image2cpp

image2cpp is a simple tool to change images into byte arrays (or arrays back into an images) for use with (monochrome) displays such as OLEDs on your Arduino or Raspberry Pi.
It was originally made to work with the Adafruit OLED library (for which your can find an example sketch for Arduino here) but has been expanded by the community to be useful in all kind of (embedded) projects.

More info (and credits) can be found in the Github repository. This is also where you can report any issues you might come across.

Did you find this tool useful? Feel free to support my open source software on Github, especially if used commercially.

Sponsor 0 Buy me a coffee

1. Select image

All processing is done locally in your browser; your images are not uploaded or stored anywhere online.

Choose Files 1.png

1.png remove

or

1. Paste byte array

128 x 64 px

Read as horizontal Read as vertical

Read images appear at step 3 below

---

2. Image Settings

Canvas size(s):

1.png (file resolution: 2048 x 1397)

128 x 64 glyph remove

Background color: (•) White ( ) Black ( ) Transparent

Invert image colors [ ]

Dithering: Binary

Brightness / alpha threshold: 128

0 - 255; if the brightness of a pixel is above the given level the pixel becomes white, otherwise they become black. When using alpha, opaque and transparent are used instead.

Scaling: scale to fit, keeping proportions

Center image: [x] horizontally [x] vertically

Centering the image only works when using a canvas larger than the original image.

Rotate image: 0 degrees

Flip image: [ ] horizontally [ ] vertically

---

3. Preview

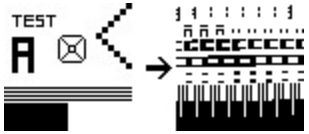
---

4. Output

Code output format Plain bytes

Draw mode: Horizontal - 1 bit per pixel

If your image looks all messed up on your display, like the image below, try using a different mode.



Swap bits in byte: [ ] swap

Useful when working with the u8g2 library.

Extra formatting options: ☐ Remove '0x' and commas from output

Generate code Copy Output Download as binary file (.bin)

```
// '1', 128x64px
0xff, 0xff, 0xff, 0xff, 0xe0, 0x07, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x07, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x3f, 0xf7, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x07, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x03, 0xff, 0xff, 0x80, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x1f, 0xf8, 0x00, 0x1f, 0xf8, 0x00, 0x1f, 0xf8, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xf8, 0x00, 0x7f, 0xff, 0x00, 0x0f, 0xf0, 0x00, 0xff, 0xfe, 0x00, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xf0, 0x01, 0xff, 0xff, 0xc0, 0x07, 0xe0, 0x03, 0xff, 0xff, 0x80, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xf0, 0x03, 0xff, 0xff, 0xe0, 0x03, 0xc0, 0x07, 0xff, 0xff, 0xc0, 0x0f, 0xff, 0xff,
```