

Using a Monochrome 1.3" 128x64 OLED display

What is the OLED monochrome display?

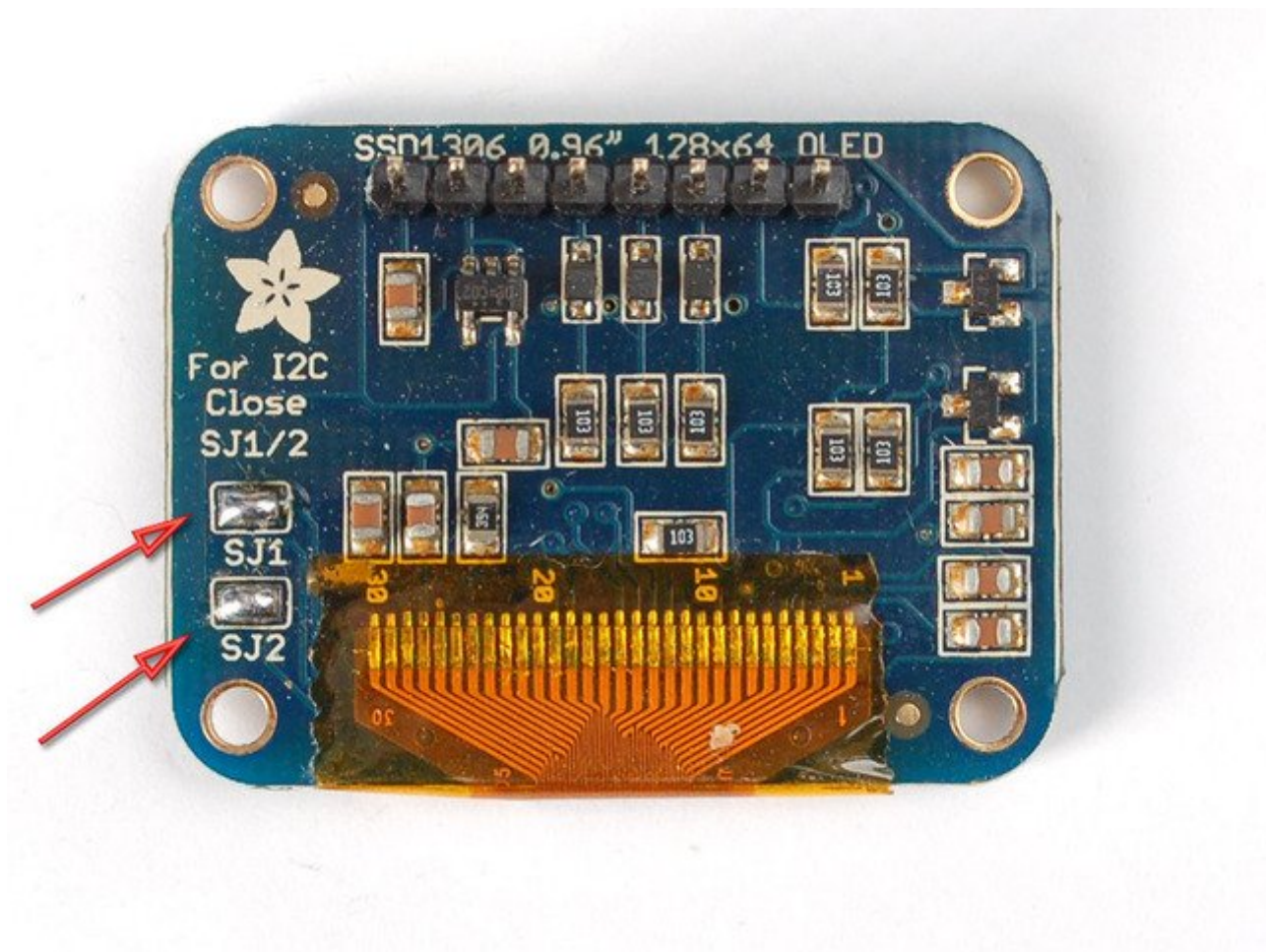
The OLED monochrome display is a small (tiny) and high-readability display. It is useful for displaying data, e.g. weather information or small graphics like what you see on Tamagotchi. For more information, please visit [here](#).



In this tutorial, we will be using Adafruit SSD1306 128 x 64 OLED with I2C communication. It can support SPI communication as well.

Display Configuration

If you have the older non-STEMMA version of the OLED, you'll need to solder the two jumpers on the back of the OLED. Both must be soldered 'closed' for I2C to work!



Wiring

Wiring up the sensor is simple:

1. Power (VIN to 5V)
2. Ground (GND to GND)
3. Data to Arduino SDA pin (A5 on Uno)
4. CLK to Arduino SCL pin (A4 on Uno)

Library

To use this code you will need the [Adafruit_SSD1306 Library](#). We have a tutorial on [how to install a library](#) here.

Getting started

This code will display two bitmap images at intervals of 1 second.

```
#include <SPI.h>
#include <Wire.h>
```

[illegible]

[illegible]

}

```
void setup() {
```

```

Serial.begin(9600);

// SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
  Serial.println(F("SSD1306 allocation failed"));
  for(;;); // Don't proceed, loop forever
}
}

void loop() {
  testdrawbitmap(); // Draw a small bitmap image
}

void testdrawbitmap(void) {

  display.clearDisplay();
  display.drawBitmap(
    0, //x coordinate
    0, //y corrdinate
    my_bitmap[counter++], //bitmap file
    128, //bitmap width
    64, //bitmap height
    1 //each '1' bit sets the corresponding pixel to 'color'
  );
  display.display();
  delay(frameRate);

  if (counter >= numImage){
    counter = 0;
  }

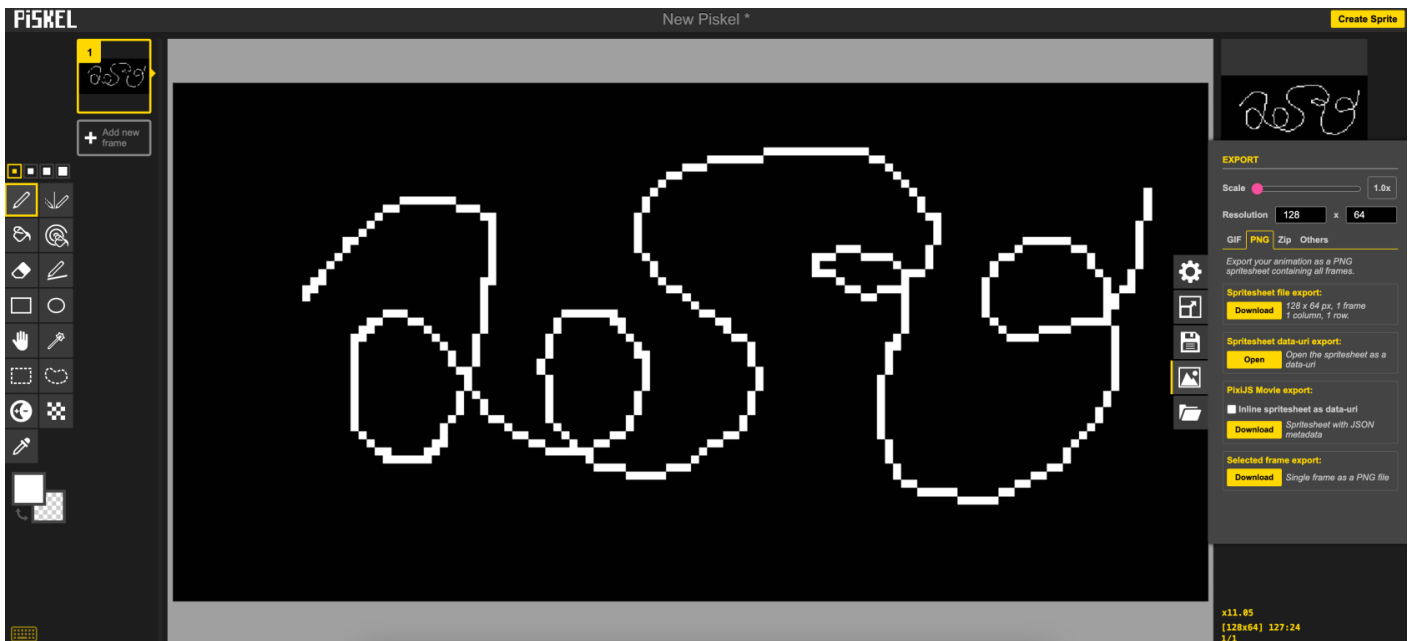
}

```

Create your own bitmap

Create your pixel art

Piskel is a free online tool for you to create pixel art. You can specify the canvas size, import images, draw your own graphics etc, and then export it as a PNG.



Convert your image into bitmap code

image2cpp was created by GitHub user *javl* and provides a handy way to create bitmaps without installing any additional software. [Know more here.](#)

Upload your image, select your preferred image settings, and generate code!

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 you 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



1. Select image

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

Choose files New Piskel.png

New Piskel.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):

New Piskel.png (file resolution: 128 x 64)

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:

original size

Center image:

☐ horizontally ☐ 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

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/Prefix:

epd_bitmap_

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.

TEST

You will see the code generated at the bottom and you can paste it in Arduino directly.

```
// 'New Piskel', 128x64px
const unsigned char epd_bitmap_New_Piskel[] 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,
    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, 0x7f, 0xfc, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0x80, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x80,
    0x00, 0x00, 0x00, 0xf, 0xe0, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x80,
    0x00, 0x00, 0x00, 0x30, 0x1c, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x80,
    0x00, 0x00, 0x00, 0x40, 0x03, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x80,
    0x00, 0x00, 0x00, 0x80, 0x01, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x01, 0x00,
    0x00, 0x00, 0x01, 0x00, 0x01, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x01, 0x00,
    0x00, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x02, 0x00, 0x0e, 0x01, 0x00,
    0x00, 0x00, 0x04, 0x00, 0x01, 0x00, 0x00, 0x40, 0x00, 0x00, 0x38, 0x04, 0x00, 0x81, 0x81, 0x00,
    0x00, 0x00, 0x08, 0x00, 0x01, 0x00, 0x00, 0x20, 0x00, 0x01, 0xc7, 0x84, 0x03, 0x00, 0x41, 0x00,
    0x00, 0x00, 0x10, 0x00, 0x01, 0x00, 0x00, 0x10, 0x00, 0x01, 0x00, 0x01, 0x00, 0x00, 0x10, 0x00
```

Create Animation

Change the parameters at the start of the code, including the number of frames, frame sizes etc. Arduino doesn't come with a huge memory, so when you are preparing the animation, you may need to think about how many frames you want and how many frames can Arduino handle, and crop the empty space to minimize the bytes used.

The below example demonstrates two animations.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET    4 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3D ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// for 1st animation
int numFrame = 13;

int frameRate = 85; // 85 = 12fps, 67 = 15fps, 42 = 24fps
int counter = 0;

int bitmapWidth = 35;
int bitmapHeight = 64;
```

[illegible]


```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x30, 0x30, 0x00, 0x07, 0x1c,
0x38, 0x78, 0x00, 0x07, 0x18, 0x1c, 0xf8, 0x00, 0x07, 0xf8, 0x0f, 0xd8, 0x00, 0x07, 0xf0, 0x07,
0x98, 0x00, 0x06, 0x60, 0x03, 0x18, 0x00, 0x06, 0x00, 0x00, 0x18, 0x00, 0x06, 0x00, 0x00, 0x18,
0x00, 0x06, 0x00, 0x00, 0x18, 0x00, 0x06, 0x00, 0x00, 0x18, 0x00, 0x06, 0x00, 0x00, 0x18, 0x00,
0x06, 0x00, 0x00, 0x30, 0x00, 0x06, 0x00, 0x00, 0x30, 0x00, 0x06, 0x00, 0x00, 0x30, 0x00, 0x03,
0x00, 0x00, 0x60, 0x00, 0x03, 0x00, 0x00, 0x60, 0x00, 0x03, 0x00, 0x00, 0xe0, 0x00, 0x03, 0x80,
0x00, 0xc0, 0x00, 0x01, 0x80, 0x01, 0x80, 0x00, 0x01, 0xc0, 0x07, 0x80, 0x00, 0x00, 0xe0, 0x1e,
0x00, 0x00, 0x00, 0x7b, 0xf8, 0x00, 0x00, 0x00, 0x3f, 0xe0, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00,
0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00,
0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00,
0x03, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x03, 0x00,
0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00,
0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00,
0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00,
0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01,
0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80,
0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00,
0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

```
// Array of all bitmaps for convenience. (Total bytes used to store images in PROGMEM = 4368)
```

```
const int epd_bitmap_allArray_LEN = 13;
const unsigned char* epd_bitmap_allArray[13] = {
    epd_bitmap_frame_00_delay_0,
    epd_bitmap_frame_01_delay_0,
    epd_bitmap_frame_02_delay_0,
    epd_bitmap_frame_03_delay_0,
    epd_bitmap_frame_04_delay_0,
    epd_bitmap_frame_05_delay_0,
    epd_bitmap_frame_06_delay_0,
    epd_bitmap_frame_07_delay_0,
    epd_bitmap_frame_08_delay_0,
    epd_bitmap_frame_09_delay_0,
    epd_bitmap_frame_10_delay_0,
    epd_bitmap_frame_11_delay_0,
    epd_bitmap_frame_12_delay_0
};
```

```
////////////////////////////////////animation1////////////////////////////////////
```

```
// '3', 78x64px
```

```
const unsigned char epd_bitmap_3 [] 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,  
    0x01, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x3f, 0xff, 0x80, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x03, 0xf1, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xcf,  
    0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x9f, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00,  
    0x00, 0x00, 0x06, 0x7f, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x04, 0xff, 0xff, 0xff,  
    0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,  
    0x03, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff,  
    0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x0f, 0xff,  
    0xff, 0xff, 0xff, 0xdf, 0xc0, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0,  
    0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff,  
    0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x0e,  
    0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x01, 0xdf, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,  
    0xff, 0x80, 0x03, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x01, 0xff, 0x7f, 0xc7,  
    0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0xfe, 0x7f, 0x83, 0xff, 0xfb, 0xff, 0xff, 0xff, 0x00,  
    0x01, 0xff, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0x00, 0x01, 0xff, 0x7f, 0xff, 0x87, 0xf0,  
    0xff, 0xff, 0xfe, 0x00, 0x01, 0xff, 0x7f, 0xff, 0xd3, 0xf9, 0xff, 0xff, 0xfc, 0x00, 0x01, 0xef,  
    0x7f, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x10, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,  
    0xfc, 0x00, 0x00, 0x11, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0xd1, 0xff, 0xff,  
    0xdf, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0xf1, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfc, 0x00,  
    0x00, 0xf1, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x71, 0xff, 0xff, 0xff, 0xff,  
    0xff, 0xff, 0xfc, 0x00, 0x00, 0x71, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x10,  
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,  
    0xf8, 0x00, 0x00, 0x1f, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x1e, 0x3f, 0xff,  
    0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x18, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00,  
    0x00, 0x10, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x10, 0x00, 0x3f, 0xff, 0xff,  
    0xff, 0xff, 0xc0, 0x00, 0x00, 0x1f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x27,  
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,  
    0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff,  
    0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x30, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,  
    0x00, 0x30, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff,  
    0xff, 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x30,  
    0x00, 0xff, 0xff, 0xfe, 0xff, 0x8f, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0x9f,  
    0x80, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xbf, 0x80, 0x00, 0x00, 0x00, 0x00, 0xff,  
    0xff, 0xfe, 0xfb, 0x9f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xd8, 0x00, 0x00,  
    0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xfb, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe,  
    0xf7, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xf7, 0xc0, 0x00, 0x00, 0x00, 0x00,  
    0x01, 0xff, 0xff, 0xff, 0x77, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x67, 0xc0,  
    0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x8f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff,
```

```
0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00
};
// '1', 78x64px
const unsigned char epd_bitmap_1 [] PROGMEM = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff,
0xff, 0xff, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xf8, 0x3f, 0xff, 0x03, 0xff,
0xff, 0xfc, 0x3f, 0xff, 0xfe, 0x01, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xf0, 0x0f,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xfc, 0x3f, 0xff, 0xc7, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xfc,
0x3f, 0xff, 0x1e, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xfc, 0x3f, 0xfe, 0x7c, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xfc, 0x3f, 0xfe, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xf8, 0xff, 0xfc, 0x3f, 0xfe,
0xf3, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xfc, 0x3f, 0xfe, 0x77, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xfc, 0x3f, 0xff, 0x27, 0xff, 0xff, 0xff, 0xff, 0xff, 0xbf, 0xfc, 0x3f, 0xff, 0x8f, 0xff,
0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x3f, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xfc,
0x3f, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe3, 0xfc, 0x3f, 0xff, 0x9f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf8, 0xfc, 0x3f, 0x9f, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7c, 0x3e, 0x07,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3c, 0x20, 0xf3, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x9c, 0x0e, 0xf3, 0x7f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x0f, 0xf3, 0x7f, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x0f, 0xf3, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xe4,
0x27, 0xf2, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0xf0, 0x37, 0xf8, 0xff, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0xf0, 0x27, 0xf8, 0xff, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xff, 0xf0, 0x27, 0xf9,
0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xff, 0xc4, 0x27, 0x01, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0x0c, 0x30, 0x79, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x3f, 0xf9, 0xff, 0xff,
0xcf, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x3f, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xfc,
0x3f, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x3f, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf9, 0xfc, 0x3f, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x3f, 0xfc,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x3f, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xfb, 0xfc, 0x3f, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x3f, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x3f, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc,
0x3f, 0xff, 0xc1, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc, 0x3f, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xfc, 0x3f, 0x9f, 0x83, 0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x3f, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0x3f, 0x2f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8,
0x7f, 0xfc, 0x3f, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x3f, 0xd7, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xfc, 0x3f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xfc,
0x3f, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xfc, 0x3f, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xff, 0x3f, 0xff, 0xfc, 0x3f, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xfc, 0x3f, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0xbe, 0x7f, 0xfc, 0x3f, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xf3, 0x9c,
0x3f, 0xfc, 0x3f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xf3, 0x91, 0x9f, 0xfc, 0x3f, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xf3, 0x87, 0x9f, 0xfc, 0x3f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xf3, 0x9f, 0xbf, 0xfc,
0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf3, 0xdf, 0x3f, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf3, 0xdc, 0x7f, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0x77, 0xd9, 0xff, 0xfc, 0x3f, 0xff,
```

```

0xfd, 0xff, 0xff, 0xff, 0x77, 0xd3, 0xff, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xc7,
0xff, 0xfc, 0x3f, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xc7, 0xff, 0xfc, 0x3f, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xff, 0xcf, 0xff, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xff, 0xfc
};
// '4', 78x64px
const unsigned char epd_bitmap_4 [] 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, 0x07, 0xc0, 0x3f, 0xff, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x0f, 0xf1, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xcf,
0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x9f, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00,
0x00, 0x00, 0x0e, 0x7f, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x0c, 0xff, 0xff, 0xff,
0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x0d, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,
0x0b, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff,
0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x0f, 0xff,
0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00,
0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x63,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xe0, 0x00, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0xff, 0xbf, 0xff,
0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x7f, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x00, 0xff, 0x7f, 0xc7, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0xff, 0xbf, 0xff, 0x87, 0xf7,
0xff, 0xff, 0xff, 0x00, 0x00, 0xff, 0xbf, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x77,
0xbf, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x08, 0x7f, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0x00, 0x00, 0x09, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x09, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x79, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfc, 0x00,
0x00, 0x79, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x79, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x39, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x08,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xf8, 0x00, 0x00, 0x0f, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x1e, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x1c, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00,
0x00, 0x18, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x10, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xc0, 0x00, 0x00, 0x1f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x10, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,
0x00, 0x10, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff,
0xff, 0x38, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x38, 0x00, 0x00, 0x00, 0x30,
0x00, 0xff, 0xff, 0xfe, 0xff, 0xb8, 0x00, 0x00, 0x00, 0x20, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xb8,
0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xb0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff,
0xff, 0xfe, 0xfb, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xd0, 0x00, 0x00,

```

```
0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xfb, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe,
0xf7, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xf7, 0xc0, 0x00, 0x00, 0x00, 0x00,
0x01, 0xff, 0xff, 0xff, 0x77, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x67, 0xc0,
0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x8f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff,
0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00
```

```
};
```

```
// '5', 78x64px
```

```
const unsigned char epd_bitmap_5 [] 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, 0x3f, 0xff, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1f, 0xe1, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xcf,
0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x9f, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00,
0x00, 0x00, 0x1e, 0x7f, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xff, 0xff, 0xff,
0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x19, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,
0x1b, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff,
0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x0f, 0xff,
0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00,
0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00,
0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x7f, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xf0, 0x00, 0x7f, 0xdf, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x7f, 0xdf, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x3f, 0xbf, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xe0,
0x00, 0x3f, 0xbf, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x7f, 0xdf, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0x80, 0x00, 0x7f, 0xdf, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xff, 0x00, 0x00, 0x3b,
0xdf, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x04, 0x3f, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0x00, 0x00, 0x05, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x3d, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x3d, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfc, 0x00,
0x00, 0x3d, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x3d, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x1d, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x0c,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xf8, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x0f, 0xbf, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x0f, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00,
0x00, 0x08, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x08, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xc0, 0x00, 0x00, 0x1f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x10, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,
0x00, 0x30, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x20, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff,
0xff, 0x70, 0x00, 0x00, 0x00, 0x30, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x30, 0x00, 0x00, 0x00, 0x30,
0x00, 0xff, 0xff, 0xfe, 0xff, 0xb0, 0x00, 0x00, 0x00, 0x20, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xb0,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xb0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff,
0xff, 0xfe, 0xfb, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0xfb, 0xd0, 0x00, 0x00,
0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xfb, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe,
0xf7, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0xf7, 0xd0, 0x00, 0x00, 0x00, 0x00,
0x01, 0xff, 0xff, 0xff, 0x77, 0xd0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x67, 0xc0,
0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x8f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff,
0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00
```

```
};
```

```
// '6', 78x64px
```

```
const unsigned char epd_bitmap_6 [] PROGMEM = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xfc, 0x01, 0xff,
0xff, 0xff, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x01, 0xff, 0xff, 0xf8, 0x3f, 0xff, 0x83, 0xff,
0xff, 0xfc, 0x01, 0xff, 0xfe, 0x01, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xfc, 0x01, 0xff, 0xf0, 0x0f,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xfc, 0x01, 0xff, 0xc7, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xfc,
0x01, 0xff, 0x1f, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xfc, 0x01, 0xfe, 0x7c, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xfc, 0x01, 0xfe, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xfc, 0xff, 0xfc, 0x01, 0xfe,
0xfb, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xfc, 0x01, 0xfe, 0x77, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xfc, 0x01, 0xff, 0x2f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xbf, 0xfc, 0x01, 0xff, 0x8f, 0xff,
0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x01, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xfc,
0x01, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe3, 0xfc, 0x01, 0xff, 0xbf, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf8, 0xfc, 0x01, 0x98, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7c, 0x00, 0x63,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3c, 0x00, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x9c, 0x00, 0xff, 0xbf, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x00, 0xff, 0xbf, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x00, 0x7f, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xf4,
0x00, 0xff, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0xf0, 0x00, 0xff, 0xbf, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0xf0, 0x00, 0xff, 0xbf, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x77,
0xbf, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xff, 0xc4, 0x01, 0x08, 0x7f, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfe, 0x0c, 0x01, 0xf9, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x01, 0xf9, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x01, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xfc,
0x01, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x01, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfd, 0xfc, 0x01, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x01, 0xfc,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xfb, 0xfc, 0x01, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x01, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x01, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xfc,
0x01, 0xff, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc, 0x01, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xfc, 0x01, 0x9f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x01, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0x01, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x7f, 0xfc, 0x01, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x01, 0xdf, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xfc, 0x01, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x1f, 0xff, 0xfc,
0x01, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x0f, 0xff, 0xfc, 0x01, 0xff, 0xe0, 0x7f, 0xff, 0xff,
```

```
0xff, 0x67, 0xff, 0xfc, 0x01, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x33, 0xff, 0xfc, 0x01, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0xbb, 0xff, 0xfc, 0x01, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xfb, 0xbb,
0xff, 0xfc, 0x01, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xb9, 0xff, 0xfc, 0x01, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xfb, 0x99, 0xff, 0xfc, 0x01, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xdd, 0xff, 0xfc,
0x01, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xfb, 0xdd, 0xff, 0xfc, 0x01, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf7, 0xdd, 0xff, 0xfc, 0x01, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf7, 0xd9, 0xff, 0xfc, 0x01, 0xff,
0xfd, 0xff, 0xff, 0xff, 0x77, 0xd9, 0xff, 0xfc, 0x01, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xdb,
0xff, 0xfc, 0x01, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xd3, 0xff, 0xfc, 0x01, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xff, 0xc7, 0xff, 0xfc, 0x01, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xc7, 0xff, 0xfc
```

```
};
```

```
// '8', 78x64px
```

```
const unsigned char epd_bitmap_8 [] PROGMEM = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xfe, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x1f, 0xff, 0xfc, 0x1f, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xf0, 0x1f, 0xff,
0xf9, 0xc7, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xf0, 0x1f, 0xff, 0xfb, 0xf0, 0x3f, 0xff, 0x83, 0xff,
0xff, 0xf0, 0x1f, 0xff, 0xf3, 0xf1, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xf0, 0x1f, 0xff, 0xf3, 0xcf,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xf0, 0x1f, 0xff, 0xf7, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xf0,
0x1f, 0xff, 0xf6, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xf0, 0x1f, 0xff, 0xe4, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xf0, 0x1f, 0xff, 0xe1, 0xff, 0xff, 0xff, 0xff, 0xfc, 0xff, 0xf0, 0x1f, 0xff,
0xe3, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xf0, 0x1f, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xf0, 0x1f, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x10, 0x1f, 0xff, 0xcf, 0xff,
0xff, 0xff, 0xff, 0xdf, 0xc0, 0x1f, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0,
0x1f, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x1f, 0xff, 0xbf, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xc0, 0x1f, 0xe7, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xd0, 0x1c, 0x13,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xd0, 0x13, 0x3d, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x90, 0x17, 0xfd, 0x7f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x90, 0x17, 0xfd, 0x7f, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xb0, 0x17, 0xf9, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0x30,
0x1b, 0xfc, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0x70, 0x13, 0xfe, 0xff, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xfe, 0x70, 0x17, 0xfe, 0xff, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xfc, 0xf0, 0x17, 0xfe,
0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xf0, 0x1b, 0x81, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0xf0, 0x1c, 0x79, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xf0, 0x1f, 0xf9, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfd, 0xf0, 0x1f, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xf0,
0x1f, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xf0, 0x1f, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfd, 0xf0, 0x1f, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xf0, 0x1f, 0xfe,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xf0, 0x1f, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xfb, 0xf0, 0x1f, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xf0, 0x1f, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xf3, 0xf0, 0x1f, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xf0,
0x1f, 0xff, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xf0, 0x1f, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xf0, 0x1f, 0x9f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xf0, 0x1f, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xf0, 0x1f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x7f, 0xf0, 0x1f, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xf0, 0x1f, 0xdf, 0xff, 0xff,
```



```
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xf0, 0x1f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xf0,
0x1f, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xf0, 0x1f, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xff, 0x3f, 0xff, 0xf0, 0x1f, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x3c, 0xff, 0xf0, 0x1f, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0xb8, 0x7f, 0xf0, 0x1f, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xfb, 0x93,
0x7f, 0xf0, 0x1f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0x97, 0x3f, 0xf0, 0x1f, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xfb, 0x97, 0x3f, 0xf0, 0x1f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xc7, 0x3f, 0xf0,
0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xfb, 0xcf, 0x7f, 0xf0, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf7, 0xde, 0x7f, 0xf0, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf7, 0xde, 0xff, 0xf0, 0x1f, 0xff,
0xfd, 0xff, 0xff, 0xff, 0x77, 0xdc, 0xff, 0xf0, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xd9,
0xff, 0xf0, 0x1f, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xc3, 0xff, 0xf0, 0x1f, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xff, 0xc7, 0xff, 0xf0, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xff, 0xf0
```

```
};
```

```
// '9', 78x64px
```

```
const unsigned char epd_bitmap_9 [] PROGMEM = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xf9, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff,
0xe0, 0x1f, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xe7, 0xc0, 0x3f, 0xff, 0x03, 0xff,
0xff, 0xfc, 0x3f, 0xff, 0xef, 0xf1, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xef, 0xcf,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xfc, 0x3f, 0xff, 0xef, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xfc,
0x3f, 0xff, 0xce, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xfc, 0x3f, 0xff, 0xcc, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xfc, 0x3f, 0xff, 0xc9, 0xff, 0xff, 0xff, 0xff, 0xfc, 0xff, 0xfc, 0x3f, 0xff,
0xcb, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xfc, 0x3f, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xfc, 0x3f, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x87, 0xfc, 0x3f, 0xff, 0xcf, 0xff,
0xff, 0xff, 0xff, 0xff, 0x80, 0x3c, 0x3f, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x0c,
0x3f, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x3f, 0xff, 0xbf, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf4, 0x3f, 0x9f, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf4, 0x3e, 0x07,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x20, 0xf3, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xe4, 0x0f, 0xfb, 0x7f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xec, 0x1f, 0xf3, 0x7f, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x0f, 0xf3, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xdc,
0x27, 0xfa, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0x9c, 0x37, 0xf8, 0xff, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0x3c, 0x27, 0xf8, 0xff, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xfe, 0x7c, 0x27, 0xf9,
0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x27, 0x01, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0xfc, 0x30, 0x79, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x3f, 0xf9, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x3f, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xfc,
0x3f, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x3f, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfd, 0xfc, 0x3f, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x3f, 0xfc,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x3f, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xfb, 0xfc, 0x3f, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x3f, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xf3, 0xfc, 0x3f, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xfc,
0x3f, 0xff, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc, 0x3f, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xfc, 0x3f, 0x9f, 0xc7, 0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x3f, 0x27,
```

```
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0x3f, 0x2f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x7f, 0xfc, 0x3f, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x3f, 0xdf, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xfc, 0x3f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xfc,
0x3f, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xfc, 0x3f, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xff, 0x3f, 0xff, 0xfc, 0x3f, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xfc, 0x3f, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0xbf, 0xff, 0xfc, 0x3f, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xfb, 0x9c,
0x3f, 0xfc, 0x3f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0x99, 0x9f, 0xfc, 0x3f, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xfb, 0x93, 0x9f, 0xfc, 0x3f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xc7, 0x9f, 0xfc,
0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xfb, 0xcf, 0xbf, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf7, 0xdf, 0x3f, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf7, 0xde, 0x7f, 0xfc, 0x3f, 0xff,
0xfd, 0xff, 0xff, 0xff, 0x77, 0xdc, 0xff, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xc1,
0xff, 0xfc, 0x3f, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xc7, 0xff, 0xfc, 0x3f, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xff, 0xcf, 0xff, 0xfc, 0x3f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xff, 0xfc
```

```
};
```

```
// '2', 78x64px
```

```
const unsigned char epd_bitmap_2 [] PROGMEM = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xfb, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xfc, 0x1f, 0xff,
0xe0, 0x0f, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x1f, 0xff, 0xef, 0xc0, 0x3f, 0xff, 0x83, 0xff,
0xff, 0xfc, 0x1f, 0xff, 0xef, 0xf1, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xfc, 0x1f, 0xff, 0xcf, 0xcf,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xfc, 0x1f, 0xff, 0xcf, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xfc,
0x1f, 0xff, 0xce, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xfc, 0x1f, 0xff, 0xcc, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xfc, 0x1f, 0xff, 0xc9, 0xff, 0xff, 0xff, 0xff, 0xfc, 0xff, 0xfc, 0x1f, 0xff,
0xcb, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xfc, 0x1f, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xfc, 0x1f, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0x8f, 0xfc, 0x1f, 0xff, 0xcf, 0xff,
0xff, 0xff, 0xff, 0xff, 0xc0, 0xfc, 0x1f, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x1c,
0x1f, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc4, 0x1f, 0xff, 0xbf, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x1f, 0xe7, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x1c, 0x13,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf4, 0x13, 0x3d, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xe4, 0x17, 0xfd, 0x7f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x17, 0xfd, 0x7f, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x17, 0xf9, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xdc,
0x1b, 0xfc, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0x9c, 0x13, 0xfe, 0xff, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0x3c, 0x17, 0xfe, 0xff, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xfe, 0x7c, 0x17, 0xfe,
0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x1b, 0x81, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0xfc, 0x1c, 0x79, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x1f, 0xf9, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x1f, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xfc,
0x1f, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x1f, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfd, 0xfc, 0x1f, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x1f, 0xfe,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x1f, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xfb, 0xfc, 0x1f, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x1f, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x1f, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xfc,
```

```
0x1f, 0xff, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc, 0x1f, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xfc, 0x1f, 0x9f, 0xef, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x1f, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0x1f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x7f, 0xfc, 0x1f, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x1f, 0xdf, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xfc, 0x1f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xfc,
0x1f, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xfc, 0x1f, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xff, 0x3f, 0xff, 0xfc, 0x1f, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xfc, 0x1f, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0xa0, 0x3f, 0xfc, 0x1f, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xfb, 0x87,
0x9f, 0xfc, 0x1f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0x9f, 0x9f, 0xfc, 0x1f, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xfb, 0x9f, 0x9f, 0xfc, 0x1f, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xdc, 0x3f, 0xfc,
0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xfb, 0xd8, 0x7f, 0xfc, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf7, 0xdb, 0xff, 0xfc, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf7, 0xd3, 0xff, 0xfc, 0x1f, 0xff,
0xfd, 0xff, 0xff, 0xff, 0x77, 0xd3, 0xff, 0xfc, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xc7,
0xff, 0xfc, 0x1f, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xc7, 0xff, 0xfc, 0x1f, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xc7, 0xff, 0xfc, 0x1f, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xc7, 0xff, 0xfc
```

```
};
```

```
// '7', 78x64px
```

```
const unsigned char epd_bitmap_7 [] PROGMEM = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xfb, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xfc, 0x07, 0xff,
0xe0, 0x0f, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x07, 0xff, 0xef, 0xc0, 0x3f, 0xff, 0x03, 0xff,
0xff, 0xfc, 0x07, 0xff, 0xef, 0xf1, 0xff, 0xff, 0xf0, 0xff, 0xff, 0xfc, 0x07, 0xff, 0xcf, 0xcf,
0xff, 0xff, 0xfe, 0x3f, 0xff, 0xfc, 0x07, 0xff, 0xcf, 0x9f, 0xff, 0xff, 0xff, 0x8f, 0xff, 0xfc,
0x07, 0xff, 0xce, 0x7f, 0xff, 0xff, 0xff, 0xe7, 0xff, 0xfc, 0x07, 0xff, 0xcc, 0xff, 0xff, 0xff,
0xff, 0xf3, 0xff, 0xfc, 0x07, 0xff, 0xc9, 0xff, 0xff, 0xff, 0xff, 0xf8, 0xff, 0xfc, 0x07, 0xff,
0xcb, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xfc, 0x07, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0xfc, 0x07, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff, 0x8f, 0xfc, 0x07, 0xff, 0xcf, 0xff,
0xff, 0xff, 0xff, 0xff, 0x80, 0xfc, 0x07, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x1c,
0x07, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc4, 0x07, 0xff, 0x9f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf0, 0x07, 0x71, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x04, 0x04,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf4, 0x01, 0xdf, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xe4, 0x01, 0xff, 0x7f, 0xef, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x01, 0xfe, 0x7f, 0xc7,
0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x04, 0xfe, 0x7f, 0x83, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xdc,
0x04, 0xfe, 0x7f, 0xc7, 0xff, 0xf0, 0xff, 0xff, 0xff, 0x9c, 0x05, 0xff, 0x7f, 0xef, 0x87, 0xf0,
0xff, 0xff, 0xff, 0x3c, 0x05, 0xff, 0x7f, 0xff, 0xd3, 0xf0, 0xff, 0xff, 0xfe, 0x7c, 0x05, 0xef,
0x7f, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x04, 0x00, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff,
0xfc, 0xfc, 0x07, 0xf9, 0xff, 0xff, 0xd7, 0xff, 0xff, 0xff, 0xfc, 0xfc, 0x07, 0xf9, 0xff, 0xff,
0xcf, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x07, 0xf9, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xfd, 0xfc,
0x07, 0xf9, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xfc, 0x07, 0xfd, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf9, 0xfc, 0x07, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x07, 0xfc,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
```

```

0xfb, 0xfc, 0x07, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x07, 0xff, 0x3f, 0xff,
0xff, 0xff, 0xff, 0xff, 0xf3, 0xfc, 0x07, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc,
0x07, 0xff, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xfc, 0x07, 0xd0, 0x00, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xcf, 0xfc, 0x07, 0x9f, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xfc, 0x07, 0x27,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0x07, 0x2f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x7f, 0xfc, 0x07, 0x87, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x07, 0xd7, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x0f, 0xff, 0xfc, 0x07, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xfc,
0x07, 0xfc, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xfc, 0x07, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xff, 0x3f, 0xff, 0xfc, 0x07, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0x31, 0xff, 0xfc, 0x07, 0xff,
0xfe, 0xff, 0xff, 0xfe, 0xff, 0x80, 0xff, 0xfc, 0x07, 0xff, 0xfe, 0xff, 0xff, 0xfe, 0xf3, 0x8e,
0xff, 0xfc, 0x07, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0x9e, 0x7f, 0xfc, 0x07, 0xff, 0xfc, 0xff,
0xff, 0xfe, 0xfb, 0x8e, 0x7f, 0xfc, 0x07, 0xff, 0xfc, 0xff, 0xff, 0xfe, 0xfb, 0xcf, 0x7f, 0xfc,
0x07, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0xf3, 0xcf, 0x7f, 0xfc, 0x07, 0xff, 0xfd, 0xff, 0xff, 0xfe,
0xf7, 0xde, 0x7f, 0xfc, 0x07, 0xff, 0xfd, 0xff, 0xff, 0xfe, 0x77, 0xde, 0x7f, 0xfc, 0x07, 0xff,
0xfd, 0xff, 0xff, 0xff, 0x77, 0xdc, 0xff, 0xfc, 0x07, 0xff, 0xfd, 0xff, 0xff, 0xff, 0x67, 0xd9,
0xff, 0xfc, 0x07, 0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0xc3, 0xff, 0xfc, 0x07, 0xff, 0xf9, 0xff,
0xff, 0xff, 0xc7, 0xff, 0xfc, 0x07, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xc7, 0xff, 0xfc
};

```

```

// Array of all bitmaps for convenience. (Total bytes used to store images in PROGMEM = 5904)

```

```

const unsigned char* epd_bitmap_allArray2[9] = {
    epd_bitmap_1,
    epd_bitmap_2,
    epd_bitmap_3,
    epd_bitmap_4,
    epd_bitmap_5,
    epd_bitmap_6,
    epd_bitmap_7,
    epd_bitmap_8,
    epd_bitmap_9
};

```

```

void setup() {

```

```

    Serial.begin(9600);

```

```

    // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally

```

```

    if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {

```

```

        Serial.println(F("SSD1306 allocation failed"));

```

```

        for(;;); // Don't proceed, loop forever
    }
}

```

```

    }
}

void loop() {
    drawbitmap1(); // Draw a small bitmap image
    drawbitmap2();

}

void drawbitmap1(void) {

for (int i = 0; i < numFrame; i++){
    display.clearDisplay();
    display.drawBitmap(
        (display.width() - bitmapWidth ) / 2, //x coordinate, center
        (display.height() - bitmapHeight) / 2, //y corrdinate, center
        epd_bitmap_allArray[i], //bitmap file
        bitmapWidth, //bitmap width
        bitmapHeight, //bitmap height
        1 //each '1' bit sets the corresponding pixel to 'color'
    );
    display.display();
    delay(frameRate);
}

}

void drawbitmap2(void) {

for (int i = 0; i < numFrame2; i++){
    display.clearDisplay();
    display.drawBitmap(
        (display.width() - bitmapWidth2 ) / 2, //x coordinate, center
        (display.height() - bitmapHeight2) / 2, //y corrdinate, center
        epd_bitmap_allArray2[i], //bitmap file
        bitmapWidth2, //bitmap width
        bitmapHeight2, //bitmap height
        1 //each '1' bit sets the corresponding pixel to 'color'
    );
    display.display();
    delay(frameRate);
}
}

```

}

}

Revision #6

Created 24 April 2024 10:18:27 by Joanne Leung

Updated 3 May 2024 13:55:25 by Joanne Leung