

Guide for displaying a custom image on e-paper display

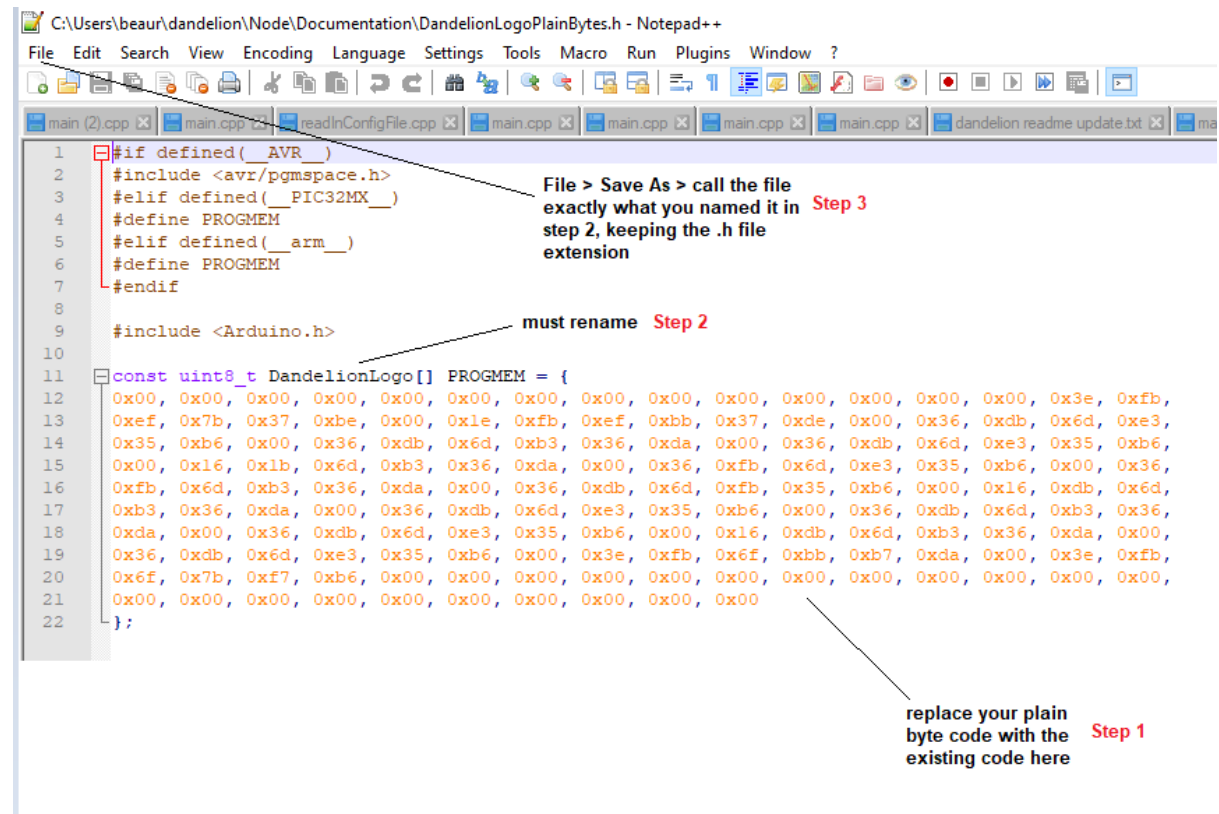
Source: https://www.youtube.com/watch?v=W-r-sF-O2xBE&list=PLoaL5wepIgmfpHDWzBej9zjJrounB8mG&index=14&ab_channel=ShotokuTech

The following **17** steps outline the process:

- 1) Download and install IrfanView from <https://www.irfanview.com/>
- 2) Open IrfanView > Open the image you want displayed on the e-paper display
- 3) In the toolbar, click Image > Resize/Resample
- 4) Set your required Width and Height. For reference, an image taking up the entire e-paper display is 250 x 112. The Dandelion logo is 50 x 22. The Wi-Fi connected & disconnected icons are 30 x 22.
- 5) Ensure "preserve aspect ratio (proportional)" is unticked
- 6) Set DPI to 16 > click OK
- 7) Click Image > Decrease colour depth
- 8) Select "2 Colors (black/white) (1 BPP)" radio button > Click OK
- 9) Save As
- 10) Browse to <http://javl.github.io/image2cpp/>
- 11) Click "Choose files" > open the file you just saved
- 12) Under "Image Settings", beside "Background color" > select "Black" radio button
- 13) Under "Output", choose "plain bytes" in the "Code output format" drop-down menu
- 14) Click "Generate code"
- 15) Highlight and copy the code

Continues on next page...

- 16) Open an existing file with plain byte code for another image in a text editor such as Notepad ++ to paste the plain bytes for this image into, replacing what's there. Change the name inside the file & the name of the file itself. These must be matching. An existing file with plain byte code can be found in the Dandelion GitHub repo/Node/Documentation/DandelionLogoPlainBytes.h.



The screenshot shows the Notepad++ editor with the file `C:\Users\beaur\dandelion\Node\Documentation\DandelionLogoPlainBytes.h` open. The code is as follows:

```
1  #if defined( __AVR__ )
2  #include <avr/pgmspace.h>
3  #elif defined( __PIC32MX__ )
4  #define PROGMEM
5  #elif defined( __arm__ )
6  #define PROGMEM
7  #endif
8
9  #include <Arduino.h>
10
11  const uint8_t DandelionLogo[] PROGMEM = {
12    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3e, 0xfb,
13    0xef, 0x7b, 0x37, 0xbe, 0x00, 0x1e, 0xfb, 0xef, 0xbb, 0x37, 0xde, 0x00, 0x36, 0xdb, 0x6d, 0xe3,
14    0x35, 0xb6, 0x00, 0x36, 0xdb, 0x6d, 0xb3, 0x36, 0xda, 0x00, 0x36, 0xdb, 0x6d, 0xe3, 0x35, 0xb6,
15    0x00, 0x16, 0x1b, 0x6d, 0xb3, 0x36, 0xda, 0x00, 0x36, 0xfb, 0x6d, 0xe3, 0x35, 0xb6, 0x00, 0x36,
16    0xfb, 0x6d, 0xb3, 0x36, 0xda, 0x00, 0x36, 0xdb, 0x6d, 0xfb, 0x35, 0xb6, 0x00, 0x16, 0xdb, 0x6d,
17    0xb3, 0x36, 0xda, 0x00, 0x36, 0xdb, 0x6d, 0xe3, 0x35, 0xb6, 0x00, 0x36, 0xdb, 0x6d, 0xb3, 0x36,
18    0xda, 0x00, 0x36, 0xdb, 0x6d, 0xe3, 0x35, 0xb6, 0x00, 0x16, 0xdb, 0x6d, 0xb3, 0x36, 0xda, 0x00,
19    0x36, 0xdb, 0x6d, 0xe3, 0x35, 0xb6, 0x00, 0x3e, 0xfb, 0x6f, 0xbb, 0xb7, 0xda, 0x00, 0x3e, 0xfb,
20    0x6f, 0x7b, 0xf7, 0xb6, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
21    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
22  };
```

Annotations in the image:

- Step 1:** An arrow points to the byte array data, with the text: "replace your plain byte code with the existing code here".
- Step 2:** An arrow points to the variable name `DandelionLogo` in the array declaration, with the text: "must rename".
- Step 3:** An arrow points to the `#if defined(__AVR__)` block, with the text: "File > Save As > call the file exactly what you named it in step 2, keeping the .h file extension".

- 17) Now you have the new header file, place it into the 'include' folder found in the directory of your C++/PlatformIO project, ensure you add it as a #include directive in the appropriate class or header file, and the image is now available to use.