


[HOGAR](#)
[CATEGORIAS](#)
[BERRYCLIP](#)
[COMPRAR](#)
[HERRAMIENTAS](#)
[TUTORIALS & HELP](#)
[CONTACT US](#)
[SITE MAP](#)
**ESTÁS EN:** [Inicio](#) » [Hardware](#) » [Interfaces](#) » [I2C](#) » [Uso de un módulo de pantalla OLED I2C con la Raspberry Pi](#)

## Uso de un módulo de pantalla OLED I2C con Raspberry Pi

[27](#)

POR MATT EN 8 DE ABRIL DE 2018

I2C , TUTORIALES Y AYUDA



Los módulos de pantalla OLED en miniatura son una excelente manera de agregar una pantalla pequeña a sus proyectos de Raspberry Pi. Están disponibles en varios tamaños, pero los tamaños comunes incluyen 128 × 32 y 128 × 64 píxeles. Los más baratos tienen píxeles de un solo color que son blanco, amarillo o azul. Mi dispositivo tiene píxeles blancos y utiliza una interfaz I2C que solo requiere cuatro cables para conectarse a la Pi.

En este tutorial explicaré cómo configuro mi módulo de pantalla OLED de 0.96 " utilizando la interfaz I2C de Pi. Una vez configurado, es fácil usar Python para colocar texto, dibujar formas o incluso mostrar imágenes y animaciones simples.

### El módulo OLED

Mi módulo de pantalla OLED es un módulo de pantalla LED LCD OLED serie 128X64 I2C IIC SPI de 0.96 ".



#### RECENT POSTS

NOVEMBER 13, 2019

[1](#)
[Ntablet an Open-source Tablet](#)

OCTOBER 21, 2019

[2](#)
[Pi-Hole OLED Status Screen](#)

SEPTEMBER 22, 2019

[0](#)
[KKSB Raspberry Pi 4 Steel Case](#)

JUNE 24, 2019

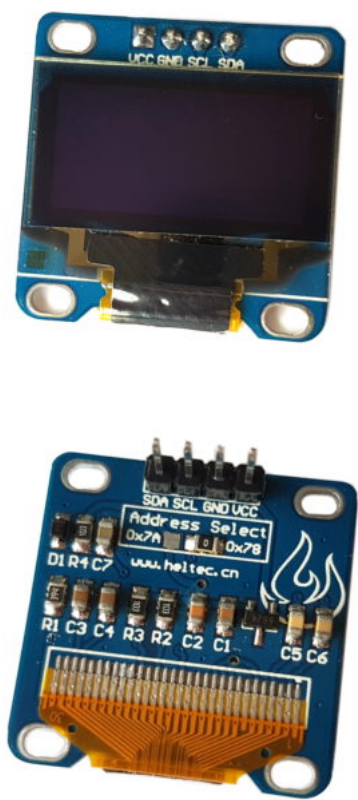
[0](#)
[Introducing the Raspberry Pi 4](#)

JUNE 19, 2019

[2](#)
[Gameboy Zero 6 Button Board from Aliexpress](#)

#### CATEGORIES

[1-wire](#)
[3D Printing](#)
[Add-ons](#)
[BBC Micro:bit](#)
[BerryClip](#)
[Books](#)
[Camera Module](#)
[Cases](#)
[Events](#)
[General](#)
[Hardware](#)
[I2C](#)



Tiene cuatro pines. Dos son de alimentación (Vcc y Gnd) y dos son para la interfaz I2C (SDA y SCL). Es posible que deba soldar el encabezado antes de poder usarlo.

### Actualizar sistema operativo

Al igual que con todos mis proyectos, comencé creando una tarjeta SD con la última imagen de Raspbian. Luego me aseguré de que esto estuviera actualizado ejecutando los siguientes comandos:

```
sudo apt-get update
sudo apt-get upgrade
```

Este paso puede tardar unos minutos si hay muchos paquetes para actualizar, pero generalmente ahorra algo de frustración en el futuro.

### Configuración del módulo de pantalla

Mi pantalla tenía cuatro pines, dos para la alimentación y dos para la interfaz I2C.

- Infographics
- Interfaces
- Minecraft
- Model A+
- Model B+
- News
- Pi Models
- Pi Zero
- Power
- Programming
- Python
- Raspbian
- RetroGaming
- Robotics
- Sensors
- Software
- SPI
- Tutorials & Help

Follow on Twitter 18.1K

Subscribe on YouTube 1.5K

Like on Facebook

Follow on Instagram

#### TAGS

3D Printingaudiobatteryberryclip

Birthdaybluetoothcambridge

cameraCamJamDigiMakers

displaygamesGPIOI2C

interfaceKickstarterLCDLED

LinuxmediaminecraftModel A

Model BmotionEyeOSPCBphotography

photosPi-Liteportablepower

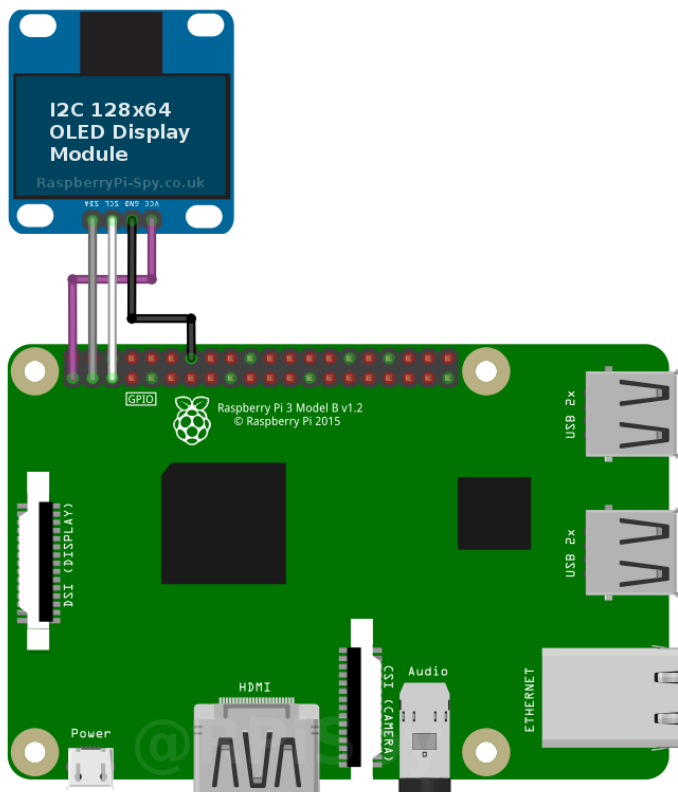
pythonRaspberry Jam

Raspberry Pi Bootcampraspbian

RetrogamingretroPiescreenSD card

securitysensormSPItemperature

ultrasonicvideowifi



Los conecté directamente al encabezado GPIO de la Raspberry Pi usando el siguiente esquema:

Pin OLED	Pi GPIO Pin	Notas
Vcc	1 *	3.3V
Gnd	14 **	Suelo
SCL	5 5	I2C SCL
SDA	3	I2C SDA

\* Puede conectar el pin Vcc al Pin 1 o 17, ya que ambos proporcionan 3.3V.

\*\* Puede conectar el pin Gnd a los pines 6, 9, 14, 20, 25, 30, 34 o 39, ya que todos proporcionan tierra.

## Habilitar interfaz I2C

La interfaz I2C está deshabilitada de manera predeterminada, por lo que debe habilitarla. Puede hacer esto dentro de la herramienta `raspi-config` en la línea de comando ejecutando:

```
sudo raspi-config
```

Para obtener detalles adicionales sobre este paso, consulte mi [Cómo habilitar la interfaz I2C en la publicación de Raspberry Pi](#).

Es posible que las siguientes bibliotecas ya estén instaladas, pero ejecute estos comandos de todos modos para asegurarse:

```
sudo apt install -y python3-dev
sudo apt install -y python-imaging python-smbus i2c-tools
sudo apt install -y python3-pil
sudo apt install -y python3-pip
sudo apt install -y python3-setuptools
sudo apt install -y python3-rpi.gpio
```

### RASPBERRY PI RELATED

[Adafruit Blog](#)

[Average Maker](#)

[Official RaspBerry Pi Site](#)

[Raspberry Pi Pod](#)

[RasPi.tv](#)

[RaspTut](#)

[Stuff About Code](#)

### TECH RESOURCES

[MattsBits - Pi Resources](#)

[Microbit Spy](#)

[Technology Spy](#)

### ARCHIVES

Select Month ▼

Si está utilizando Python 2, use estos comandos en su lugar:

```
sudo apt install -y python-dev
sudo apt install -y python-imaging python-smbus i2c-tools
sudo apt install -y python-pil
sudo apt install -y python-pip
sudo apt install -y python-setuptools
```

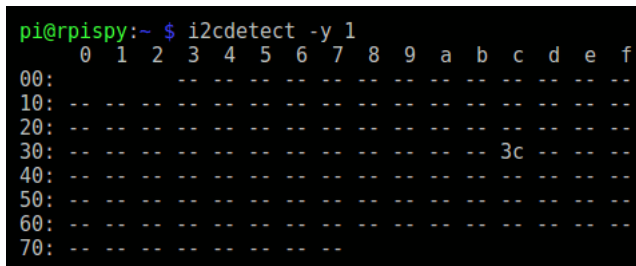
Recomendaría usar Python 3 a menos que tenga una buena razón para usar Python 2.

## Encontrar la dirección del módulo de pantalla OLED

Con las bibliotecas I2C instaladas, utilicé el comando `i2cdetect` para encontrar el módulo en el bus I2C.

```
i2cdetect -y 1
```

y obtuve el siguiente resultado:



```
pi@rpi3py:~$ i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  3c  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Esta fue una buena noticia, ya que mostró que el dispositivo había sido detectado con una dirección de "0x3c". Esta es la dirección hexadecimal predeterminada para este tipo de dispositivo. No tengo idea de por qué la PCB del dispositivo sugiere que la dirección es "0x78" cuando es claramente "0x3c".

Si tiene un Modelo B Rev 1 Pi original, escriba el siguiente comando en su lugar:

```
i2cdetect -y 0
```

## Instalar la biblioteca OLED Python

Para mostrar texto, formas e imágenes, puede usar la biblioteca Adafruit Python. Debería funcionar con todas las pantallas basadas en SSD1306, incluidos sus propios **dispositivos de 128 × 32 y 128 × 64**.

Para instalar la biblioteca, clonaremos el repositorio Adafruit git. Asegúrese de que git esté instalado ejecutando:

```
sudo apt install -y git
```

Luego clone el repositorio usando el siguiente comando:

```
git clone https://github.com/adafruit/Adafruit_Python_SSD1306.git
```

Una vez que se complete, navegue al directorio de la biblioteca:

```
cd Adafruit_Python_SSD1306
```

e instale la biblioteca para Python 2:

```
instalación de sudo python setup.py
```

y / o Python 3:

```
sudo python3 setup.py install
```

Este proceso le dará la capacidad de incluir la biblioteca dentro de sus propios scripts de Python.

## Ejemplos de scripts de Python

Ahora estamos listos para probar algunos scripts de ejemplos. Navegue al directorio "ejemplos":

```
ejemplos de cd
```

Allí debería encontrar una serie de scripts de ejemplo, tales como:

- animate.py
- botones.py
- image.py
- formas.py
- stats.py

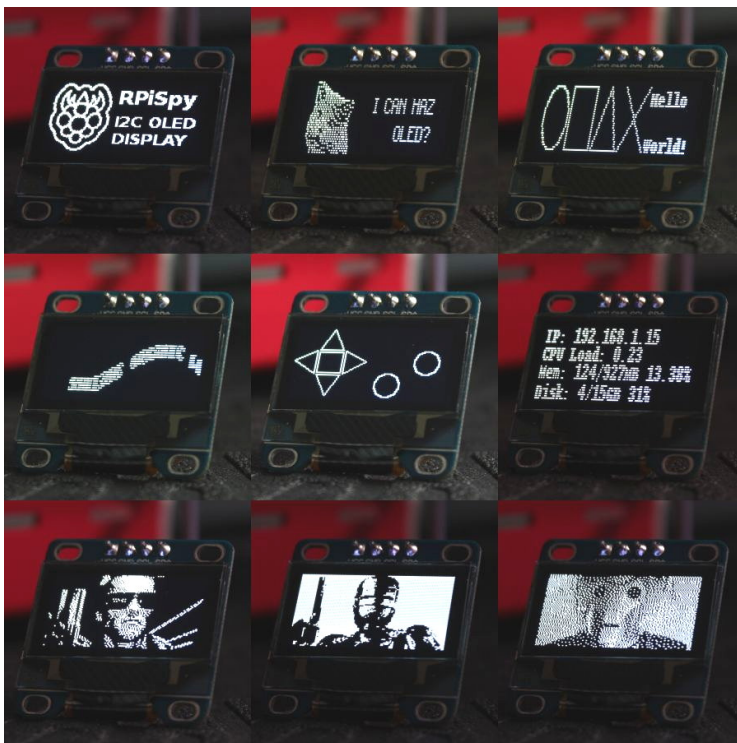
Estos ejemplos se pueden ejecutar con:

```
python shapes.py
```

o usando Python 3:

```
python3 shapes.py
```

Los ejemplos deberían proporcionarle pantallas que aparecen en los ejemplos a continuación:



Al modificar estos scripts, puede crear sus propios gráficos con formas, imágenes y texto según su proyecto. ¡Vea si puede adivinar cuáles fueron las fotos que descargué a mi Pi desde Google Images!

## Ajuste de tamaño de pantalla

Los ejemplos de Adafruit suponen que tiene una pantalla de  $128 \times 32$ . Todavía se ejecutan con una pantalla de  $128 \times 64$  píxeles, pero es mejor cambiarlos antes de pasar a algo más complicado. Para hacer esto, simplemente edite los scripts y desactive la línea de configuración  $128 \times 32$  colocando un carácter # en la parte delantera, y habilite la línea  $128 \times 64$  eliminando el carácter # en la parte frontal. La sección en el script ahora debería verse así:

```
# 128x32 display with hardware I2C:  
#disp = Adafruit_SSD1306.SSD1306_128_32(rst=RST)  
  
# 128x64 display with hardware I2C:  
disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST)
```

Este paso se vuelve esencial si desea comenzar a crear sus propias imágenes para mostrar en la pantalla.

## Crear nuevas imágenes

¿Entonces probaste el ejemplo image.py y te preguntaste cómo puedes crear tus propias imágenes? Es bastante fácil si tiene una aplicación de edición de imágenes como Photoshop o GIMP. Prefiero usar GIMP porque es gratis.

Idealmente quieres que las imágenes sean:

- Resolución  $128 \times 64$
- Color de 1 bit (es decir, blanco y negro)

Por defecto, el ejemplo image.py convertirá la imagen a 1 bit, pero supone que la resolución es correcta.

Notará en el script una línea alternativa que cambia el tamaño y convierte una imagen para que pueda cargar imágenes sin preocuparse por su tamaño y color.

Cargar imagen y convertir a color de 1 bit:

```
image = Image.open ('happycat_oled_64.ppm'). convert ('1')
```

Cargue una imagen, cambie el tamaño y convierta a 1 bit:

```
image = Image.open ('example.png'). resize ((disp.width, disp.height), Image.ANTIALIAS) .convert ('1')
```

Which technique you use is up to you. Resizing and converting takes extra processing time so in high performance applications you are better feeding the script images that have already been resized.

## Resizing and Converting Images

If you want to load an image or photo then load it into your graphics application and perform the following steps :

- Load image
- Resize/scale to  $128 \times 64$
- Convert to 1-bit colour (monochrome)
- Export as a “.pbm” or “.png” file
- Copy to your Pi in the same location as your Python script
- Update the Python script to use your new file

The Adafruit example image is a “ppm” file because it is colour although it is converted to monochrome at the point it is displayed on the screen. Adafruit use ppm as the library also supports their colour OLED modules. If you don't have a colour screen you can switch to pbm or png.

I prefer creating “pbm” files as they are black and white and much smaller files. It also means your Python script doesn't need to convert them. The library can handle both just make sure you use the correct filename and extension in your scripts.

## Increasing I2C Bus Speed

If you are displaying multiple images per second it is worth increasing the bus speed of the interface as it can improve performance. Please see the [Change Raspberry Pi I2C Bus Speed](#) post .

## Troubleshooting

If your screen isn't working you should start at the beginning of this tutorial and work through it. Here are some thing to consider :

- Did you enable I2C and instal "python-smbus" and "i2c-tools"?
- Are the four module connections correct? Did you get SDA and SCL mixed up?
- Did "i2cdetect -y 1" give you the address of the display on the I2C bus?
- If your screen is using an address other than 0x3c did you adjust the Python script?

## Buy a Miniature OLED Screen

These screens are available from a number of retailers so take a look and pick one that is convenient for your location :

- [Adafruit](#)
- [The PiHut](#)
- [ModMyPi](#)
- [eBay](#)
- [Amazon](#)

Read the descriptions carefully as some OLED display modules use the SPI interface rather than I2C. Those are fine but you'll need to follow a different tutorial to use that style.

SHARE.



[← PREVIOUS ARTICLE](#)

[Raspberry Pi RetroPie Shutdown Button](#)

[NEXT ARTICLE >](#)

[Cheap SD Cards from eBay are Fake](#)

### RELATED POSTS

OCTOBER 21, 2019

2

[Pi-Hole OLED Status Screen](#)

JUNE 4, 2019

4

[Using a USB Audio Device with the Raspberry Pi](#)

FEBRUARY 5, 2019

1

[Setting up SSH Keys on the Raspberry Pi](#)

DECEMBER 15, 2018

3

[Running Flask under NGINX on the Raspberry Pi](#)

DECEMBER 8, 2018

3

[Remote Access to a Raspberry Pi using MobaXterm](#)

NOVEMBER 5, 2018

0

[Raspberry Pi 7-Segment LED Display Module using Python](#)

SEPTEMBER 4, 2018

0

JUNE 16, 2018

0

MAY 13, 2018

1