

Grafik erstellen für die Verwendung eines Oled Displays 0,96" mit einem Arduino

Inhalt

Beschreibung:	1
Grafik erstellen:.....	1
Grafik in Arduino-Code umwandeln:	2
Fotos:.....	5

Beschreibung:

Wenn Sie eine eigene Grafik als Hintergrund auf einem Oled-Display in Verbindung mit einem Arduino-Board verwenden wollen, müssen Sie zunächst eine Grafik in der entsprechenden Größe erstellen. Wir führen hier als Beispiel ein 0,96" Oled-Display mit einer Auflösung von 128x64 Pixels auf. Wenn Sie ein Display mit einer anderen Auflösung verwenden wollen, müssen Sie lediglich die Größe Ihrer Grafik entsprechend anpassen.

Grafik erstellen:

Sie benötigen eine schwarzweiße Grafik in der Größe 128x64 Pixels. Bei der Erstellung der Grafik sollten Sie sich bereits überlegen, welche Bereiche später in der Displayfarbe erscheinen sollen und welche Bereiche dunkel bleiben sollen. Speichern Sie die Grafik nicht im JPG-Format, sondern im PNG-Format. Im JPG-Format kann die Grafik komprimiert werden, was zu Pixeln an der falschen Stelle führen kann.



In unserer Beispielgrafik haben wir rund um die Grafik 2 Pixels Rand gelassen. Für unser Logo machte das Sinn. Sie können aber die gesamte Fläche ausnutzen, wenn Sie das für richtig erachten. Zwischen dem Logo und der Web-Adresse ist in unserem

Beispiel eine kleine Lücke. Die ist nur aus einem Grund genau an dieser Stelle. Es gibt die Oled-Displays ja auch in zweifarbig. Bei den zweifarbigen Displays ist der untere Bereich blau und der obere Bereich gelb. Zwischen diesen beiden Farbbereichen gibt es eine Lücke von einem Pixel Höhe. Bei der Erstellung der Hintergrundgrafik für zweifarbige Displays muss man das natürlich berücksichtigen, sonst fehlt dann nachher eine Zeile im Bild oder ein Teil ist in der „falschen“ Farbe. Wenn Sie ein monochromes Display verwenden, müssen Sie sich darüber natürlich keine Gedanken machen. Bei unserem Beispiel galt aber die Devise: Eine Grafik für alle Displays.

Grafik in Arduino-Code umwandeln:

Das Umwandeln dieser Grafik in einen Code, den der Arduino versteht ist über das Internet problemlos und schnell möglich. Besuchen Sie dazu die 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 additional code by [wiredolphin](#), [davidalim](#) and [whoisian](#)

1. Select image

Keine Dateien ausgewählt.

2. Image Settings

Canvas size/s: No files selected

Background color: White Black


Invert image colors

Brightness threshold:

Scaling:

Center: horizontally vertically

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



Die Oberfläche ist zwar in Englisch aber auch wenn man der englischen Sprache nicht so mächtig ist leicht zu verstehen, da man die Resultate sofort sieht. Klicken Sie auf die Schaltfläche „Durchsuchen“ und laden Sie Ihre vorbereitete Grafik.

Seite 2

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)

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 der Vorschau (Preview) können Sie sofort sehen, wie Ihre Grafik später auf dem Display aussieht. Naja fast 😊. Was Sie in der Vorschau in schwarz sehen, ist der Bereich, der später in der Display-Farbe leuchtet. Die weißen Bereiche werden dann auf dem Display schwarz bleiben. Spielen Sie ruhig mal etwas mit den Einstellungen herum. Sie sehen sofort, wie sich das auf Ihre Grafik auswirkt. Uns sind dabei jedenfalls ein paar neue Ideen gekommen. Wenn Sie mit den Einstellungen gespielt haben und diese wieder verwerfen wollen, empfehlen wir Ihnen zunächst auf Remove (Entfernen)

zu klicken und ganz von vorne anzufangen. Die Website scheint manchmal an den Einstellungen festzuhalten 😊.

3. Preview



4. Output

Code output format:

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:

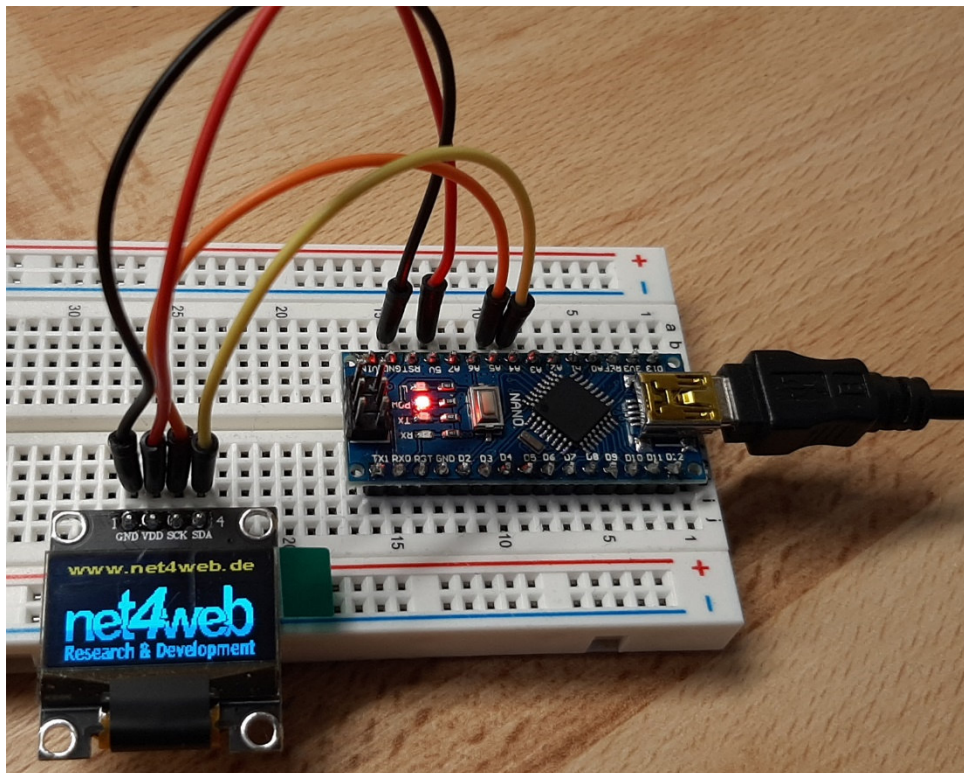
Draw mode: Horizontal Vertical

Generate code

```
// '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, 0xfe, 0xfc, 0xff, 0xff, 0xfc, 0xff, 0xff, 0x3f, 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.
```

Wählen Sie nun als Code Output Format Arduino Code und vergeben Sie bei Identifier: einen Namen Ihrer Wahl (Nur alphanumerische Zeichen und den Unterstrich, keine Sonderzeichen). Wir haben als Identifier *Logo* verwendet. Mit dem Identifier rufen Sie später im Quellcode des Arduino-Sketches die Grafik auf. Kopieren Sie sich den Code und fügen Sie ihn in Ihren Sketch ein.

Fotos:



Besuchen Sie uns im Internet: www.net4web.de

Im Bereich **Downloads** finden Sie die Konformitätserklärungen zu RoHS und CE, sowie weitere technische Informationen zu unseren Produkten.