



اعرض صورك على الشاشة (OLED) باستخدام الاردوينو

مقدمة

تمكنا الشاشات من عرض الصور والرسوم الثابتة على اختلافها سواء كانت خرائط، جداول، رمز شريطي (Barcode)، رسوم بيانية أو رسوم توضيحية، في هذا الدرس سنتعلم كيفية عرض الصور على الشاشة (OLED) باستخدام الاردوينو.



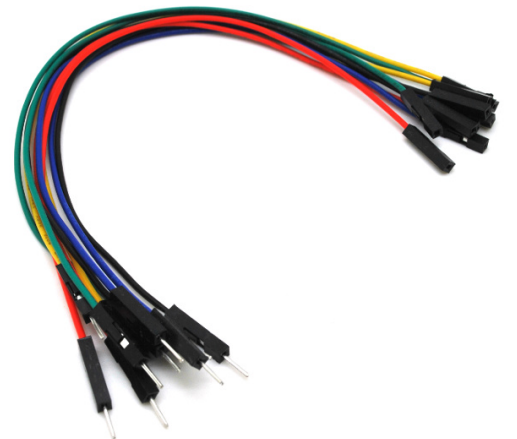
المواد والأدوات



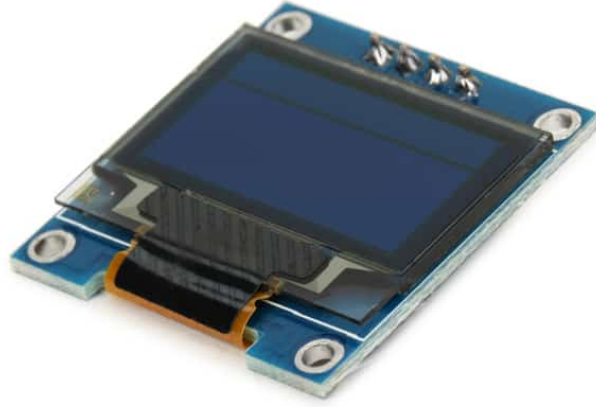
1 × اردوينو اونو



1 × سلك الاردوينو



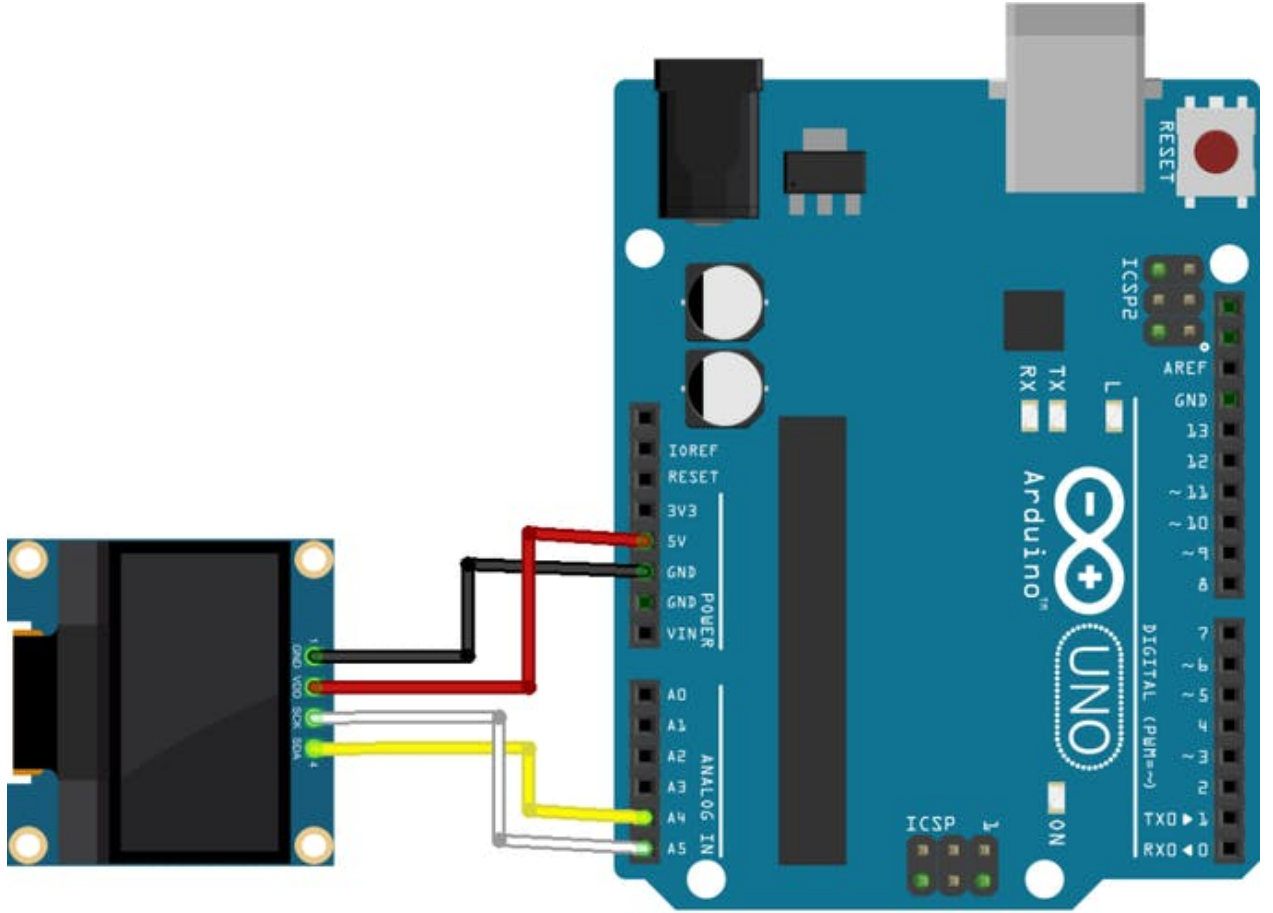
حزمة أسلاك توصيل (ذكر - أنثى)



×1 شاشة (OLED)

توصيل الدائرة

للمزيد حول الشاشة (OLED) يمكنك الرجوع للدرس التالي: شاشة عرض (OLED) Display.



تحويل الصورة إلى كود بنظام (HEX)

في البداية عليك تحويل الصورة لكود (HEX).

نظام (HEX) أو نظام العد الستة عشري سمي بذلك لأنه يحتوي على 16 (ستة عشر) رمزاً لتمثيله (كتابته)، وهم بالترتيب من الرمز الأصغر (0 - صفر) إلى الرمز الأكبر (F).

انقر على الرابط التالي: تحويل الصور إلى كود بنظام (HEX).

1- اختر الصورة المناسبة لك.

1. Select image

اختيار الملفات Transparant ...ks (new).png

2. عيّن الإعدادات التالية: لابد أن يكون مقاس الصورة 64×128 حتى يتلائم مع حجم الشاشة (OLED).

يمكنك اختيار لون الخلفية المناسب لك أبيض أو أسود أو شفاف.

درجة السطوع: 184 أو حسب الدرجة المناسبة للصورة.

Scaling تتلائم مع حجم الشاشة لتملأها.

2. Image Settings

Canvas size(s): Transparent Background Geeks (new).png (file resolution: 400 x 400)
 X glyph remove

Background color: White Black Transparent

Invert image colors

Brightness / alpha threshold:
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

Center: horizontally vertically

Rotate image: rotate 180 degrees

Flip: horizontally vertically

Note: centering the image only works when using a canvas larger than the original image.

3. Preview



3. Arduino code output format اختر

ثم انقر على Generate code حتى يتم إنشاء الكود.

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/Prefix:

Draw mode:

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



انسخ الكود البرمجي الذي داخل دالة `epd_bitmap_Transparant_Background_Geeks_new_[]` = PROGMEM
 ستحتاج إليه في قسم الكود البرمجي.

Generate code

```
// 'Transparent Background Geeks (new)', 64x128px
const unsigned char epd_bitmap_Transparant_Background_Geeks_new_[] 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, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xf0, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x0f, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xfe, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x03, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xf0, 0x00, 0x01, 0x07, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x01, 0xff, 0xff,
  0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x1f, 0x80, 0x00, 0x7f, 0xff,
  0xff, 0xff, 0x80, 0x03, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x80, 0x01, 0x00, 0x03, 0xff, 0xff,
  0xff, 0xff, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff,
  0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0x07, 0xe0, 0x1f, 0x01, 0xff, 0xff,
```

الكود البرمجي

حمّل مكتبة <Adafruit_SSD1306.h> و <Adafruit_GFX.h>.

قبل رفع كود عرض الصور على الشاشة (OLED) إلى لوحة الاردوينو هناك أمور مهمة عليك تحريرها بعد قراءة شرح الكود البرمجي.

```
#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
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define LOGO_HEIGHT 64
#define LOGO_WIDTH 128

//image bitmap data
const unsigned char logo_bmp [] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0x00, 0x07, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xf8, 0x00, 0x01, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x7f, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x0f, 0x07, 0xff, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xc1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0xc0, 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x07, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0x00, 0x00, 0x07, 0xc0, 0x00, 0x00, 0x03, 0x00, 0x00,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x00, 0x03, 0x00, 0x00, 0x01, 0xff, 0x00, 0x00,
  0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0xff, 0x00, 0x03, 0xf8, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0xc7, 0x00, 0x03, 0x8c, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0x83, 0x00, 0x03, 0x04, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0x03, 0x00, 0x03, 0x00, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0x9f, 0x00, 0x03, 0xe4, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0xff, 0x00, 0x03, 0xfc, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x03, 0xff, 0x00, 0x03, 0xf8, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x01, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0x80, 0x0f, 0xff, 0x02, 0x00, 0x07, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x03, 0xf8, 0x06, 0x01, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc0, 0x78, 0x38, 0x0f, 0x7f, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfc, 0x00, 0x01, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0x00, 0x03, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x03, 0x00, 0x40, 0x18, 0x60, 0x80, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x18, 0x41, 0x80, 0x1f, 0x00, 0x00,

```

0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xe1, 0x0f, 0xc3, 0xf8, 0x43, 0x87, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xff, 0x00, 0xc0, 0x18, 0x07, 0x81, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x81, 0x00, 0x40, 0x18, 0x0f, 0x80, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x18, 0x07, 0xf0, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x81, 0x0f, 0xc3, 0xf8, 0x03, 0x0e, 0x0f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xe1, 0x0f, 0xc3, 0x38, 0x41, 0x0e, 0x0f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x08, 0x60, 0x80, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x07, 0x00, 0x52, 0x08, 0xe0, 0x40, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfc, 0xe2, 0x7e, 0x27, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc3, 0xe4, 0xc0, 0x67, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xcc, 0x44, 0x80, 0x4f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x48, 0x49, 0x94, 0xcf, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0x92, 0x49, 0x0c, 0x9f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0x86, 0x99, 0x38, 0x1f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfd, 0xac, 0x90, 0x86, 0x80, 0x7f, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xf9, 0x2c, 0x0d, 0xfc, 0x1f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xf8, 0x63, 0xff, 0xf9, 0x3f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0xff, 0xff, 0xfc, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
};

```

```

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
  }
}

```



```

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

testdrawbitmap(); // Draw the required image
}

void loop() {
}

void testdrawbitmap(void) {
    display.clearDisplay();

    display.drawBitmap(
        (display.width() - LOGO_WIDTH ) / 2,
        (display.height() - LOGO_HEIGHT) / 2,
        logo_bmp, LOGO_WIDTH, LOGO_HEIGHT, 1);
    display.display();
    delay(1000);
}

```

شرح الكود البرمجي

في هذه الأسطر يتم استدعاء المكتبات التالية: <Wire.h> و <Adafruit_GFX.h> و <Adafruit_SSD1306.h>.

```

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

```

هنا حجم الشاشة والصورة المستخدمة 64*128.

```

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

```

هنا نكتب بروتوكول إعادة الضبط.

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

الصق كود (HEX) الخاص بالصورة التي قمت باختيارها هنا استخدمنا شعار وادي العباقرة.

```

static const uint8_t PROGMEM image_data_20190410_114707[] = {
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,

```


0x00, 0x00, 0x00, 0x00, 0xfc, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x01, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0x80, 0x0f, 0xff, 0x02, 0x00, 0x07, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x03, 0xf8, 0x06, 0x01, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc0, 0x78, 0x38, 0x0f, 0x7f, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x3f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfc, 0x00, 0x01, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0x00, 0x03, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfe, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xf0, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x03, 0x00, 0x40, 0x18, 0x60, 0x80, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x18, 0x41, 0x80, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xe1, 0x0f, 0xc3, 0xf8, 0x43, 0x87, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xff, 0x00, 0xc0, 0x18, 0x07, 0x81, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x81, 0x00, 0x40, 0x18, 0x0f, 0x80, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x18, 0x07, 0xf0, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x81, 0x0f, 0xc3, 0xf8, 0x03, 0x0e, 0x0f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0xe1, 0x0f, 0xc3, 0x38, 0x41, 0x0e, 0x0f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x40, 0x08, 0x60, 0x80, 0x1f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xf8, 0x07, 0x00, 0x52, 0x08, 0xe0, 0x40, 0x3f, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfc, 0xe2, 0x7e, 0x27, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc3, 0xe4, 0xc0, 0x67, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xcc, 0x44, 0x80, 0x4f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x48, 0x49, 0x94, 0xcf, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0x92, 0x49, 0x0c, 0x9f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0x86, 0x99, 0x38, 0x1f, 0xff, 0xff, 0x00, 0x00,

```

0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfd, 0xac, 0x90, 0x86, 0x80, 0x7f, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xf9, 0x2c, 0x0d, 0xfc, 0x1f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xf8, 0x63, 0xff, 0xf9, 0x3f, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xfc, 0xff, 0xff, 0xfc, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00,
};

```

في دالة void setup() نهيئ شاشة العرض (OLED) ثم نمسح الشاشة استعدادًا لعرض الصور والرسومات عليها تأكد من كتابة عنوان I2C لشاشة العرض.

```

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  testdrawbitmap(); // Draw the required image
}

```

دالة void loop() بدون أوامر برمجية حيث سينفذ الكود أمر التشغيل في void setup().

```

void loop() { }

```

بعد تحرير الكود البرمجي يمكنك رفعه على لوحة الاردوينو.

تأكد بأن النظام يعمل بالشكل الصحيح.

لا تنسَ فصل وحدة الطاقة بعد الانتهاء من استخدام نظام عرض الصور على الشاشة (OLED).