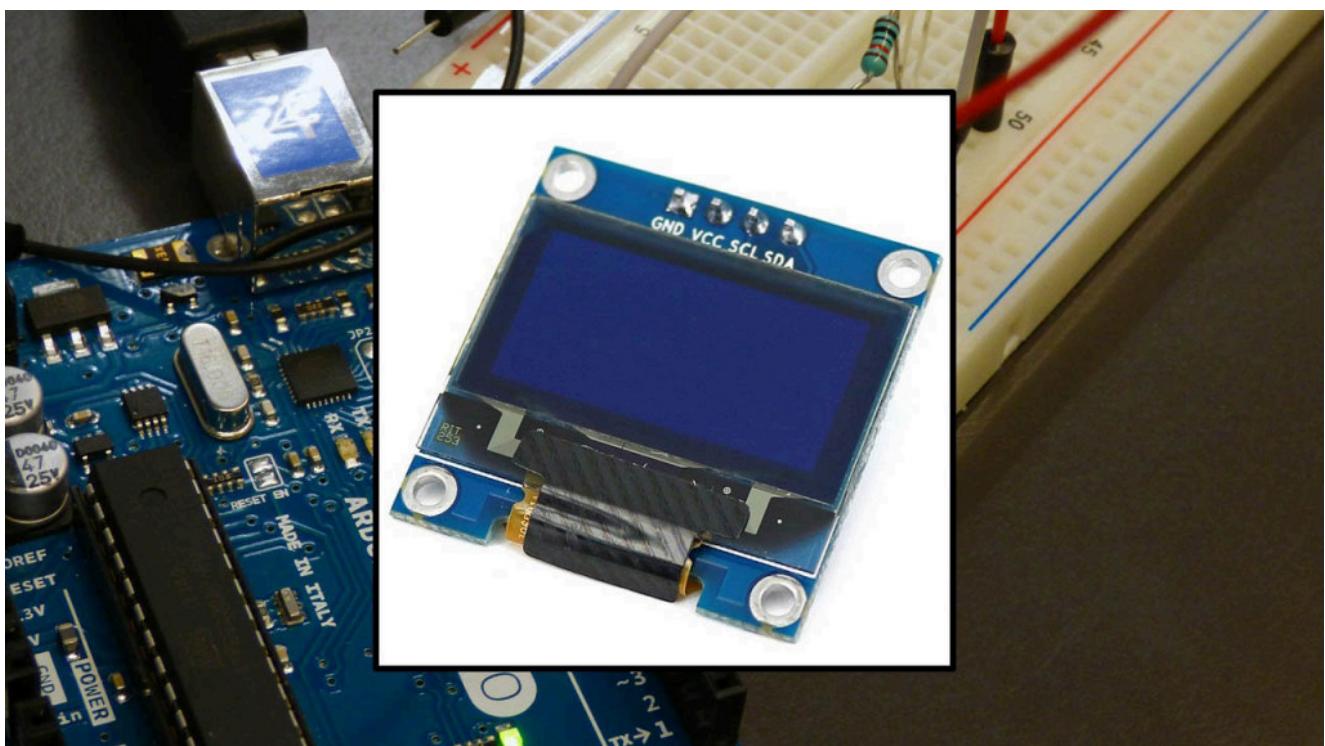


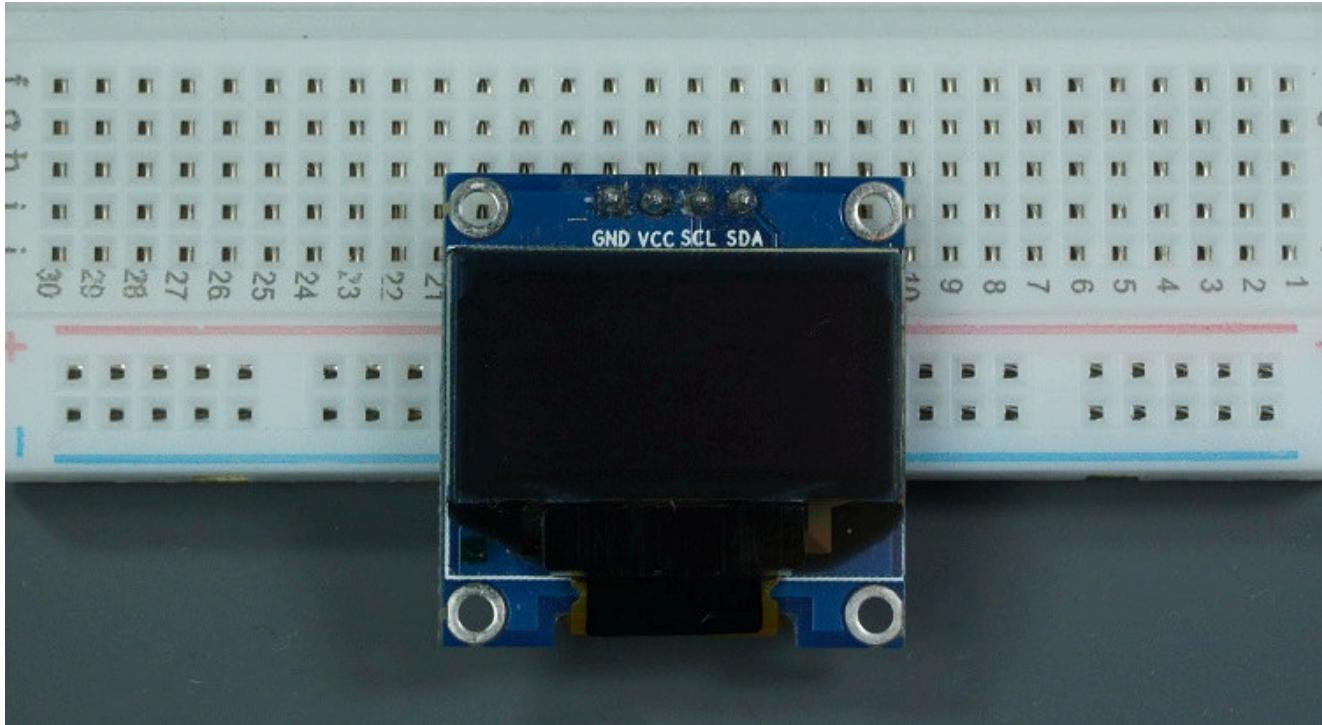
# Guide for I2C OLED Display with Arduino

This article shows how to use the SSD1306 0.96 inch I2C OLED display with the Arduino. We'll show you some features of the OLED display, how to connect it to the Arduino board, and how to write text, draw shapes and display bitmap images. Lastly, we'll build a project example that displays temperature and humidity readings.



## Introducing the 0.96 inch OLED display

The *organic light-emitting diode* (OLED) display that we'll use in this tutorial is the SSD1306 model: a monicolor, 0.96-inch display with 128×64 pixels as shown in the following figure.



The OLED display doesn't require backlight, which results in a very nice contrast in dark environments. Additionally, its pixels consume energy only when they are on, so the OLED display consumes less power when compared with other displays.

The model we're using here has only four pins and communicates with the Arduino using I2C communication protocol. There are models that come with an extra RESET pin. There are also other OLED displays that communicate using SPI communication.

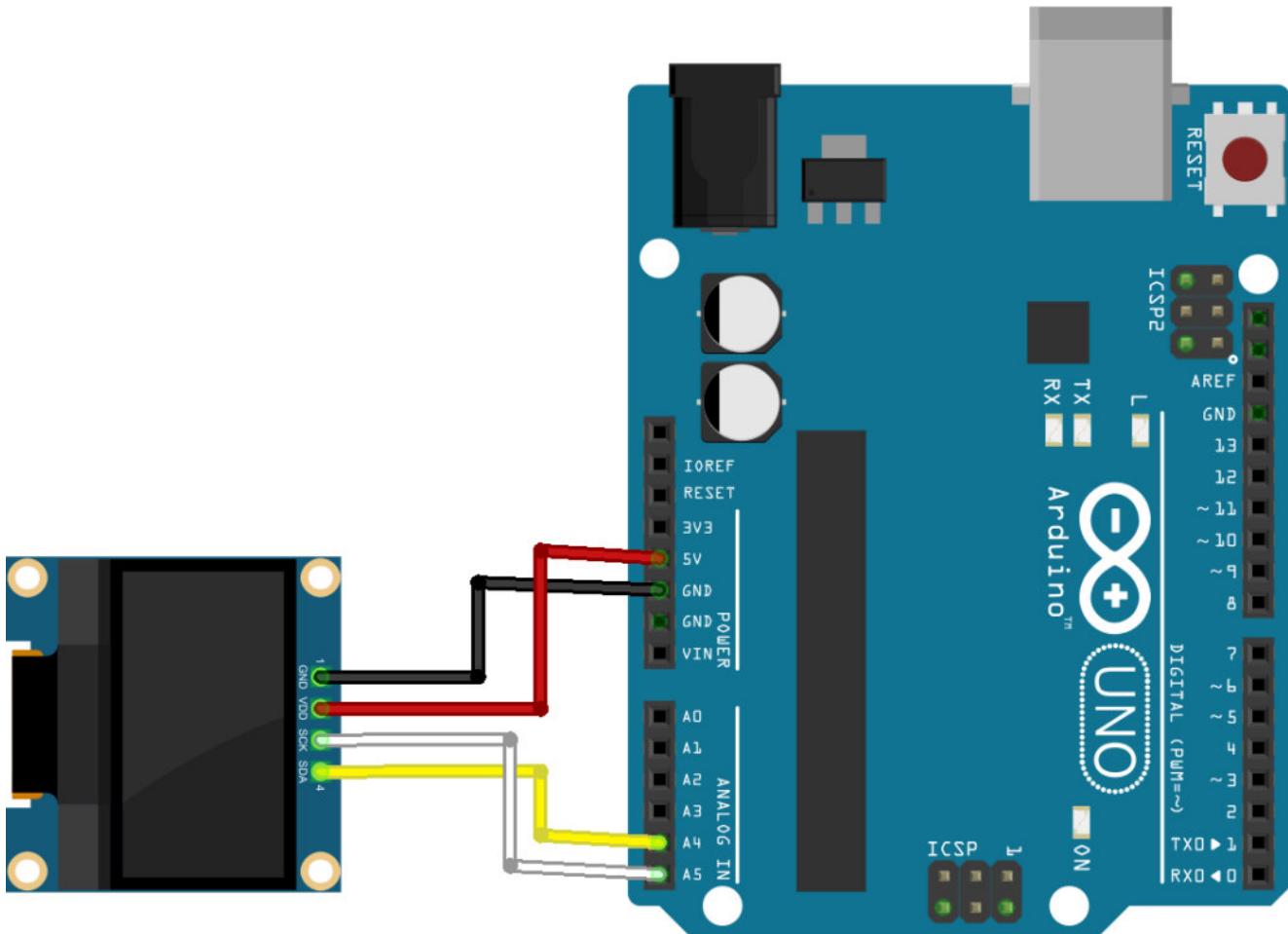
## Pin wiring

Because the OLED display uses I2C communication protocol, wiring is very simple. You just need to connect to the Arduino Uno I2C pins as shown in the table below.

Pin	Wiring to Arduino Uno
Vin	5V
GND	GND
SCL	A5
SDA	A4

If you're using a different Arduino board, make sure you check the correct I2C pins:

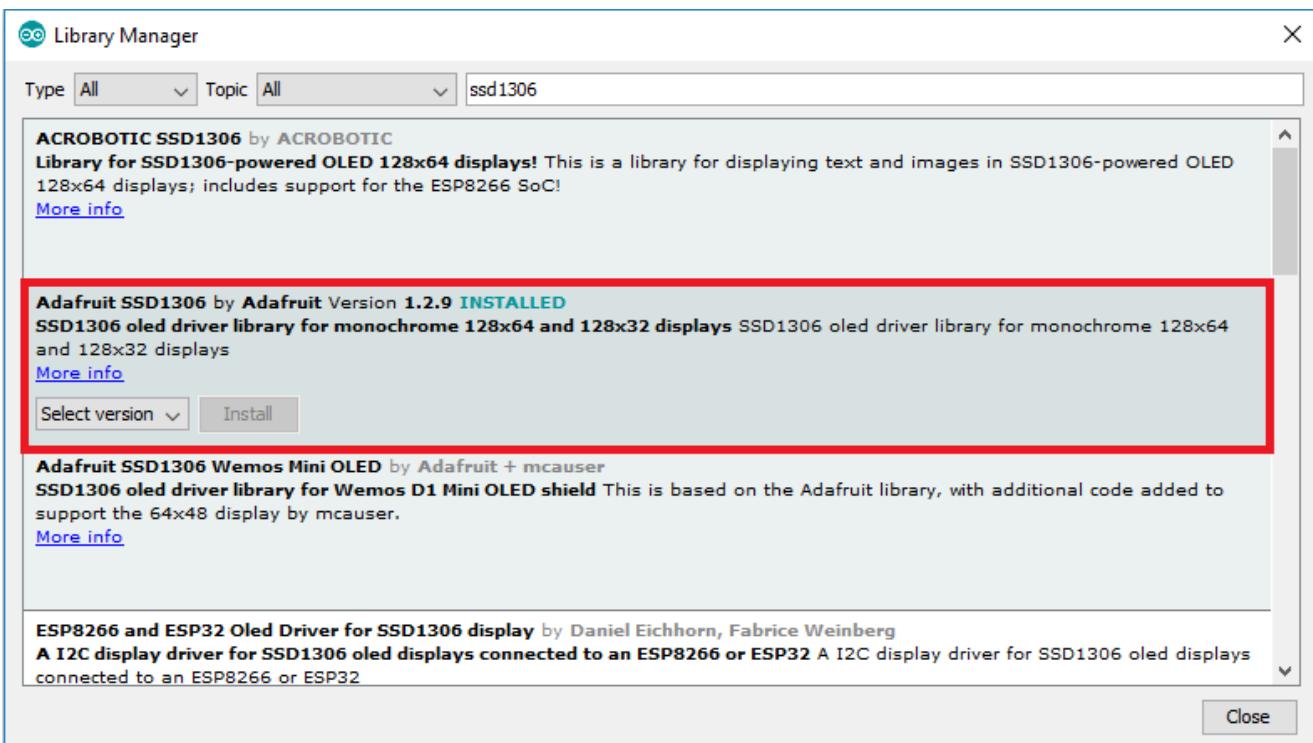
- Nano: SDA (A4); SCL (A5);
- MEGA: SDA (20); SCL (21);
- Leonardo: SDA (20); SCL (21);



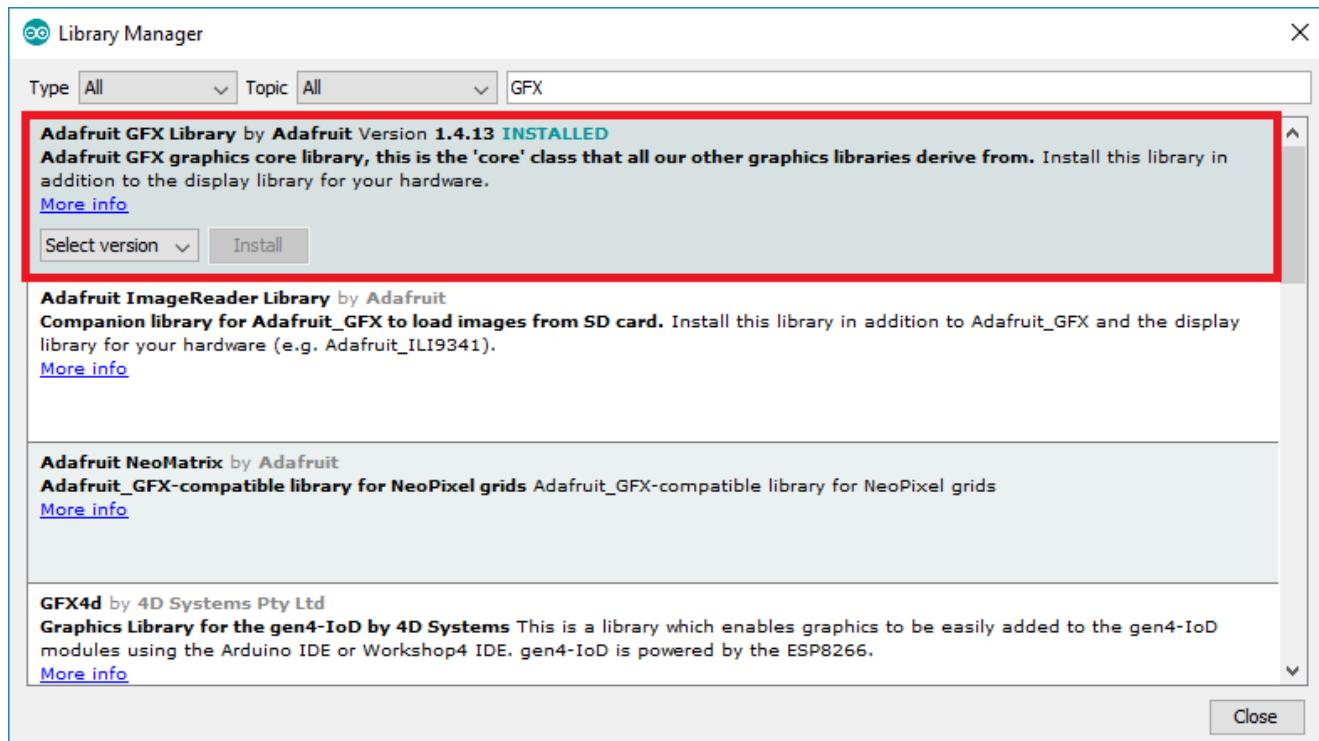
## Libraries

To control the OLED display you need the [adafruit\\_SSD1306.h](#) and the [adafruit\\_GFX.h](#) libraries. Follow the next instructions to install those libraries.

1. Open your Arduino IDE and go to **Sketch > Include Library > Manage Libraries**. The Library Manager should open.
2. Type “**SSD1306**” in the search box and install the SSD1306 library from Adafruit.



3. After installing the SSD1306 library from Adafruit, type “**GFX**” in the search box and install the library.



4. After installing the libraries, restart your Arduino IDE.

## Tips for writing text using these libraries

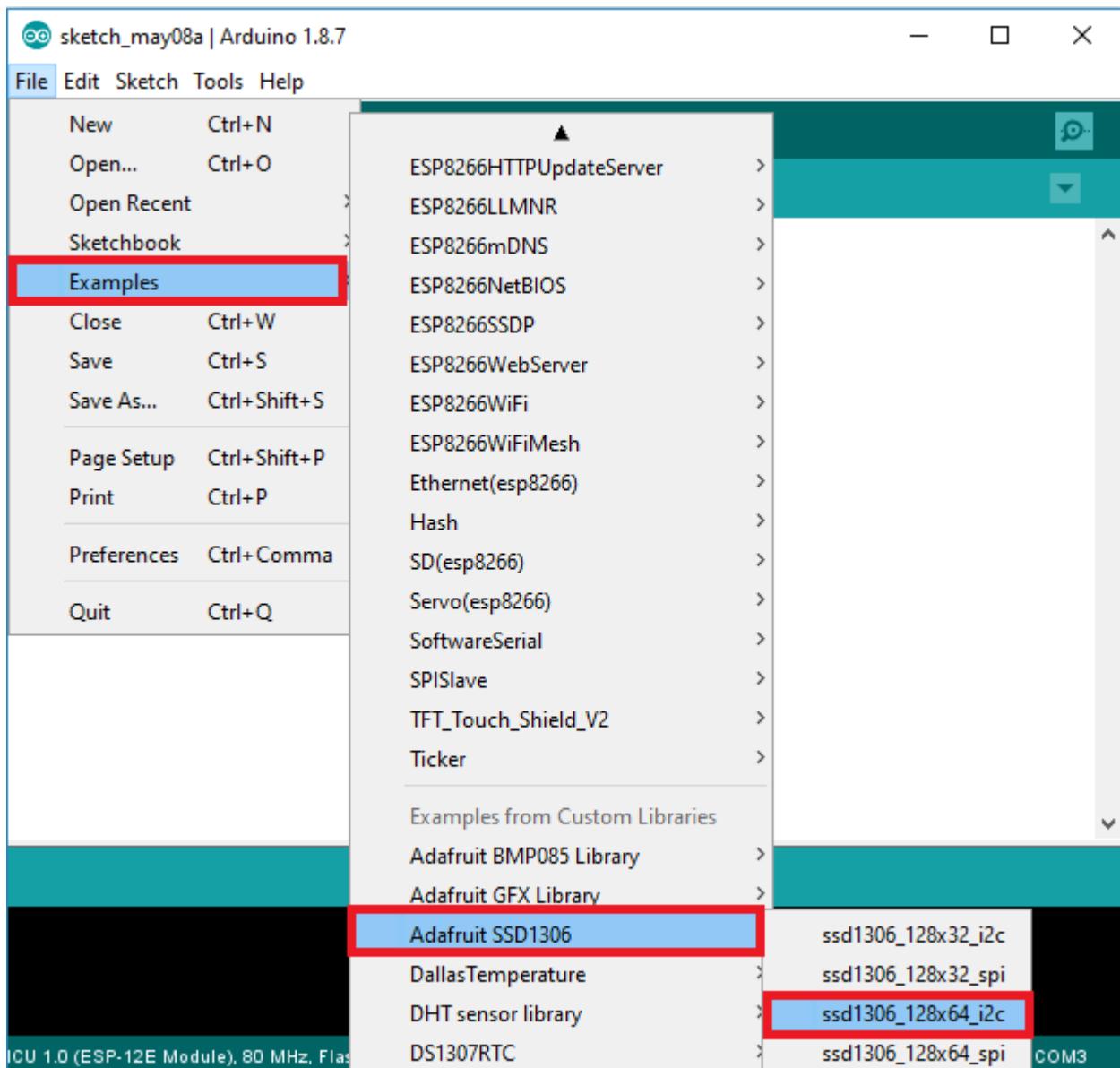
Here's some functions that will help you handle the OLED display library to write text or draw simple graphics.

- `display.clearDisplay()` – all pixels are off
- `display.drawPixel(x, y, color)` – plot a pixel in the x,y coordinates
- `display.setTextSize(n)` – set the font size, supports sizes from 1 to 8
- `display.setCursor(x, y)` – set the coordinates to start writing text
- `display.print("message")` – print the characters at location x,y
- `display.display()` – call this method for the changes to make effect

## Testing the OLED Display

After wiring the OLED display to the Arduino and installing all required libraries, you can use one example from the library to see if everything is working properly.

In your Arduino IDE, go to **File > Examples > Adafruit SSD1306** and select the example for the display you're using.



The following code should load:

```
// display.display(). These examples demonstrate both approaches

testdrawline();           // Draw many lines

testdrawrect();           // Draw rectangles (outlines)

testfillrect();           // Draw rectangles (filled)

testdrawcircle();          // Draw circles (outlines)

testfillcircle();          // Draw circles (filled)

testdrawroundrect(); // Draw rounded rectangles (outlines)

testfillroundrect(); // Draw rounded rectangles (filled)

testdrawtriangle(); // Draw triangles (outlines)

testfilltriangle(); // Draw triangles (filled)

testdrawchar();           // Draw characters of the default font

testdrawstyles();          // Draw 'stylized' characters

testscrolltext();          // Draw scrolling text

testdrawbitmap();          // Draw a small bitmap image
```

[View raw code](#)

If your OLED doesn't have a RESET pin, you should set the `OLED_RESET` variable to -1 as shown below:

```
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino re
```

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

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

Upload the code to your Arduino board. Don't forget to select the right board and COM port in the **Tools** menu.

You should get a series of different animations in the OLED as shown in the following short video.

00:26

If your OLED display is not showing anything:

- Check that the OLED display is properly wired to the Arduino
- Double-check the OLED display I2C address: with the OLED connected to the Arduino, [upload this code](#) and check the I2C address in the Serial Monitor

You should change the OLED address in the following line, if necessary. In our case, the address is 0x3C.

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
```

## Write Text – OLED Display

The Adafruit library for the OLED display comes with several functions to write text. In this section, you'll learn how to write and scroll text using the library functions.

### “Hello, world!” OLED Display

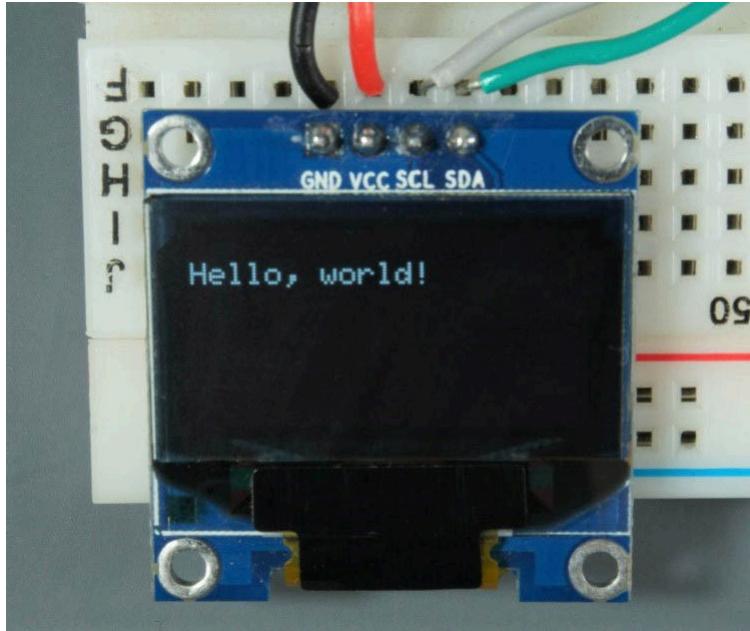
The following sketch displays Hello, world! message in the OLED display.

```
*****  
Rui Santos  
Complete project details at https://randomnerdtutorials.com  
*****  
  
#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  
  
// Declaration for an SSD1306 display connected to I2C (SDA, SC  
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1  
  
void setup() {  
    Serial.begin(115200);  
  
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0  
        Serial.println(F("SSD1306 allocation failed"));  
        for(;;);  
    }  
    delay(2000);  
    display.clearDisplay();
```

```
display.setTextSize(1);
```

[View raw code](#)

After uploading the code, this is what you'll get in your OLED:



Let's take a quick look on how the code works.

## Importing libraries

First, you need to import the necessary libraries. The `Wire` library to use I2C and the Adafruit libraries to write to the display: `Adafruit_GFX` and `Adafruit_SSD1306`.

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

## Initialize the OLED display

Then, you define your OLED width and height. In this example, we're using a 128×64 OLED display. If you're using other sizes, you can change that in the `SCREEN_WIDTH`, and `SCREEN_HEIGHT` variables.

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

Then, initialize a display object with the width and height defined earlier with I2C communication protocol ( `&Wire` ).

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

The (-1) parameter means that your OLED display doesn't have a RESET pin. If your OLED display does have a RESET pin, it should be connected to a GPIO. In that case, you should pass the GPIO number as a parameter.

In the `setup()`, initialize the Serial Monitor at a baud rate of 115200 for debugging purposes.

```
Serial.begin(115200);
```

Initialize the OLED display with the `begin()` method as follows:

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    for(;;); // Don't proceed, loop forever
}
```

This snippet also prints a message on the Serial Monitor, in case we're not able to connect to the display.

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

In case you're using a different OLED display, you may need to change the OLED address. In our case, the address is `0x3C`.

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
```

If this address doesn't work, you can run an I2C scanner sketch to find your OLED address. You can find the I2C scanner sketch [here](#).

After initializing the display, add a two second delay, so that the OLED has enough time to initialize before writing text:

```
delay(2000);
```

## Clear display, set font size, color and write text

After initializing the display, clear the display buffer with the `clearDisplay()` method:

```
display.clearDisplay();
```

Before writing text, you need to set the text size, color and where the text will be displayed in the OLED.

Set the font size using the `setTextSize()` method:

```
display.setTextSize(1);
```

Set the font color with the `setTextColor()` method:

```
display.setTextColor(WHITE);
```

`WHITE` sets white font and black background.

Define the position where the text starts using the `setCursor(x,y)` method. In this case, we're setting the text to start at the (0,10) coordinates.

```
display.setCursor(0,10);
```

Finally, you can send the text to the display using the `println()` method, as follows:

```
display.println("Hello, world!");
```

Then, you need to call the `display()` method to actually display the text on the screen.

```
display.display();
```

## Scrolling Text

The Adafruit OLED library provides useful methods to easily scroll text.

- `startscrollright(0x00, 0x0F)` : scroll text from left to right
- `startscrollleft(0x00, 0x0F)` : scroll text from right to left
- `startscrolldiagright(0x00, 0x07)` : scroll text from left bottom corner to right upper corner
- `startscrolldiagleft(0x00, 0x07)` : scroll text from right bottom corner to left upper corner

The following sketch implements those methods.

```
*****
Rui Santos
Complete project details at https://randomnerdtutorials.com
*****  

#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

// Declaration for an SSD1306 display connected to I2C (SDA, SC
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1

void setup() {
    Serial.begin(115200);

    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0
        Serial.println(F("SSD1306 allocation failed"));
        for(;;);
    }
    delay(2000);
    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(WHITE);
```

[View raw code](#)

The text scrolls as shown in the following short video.

## Using Other Fonts – OLED Display

The Adafruit GFX library allows us to use some alternate fonts besides the built-in fonts. It allows you to chose between Serif, Sans, and Mono. Each font is available in bold, italic and in different sizes.

The sizes are set by the actual font. So, the `setTextSize()` method doesn't work with these fonts. The fonts are available in 9, 12, 18 and 24 point sizes and also contain 7-bit characters (ASCII codes) (described as 7b in the font name).

You can chose from the next selection of fonts:

FreeMono12pt7b.h	FreeSansBoldOblique12pt7b.h
FreeMono18pt7b.h	FreeSansBoldOblique18pt7b.h
FreeMono24pt7b.h	FreeSansBoldOblique24pt7b.h
FreeMono9pt7b.h	FreeSansBoldOblique9pt7b.h
FreeMonoBold12pt7b.h	FreeSansOblique12pt7b.h
FreeMonoBold18pt7b.h	FreeSansOblique18pt7b.h
FreeMonoBold24pt7b.h	FreeSansOblique24pt7b.h
FreeMonoBold9pt7b.h	FreeSansOblique9pt7b.h
FreeMonoBoldOblique12pt7b.h	FreeSerif12pt7b.h
FreeMonoBoldOblique18pt7b.h	FreeSerif18pt7b.h
FreeMonoBoldOblique24pt7b.h	FreeSerif24pt7b.h
FreeMonoBoldOblique9pt7b.h	FreeSerif9pt7b.h
FreeMonoOblique12pt7b.h	FreeSerifBold12pt7b.h
FreeMonoOblique18pt7b.h	FreeSerifBold18pt7b.h
FreeMonoOblique24pt7b.h	FreeSerifBold24pt7b.h

FreeMonoOblique9pt7b.h	FreeSerifBold9pt7b.h
FreeSans12pt7b.h	FreeSerifBoldItalic12pt7b.h
FreeSans18pt7b.h	FreeSerifBoldItalic18pt7b.h
FreeSans24pt7b.h	FreeSerifBoldItalic24pt7b.h
FreeSans9pt7b.h	FreeSerifBoldItalic9pt7b.h
FreeSansBold12pt7b.h	FreeSerifItalic12pt7b.h
FreeSansBold18pt7b.h	FreeSerifItalic18pt7b.h
FreeSansBold24pt7b.h	FreeSerifItalic24pt7b.h
FreeSansBold9pt7b.h	FreeSerifItalic9pt7b.h

The fonts that work better with the OLED display are the 9 and 12 points size.

To use one of those fonts, first you need to include it in your sketch, for example:

```
#include <Fonts/FreeSerif12pt7b.h>
```

Next, you just need to use the `setFont()` method and pass as argument, the specified font:

```
display.setFont(&FreeSerif12pt7b);
```

After specifying the font, all methods to write text will use that font. To get back to use the original font, you just need to call the `setFont()` method with no arguments:

```
display.setFont();
```

Upload the next sketch to your board:

```
*****
Rui Santos
Complete project details at https://randomnerdtutorials.com
*****
```

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

```
#include <Fonts/FreeSerif9pt7b.h>

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

// Declaration for an SSD1306 display connected to I2C (SDA, SC
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1

void setup() {
    Serial.begin(115200);

    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println("SSD1306 allocation failed");
        for(;;);
    }
    delay(2000);

    display.setFont(&FreeSerif9pt7b);
    display.clearDisplay();
```

[View raw code](#)

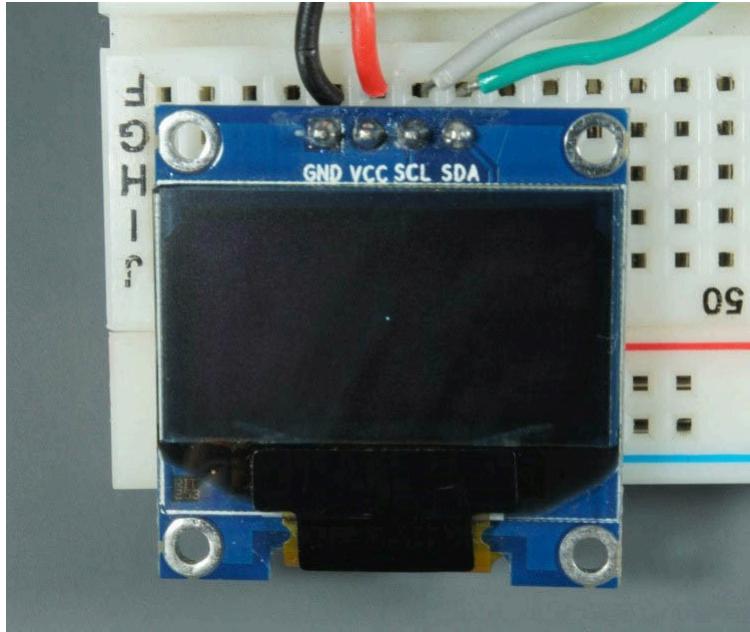
Now, your display prints the “Hello, world!” message in FreeSerif font.



## Draw Shapes in the OLED Display

The Adafruit OLED library provides useful methods to draw pixels, lines and shapes. Let's take a quick look at those methods.

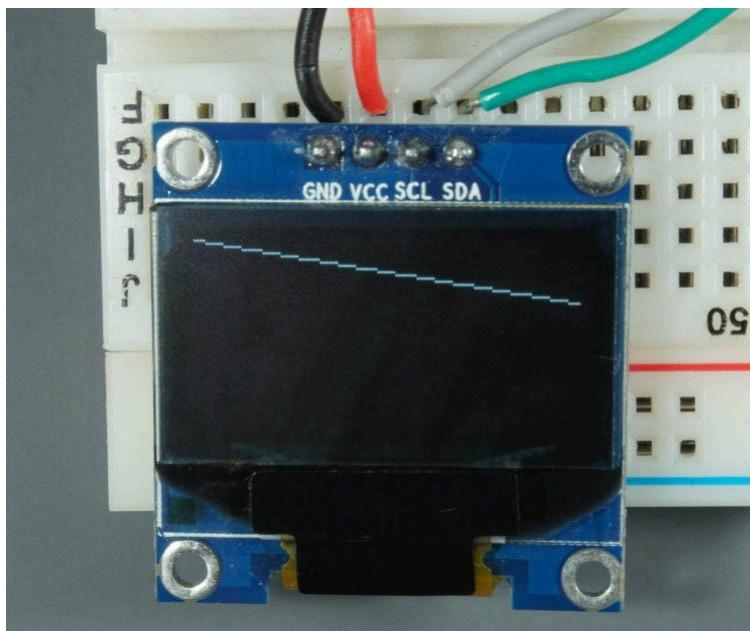
## Draw a pixel



To draw a pixel in the OLED display, you can use the `drawPixel(x, y, color)` method that accepts as arguments the x and y coordinates where the pixel appears, and color. For example:

```
display.drawPixel(64, 32, WHITE);
```

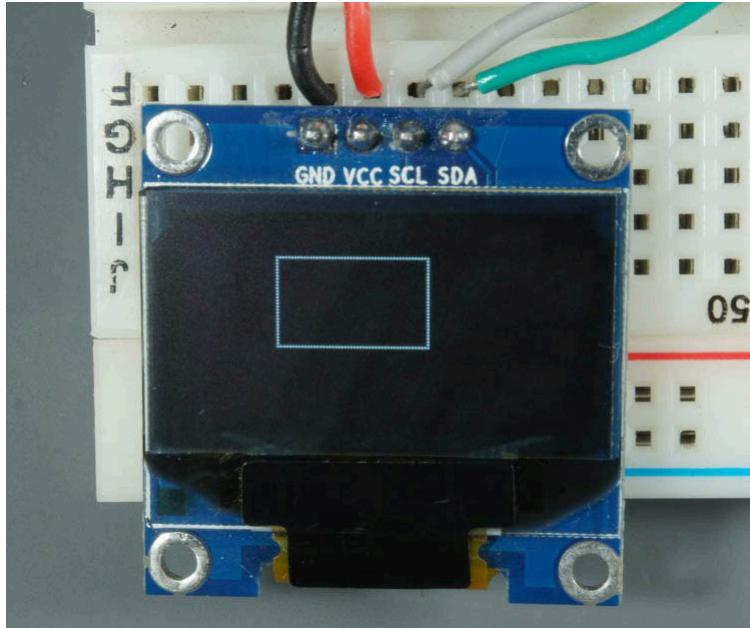
## Draw a line



Use the `drawLine(x1, y1, x2, y2, color)` method to create a line. The (x1, y1) coordinates indicate the start of the line, and the (x2, y2) coordinates indicates where the line ends. For example:

```
display.drawLine(0, 0, 127, 20, WHITE);
```

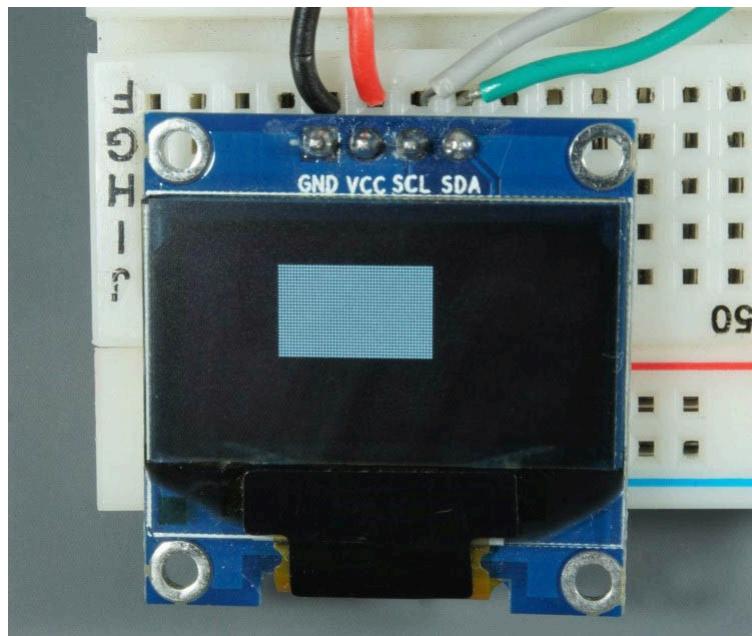
## Draw a rectangle



The `drawRect(x, y, width, height, color)` provides an easy way to draw a rectangle. The (x, y) coordinates indicate the top left corner of the rectangle. Then, you need to specify the width, height and color:

```
display.drawRect(10, 10, 50, 30, WHITE);
```

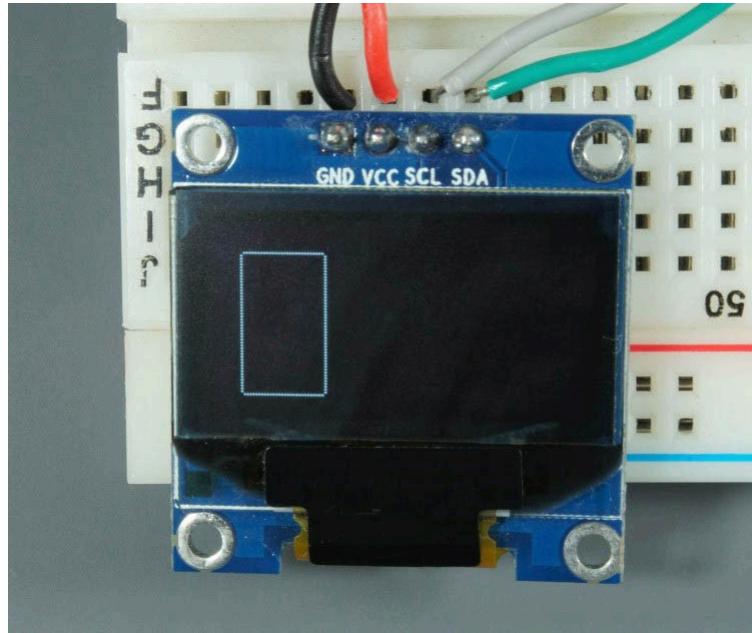
You can use the `fillRect(x, y, width, height, color)` to draw a filled rectangle. This method accepts the same arguments as `drawRect()`.



The library also provides methods to displays rectangles with round corners:

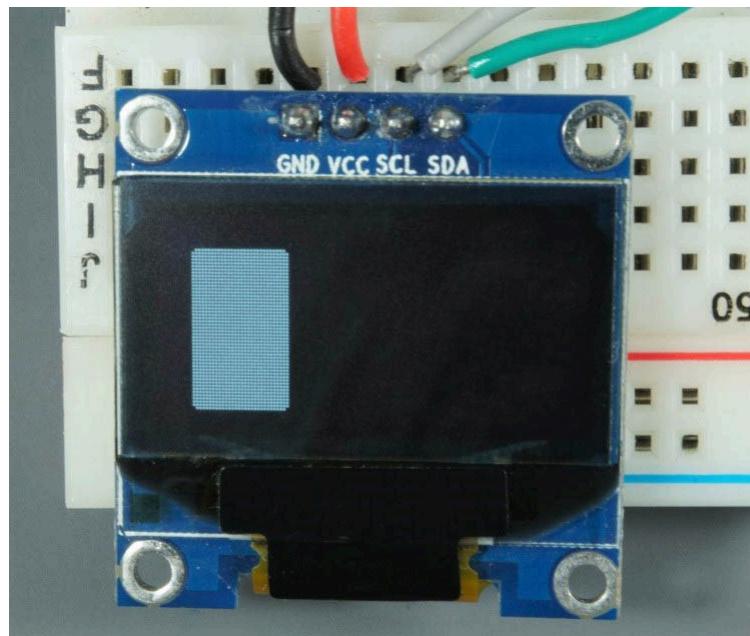
`drawRoundRect()` and `fillRoundRect()`. These methods accepts the same arguments as previous methods plus the radius of the corner. For example:

```
display.drawRoundRect(10, 10, 30, 50, 2, WHITE);
```

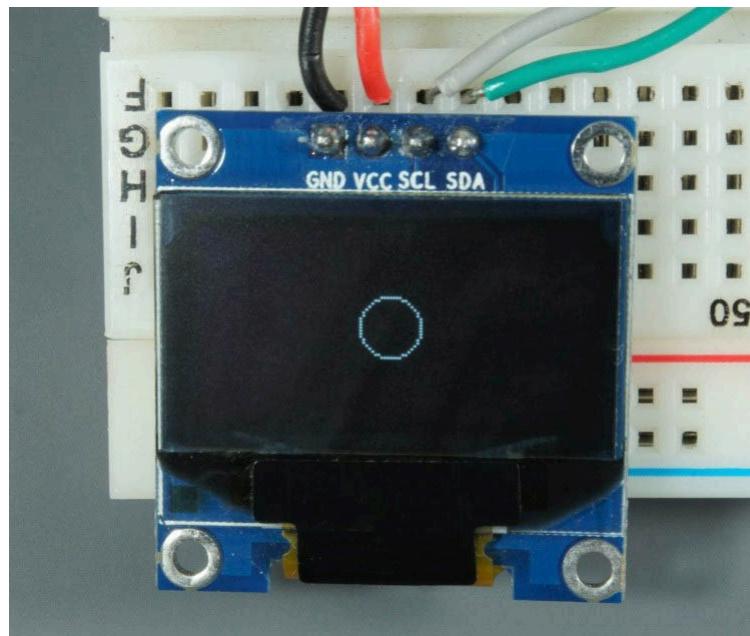


Or a filled round rectangle:

```
display.fillRoundRect(10, 10, 30, 50, 2, WHITE);
```



## Draw a circle

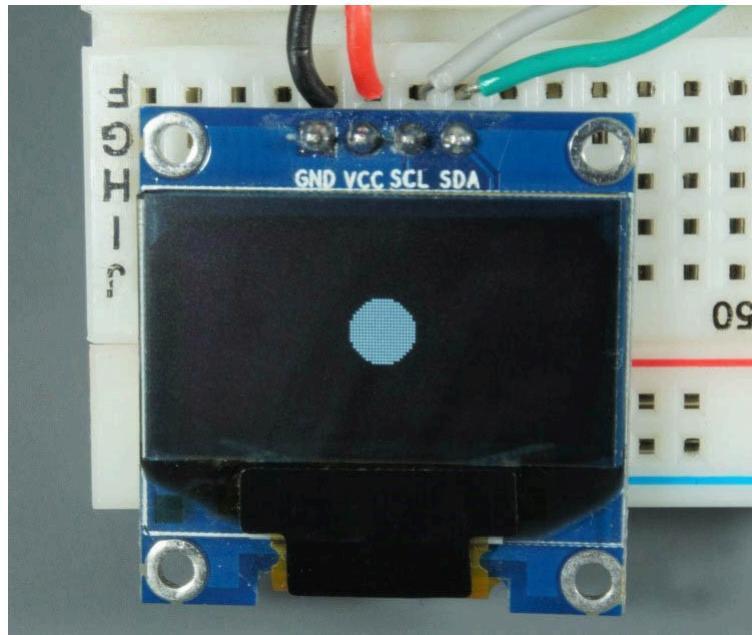


To draw a circle use the `drawCircle(x, y, radius, color)` method. The (x,y) coordinates indicate the center of the circle. You should also pass the radius as an argument. For example:

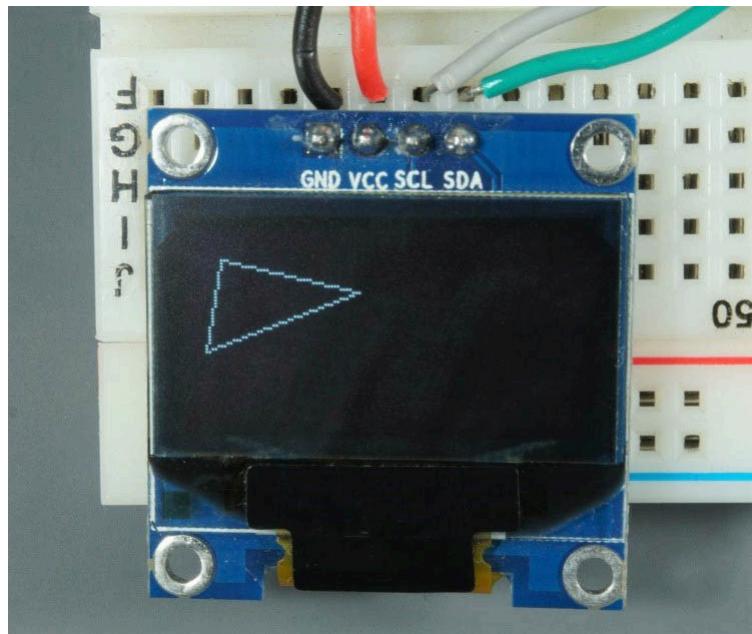
```
display.drawCircle(64, 32, 10, WHITE);
```

In the same way, to build a filled circle, use the `fillCircle()` method with the same arguments:

```
display.fillCircle(64, 32, 10, WHITE);
```



## Draw a triangle

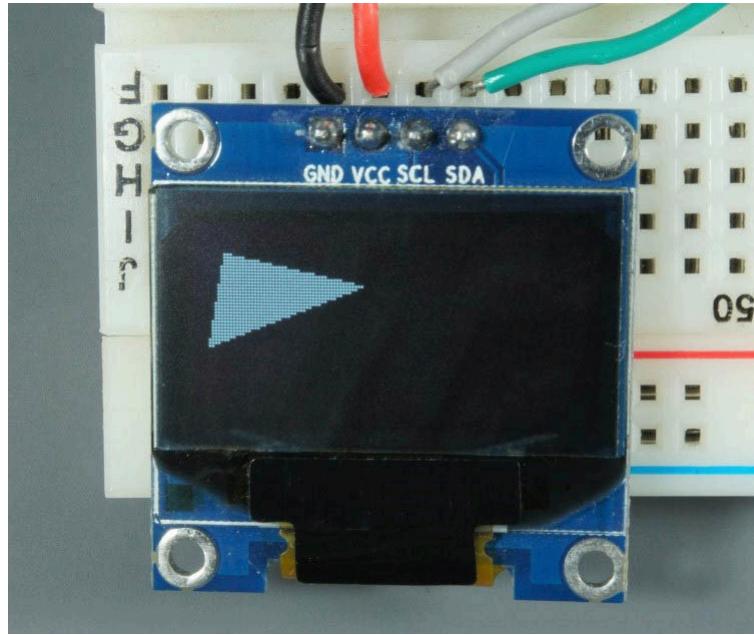


Use the `drawTriangle(x1, y1, x2, y2, x3, y3, color)` method to build a triangle. This method accepts as arguments the coordinates of each corner and the color.

```
display.drawTriangle(10, 10, 55, 20, 5, 40, WHITE);
```

Use the `fillTriangle()` method to draw a filled triangle.

```
display.fillTriangle(10, 10, 55, 20, 5, 40, WHITE);
```



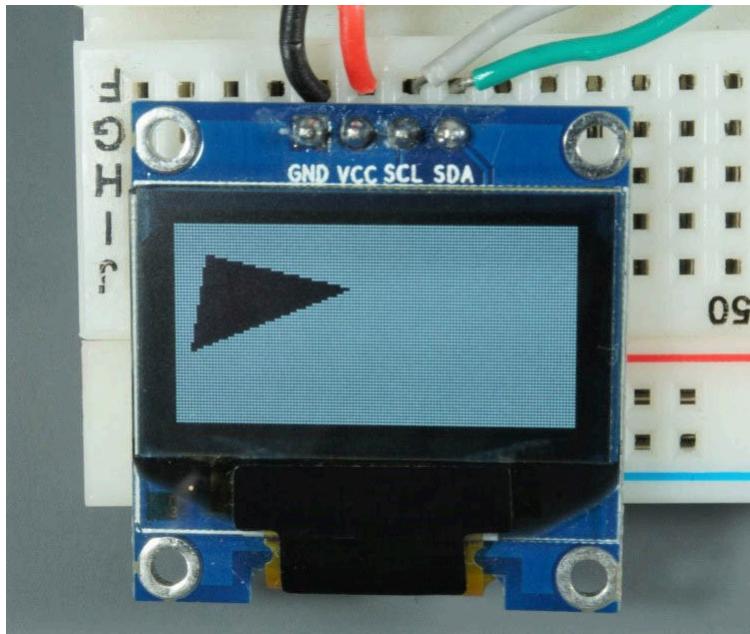
## Invert

The library provides an additional method that you can use with shapes or text: the `invertDisplay()` method. Pass `true` as argument to invert the colors of the screen or `false` to get back to the original colors.

If you call the following command after defining the triangle:

```
display.invertDisplay(true);
```

You'll get an inverted triangle as follows:



## Code – Draw Shapes

Upload the following sketch that implements each snippet of code we've covered previously and goes through all the shapes.

```
*****  
Rui Santos  
Complete project details at https://randomnerdtutorials.com  
*****  
  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
  
#define SCREEN_WIDTH 128  
#define SCREEN_HEIGHT 64  
  
// Declaration for an SSD1306 display connected to I2C (SDA, SC  
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1  
  
void setup() {  
    Serial.begin(115200);  
  
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {  
        Serial.println(F("SSD1306 allocation failed"));  
        for(;;);  
    }  
}
```

```
}
```

```
delay(2000); // Pause for 2 seconds
```

```
// Clear the buffer
```

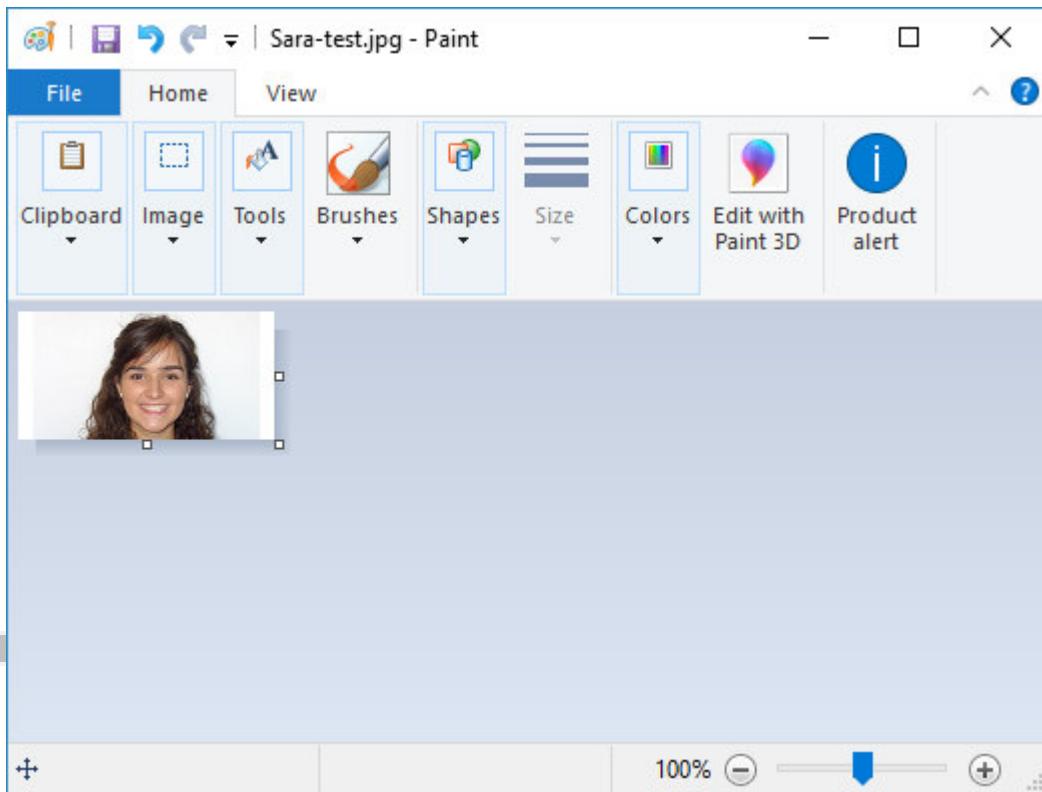
```
display.clearDisplay();
```

[View raw code](#)

## Display Bitmap Images in the OLED

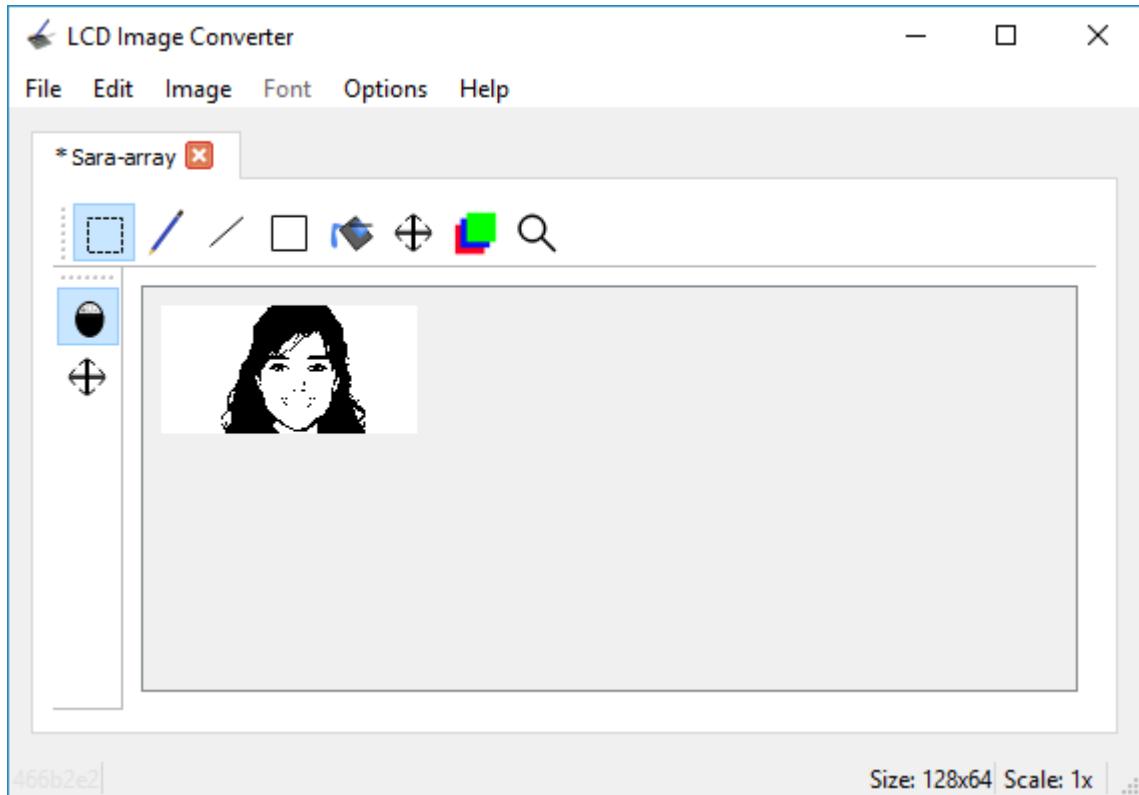
You can display 128×64 bitmap monocolored images on the OLED display.

First, use an imaging program to resize a photo or picture and save it as monochrome bitmap. If you're on a Windows PC, you can use Paint.



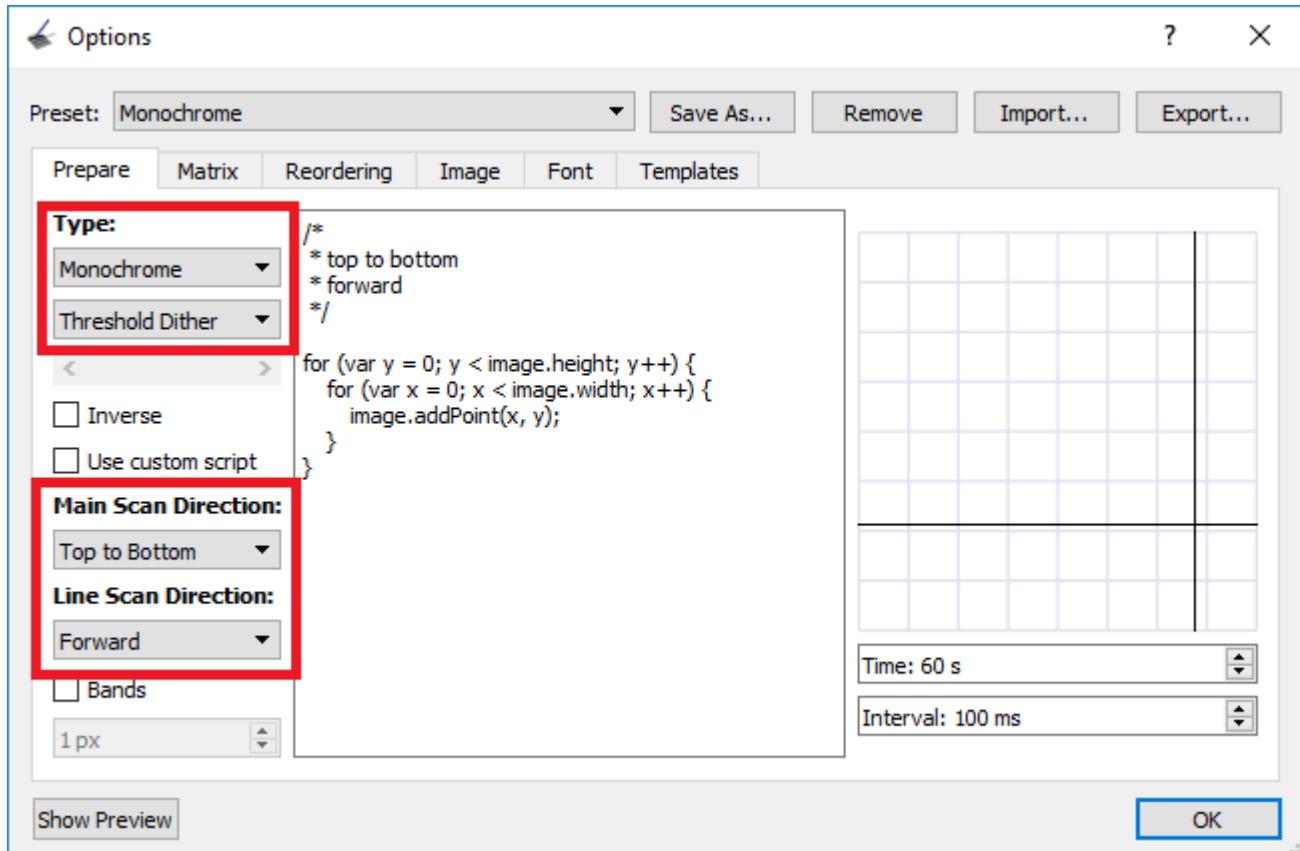
Then, use a Image to C Array converter to convert the image into an array. I've used [LCD Image Converter](#).

Run the program and start with a new image. Go to **Image > Import** and select the bitmap image you've created earlier.



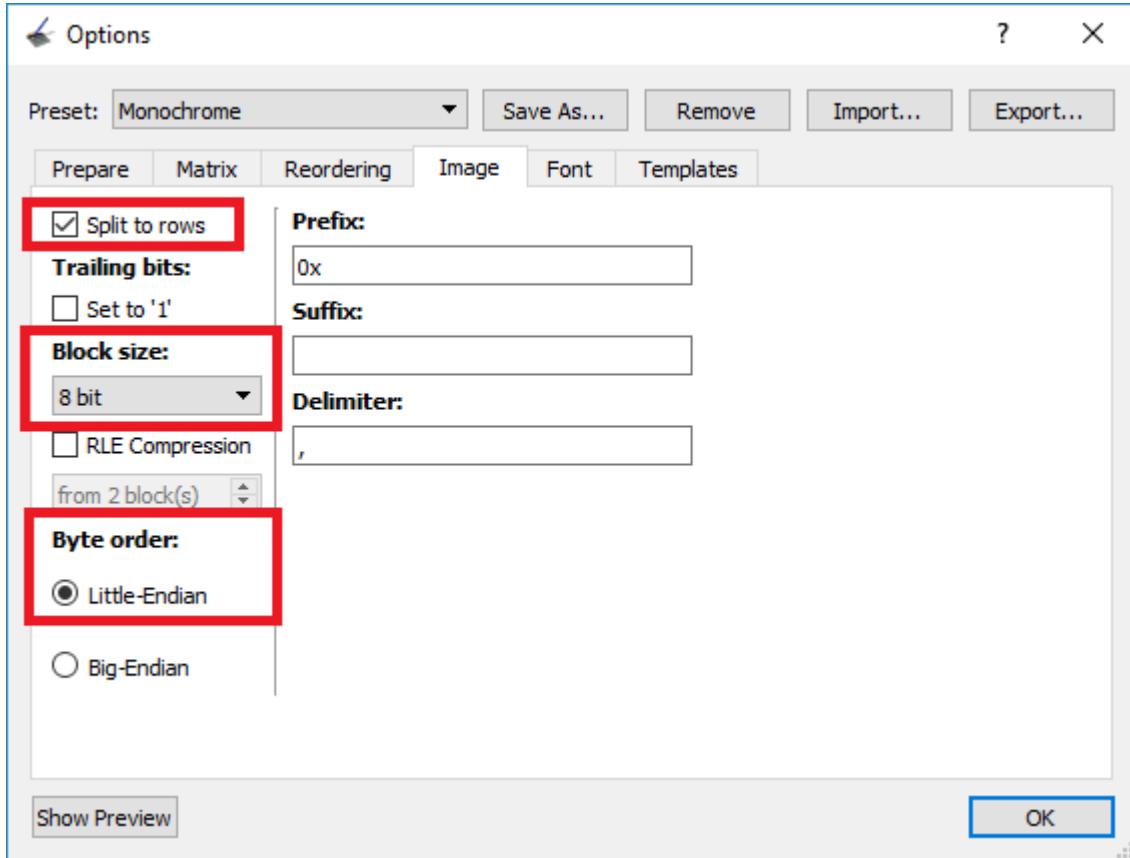
Go to **Options > Conversion** and in the **Prepare** tab, select the following options:

- **Type:** Monochrome, Threshold Dither
- **Main Scan Direction:** Top to Bottom
- **Line Scan Direction:** Forward



Go to the Image tab and select the following options:

- Split to rows
- **Block size:** 8 bit
- **Byte order:** Little-Endian



Then, click **OK**. Finally, in the main menu, go to **File > Convert**. A new file with .c extension should be saved. That file contains the C array for the image. Open that file with a text editor, and copy the array.

In our case, this is the array that we get:

```
static const uint8_t image_data_Saraarray[1024] = {
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00,
```

```
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x0a,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x14, 0x9e,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x36, 0x3f,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x6d, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0xfb, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x03, 0xd7, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x07, 0xef, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x0f, 0xdf, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x0f, 0xbf, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x1d, 0x7f, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xf0, 0x01, 0x1b, 0xff, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xf0, 0x02, 0xa7, 0xff, 0xff,
```

[View raw code](#)

Copy your array to the sketch. Then, to display the array, use the `drawBitmap()` method that accepts the following arguments (x, y, image array, image width, image height, rotation). The (x, y) coordinates define where the image starts to be displayed.

Copy the code below to display your bitmap image in the OLED.

```
*****  
Rui Santos  
Complete project details at https://randomnerdtutorials.com  
*****/
```

```
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
  
#define SCREEN_WIDTH 128
```

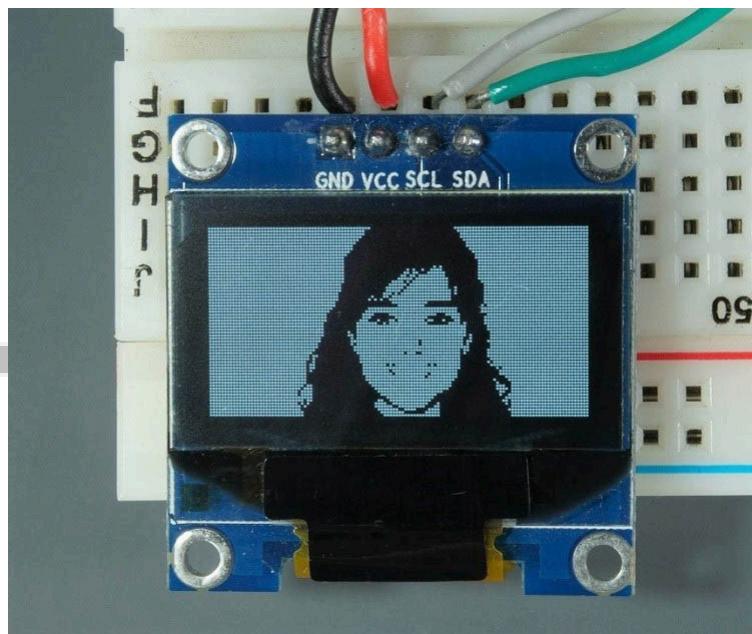
```
#define SCREEN_HEIGHT 64
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1

static const unsigned char PROGMEM image_data_Saraarray[] = {
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00,
    0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00,
```

[View raw code](#)

After uploading the code, this is what we get on the display.



## Display Temperature and Humidity in the OLED Display with Arduino

In this section we'll build a project that displays temperature and humidity readings on the OLED display. We'll get temperature and humidity using the [DHT11 temperature and humidity sensor](#). If you're not familiar with the DHT11 sensor, read the following article:

- [Complete Guide for DHT11/DHT22 Humidity and Temperature Sensor With Arduino](#)

## Parts required

To complete this project you need the following components:

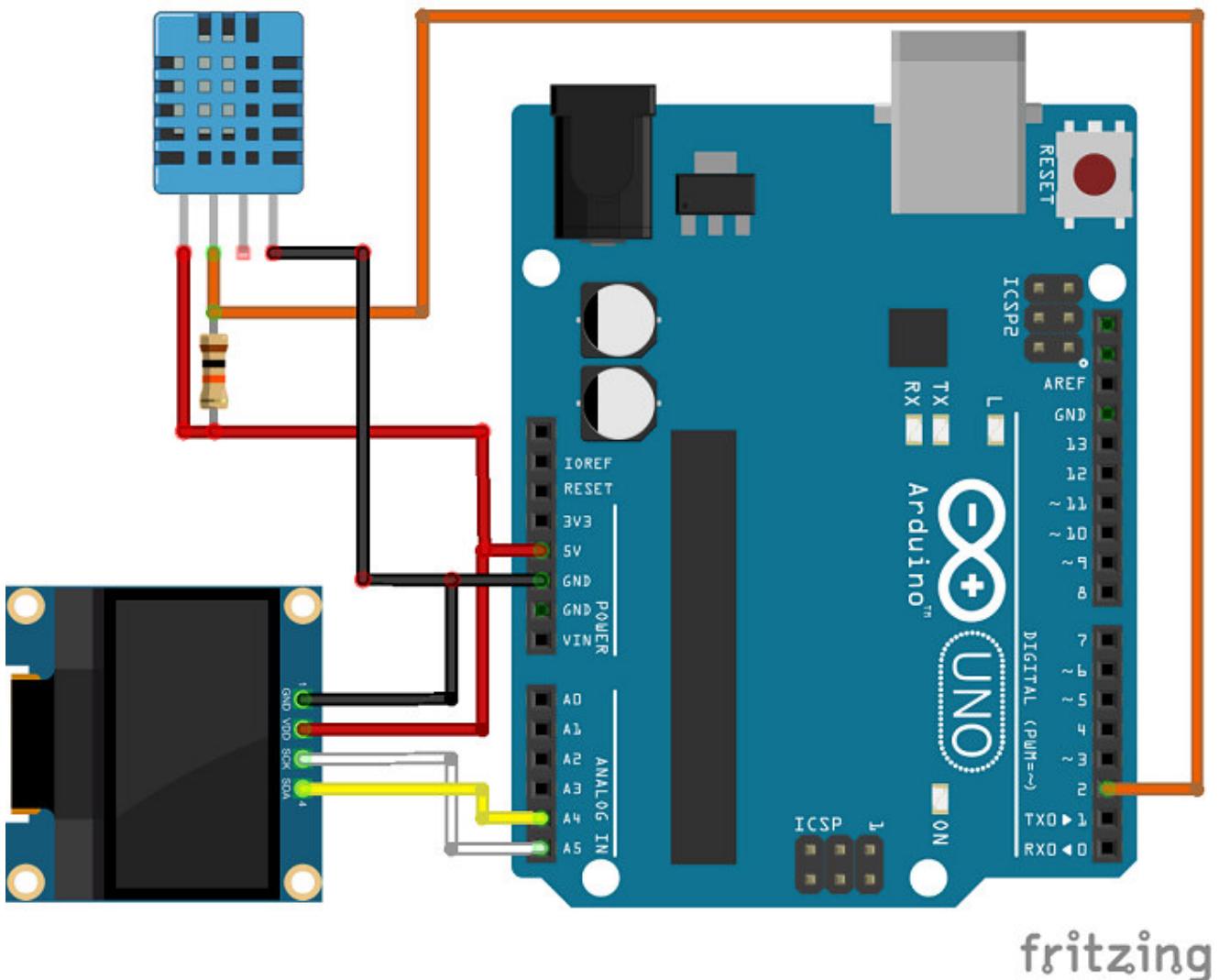
- [0.96 inch OLED display](#)
- [Arduino](#) – Read [Best Arduino Starter Kits](#)
- [DHT11](#) or [DHT22](#) temperature and humidity sensor
- Breadboard
- [4.7k Ohm resistor](#) (or [10k Ohm resistor](#))
- Jumper wires

You can use the preceding links or go directly to [MakerAdvisor.com/tools](#) to find all the parts for your projects at the best price!



## Schematic

Assemble the circuit by following the next schematic diagram.



fritzing

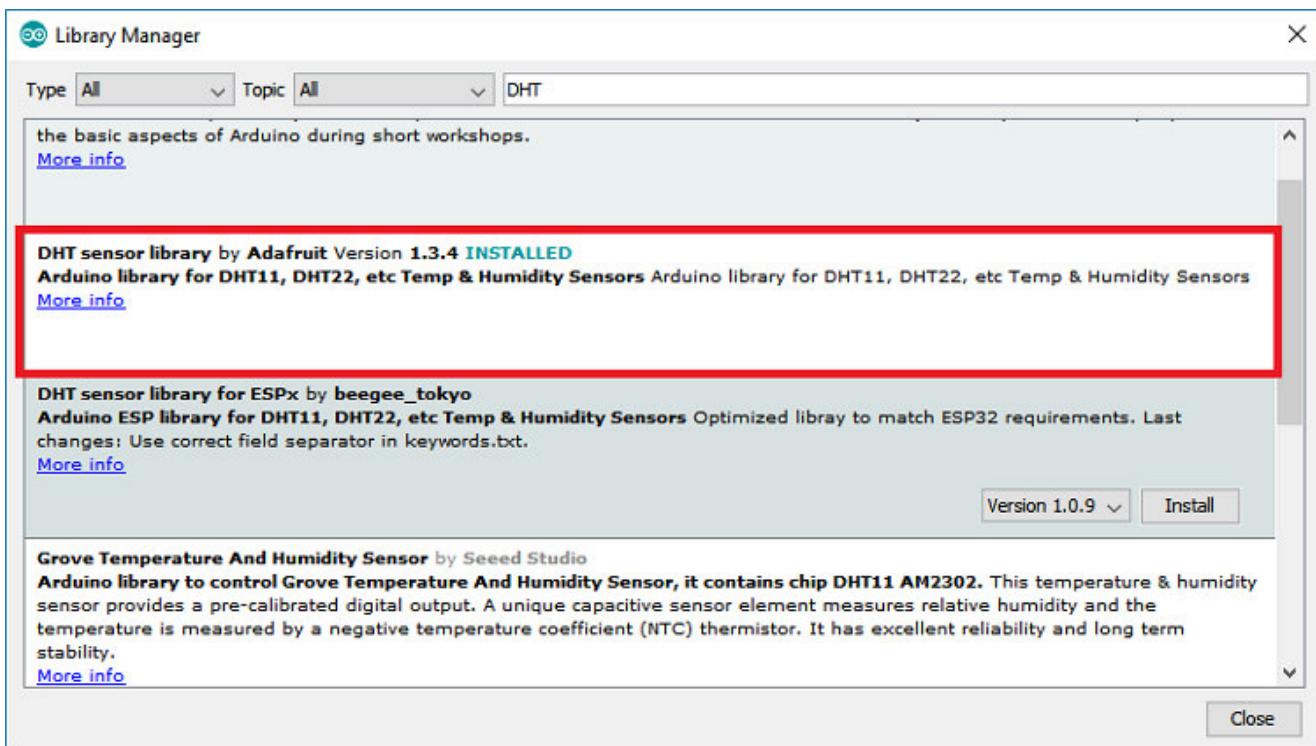
**Note:** if you're using a module with a DHT sensor, it normally comes with only three pins. The pins should be labeled so that you know how to wire them. Additionally, many of these modules already come with an internal pull up resistor, so you don't need to add one to the circuit.

## Installing Libraries

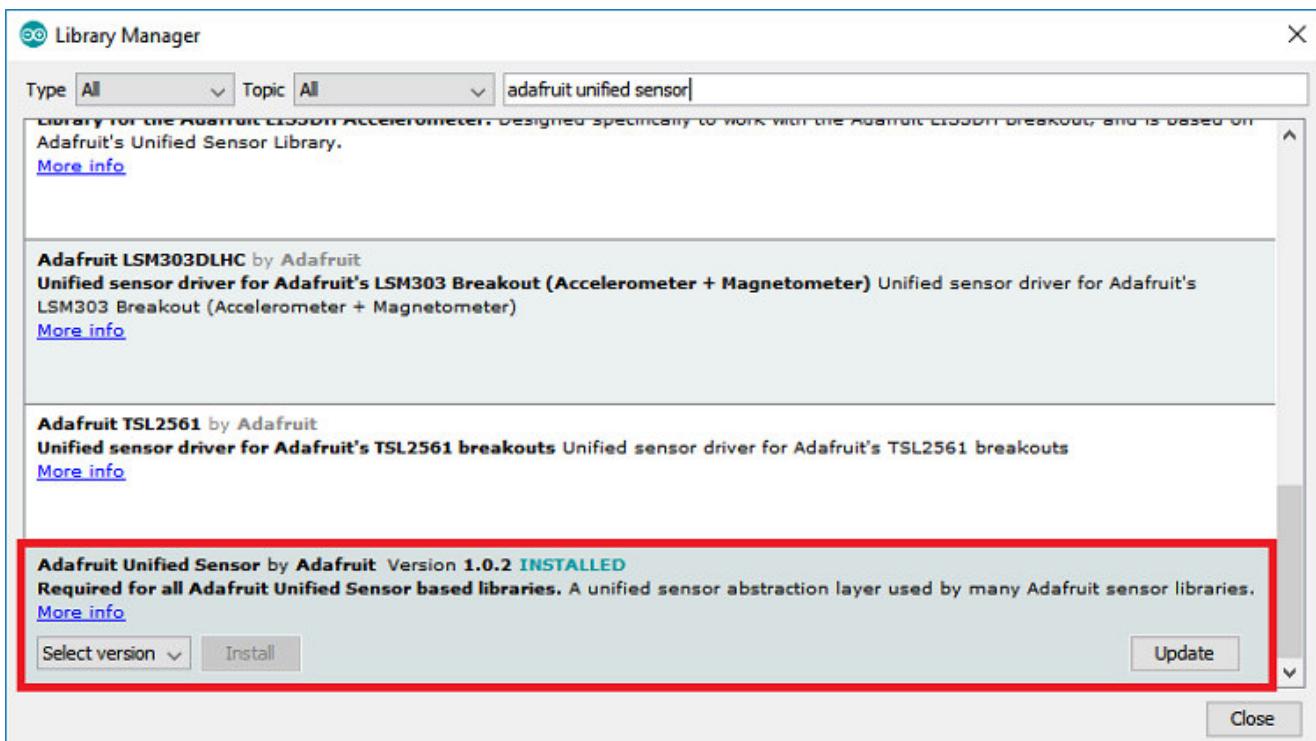
Before proceeding, make sure you have installed the “[adafruit\\_GFX.h](#)” and the “[adafruit\\_SSD1306.h](#)” libraries to control the OLED display.

For this project you also need two additional libraries to read from the DHT sensor: the [DHT](#) library and the [Adafruit\\_Sensor](#) library. Follow the next steps to install those libraries

1. Open your Arduino IDE and go to **Sketch > Include Library > Manage Libraries**. The Library Manager should open.
2. Search for “**DHT**” on the Search box and install the DHT library from Adafruit.



**3.** After installing the DHT library from Adafruit, type “**Adafruit Unified Sensor**” in the search box. Scroll all the way down to find the library and install it.



**4.** Restart your Arduino IDE.

## Code

After installing all the necessary libraries, you can upload the following code.

```
*****
Rui Santos
Complete project details at https://randomnerdtutorials.com
*****
```

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

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

// Declaration for an SSD1306 display connected to I2C (SDA, SC
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1

#define DHTPIN 2      // Digital pin connected to the DHT sensor

// Uncomment the type of sensor in use:
#define DHTTYPE     DHT11      // DHT 11
//#define DHTTYPE     DHT22      // DHT 22 (AM2302)
//#define DHTTYPE     DHT21      // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
```

[View raw code](#)

## How the Code Works

Read this section if you want to learn how the code works. Otherwise, you can skip to the “Demonstration” section.

### Importing libraries

The code starts by including the necessary libraries. The `Wire`, `Adafruit_GFX` and `Adafruit_SSD1306` are used to interface with the OLED display. The `Adafruit_Sensor` and the `DHT` libraries are used to interface with the DHT22 or DHT11 sensors.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

## Create a display object

Then, define your OLED display dimensions. In this case, we're using a 128×64 pixel display.

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

Then, initialize a display object with the width and height defined earlier with I2C communication protocol (`&Wire`).

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```



The (-1) parameter means that your OLED display doesn't have a RESET pin. If your OLED display does have a RESET pin, it should be connected to a GPIO. In that case, you should pass the GPIO number as a parameter.

## Create a DHT object

Then, define the DHT sensor type you're using. If you're using a DHT11 you don't need to change anything on the code. If you're using another sensor, just uncomment the sensor you're using and comment the others.

```
#define DHTTYPE DHT11      // DHT 11
//#define DHTTYPE DHT22      // DHT 22 (AM2302)
//#define DHTTYPE DHT21      // DHT 21 (AM2301)
```

Initialize a DHT sensor object with the pin and type defined earlier.

```
DHT dht(DHTPIN, DHTTYPE);
```

## setup()

In the `setup()`, initialize the serial monitor for debugging purposes.

```
Serial.begin(115200);
```

Initialize the DHT sensor:

```
dht.begin();
```

Then, initialize the OLED display.

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
}
```



In this case, the address of the OLED display we're using is **0x3C**. If this address doesn't work, you can run an I2C scanner sketch to find your OLED address. You can [find the I2C scanner sketch here](#).

Add a delay to give time for the display to initialize, clear the display and set the text color to white:

```
delay(2000);
display.clearDisplay();
display.setTextColor(WHITE)
```

In the `loop()` is where we read the sensor and display the temperature and humidity on the display.

## Get temperature and humidity readings from DHT

The temperature and humidity are saved on the `t` and `h` variables, respectively. Reading temperature and humidity is as simple as using the `readTemperature()` and `readHumidity()` methods on the `dht` object.

```
float t = dht.readTemperature();
float h = dht.readHumidity();
```

In case we are not able to get the readings, display an error message:

```
if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
}
```

If you get that error message, read our troubleshooting guide: [how to fix “Failed to read from DHT sensor”](#).

## Display sensor readings on the OLED display

The following lines display the temperature on the OLED display.

```
display.setTextSize(1);
display.setCursor(0,0);
display.print("Temperature: ");
display.setTextSize(2);
display.setCursor(0,10);
```

```
display.print(t);
display.print(" ");
display.setTextSize(1);
display.cp437(true);
display.write(167);
display.setTextSize(2);
display.print("C");
```

We use the `setTextSize()` method to define the font size, the `setCursor()` sets where the text should start being displayed and the `print()` method is used to write something on the display.

To print the temperature and humidity you just need to pass their variables to the `print()` method as follows:

```
display.print(t);
```

The “Temperature” label is displayed in size 1, and the actual reading is displayed in size 2.

To display the ° symbol, we use the [Code Page 437](#) font. For that, you need to set the `cp437` to `true` as follows:

```
display.cp437(true);
```

Then, use the `write()` method to display your chosen character. The ° symbol corresponds to character 167.

```
display.write(167);
```

A similar approach is used to display the humidity:

```
display.setTextSize(1);
display.setCursor(0, 35);
```

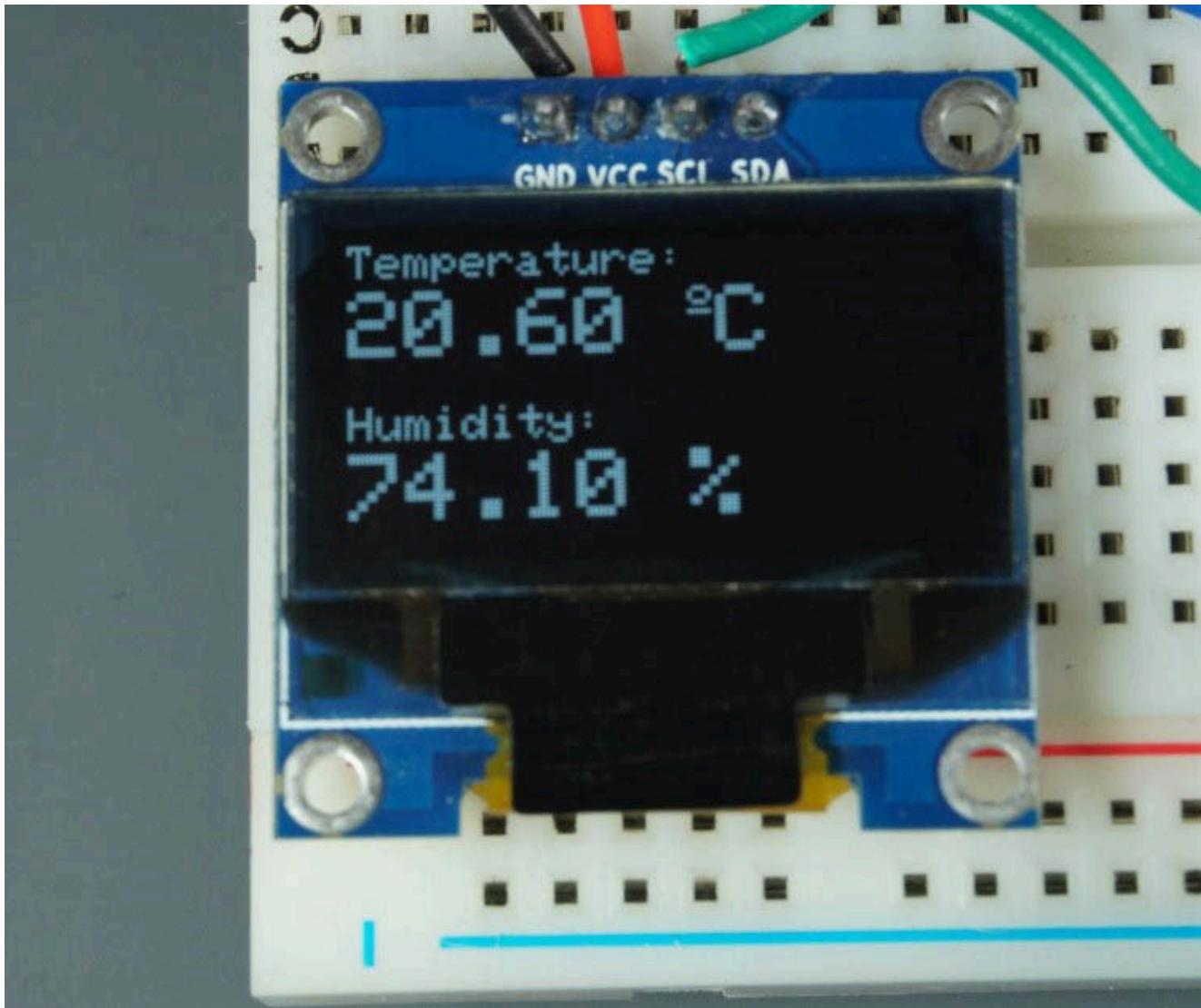
```
display.print("Humidity: ");
display.setTextSize(2);
display.setCursor(0, 45);
display.print(h);
display.print(" %");
```

Don't forget that you need to call `display.display()` at the end, so that you can actually display something on the OLED.

```
display.display();
```

## Demonstration

After wiring the circuit and uploading the code, the OLED display shows the temperature and humidity readings. The sensor readings are updated every five seconds.



## Troubleshooting

If your DHT sensor fails to get the readings or you get the message “Failed to read from DHT sensor”, read our [DHT Troubleshooting Guide](#) to help you solve that problem.

If you get the “**SSD1306 allocation failed**” error or if the OLED is not displaying anything in the screen, it can be one of the following issues:

### Wrong I2C address

The I2C address for the OLED display we are using is 0x3C. However, yours may be different. So, make sure you check your display I2C address using an [I2C scanner sketch](#).

### SDA and SCL not connected properly

Please make sure that you have the SDA and SCL pins of the OLED display wired correctly.

## Wrapping Up

The OLED display provides an easy and inexpensive way to display text or graphics using an Arduino. We hope you've found this guide and the project example useful.

If you like Arduino, make sure you check all our Arduino resources:

- [Arduino Mini Course \(Free\)](#)
- [Arduino Step-by-step Projects \(course\)](#)
- [60+ Arduino Projects and Tutorials](#)
- [Free electronics resources and eBooks](#)

Don't miss our next tutorials and projects! Make sure you [subscribe the RNT blog](#).

Thanks for reading



*PCB Fabrication & Assembly*

# ONLY \$5 for 10 PCBs

✓ 24-hour Build Time   ✓ Quality Guaranteed  
 ✓ Most Soldermask Colors:  













































































































































































































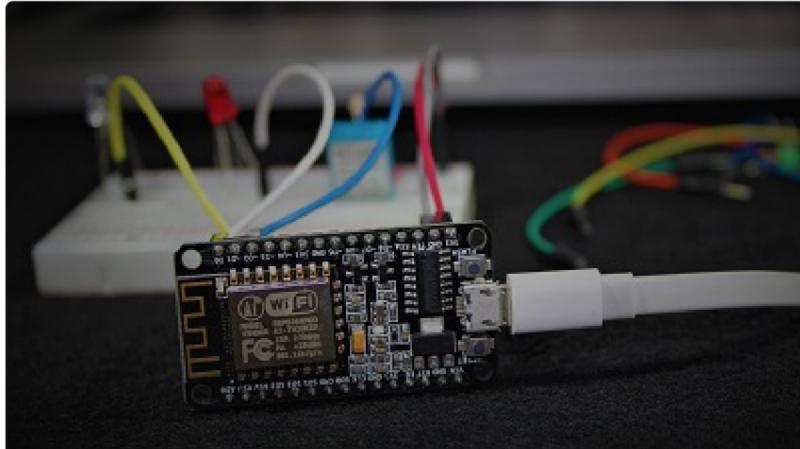

<span style="display: inline-block; width: 200px; height: 15px

Learn how to build a home automation system and we'll cover the following main subjects: Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT, and InfluxDB database [DOWNLOAD »](#)

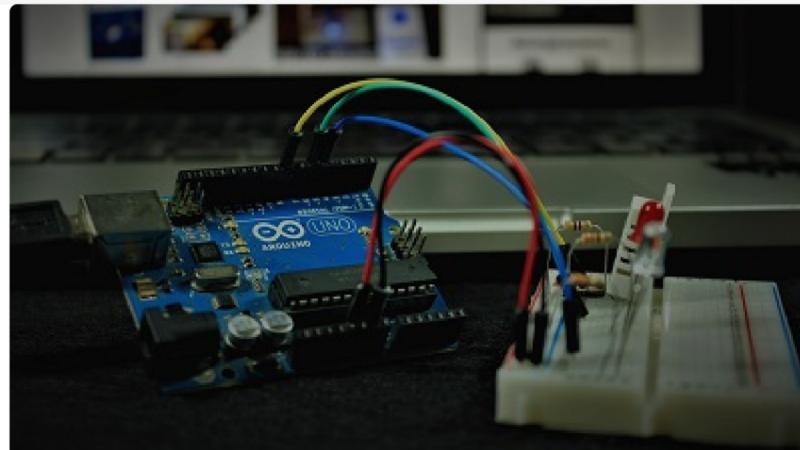
## Recommended Resources



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.



[Arduino Step-by-Step Projects »](#) Build 25 Arduino projects with our course, even with no prior experience!

## What to Read Next...

[Build an All-in-One ESP32 Weather Station Shield](#)

[Power ESP32/ESP8266 with Solar Panels \(includes battery level monitoring\)](#)

[MicroPython IDEs for ESP32 and ESP8266](#)

**Enjoyed this project? Stay updated by subscribing our newsletter!**

Your Email Address

 SUBSCRIBE

## 132 thoughts on “Guide for I2C OLED Display with Arduino”



**Steve Tripoli**

September 29, 2016 at 4:30 pm

Thanks for the guide.

[Reply](#)

**Rui Santos**

October 2, 2016 at 11:26 am

No problem! Thanks for reading Steve

[Reply](#)**John**

September 29, 2016 at 4:51 pm

Awesome Rui, I've been looking at these displays with much interest. This looks like fun! 😊

[Reply](#)**Rui Santos**

October 2, 2016 at 11:27 am

They work great, but keep in mind that they are very very small. Thanks!

[Reply](#)**cesar marin**

September 29, 2016 at 8:48 pm

muchas gracias por la contribucion. felicitaciones por su labor. siga adelante 😊

[Reply](#)



**Rui Santos**

October 2, 2016 at 11:26 am

You're welcome Cesar!

[Reply](#)



**Shehu Bello**

October 1, 2016 at 12:04 am

Very interesting! I appreciate your effort Mr. Rui.

[Reply](#)



**Rui Santos**

October 2, 2016 at 11:25 am

Thanks Shehu!

[Reply](#)

**Gorki**

October 2, 2016 at 11:21 am

It is too advanced for me right now, I keep it for when the time is right. It seems very simply explained, Congratulations

[Reply](#)**Rui Santos**

October 2, 2016 at 11:23 am

Thanks for reading, if you follow each step at a time, you'll see it's very easy.

Rui

[Reply](#)**Tom Evans**

December 27, 2016 at 4:41 pm

Rui

I could use a little guidance with MULTIPLE OLED displays :-).. I have two OLEDs (.96" I2C) but with different addresses (03c and 03d). I need to print to one with some fixed data (ie a string) and the other with variable data (ie `read.analog(0)`..) to make one "large" display. Can't figure out the arduino code to do this beyond the  
`"display.begin(SSD1306_SWITCHCAPVCC, 0x3D);"` statement. Any ideas suggestions would be greatly appreciated Thanks  
TOM

[Reply](#)**Rui Santos**

December 28, 2016 at 1:47 pm

Hi Tom,

I've never tried that, so I can't give you a clear explanation of how to make it work without trying myself.

Thanks and good luck with your project,

Rui

[Reply](#)**JayAchTee**

July 29, 2018 at 11:47 am

This is a good guide to get you going. To use two displays, you can create two different "display" instances, e.g. `displayFixed` and `displayVariable` at the discovered addresses then handle writing the data in code. Another approach would be to create a C++ class that automatically does the Wire scan in the constructor to get the device addresses dynamically and has methods that handle writing to each I2C OLED.

[Reply](#)**james**

February 8, 2021 at 8:50 pm

it looks like we have the same module, and if i understand it correctly there is a very small spot on the back, where you can solder two small spots to enable a different address. on mine, one is already soldered with a tiny resistor for address 3D, i assume we can just desolder the one side and add a small glob to the other. doesnt look like we can use more than two, though. oh crap i just realised i'm like 7 years too late hah

[Reply](#)



**anton buts**

June 2, 2021 at 6:57 am

You have to measure the resistance of the resistor first. in my case it was a 4k7 resistor. if I didn't solder a resistor between the contacts it could burn the oled. Been there, done that, never again.

[Reply](#)



**muc**

June 2, 2021 at 9:45 am

Hi,

(I don't think it's a resistor, but a capacitor.)

I desoldered it (left and middle contacts) and soldered it to middle and right.

So only 2 addresses are possible.

[Reply](#)

**Aubrey**

January 28, 2023 at 4:17 pm

Sounds too advanced for my short, fat fingers to handle. At least I know it can be done. Maybe when I get more comfortable at soldering.....

Thanks guys.

[Reply](#)**MR**

March 16, 2024 at 12:12 pm

Doesn't matter if you were late for the OP, other people may come here years later and get very useful information, like me, I would never think of looking under the PCB, so thanks a lot for your comment !

[Reply](#)**Menel**

January 19, 2017 at 7:43 am

Very, very interesting !!!!!!!!!

[Reply](#)

**Rui Santos**

January 24, 2017 at 7:12 pm

Thanks for your feedback Menel!

Regards,

Rui

[Reply](#)**Harold Lacadie**

December 15, 2017 at 10:36 pm

This is a great tutorial. You've made the OLED display simple even for me. The examples in the library are too advanced to grasp.

[Reply](#)**Sara Santos**

December 21, 2017 at 11:06 am

Hi. We're glad you've enjoyed.

Thanks.

[Reply](#)

**Steve Fraser**

March 7, 2018 at 12:35 pm

Thank you for sharing your time and knowledge with us. I have enjoyed the tutorials and I have learned quite a bit.

Cheers,  
Steve

[Reply](#)**Sid Marker**

April 23, 2018 at 5:15 am

I'd been searching for nearly 3 months that how to properly display the content I want on a OLED screen; came across this site, really helped me in my project. Thanks!

[Reply](#)**Sara Santos**

April 23, 2018 at 9:21 am

Hi.

I'm glad you've found our content useful.

Good luck with your projects.

Regards,  
Sara 😊

[Reply](#)

**Daniel F**

May 11, 2018 at 12:19 am

Thanks for the guide. you are the best!!!!

[Reply](#)**Sara Santos**

May 13, 2018 at 9:40 am

You're welcome 😊

[Reply](#)**Glenn Anderson**

July 19, 2018 at 7:01 am

After some sketch issues...I was able to get this working with the OLED .96 display... It works great and I find the display is adequate enough...

Thanks for the projects!!!

[Reply](#)**Sara Santos**

July 30, 2018 at 10:46 am

Great! 😊

[Reply](#)



**John**

August 8, 2018 at 4:10 am

Hi Rui, I finally got my hands on some of these displays. The first one was failed on arrival. the second one worked for 2 days and failed, I now have a third one received today and it works but IDK for how long. I'm wondering if others are having a high failure rate with these. I bought mine from several different ebay vendors. The first one never illuminated, but I could see data and clock on the SDA & SCL lines. The second one worked on both 3.3v and 5v as the listing stated, but failed. The one I have now I plan to run from a pro mini 3.3v version just to see if it lasts longer. But all 3 vendors stated in their listings that they should run on either 3.3 or 5 volts. Any thoughts?

[Reply](#)



**Leonor**

September 16, 2018 at 7:21 pm

Excuse me, where can i get the libraries for the OLED display??

[Reply](#)

**Sara Santos**

September 17, 2018 at 8:46 am

Hi.

You can find the libraries in the following links:

- [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306)
- <https://github.com/adafruit/Adafruit-GFX-Library>

[Reply](#)**KG**

November 29, 2018 at 1:48 am

I'm new to all this. When I put in the code I get an error message. fatal error: vector: No such file or directory. I downloaded the libraries. Thanks for your time!

[Reply](#)**Sara Santos**

November 29, 2018 at 10:38 am

Hi Kg.

You also need to install the adafruit sensor library:

[https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor)

If you're using web editor take a look at this topic:

[forum.arduino.cc/index.php?topic=574940.msg3915277#msg3915277](http://forum.arduino.cc/index.php?topic=574940.msg3915277#msg3915277)

Regards,

Sara

[Reply](#)**Dieter Kögler**

February 14, 2019 at 11:04 am

Thanks for the free Downloads.I want to Donate you but the link i found doesnt work.

Please send me your Adresse that I can Donate you with PayPal.

You now lot from this Electronic and programming Stuff.Yor little Sister i like

to see so much as well.

Nice Regards from Germany

[Reply](#)**Sara Santos**

February 17, 2019 at 10:57 am

Hi Dieter.

The link should be working fine: <https://www.paypal.me/RuiFSantos>

The best way to support our work is by getting one of our courses.

Besides supporting our work, you also get access to our exclusive resources to learn more.

You can find our courses here: <https://randomnerdtutorials.com/courses/>

Thank you so much for supporting our work and I'm glad that you like our content.

Regards,

Sara

[Reply](#)

**Nahireen**

February 20, 2019 at 6:41 am

M. Rui,

I learn lots and want to learn more . how i will create a Number writing by LED or square? need more books. all the time on online, its make problem of my eyes. soooooooooo, what you kind suggestion

[Reply](#)**Sara Santos**

February 20, 2019 at 5:11 pm

Hi.

I'm sorry, but I didn't understand your question.

Regards,

Sara

[Reply](#)**Nisaar**

April 30, 2019 at 6:58 pm

Nicely explained..Easy to understand..

Thanks

[Reply](#)

**Rui Santos**

May 1, 2019 at 10:41 am

Thanks for reading! I'm glad you found it helpful.

[Reply](#)**drake**

June 6, 2019 at 7:15 am

i need help with displaying a bitmap image it keeps saying expected primary expression before display referring to  
Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire,  
-1);  
and i dont know why. help would be appreciated.

[Reply](#)**Sara Santos**

June 11, 2019 at 9:05 am

Hi.

Can you copy the exact error you're getting?

[Reply](#)



ชัชชาลย์ เจียเจริญ

June 8, 2019 at 3:16 pm

your web infor. help me a lot , I can do many topics on OLED eg. scrolling text,Hello world ,draw shape, Display DHT T ,RH But for display bitmap ,I Can't get the picture on screen .

I think I have done something wrong on conversion process . the reason is when I copy saraarray to sketch ,I can get the picture on OLED screen . How to convert picture to 128×64 with 1024 pixels ?

[Reply](#)



Sara Santos

June 8, 2019 at 4:15 pm

Hi.

If you're using windows, you can open your image in Paint. It has a resize option where you can resize your image. You can also crop your image so that you have the same dimensions. You can also try an online image resizer.

You also need to save your picture as monochrome bitmap format.

Regards,

Sara

[Reply](#)



chatchwal

June 9, 2019 at 2:58 am

Hi

thank you for your immediate response .

I think may be I do mistakes in photo file to C array conversion and did error in programming .

I use paint program to resize file to 128×64 and convert to C array ,copy code to sketch upload file and get unexpected display on half part of screen , 4 column of small line .

in program part ,

your program “static const unsigned char PROGMEM

image\_data\_Saraarray[1024] = { ” .

My program “static const uint8\_t image\_data\_ta\_eu\_1c[400] = {” but I had mark ” // “to cancel this .

Can I have your email address to send conversion file ?

please help me

chat

[Reply](#)



**Sara Santos**

June 11, 2019 at 10:20 am

Hi.

To have your image displayed properly, it should be an array with 1024 size. Yours have 400.

You are probably doing something wrong in the conversion to C array.

Are you using the same program as described in the tutorial?

[sourceforge.net/projects/lcd-image-converter/files/](http://sourceforge.net/projects/lcd-image-converter/files/)

Regards,

Sara

[Reply](#)

**chatchwal**

June 14, 2019 at 8:33 am

Hi

I use same program to convert

in my last conversion I can have 1024 size but the c array look strange

0xffempty , 0xffempty ,0xffempty ,.....

I don't know why .

thanks

chat

[Reply](#)**chatchwal**

June 14, 2019 at 8:37 am

Hi

I use same program to convert

in my last conversion I can have 1024 size but the c array look strange

0xffempty , 0xffempty ,0xffempty ,.....

I don't know why .

thanks

chat

[Reply](#)**Sara Santos**

June 14, 2019 at 6:13 pm

I'm sorry, but I don't know what can be wrong.

 [Reply](#)**Noman sarwar**

July 19, 2019 at 6:11 am

Hi, can I use different colors to display the text like “BLUE” or “RED”?

[Reply](#)**Sara Santos**

July 19, 2019 at 2:53 pm

Hi.

This OLED display is monochrome.

But there are other models that can display colors.

Regards,

Sara

[Reply](#)**JOSE GUSTAVO ABREU MURTA**

July 28, 2019 at 10:56 pm

Rui e Sara,

Excellent tutorial. Complete and functional!

Thank you very much.

“Excelente tutorial. Completo e funcional!”

Muito obrigado.”

Gustavo Murta

[Reply](#)



**Rui Santos**

July 29, 2019 at 3:42 pm

Thanks for reading!

[Reply](#)



**JOSE GUSTAVO ABREU MURTA**

August 17, 2019 at 1:34 am

Caro Rui Santos,

Baseando-me no seu excelente tutorial sobre OLED , desenvolvi um outro tutorial:

Sensor de nível de caixa d’água (Arduino) – sem fio !

<https://blog.eletrogate.com/sensor-de-nivel-de-caixa-dagua-sem-fio/>

Muito obrigado pelo compartilhamento do seu conhecimento!

Abraços além mar,

Gustavo Murta (Brasil)

[Reply](#)



**Harry**

September 7, 2019 at 11:06 pm

Thanks, Rui & Sara! Your examples \*always\* work for me. 😊

The equivalent examples at Adafruit are using their OLED, which is addressed differently than just about everything you'd get from eBay or Ali, and they never once mention changing the I2C address. You not only warn about different addresses, you give us the I2C\_scanner tool to figure out which one we've bought. The address on mine didn't match what's printed on the bottom silkscreen (nor the Adafruit default), so I'd have been running in circles with a blank display. VERY well thought-out demonstrations.

(chuckling) Spell check messed you up: two different places above it says 'monologue' where you meant 'monochrome'. 😊

[Reply](#)



**Sara Santos**

September 8, 2019 at 4:20 pm

Hi Harry.

Thank you for your nice words.

Were you able to get your display working?

Thanks for telling me about the "monologue". I haven't noticed it, but it is fixed now.

Regards,  
Sara

[Reply](#)

**Fausto Galanti**

September 19, 2019 at 10:30 pm

Congratulations guys, great tutorial !!

One question: can you use other pins than those set (A4 and A5) for i2c bus communication?

[Reply](#)**Sara Santos**

September 21, 2019 at 9:36 am

Hi Frausto.

Yes. There are some libraries that do that.

For example: <https://github.com/todbot/SoftI2CMaster>

Regards,

Sara

[Reply](#)**mohammad**

October 1, 2019 at 11:04 am

thank you!!!

that is very useful

thanks a lot

[Reply](#)

**Pooja**

October 31, 2019 at 5:58 pm

can we scroll text vertically ?

[Reply](#)**Sara Santos**

November 5, 2019 at 6:31 pm

Yes.

You can do that. But we don't have the function to do it.

[Reply](#)**reza**

November 13, 2019 at 11:10 am

Hmm can it become colorful?

Can it show a painting with colors?

[Reply](#)**Sara Santos**

November 13, 2019 at 1:30 pm

Hi.

This OLED display model only displays black and white.

There are other models that can display colors, but not this one.

Regards,

Sara

[Reply](#)



**Ron Mason**

November 22, 2019 at 4:58 am

Great tutorial! Thank you for this.

Is it possible to display the Temperature in Fahrenheit?

Thanks again,

Ron

[Reply](#)



**Sara Santos**

November 22, 2019 at 11:27 am

Yes.

To read temperature in Fahrenheit use:

```
float f = dht.readTemperature(true);
```

Regards,

Sara

[Reply](#)**Ron Mason**

November 22, 2019 at 5:10 pm

Thank you Sara! Much appreciated!

Ron

[Reply](#)**Ron Mason**

November 23, 2019 at 12:22 am

Hi Sara,

I have one more question and my apologies for asking multiple questions.

I have one of the oled that are yellow and blue. Basically the temperature is displaying half yellow and half blue. I know this is how these particular models are made, but is there any code I can use that would possibly shift the letters lower on the screen or eliminate the yellow color in a sketch?

Thanks again,

Ron

[Reply](#)

**Sara Santos**

November 24, 2019 at 12:09 pm

Hi Ron.

You can place the letter lower on the screen if you choose a different place to place the cursor to start writing the text. Increase the y coordinate.

```
display.setCursor(x,y)
```

Regards,

Sara

[Reply](#)**dusko**

December 27, 2019 at 10:53 pm

Excellent tutorial—thank you. A small thing: in the picture showing the temperature and humidity, I think that what is shown is not the degree sign (°), but something called “masculine ordinal indicator” (ASCII 167). The degree sign is ASCII 248. (In some fonts they look the same.)

[Reply](#)**Sara Santos**

December 28, 2019 at 11:24 am

Hi.

Yes, I think you're right.

Thanks for sharing that. We'll update the tutorial.

Regards,  
Sara

[Reply](#)



**Georg Berthelsen**

January 7, 2020 at 2:34 pm

Thanks for a nice tutorial on OLED, but a significant section is missing! How do I replace the 0.96 inch OLED display with a different size and controller? There are a great deal of sketches that are only written for the 0.96 inch OLED display, but how can I replace this display with e.g. and 1.5 inch 128×128 SSD 1327 display?

- \* Is it necessary to write a new sketch?
- \* If the same sketch can be used, where should it be corrected and with what values?
- \* You can use the “Display Temperature and Humidity in the OLED Display with Arduino” sketch as an example!

With best regards

Georg Berthelsen

[Reply](#)



**Rohit**

February 13, 2020 at 4:32 pm

Really amazing! I am 12 years old and I am able to do this as you explain it so well!

[Reply](#)

**Sara Santos**

February 14, 2020 at 11:31 am

That's great! 😊

[Reply](#)**Rodrigo Velazquez**

March 22, 2020 at 5:49 am

Great!.

The OLED Display module include pull up resistors for i2c communication?  
I'll use a RTC DS3231 and OLED Display on the ESP32 in the same port  
i2c.

[Reply](#)**Sara Santos**

March 22, 2020 at 12:32 pm

Hi Rodrigo.

They include the pull-up resistors for I2C.

You can connect them directly to the ESP32 I2C port. And yes, you can use the same port.

Learn more about I2C with the ESP32:

<https://randomnerdtutorials.com/esp32-i2c-communication-arduino-ide/>

Regards,

Sara

[Reply](#)**Eugenio Veloso E.**

May 7, 2020 at 2:33 am

Great tutorial. Very clear and helpful. Step by step. Thanks for the effort and all the info 😊

[Reply](#)**Sara Santos**

May 7, 2020 at 9:57 am

Thank you 😊

[Reply](#)**Rizwan**

July 16, 2020 at 12:29 pm

Very nice tutorial.....!

[Reply](#)**Rui Santos**

July 25, 2020 at 10:17 am

Thanks!

[Reply](#)



**Sia**

July 29, 2020 at 8:39 am

That was a brilliant guide. Thank you so much 😊

[Reply](#)



**KJ Archer**

September 6, 2020 at 3:12 am

Hi again i have been looking around but am unable to find an example of how to scroll long text (longer than the display) i assume that must be difficult since no one has any examples of it. Can you point me anywhere ?

[Reply](#)



**Sara Santos**

September 6, 2020 at 10:34 am

Hi.

We've created some scrolling functions, but for MicroPython.

Maybe you want to take a look at the tutorial and try to "translate" the functions to C.

<https://randomnerdtutorials.com/micropython-ssd1306-oled-scroll-shapes-esp32-esp8266/>

Regards,  
Sara

[Reply](#)



**Kj Archer**

September 6, 2020 at 8:55 pm

Hi thanks for answering I found the solution it involves moving the cursor position to a negative number and printing the string in a loop.

[Reply](#)



**Horst Grobe**

October 27, 2020 at 4:56 pm

I have a problem with the program line:

Adafruit\_SSD1306 display (SCREEN\_WIDTH, SCREEN\_HEIGHT, & Wire, -1); . The compiler writes: 'Adafruit\_SSD1306` does not name a type; ... I can't find any errors. I ask for help! thank you and  
Greetings from Horst.

[Reply](#)



**Sara Santos**

October 27, 2020 at 6:32 pm

Hi.

Did you copy the entry code?

Make sure that you've copied the lines to include the library.

You can copy the code from here:

[https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/OLED/oled\\_hello\\_world.ino](https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/OLED/oled_hello_world.ino)

Regards,

Sara

[Reply](#)



**Samuel**

October 31, 2020 at 6:03 pm

My display is filled with black dots on a white background. ): I followed the tutorial and no matter which sketch I load it stays the same. Help please my display is new. ): I'm using an original Arduino nano and a SSD1306 128×64 display.

[Reply](#)



**Judson Belmont**

May 3, 2023 at 10:56 pm

Good day. I'm having the same problem your reported. I can tell that the display is correct but only showing in the top 3 mm of the display. the bottom 80% is white with black dots. I'm wondering if you found a solution? If so, help me please.

[Reply](#)

**Julian**

November 9, 2020 at 3:03 am

I have a problem with the display that I don't know if it can help me. After a few seconds of uploading the code to the NANO, the image freezes or all the code restarts very very often.

After reading forums for several hours, I realized that the problem is not the code but the Wire.h library, which causes the reading to hang through the I2c port. Apparently the solution is to use SBWire.h, but so far I have not been able to get it to work. Do you know anything about this problem?

[Reply](#)**Mahwish**

November 19, 2020 at 4:13 am

Hello,

This tutorial was great, thank you! I am coming across one issue where I am getting unexpected display when I use PROGMEM... if I do not include it everything is displayed properly but I run low on memory and when I do use it, the graphics are not at all displayed correctly. Would you happen to have any insight about this?

The difference is between

```
static const unsigned char frame1_bits[] = { ...}; and  
static const unsigned char frame1_bits[] PROGMEM = { .. };
```

and I am sending it:

```
u8g2.clearBuffer();  
u8g2.drawXBM( 0, 0, frame1_width, frame1_height, frame1_bits );  
u8g2.sendBuffer();  
delay( 500 );
```

Thank you for taking your time to read and help out 😊

[Reply](#)



**gerritp**

February 6, 2021 at 11:52 am

I try to learn more about I2C and bought an Oled. To handle an Oled without documentation and all its addresses is a next step after sensors. Thanks to this article the display is working and the used code gives a next step to investigate.

[Reply](#)



**muc**

February 10, 2021 at 8:24 pm

Hello,

my display runs, nevertheless I read in this article with great interest, thank you for this great work!

My ‘new’ problem: I want to drive 2 displays (addresses 60 and 61).

In July 29, 2018 at 11:47 am JayAchTee wrote

... To use two displays, you can create two different “display” instances, e.g. `displayFixed` and `displayVariable` at the discovered addresses ...

I tried

```
#include <Wire.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 oled(128, 64, &Wire, -1);
Adafruit_SSD1306 oled_2(128, 64, &Wire, -1);

...
void setup() {
if(!oled.begin(SSD1306_SWITCHCAPVCC, 61)) {
Serial.println("SSD1306 allocation failed");
for(;;) // Don't proceed, loop forever
}
// adafruit "logo" is shown on device 61

if(!oled_2.begin(SSD1306_SWITCHCAPVCC, 60)) {
Serial.println("SSD1306 allocation2 failed");
for(;;) // Don't proceed, loop forever
}
// program stops here
```

No success with oled\_2

Who knows how to fit ?

Thanks in advance!

[Reply](#)



**Marcel**

April 22, 2021 at 11:16 am

you gotta remove:

```
if(!oled.begin(SSD1306_SWITCHCAPVCC, 61)) {
Serial.println("SSD1306 allocation failed");
for(;;) // Don't proceed, loop forever
```

```
}
```

// adafruit "logo" is shown on device 61

```
if(!oled_2.begin(SSD1306_SWITCHCAPVCC, 60)) {  
    Serial.println("SSD1306 allocation2 failed");  
    for(;;) // Don't proceed, loop forever  
}
```

cause it checks both addresses with a true or false statement.

And if the first one is true the second one can't be cause it still checks the first address.

So it's just gonna be stuck in the infinite loop when it comes to the second one.

Just write:

```
oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
oled_2.begin(SSD1306_SWITCHCAPVCC, 0x3D);
```

and you should be fine.

[Reply](#)



**erden furat**

March 21, 2021 at 9:37 am

Thank you very much

Now I understand how to use Oled display in my projects.

Very useful information.

[Reply](#)

**Sriram**

April 21, 2021 at 3:00 pm

Hi,

I use 128X64 OLED display in my project to display some parameters. I do not use Arduino, but I use XMC controller for the same. Everything is fine, but when I switch off and Switch on the Micro, the display is getting reset. Can you help me how can I keep the last display on the memory?

[Reply](#)**t.rajeswara rao**

June 1, 2021 at 10:47 am

“bit\_test” function is not available in SSD1306\_ROMBMP

[Reply](#)**Sara Santos**

June 1, 2021 at 2:39 pm

Hi.

What do you mean?

Regards,

Sara

[Reply](#)

**Soonkoo Kwon**

June 4, 2021 at 3:45 pm

Dear Rui Santos

Thank you for your site.

Sincerely yours,  
Soonkoo Kwon

[Reply](#)**Sara Santos**

June 4, 2021 at 8:20 pm

Thank you 😊

[Reply](#)**stuart**

June 11, 2021 at 2:25 am

How do I stop the logo bitmap from loading at startup on the ssd1306 using the 'adafruit\_ssdl306' library?

[Reply](#)

**Daniel**

June 26, 2021 at 9:34 am

Hello, great tutorial. I am having a problem, the screen seems to be displaying only a top stripe about 1/5 wide. I can see that the animation only occurs there, in the rest of the display there is only text I used in a previous animation with a different library (U8g2). How can I fix this?  
Thank you, Daniel.

[Reply](#)**Elliott Senior**

December 4, 2021 at 6:27 pm

I also have this issue, just the top few lines seem to work, I believe it may be an issue with the library's though. Help!

[Reply](#)**Sara Santos**

December 6, 2021 at 11:45 am

Hi.

Make sure the OLED display you're using is an SSD1306 and that you're passing the right I2C address to initialize the display.

You can check the I2C address of your display with this sketch:

[https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/LCD\\_I2C/I2C\\_Scanner.ino](https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/LCD_I2C/I2C_Scanner.ino)

Regards,

Sara

[Reply](#)**Joao Silva**

July 3, 2021 at 1:15 pm

Hi,

Congrats for the great content!

I have a motorbike odometer working fine with an arduino mini pro and a lcd 12864.

Now I wanted to reduce its footprint by using a smaller LCD and I'm trying these examples in a ssd1306 with 128x32.

It is working great, but the "hello world" example consumes 82% of program storage space, leaving not enough space for the practical use I need.

Is there any way to reduce the storage space (by using some optimized font or other lib?), or will I need to move on to other microcontroller?

If other micro, what would be a suitable one? I don't need wireless neither too many IOs, just more storage space...

Cheers,

João Silva, Portugal

[Reply](#)**Christof**

August 5, 2021 at 5:46 pm

Hello!

A very helpful publication, but I have one question.

Even this publication is some years old, may I have the chance to rise a question. Is it possible to show a fixed text (not moving) and at the same time

scroll a text some lines under the fixed text. Or can only the whole display scrolled? Thanks in advance.

Greetings Christof Müller

[Reply](#)



**Sara Santos**

August 6, 2021 at 11:31 am

Hi.

I'm not sure if that's possible with the library's built-in functions.

You may need to create your own functions to do that.

Regards,

Sara

[Reply](#)



**Mgmg Myint**

October 14, 2021 at 2:41 pm

Very complete guidelines for beginners and me. Thank you so much.

[Reply](#)



**Rejza**

January 8, 2022 at 6:58 pm

Hello,  
is there a way to rotate text upside down?

[Reply](#)



**Volker Paap**

February 11, 2022 at 11:03 am

Hello Sara & Rui!  
Thank you very much for your Tutorials!  
I do have a 128x64 oled display in blue/yellow, and the “Hello world”-example is running without problems.  
What I intend to do is:  
1. turn the display 180°, so the yellow part is down  
2. create a graph/curve which shows values taken from a thermometer or a pressure sensor and slowly scrolls along the screen.  
Can I somewhere find a list of all commands that are available for this lib?

Regards, Volker

[Reply](#)



**PapaK**

February 27, 2022 at 6:56 pm

If you are using the adafruit\_SSD1306.h and the adafruit\_GFX.h libraries, locate them in the Arduino/libraries directory. Look at the corresponding “.cpp” files for the list of public calls. This will give you an idea of what commands are available.

[Reply](#)

**Arta**

July 19, 2022 at 2:51 pm

Hello Sara & Rui!

Thank you very much for your Tutorials!

The function “display.drawBitmap(x, y, Name, Width, Height,Color);” draw the bitmaps which are horizontal but I had many custom bitmaps which were vertical. is there any function to use those bitmaps or any way to convert them to horizontal ones?

[Reply](#)**Edy**

July 29, 2022 at 10:23 pm

Please help me. Does the OLED work in looping. For example:

```
while(1){  
    display.println("test");  
    display.display();  
}
```

It shows text “test”, but the text keep copying down. Like,

test  
test  
test  
test

[Reply](#)

**Sara Santos**

August 1, 2022 at 1:31 pm

Hi. I'm not sure I understood your issue.  
Can you better explain?  
Regards,  
Sara

[Reply](#)**muc**

August 1, 2022 at 1:49 pm

use `display.setCursor(x,y);` e.g. (0,10)  
then print

[Reply](#)**abd Jalil**

September 11, 2022 at 11:11 pm

it's been years you still around  
keep the good job...  
tq.

[Reply](#)

**Omer**

October 4, 2022 at 9:34 pm

Hey sara

Can you explain how we can add any other sensor instead of humidity sensor?

Like i want to add a water level sensor sl067 with oled.

Can you let me know how to fo that?

[Reply](#)**Pip Took**

December 27, 2022 at 8:47 pm

Thank you for the great step-by-step tutorial. I appreciate you walking us through each line of code. This was a great way to learn about using my new OLED screen.

I learned about the very useful PROGMEM keyword through your guide on bitmaps. If others ever have problems with the Arduino's memory, look into saving storage space by using PROGMEM for arrays.

[Reply](#)**Aubrey**

January 28, 2023 at 4:19 pm

This gives me ideas for a cyberdeck I want to build. Love these tutorials.

Thanks TONS!!!

[Reply](#)



**EKS**

February 25, 2023 at 4:20 am

Hi Sara and Rui..

It is a comprehensive tutorial, and as beginner, I did learn much from it, many thanks.

Maybe I missed the content on how to shift a long text to the next line, with cut in the space, so every line (row) consist of full word and not cut in a part of word (by OLED pixel limit).

Please explain with a simple code example.

Thank you for your help.

[Reply](#)



**Eric**

March 24, 2023 at 1:26 pm

There is a step missing, which can cause the generated array to throw a “too large” error. The missing step, which is shown in the image but not the text, is to set the Preset to Monochrome from the dropdown menu, before setting the type and scan direction.

[Reply](#)



**Sara Santos**

March 25, 2023 at 11:23 am

Hi.

Thanks for pointing that out. I'll highlight that option in the image.

Regards,

Sara

[Reply](#)



**Guy Gaudreau**

April 28, 2023 at 8:29 pm

Hi, I am very green with Arduino and everything around it. I purchased the display board 1.3" OLED Module: Bright White/Blue Display with 128×64 Resolution & I2C/SPI Communication and I want to connect it to an Arduino Nano board, can it be done?

Also I was wondering if there was an how to for using clock functions?

[Reply](#)



**Guy Gaudreau**

April 28, 2023 at 8:31 pm

Hi, I am very green with Arduino and everything around it. I purchased the display board 1.3" OLED Module: Bright White/Blue Display with 128×64 Resolution & I2C/SPI Communication and I want to connect it to an Arduino Nano board, can it be done?

Also I was wondering if there was an how to for using clock library functions?

thank you

Guy

[Reply](#)



**Sumit Batra**

September 28, 2023 at 3:35 am

Is there a way to read the OLED dimensions using a READ command

[Reply](#)



**Henrik**

October 20, 2023 at 8:28 pm

Thanks for a super good guide.

Can you use the new Arduino Uno Rev4 wifi for the same guide and if so, which Libraries should you use??

[Reply](#)



**Perlchamp**

November 26, 2023 at 11:45 pm

Hi, hallo.

Newest Arduino IDE, newest Versions of libraries.

THIS doesn't work for me, cause in YOUR circuit you've selected the 5V-Output for the display.

THIS forced a compiling error in adafruit\_ssdl1306.h line 148 (bool begin...) The parameter 'SSD1306\_SWITCHCAPVCC' is ONLY for the 3.3V output (to read also in the file itself – as comment).

I'm looking for the parameter if you use an EXTERNAL power source (5V or external Power-Adapter).

You can test it !

So long

Perlchamp

[Reply](#)



**Perlchamp**

November 27, 2023 at 12:03 am

Sorry:

I use a nano V3...

I found, what i'm looking for:

begin():

Parameters:

vcs:

VCC selection. Pass SSD1306\_SWITCHCAPVCC to generate the display voltage (step up) from the 3.3V source, or SSD1306\_EXTERNALVCC otherwise. Most situations with Adafruit SSD1306 breakouts will want SSD1306\_SWITCHCAPVCC.

addr

I2C address of corresponding SSD1306 display (or pass 0 to use default of 0x3C for 128×32 display, 0x3D for all others). SPI displays (hardware or software) do not use addresses, but this argument is still required (pass 0 or any value really, it will simply be ignored). Default if unspecified is 0.

**reset**

If true, and if the reset pin passed to the constructor is valid, a hard reset will be performed before initializing the display. If using multiple SSD1306 displays on the same bus, and if they all share the same reset pin, you should only pass true on the first display being initialized, false on all others, else the already-initialized displays would be reset. Default if unspecified is true.

**periphBegin**

If true, and if a hardware peripheral is being used (I2C or SPI, but not software SPI), call that peripheral's begin() function, else (false) it has already been done in one's sketch code. Cases where false might be used include multiple displays or other devices sharing a common bus, or situations on some platforms where a nonstandard begin() function is available (e.g. a TwoWire interface on non-default pins, as can be done on the ESP8266 and perhaps others).

Good luck !

So long

[Reply](#)



TR

December 2, 2023 at 6:49 pm

I have code here and I have some troubles with it.

It should show chess board on display.

There is code:

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

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

```
#define OLED_ADDR 0x3C

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_ADDR);

void setup() {
Serial.begin(115200);

if (!display.begin(SSD1306_I2C, OLED_ADDR, &Wire)) {
Serial.println(F("SSD1306 allocation failed"));
for (;;) {
;
}

display.display(); // Zobrazí prázdný displej
delay(2000);
display.clearDisplay(); // Vymaže obsah displeje

drawChessboard();
}

void loop() {
// Váš loop kód zde
}

void drawChessboard() {
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);

for (int i = 0; i < 8; i++) {
for (int j = 0; j < 8; j++) {
if ((i + j) % 2 == 0) {
display.fillRect(i * 16, j * 8, 16, 8, SSD1306_WHITE);
} else {
display.fillRect(i * 16, j * 8, 16, 8, SSD1306_BLACK);
}
}
}
}
```

```
display.display();  
}
```

After Verify it show message: Compilation error: 'SSD1306\_I2C' was not declared in this scope.

How can I fix it? Can you help me please.

[Reply](#)



**Sara Santos**

December 3, 2023 at 10:18 am

Hi.

Use

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
```

instead of

```
if (!display.begin(SSD1306_I2C, OLED_ADDR, &Wire)) {
```

Regards,

Sara

[Reply](#)



**brandon**

January 2, 2024 at 4:49 am

You guys seem like you are well complimented, but I wanted to be the first in 2024... Great/clear/thorough tutorial Happy new year to the Santos family!

[Reply](#)**Sara Santos**

January 2, 2024 at 10:07 am

Hi.

Thank you so much.

We wish you a happy new year.

Regards,

Sara

[Reply](#)**Maxx Eastick**

January 19, 2024 at 12:20 pm

Thank you for your tutorials.

I started to do this project as the aircon at the office is set a little too high (or cold, btw I'm from Australia and it is summer here).

Just wanted to see how cold it is getting at my desk. I am now trying to expand it to writing the data to an SD card but keep it as small as possible and run on a Lipo. I have a small tin ready to encase it as well.

Had to adjusted for the 128×32 display, DHT11 instead, but your tutorials are real easy to follow so no issue there.

I started this project on the Raspberry Pico but the Nano fits in the tin better.

Thanks again for all your work.

[Reply](#)

**Ian Mather**

January 25, 2024 at 11:40 am

Hi

I am displaying the output of a BME680 on a oled display, I would like the screen to only be on when I press a button.

Please advise

[Reply](#)

## Leave a Comment

 Name \* Email \* Website Notify me of follow-up comments by email. Notify me of new posts by email.

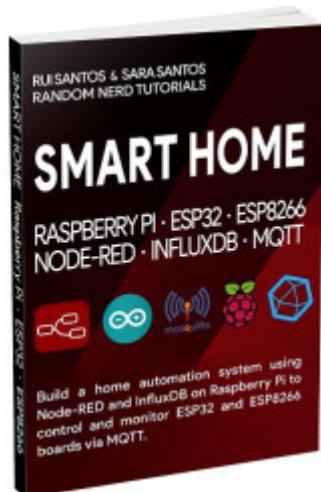
**Affiliate Disclosure:** Random Nerd

*Tutorials is a participant in affiliate advertising programs designed to provide a means for us to earn fees by linking to Amazon, eBay, AliExpress, and other sites.*

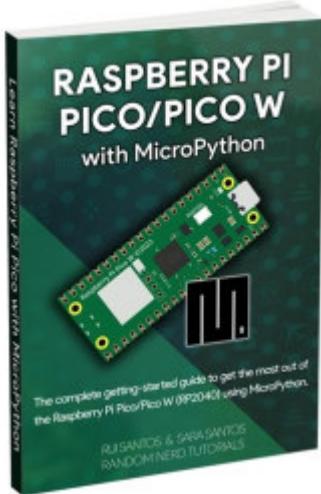
*We might be compensated for referring traffic and business to these companies.*

**[Learn ESP32 with Arduino IDE \(2nd](#)**

**[Edition\) Course](#)** » Complete guide to program the ESP32 with Arduino IDE!



**[Smart Home with Raspberry Pi, ESP32, and ESP8266](#)** » learn how to build a complete home automation system.



🔥 **Learn Raspberry Pi Pico/Pico W with MicroPython** » The complete getting started guide to get the most out of the the Raspberry Pi Pico/Pico W (RP2040) microcontroller board using MicroPython programming language.

[About](#)   [Support](#)   [Terms and Conditions](#)   [Privacy Policy](#)   [Refunds](#)   [Complaints' Book](#)

[MakerAdvisor.com](#)   [Join the Lab](#)

Copyright © 2013-2024 · RandomNerdTutorials.com · All Rights Reserved

Aktualisieren Sie die Datenschutzeinstellungen