Franz Hansmann
Albrecht-Dürer-Str. 6
D-40764 Langenfeld
☎ ++49 (0)2173-202 7007
🖷 ++49 (0)2173-202 7269
✉ info@net4web.de
🌍 www.net4web.de

Create a graphic for using an OLed display 0.96" with an Arduino

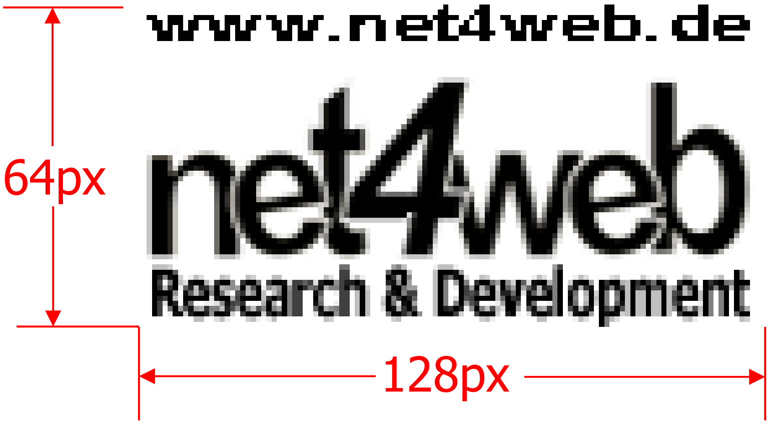
Table of contents

Description:

If you want to use your own graphic as a background on an OLed display in connection with an Arduino board, you must first create a graphic of the appropriate size. As an example, we are listing a 0.96" OLed display with a resolution of 128x64 pixels. If you want to use a display with a different resolution, all you have to do is resize your graphic accordingly.

Create graphic:

You need a black and white graphic with a size of 128x64 pixels. When creating the graphic, you should already consider which areas should later appear in the display color and which areas should remain dark. Do not save the graphic in JPG format but in PNG format. In JPG format, the graphic can be compressed, which can lead to pixels in the wrong place.



In our sample graphic, we left a 2 pixel border around the graphic. That made sense for our logo. But you can use the entire area if you think it's right. In our example, there is a small gap between the logo and the web address. It's right here for one reason only. The OLed displays are also available in two-colors. With the two-color displays, the lower area is blue and the upper area is yellow. There is a gap of one pixel in height between these two color areas. When creating the background graphics for two-color displays, you have to take this into account, of course, otherwise a line will be missing in the image afterwards or a part will be in the "wrong" color. Of course, if you're using a monochrome display, you don't have to worry about that. In our example, however, the motto was: One graphic for all displays.

Convert graphic to Arduino code:
Converting this graphic into a code that the Arduino understands is quick and easy over the internet. To do this, visit the website:

https://diyusthad.com/image2cpp

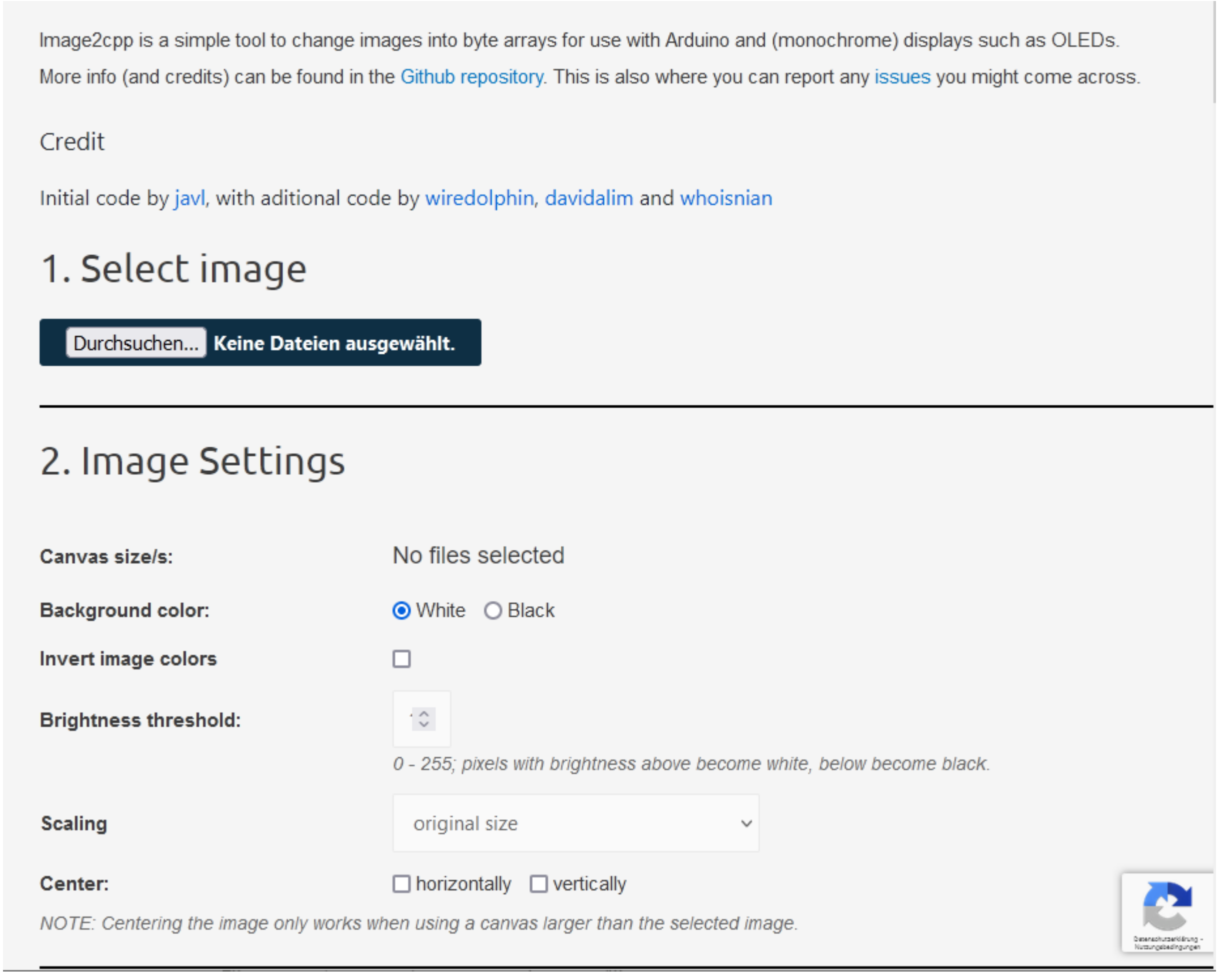


Image2cpp is a simple tool to change images into byte arrays for use with Arduino and (monochrome) displays such as OLEDs.
More info (and credits) can be found in the Github repository. This is also where you can report any issues you might come across.

Credit

Initial code by javl, with aditional code by wiredolphin, davidalim and whoisnian

1. Select image

Durchsuchen... Keine Dateien ausgewählt.

2. Image Settings

Canvas size/s: No files selected

Background color: ● White ○ Black

Invert image colors □

Brightness threshold:

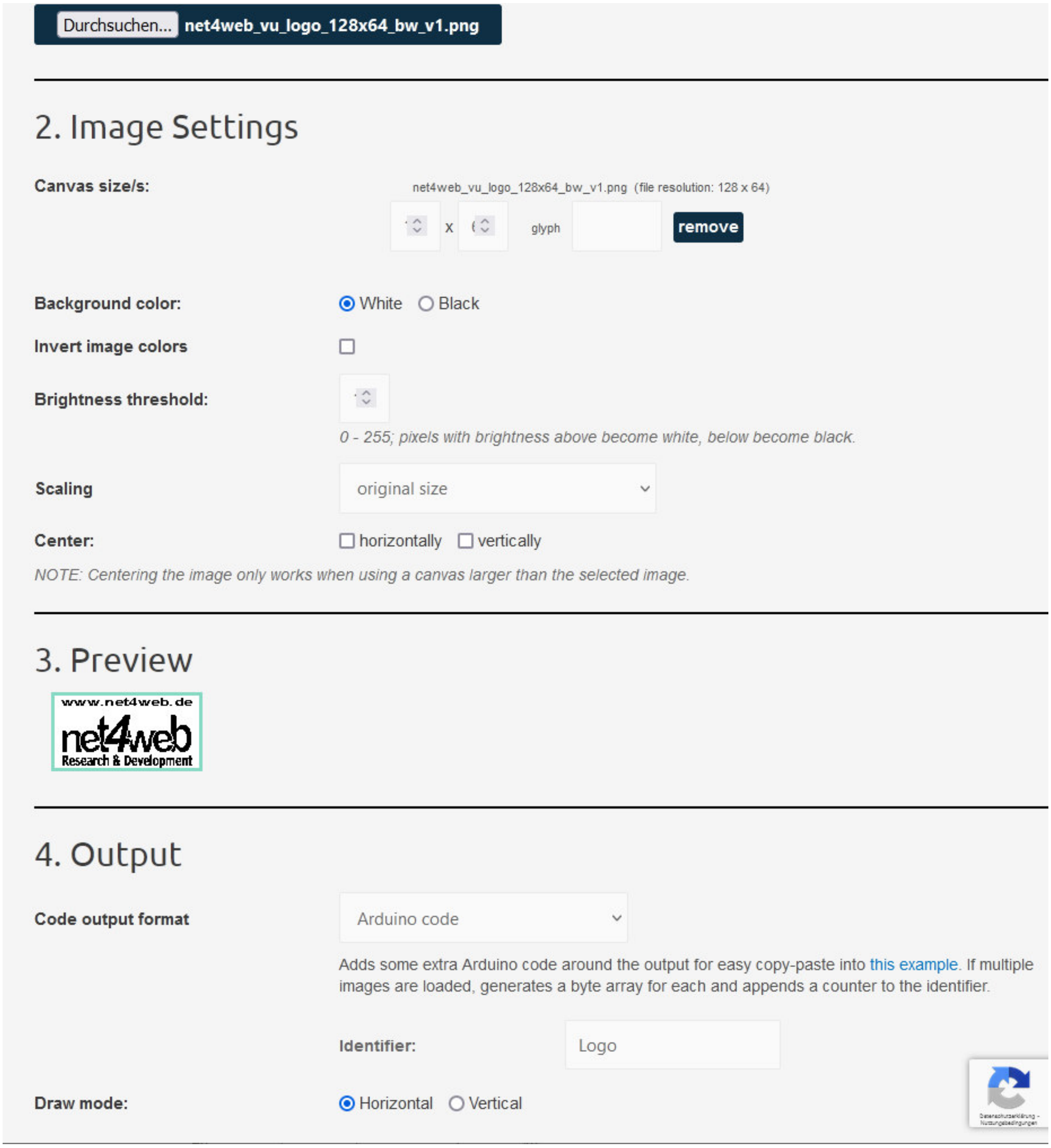
0 - 255; pixels with brightness above become white, below become black.

Scaling original size

Center: □ horizontally □ vertically

NOTE: Centering the image only works when using a canvas larger than the selected image.

Click the "Browse" button and load your prepared graphic.



Durchsuchen... net4web_vu_logo_128x64_bw_v1.png

2. Image Settings

Canvas size/s: net4web_vu_logo_128x64_bw_v1.png (file resolution: 128 x 64)
x glyph **remove**

Background color: ● White ○ Black

Invert image colors □

Brightness threshold:

0 - 255; pixels with brightness above become white, below become black.

Scaling original size

Center: □ horizontally □ vertically

NOTE: Centering the image only works when using a canvas larger than the selected image.

3. Preview

4. Output

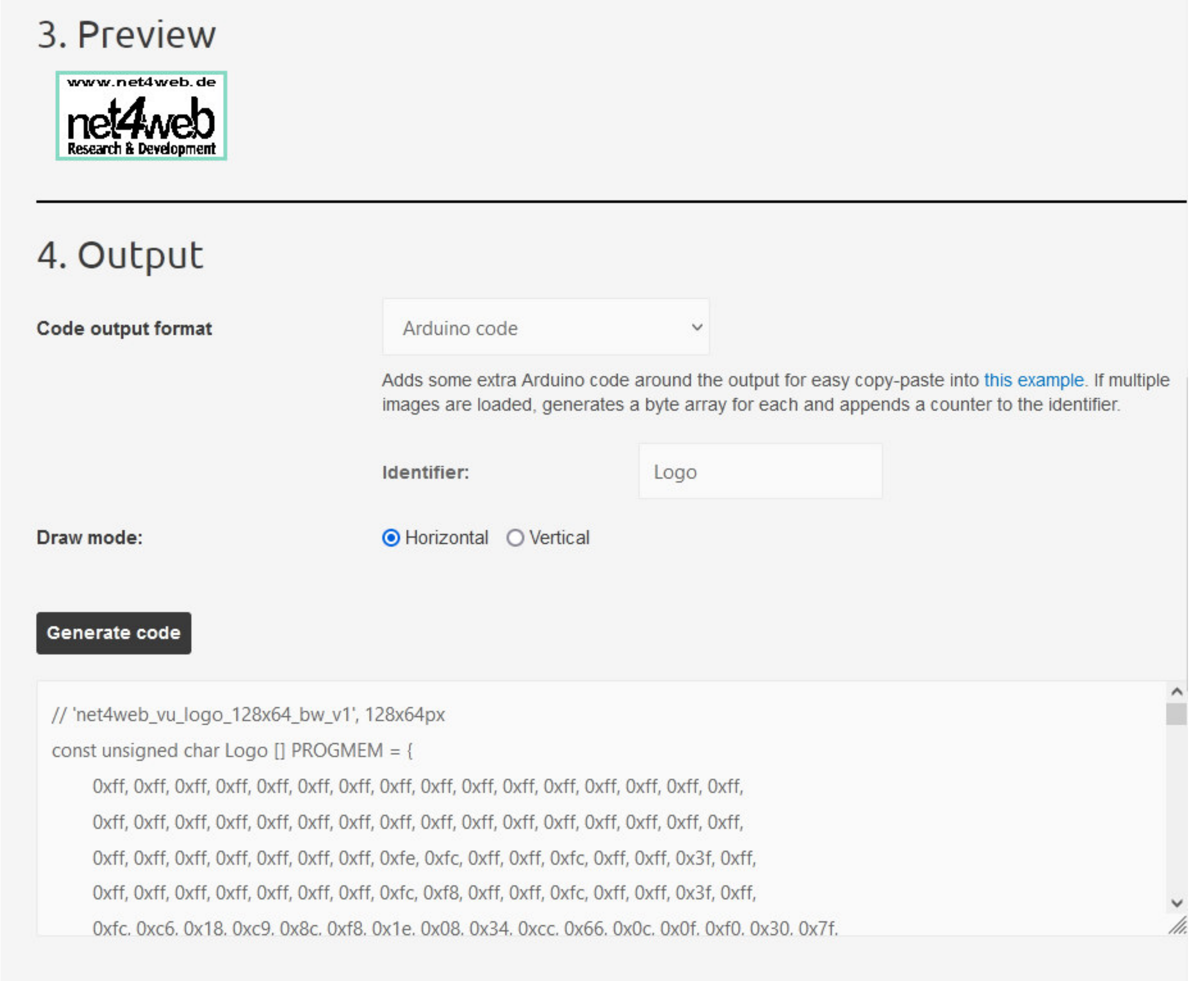
Code output format Arduino code

Adds some extra Arduino code around the output for easy copy-paste into this example. If multiple images are loaded, generates a byte array for each and appends a counter to the identifier.

Identifier: Logo

Draw mode: ● Horizontal ○ Vertical

In the preview you can immediately see how your graphic will look later on the display. Well almost ☺. What you see in black in the preview is the area that will later light up in the display color. The white areas will then remain black on the display. Feel free to play around with the settings. You can immediately see how this affects your graphics. In any case, we came up with a few new ideas. If you've played with the settings and want to discard them, we recommend that you first click Remove and start over. The website sometimes seems stuck on the settings ☺.



3. Preview

4. Output

Code output format: Arduino code

Adds some extra Arduino code around the output for easy copy-paste into this example. If multiple images are loaded, generates a byte array for each and appends a counter to the identifier.

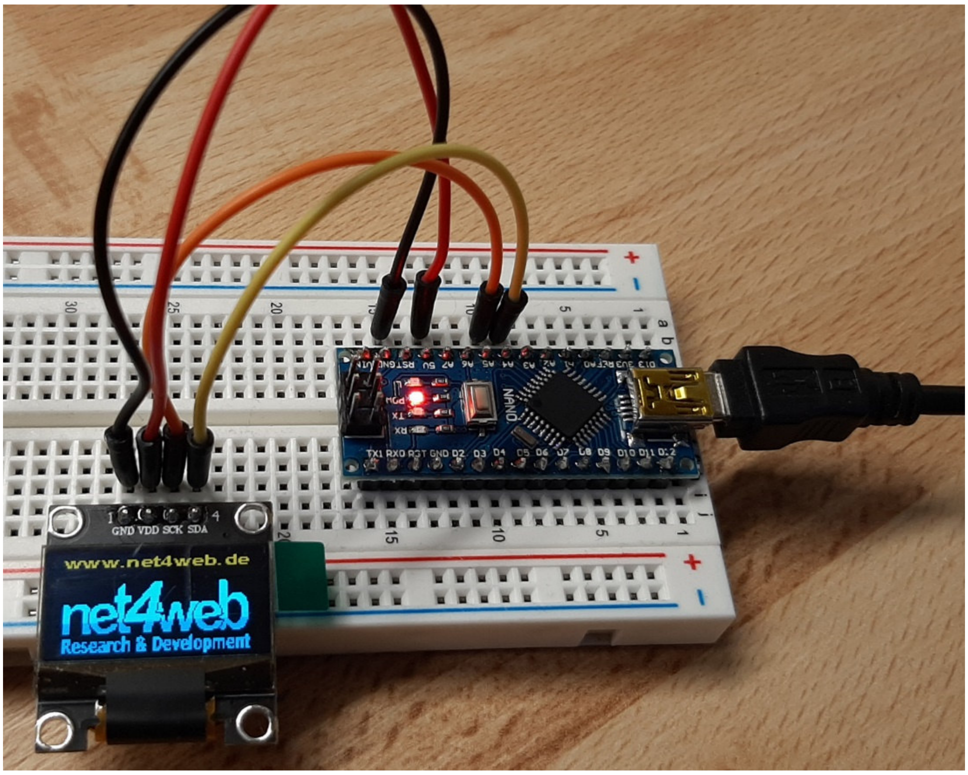
Identifier: Logo

Draw mode: ● Horizontal ○ Vertical

```
// 'net4web_vu_logo_128x64_bw_v1', 128x64px
const unsigned char Logo [] PROGMEM = {
	0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
	0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
	0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0xfc, 0xff, 0xff, 0xfc, 0xff, 0xff, 0x3f, 0xff,
	0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0xf8, 0xff, 0xff, 0xfc, 0xff, 0xff, 0x3f, 0xff,
	0xfc, 0xc6, 0x18, 0xc9, 0x8c, 0xf8, 0x1e, 0x08, 0x34, 0xcc, 0x66, 0x0c, 0x0f, 0xf0, 0x30, 0x7f,
```

Now select Arduino Code as Code Output Format and enter a name of your choice for Identifier: (Only alphanumeric characters and the underscore, no special characters). We used Logo as identifier. You use the identifier later to call up the graphic in the source code of the Arduino sketch. Copy the code and paste it into your sketch.

Photos:



Visit us online: www.net4web.de

In the Downloads area you will find the RoHS and CE declarations of conformity, as well as further technical information about our products.