and choose **Open Font** from the list. Navigate to a directory where your desired font resides. Select the font and open it.



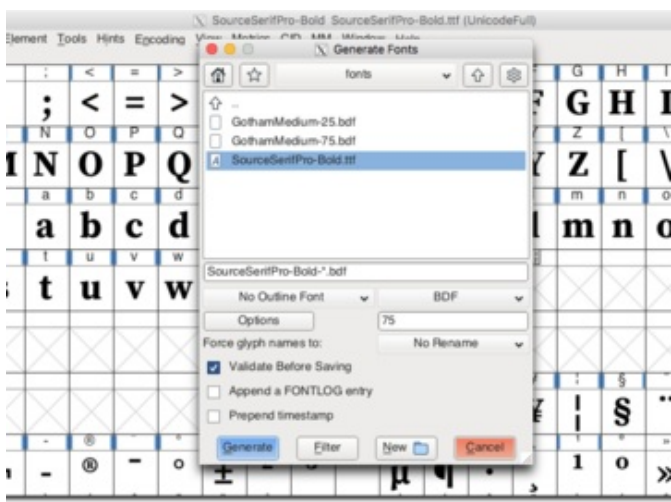
Set Font Size

From the **element** menu, select **Bitmap Strikes Available**. In this dialog, you will need to specify how large you want your font to be. The font size is fixed with Bitmap fonts, so if you want to use different sizes, you'll need to make separate files.



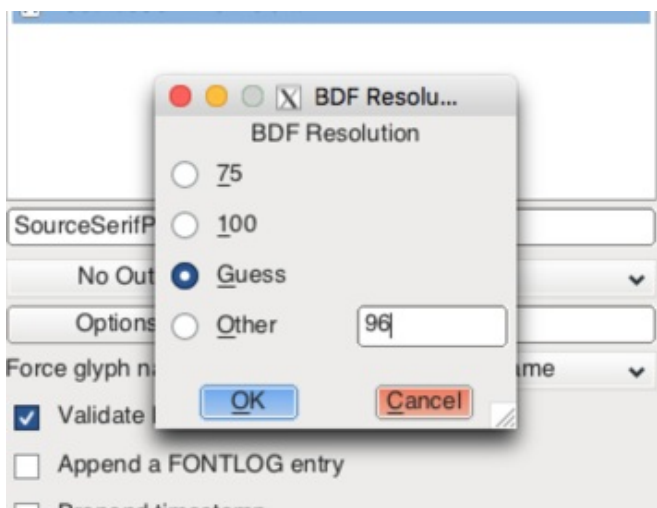
Generate Bits

From the **element** menu, select **Generate Bitmap**. Similar to the previous dialog, enter the font size of your liking. You can make it smaller here. Be aware, values too small will not generate BDF's.



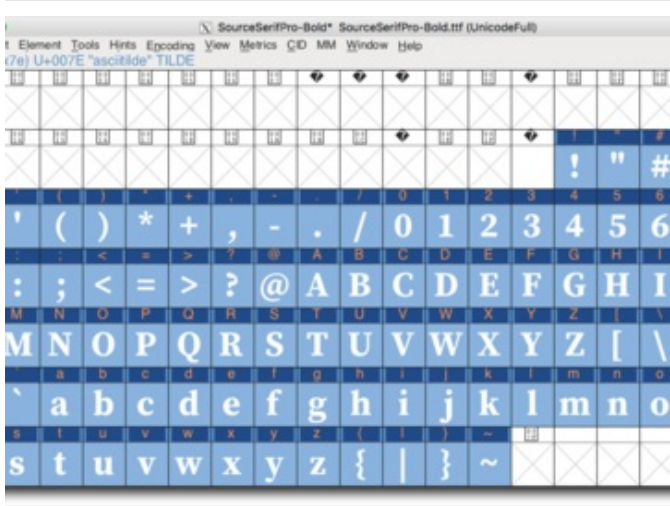
Export Converted Font

From the **file** menu, select **Generate Fonts**. In the dialog, select **No Outline Font** and **BDF** from the dropdown options. Use the navigation UI to save the file in your directory of choice. Click the **generate** button to save the file.



BDF Resolution

This dialog menu will pop up after clicking generate. You can choose one of the options from the list. If you'd like a different font size, you can enter that in the **Other** labeled input box. Click **OK** to save it!



Optimize File Size

If you take a look at the file size of the .bdf, it's roughly around 900K – That can be a bit larger than needed, especially if you plan to store a lot of image and sound assets. In cases where you need to save on every byte, you can optimize the file size of your fonts by selecting only the characters you want to use. If you scroll through the full list of glyphs, you'll see there's extra special characters – A whole bunch of them! If you don't need them in your project, just select the basic set of upper/lower and alphanumeric characters. You can click + hold and drag to make selections easier. With them selected, use the **Regenerate Bitmap** option under the **element** menu to reprocess the glyphs. Then, select **generate font** from the **file** menu and save it out.

Font Colors

The color of the fonts can be setup in your code. The CircuitPython library uses HEX color codes. This is similar to web color pickers but formatted slightly different. Most HEX color pickers use a hashtag in the front of the value, like, **#000000**. In CircuitPython, instead of a hashtag, **0x** is used. Here's a few examples.

- Black = 0x000000
- White = 0xFFFFFFFF
- Purple = 0x8f42f4