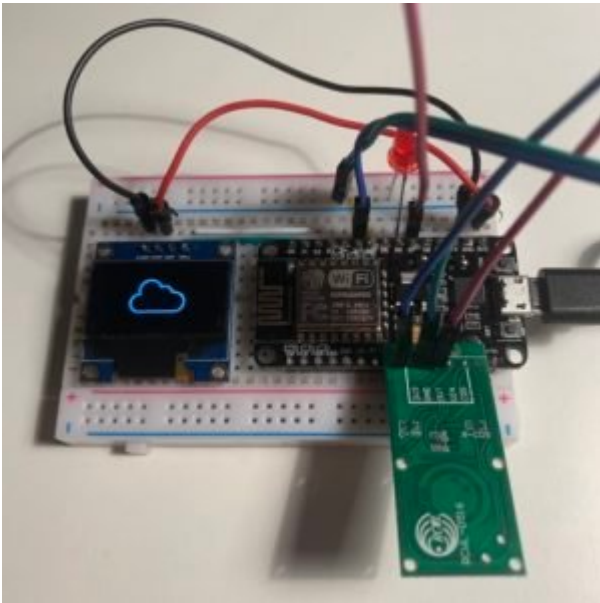


Bilder auf einem OLED-Display anzeigen



Du kannst mit deinem Arduino (oder ESP8266, ESP32 etc.) neben Texten auch Bilder auf einem monochromen OLED-Display anzeigen lassen. Allerdings kannst du nicht einfach eine Bilddatei wie zum Beispiel ein JPEG dort erscheinen lassen. Die Sache ist etwas komplexer, aber mit den richtigen Tools trotzdem kein Problem.

Grundsätzliche Überlegungen

Das vermutlich gängigste OLED-Display für den Arduino hat eine Diagonale von 0.96" und 128×64 Pixel. Das bedeutet, dass ein Bild für das Display natürlich auch nur über diese Größe verfügen kann. Soviel ist klar. □

Hinzu kommt, dass dieses OLED-Display nur weiße Pixel anzeigen kann, also monochrom ist. Mit dem schwarzen Hintergrund macht das dann jedoch immerhin zwei Farben. Das bedeutet, du kannst entweder etwas weißes auf schwarzem Hintergrund zeichnen – oder den Hintergrund weiß einfärben und darüber in Schwarz zeichnen.

Es ist also naheliegend, weniger von einem „Bild“ zu sprechen, sondern eher von einem „Icon“.

Bilder konvertieren

Dein erster Schritt ist es also, ein Bild in Code zu konvertieren, damit dein Arduino es aufs Display bringen kann. Hierfür verwendest du am besten ein Online-Tool, wie zum Beispiel [image2cpp](#), das du im folgenden kennenlernenst.

Nehmen wir zum Beispiel dieses schicke UF0:



Bild von [smalllikeart](#) von [www.flaticon.com](#)

Dieses Icon ist bereits schwarz-weiss und hat mit 64×64 Pixeln eine geeignete Größe – es ist genauso hoch wie das Display. Jetzt sind es nur 4 einfache Schritte, um dieses Bild in Code konvertieren.

Lade zunächst dein Icon hoch:

1. Select image

Dateien auswählen alien.png

Anschließend musst du ein paar kleinere Einstellungen vornehmen. Angenommen, du möchtest ein weißes Ufo auf schwarzem Hintergrund in der Mitte des Displays zentriert:

- Stelle die **Canvas size** auf die Größe deines Displays (z.B. 128×64)
- Wähle **Black** als **Background color**
- Mach einen Haken bei **Invert image colors**
- Mach einen weiteren Haken unter **Center** -> **horizontally**

2. Image Settings

Canvas size/s:

alien.png (file resolution: 64 x 64)

128 x 64 glyph

Background color:

☐ White ☒ Black

Invert image colors



Brightness threshold:

128

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.

Spiele etwas mit den Werten herum, um herauszufinden, was sie bewirken. Unter **Brightness threshold** kannst du die Linien deines Icons zum Beispiel feinjustieren. So ungefähr sollte dein Icon jetzt aussehen:

3. Preview



Fast geschafft! Jetzt fehlt nur noch der Code. Wähle im nächsten Bereich unter **Code output format** die Einstellung **Arduino code**. Unter **Identifizier** kannst du deinem Icon einen Namen geben, mit dem du es später in deinem Sketch aufrufen und einbinden kannst.

Klicke zuletzt auf den Button **Generate code** und kopiere dir genau diesen Code aus dem Feld darunter heraus.

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:

alien

Draw mode:

☒ Horizontal ☐ Vertical

Generate code

```
// 'alien', 128x64px
const unsigned char alien [] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
```

Den Code im Sketch einbinden und das Icon anzeigen

Den frisch kopierten Code setzt du am besten gleich in deinen Sketch ein – und zwar noch vor die Funktion **void setup()**.

Bevor du etwas auf deinem OLED-Display anzeigen lassen kannst, musst du in deinem Sketch ein paar Vorbereitungen treffen, wie zum Beispiel die nötigen Bibliotheken einbinden. Wenn du das noch nicht kennst, [erfährst du alles weitere in diesem Projekt](#).

Nach deinen Vorbereitungen zauberst du das Icon nun wie folgt auf dein Display:

```
display.drawBitmap(0, 0, alien, 128, 64, WHITE);
display.display();
```

Das war's! Jetzt lass deiner Kreativität freien Lauf! ☐

Wie wäre es mit die Projekt: [Sende von deinem Smartphone Emojis nach Hause und zeige sie dort auf einem OLED-Display an.](#)