

# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

## OLED

Las OLED son uno de los tipos de pantalla disponibles para emplear en nuestros proyectos de electrónica y Arduino.

Un OLED (Organic light-emitting diode) es un tipo de LED en el que la capa emisiva es está formada por un compuesto orgánico que emite luz en respuesta a la electricidad.

Al igual que el resto de tipos de pantallas, las OLED necesitan un controlador específico que convierta los datos recibidos en las señales electrónicas para controlar la pantalla.

En el campo de la electrónica casera, se comercializan **unas pequeñas pantallas OLED de 0.96" ó 1,30" listas para conectar a Arduino**. Estas pantallas incorporan el controlador SDD1306 ó SH1106 y tienen un tamaño muy reducido de 25mm x 14mm. Son monocromas y tienen una resolución de 128x64 pixels.

Las pantallas OLED tienen la **ventaja de tener un consumo muy bajo**, en torno a 20mA, dado que solo se enciende el pixel necesario y no requieren de backlight. Esto es especialmente interesante en aplicaciones que funcionan con baterías.

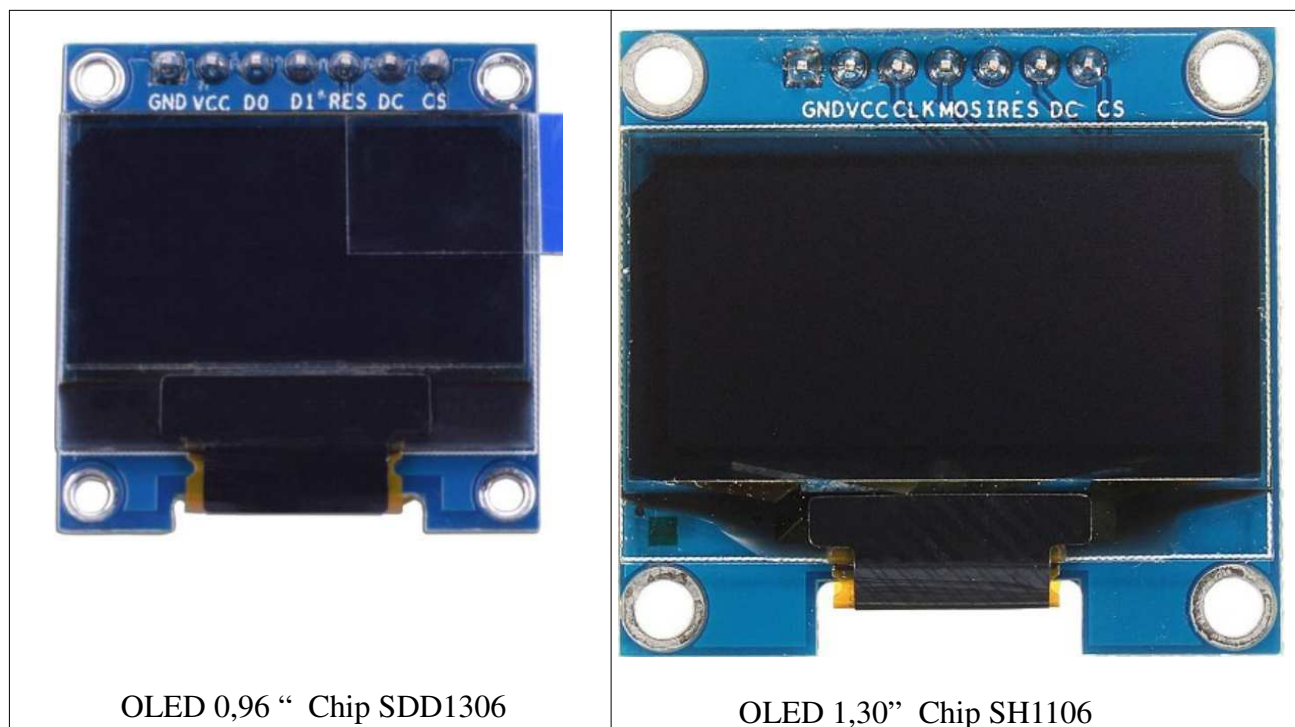
Además, **tienen una mejor visibilidad en ambientes luminosos**, como bajo el sol. Aunque el pequeño tamaño de estas pantallas OLED de 0.96"(1,30") pueden ser un problema para su correcta visualización.

La comunicación puede realizarse, según modelos, por [bus SPI](#) o por [bus I2C](#) por lo que es sencillo obtener los datos medidos. La tensión de alimentación admite voltajes de 3.3V y 5V.

**NOTA:** De fábrica vienen configuradas para trabajar en modo SPI .- (Imagen derecha)

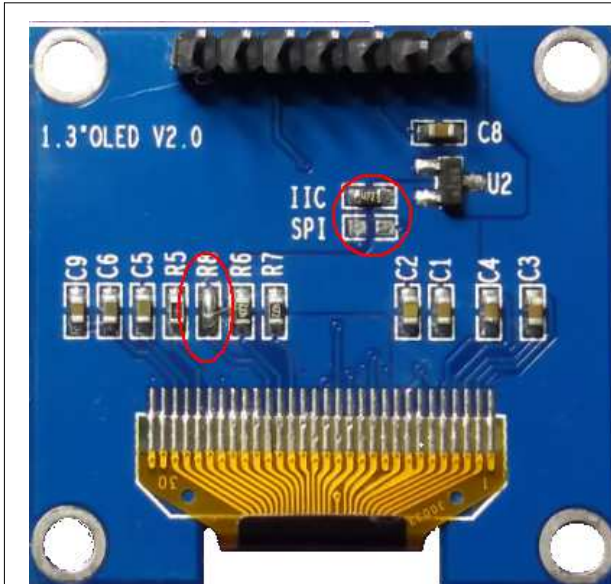
Para usarlas en modo I2C debe configurarse cambiando una resistencia de lugar y un puente.

Este tipo de pantallas OLED son interesantes para proyectos donde podamos aprovechar su pequeño tamaño.

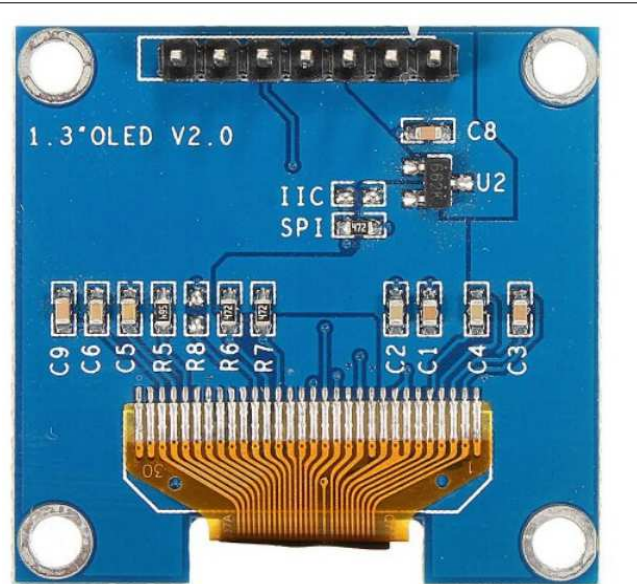


# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

## Configuración OLED 1,30"



Configuración modo I2C OLED - 1,30"

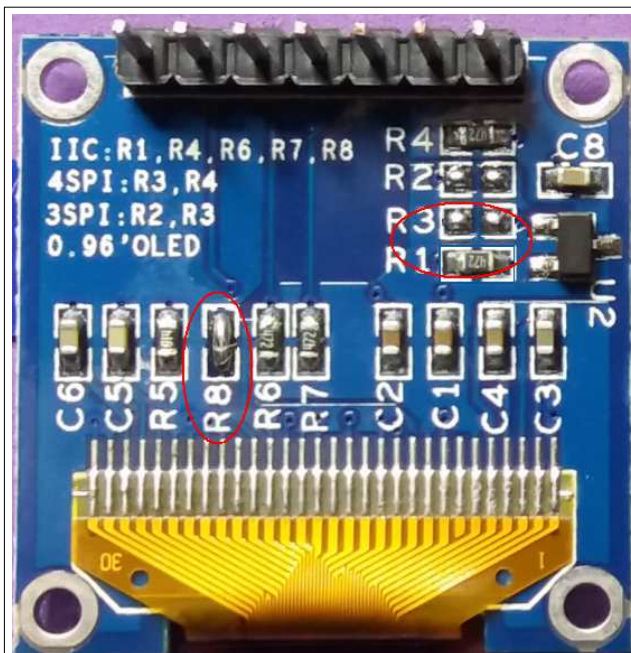


Configuración modo SPI OLED - 1,30"

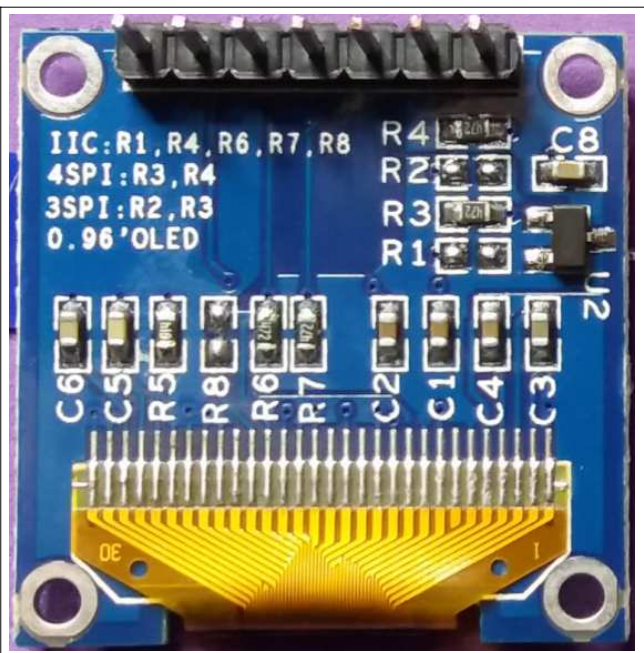
En este caso se pasa la resistencia de 4K7(SMD 472) del SPI a IIC y se realiza un puente con estaño en R8.-

## Configuración OLED 0,96"

En este caso es similar. En R8 realizamos el puente y trasladamos R3 a la posición libre de R1.-



Configuración modo I2C OLED - 0,96"



Configuración modo SPI OLED - 0,96"

# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

Este tipo de pantallas OLED son interesantes para proyectos donde podamos aprovechar su pequeño tamaño.

## Correspondencia de pines OLEDs

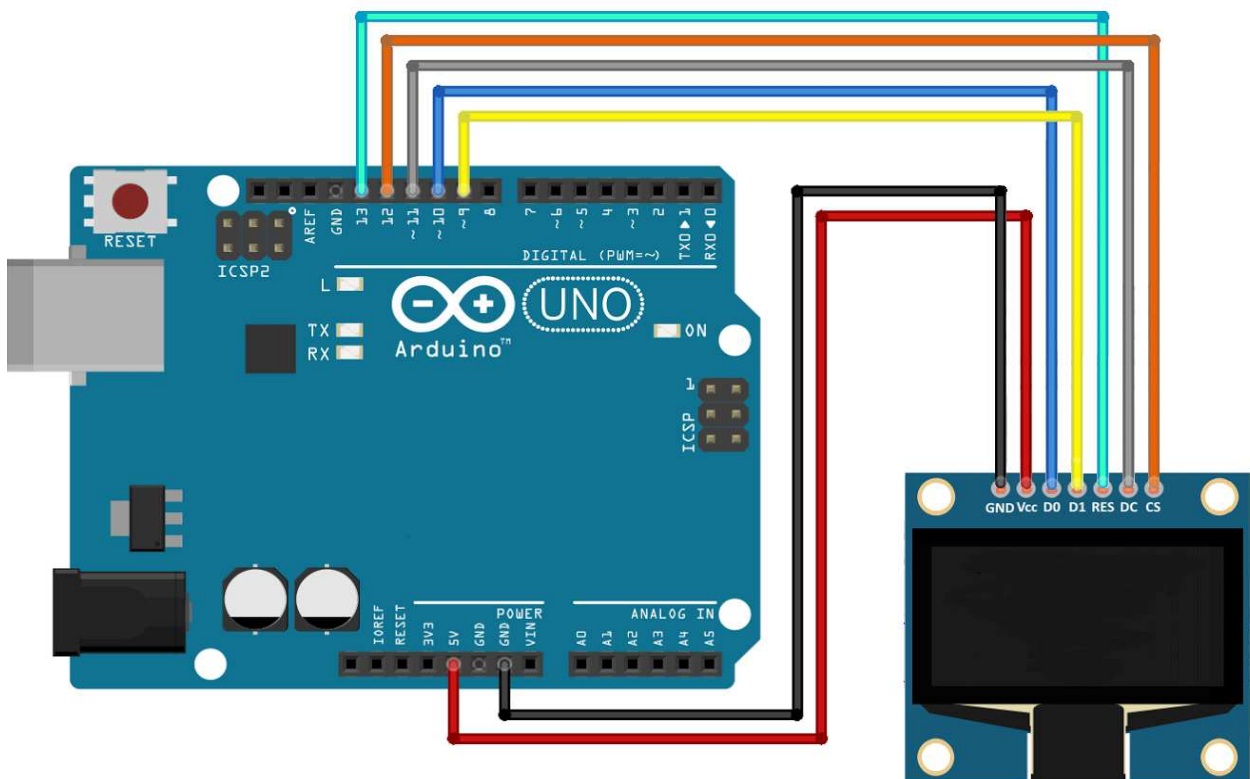
La correspondencia de pines entre ambos Oled de este artículo es la siguiente:

- > OLED 1.30" GND <-> OLED 0,96" GND
- > OLED 1.30" VCC <-> OLED 0,96" VCC
- > OLED 1.30" CLK <-> OLED 0,96" D0
- > OLED 1.30" MOSI <-> OLED 0,96" D1
- > OLED 1.30" RES <-> OLED 0,96" RES
- > OLED 1.30" DC <-> OLED 0,96" DC
- > OLED 1.30" CS <-> OLED 0,96" CS

## Conexión SPI

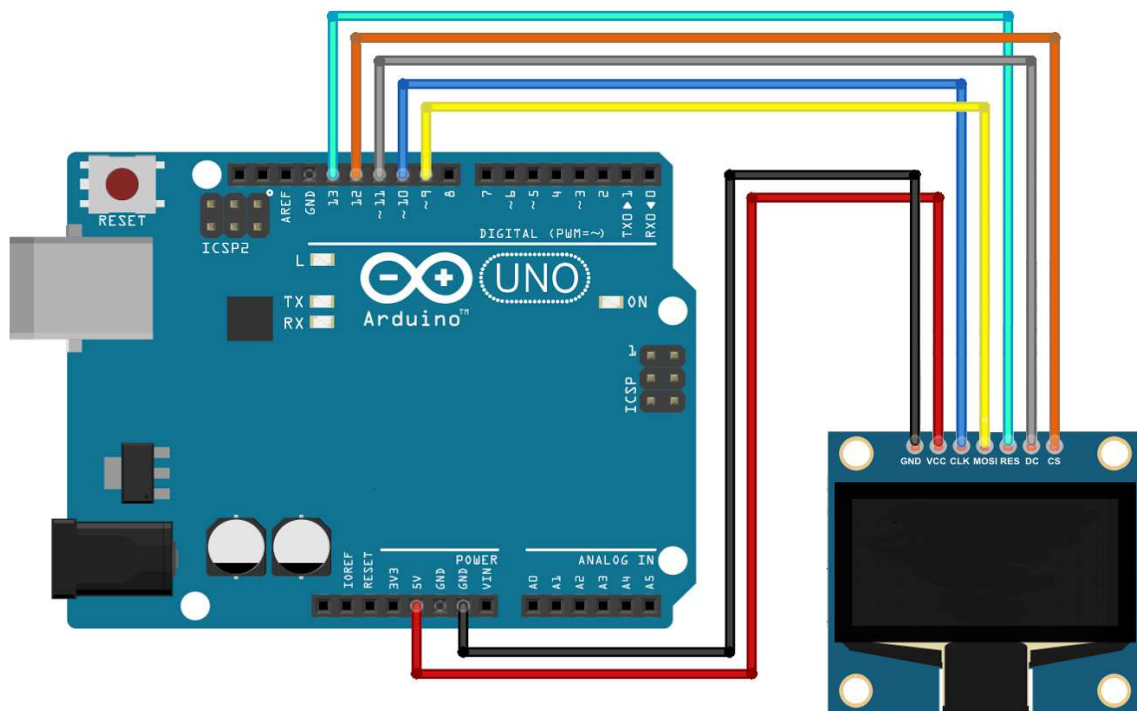
- > OLED GND - Arduino GND
- > OLED VCC - Arduino +5V
- > OLED CLK/(D0) - Arduino Uno Pin 10
- > OLED MOSI/(D1) - Arduino Uno Pin 9
- > OLED RES - Arduino Uno Pin 13
- > OLED DC - Arduino Uno Pin 11
- > OLED CS - Arduino Uno Pin 12

## Diagramas de conexionado modo SPI OLED 0,96" Driver SSD1306



# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

## Diagramas de conexionado modo SPI OLED 1,30" Driver SH1106

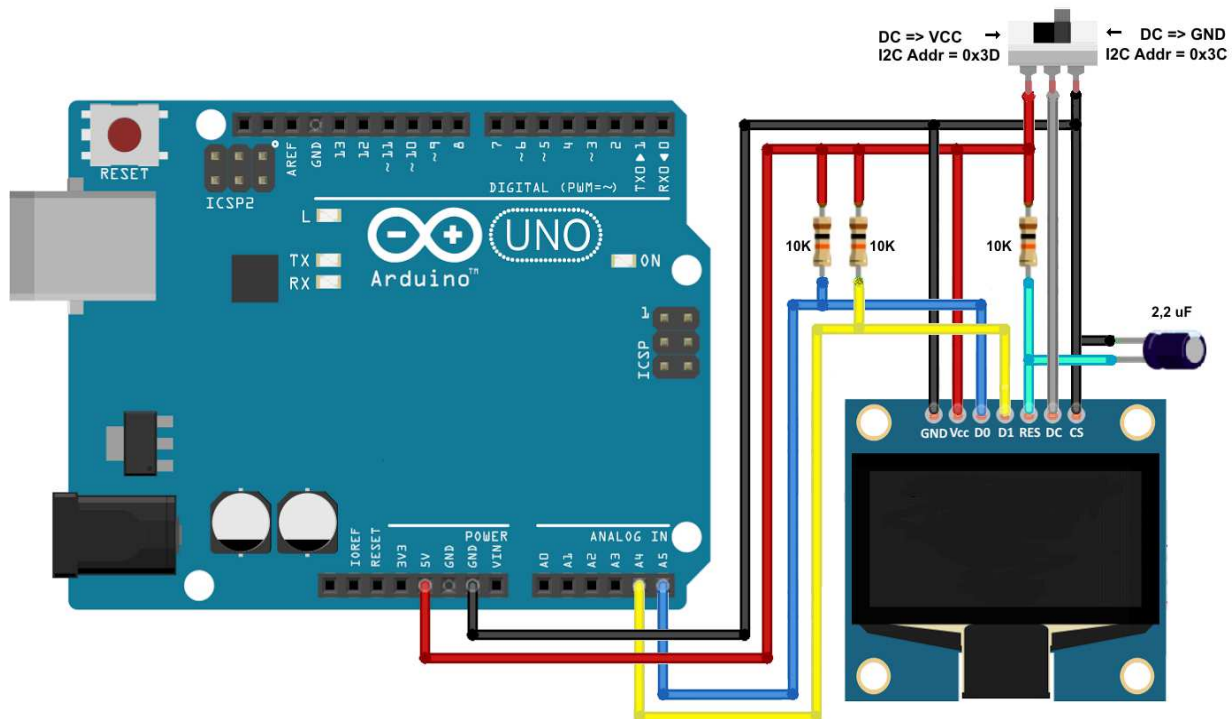


### Conexión I2C

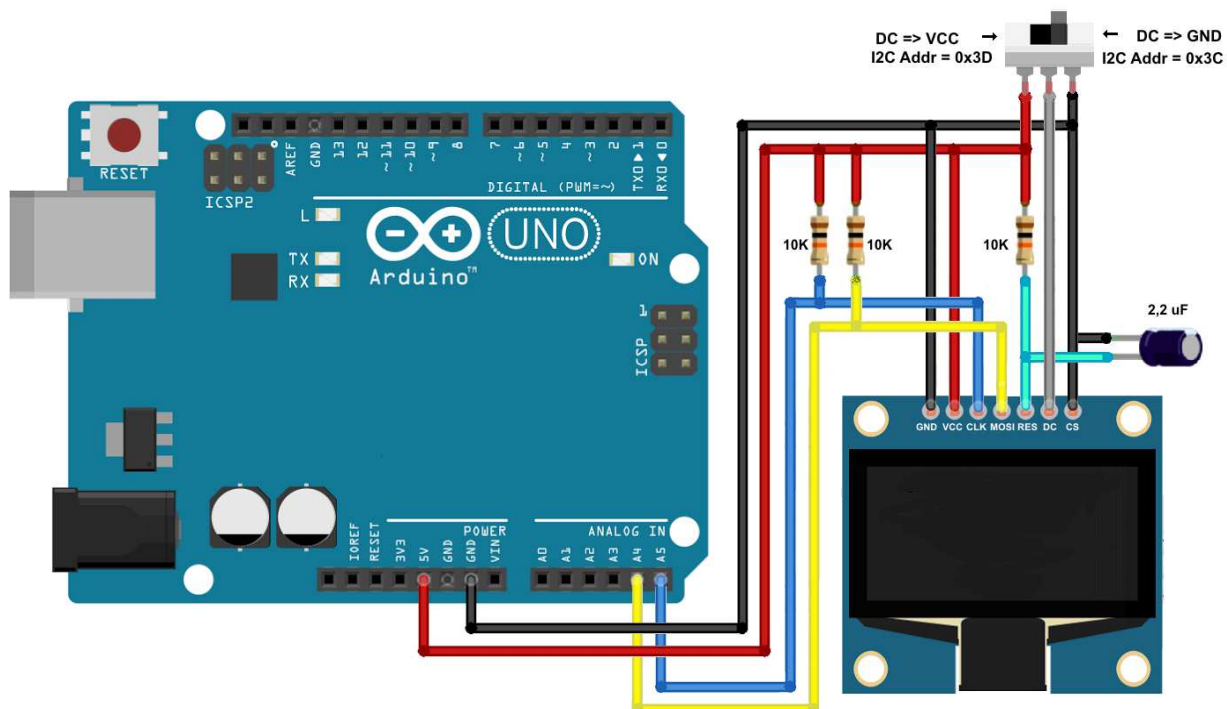
- > OLED GND - Arduino GND
- > OLED VCC - Arduino +5V
- > OLED CLK/ (D0) - Arduino Uno A5 - SCL
- > OLED MOSI7 (D1) - Arduino Uno A4 - SDA
- > OLED RES - Circuito Reset R/C ó pin Arduino por soft
- > OLED DC - Arduino GND
- > OLED CS - Arduino GND

# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

## Diagramas de conexionado modo I2C OLED 0,96" Driver SSD1306



## Diagramas de conexionado modo I2C OLED 1,30" Driver SH1106



# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

## Librerías para IDE Arduino

### **SSD1306**

#### Adafruit SSD1306

[https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306)

#### Adafruit GFX

<https://github.com/adafruit/Adafruit-GFX-Library>

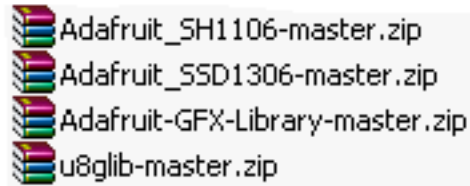
### **SH1106**

### **U8glib - Universal Graphics Library for 8 Bit Embedded Systems**

<https://github.com/olikraus/u8glib>

## Instalación de las Librerías en IDE de Arduino

Cuando bajemos las librerías, debemos guardarlas en una carpeta temporal para luego poder instalarlas.



En la figura se muestran los distintos archivos ZIP de las librerías. En el procedimiento de instalación es el mismo para todas.

### Instalar la librería Adafruit SSD1306

En el computador quedará el archivo: **Adafruit\_SSD1306-master.zip**.

Este archivo debemos descomprimirlo en la carpeta:

**Documents → Arduino → libraries**

conviene renombrar la carpeta que se genera :**Adafruit\_SSD1306-master**

por : **Adafruit\_SSD1306**

### Instalar la librería GFX

El procedimiento es igual que para la librería anterior.

Cuando bajemos la librería, en el computador quedará el archivo:

**Adafruit-GFX-Library-master.zip**.

Este archivo debemos descomprimirlo en la carpeta:

**Documents → Arduino → libraries**

conviene renombrar la carpeta que se genera :**Adafruit-GFX-Library-master**

# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

por : Adafruit\_GFX

## Instalar la librería Adafruit\_SH1106

En el computador quedará el archivo: **Adafruit\_SH1106-master.zip**.

Este archivo debemos descomprimirlo en la carpeta:

**Documents → Arduino → libraries**

conviene renombrar la carpeta que se genera :**Adafruit\_SH1106-master**

por : **Adafruit\_SH1106**

## Instalar la librería u8glib

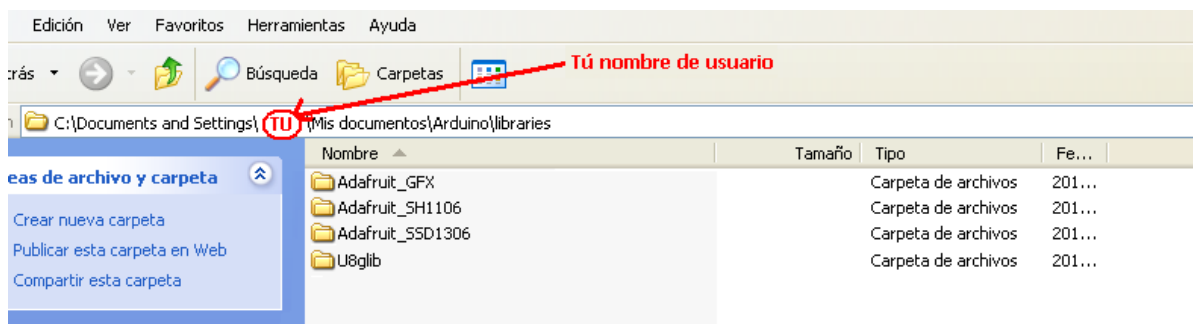
En el computador quedará el archivo: **u8glib-master.zip**.

Este archivo debemos descomprimirlo en la carpeta:

**Documents → Arduino → libraries**

conviene renombrar la carpeta que se genera : **u8glib-master** por : **u8glib**

Nos debería quedar así:



## Consideraciones previas

Las librerías tienen definidos en forma pre determinada tamaño del oled y tipo de comunicación en particular. Por lo cual es necesario de acuerdo a nuestro circuito modificarlas para funcionen correctamente.

## Modificar la librería SSD1306

El controlador SSD1306 no está configurado para la pantalla OLED de forma predeterminada. El tamaño de visualización debe cambiarse en el controlador antes de poder usarse. Si no se cambia, aparecerá un mensaje de error al intentar verificar el boceto de ejemplo (consulte la sección a continuación) en el IDE de Arduino:

```
#error ("Altura incorrecta, corrija Adafruit_SSD1306.h!");
```

Abra la carpeta Adafruit\_SSD1306 que acaba de instalar en la carpeta de bibliotecas Arduino. Encuentra [Adafruit\\_SSD1306.h](#) y ábrelo en un editor de texto. Desplácese por el archivo para buscar la sección con el encabezado SSD1306 Displays o busque este término en el editor de texto para encontrarlo rápidamente.

```
#define SSD1306_128_32 dejarlo como comentario ... // #define SSD1306_128_32
```

# Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

```
// #define SSD1306_128_64 eliminar // comentario ... #define SSD1306_128_64
```

Nos debería quedar así:

```
*=====
SSD1306 Displays
-----
The driver is used in multiple displays (128x64, 128x32, etc.).
Select the appropriate display below to create an appropriately
sized framebuffer, etc.

SSD1306_128_64 128x64 pixel display

SSD1306_128_32 128x32 pixel display

SSD1306_96_16

-----*/
#define SSD1306_128_64
// #define SSD1306_128_32
// #define SSD1306_96_16
/*=====*/
```

## Programa para controlar por medio de SPI SSD1306 ó SH1106 7 Pines

**El circuito es el descrito anteriormente para modo SPI**

Luego de instalar las librerías, re-iniciamos el IDE de Arduino abrimos un sketck nuevo y copiamos el siguiente programa :

Código:

```
/*****
```

Este ejemplo es para los oled de 128x64 usando SPI y Drivers  
SSD1306 o SH1106 ambos con 7 pines de conexión.-

Se utilizan las librerias de Adafruit.

```
*****/
```

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

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

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

```
// #include <Adafruit_SH1106.h>
```

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

```
// Definir pines SPI en Arduino
```

```
#define OLED_MOSI 9
```

```
#define OLED_CLK 10
```

```
#define OLED_DC 11
```

```
#define OLED_CS 12
```



## **Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines**

```
#define OLED_RESET 13

// A continuación informamos al soft del driver,cuales pines del arduino usamos
// Para SH1106 descomentar siguiente comando
//Adafruit_SH1106 display(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
// Para SSD1306 descomentar siguiente comando
Adafruit_SSD1306 display(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);

// Para SH1106 descomentar siguiente comando
//#if (SH1106_LCDHEIGHT != 64)
// Para SSD1306 descomentar siguiente comando
#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SH1106.h!");
#endif

void setup()
{
  // Serial.begin(9600); // Si se va usar comunicación serial, Descomentar
  // Para SH1106 descomentar siguiente comando
  // display.begin(SH1106_SWITCHCAPVCC); // Inicia el display OLED (Carga Buffer)
  // Para SSD1306 descomentar siguiente comando
  display.begin(SSD1306_SWITCHCAPVCC); // Inicia el display OLED (Carga Buffer)
  display.clearDisplay(); // Borrar imagen en el OLED
  display.setTextSize(1); // Definir Tamaño del Texto
  display.setTextColor(WHITE); // Definir color del texto. (mono=>Blanco)
  display.setCursor(0,10 ); // Definir posición inicio texto Columna (0) Fila (10)
  display.println("Hola, mundo!"); // Carga la información al buffer
  display.display(); // Actualiza display con datos en Buffer
  delay(2000); // Demora de 2 segundos.
}

void loop() {

  // Aquí irá el programa

  // Cuando se quiera mostrar nueva información
  display.clearDisplay(); // Borrar imagen en el OLED
  display.setTextSize(1); // Definir Tamaño del Texto
  display.setTextColor(WHITE); // Definir color del texto. (mono=>Blanco)
  display.setCursor(0,10 ); // Definir posición inicio texto Columna (0) Fila (10)
  display.println("Nuevo Mensaje"); // Carga la información al buffer
  display.display(); // Actualiza display con datos en Buffer
```

## Pantalla OLED SSD1306-SH1106 I2C-SPI 7 Pines

```
delay(2000); // Demora de 2 segundos.
```

```
}
```

```
// Aquí van las funciones
```

```
// ----- FIN PROGRAMA -----
```

Lo único a considerar en este ejemplo es que está des-comentado para correr con el driver **SSD1306** es el **texto en marrón**.

Si se fuera a utilizar el driver **SH1106** hay que corregirlo colocando las **// delante del comando en marrón** y sacar las **// en texto en azul**.-

Resultado:

