**Conectando Pantalla Oled a Raspberry**

Como parte de todo Proyecto, la visualización de datos es una parte esencial en la presentación, en este caso la visualización en físico se hará mediante una pantalla Oled. Mi módulo de pantalla OLED es un módulo de pantalla LED LCD OLED de 0,96″ I2C IIC SPI Serial 128X64. 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.

**Materiales Necesarios.**

1. Raspberry pi
2. Pantalla Oled
3. Conectores tipo jumper

**Configuración del módulo de visualización.**

Mi pantalla tenía cuatro pines, dos para alimentación y dos para la interfaz I2C. Los conecté directamente al encabezado GPIO de Raspberry Pi usando el siguiente esquema:

OLED Pin Pi GPIO Pin Notas

vcc 1 \* 3,3 V

Tierra 14 \*\* Terrestre

SCL 5 SCL I2C

ASD 3 SCA I2C

\* Puede conectar el pin Vcc al pin 1 o al 17, ya que ambos proporcionan 3,3 V.

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

**Circuito sugerido para conectar la pantalla con la Raspberry**

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

**El circuito ya integrado a los demás elementos.**

Diagrama

Descripción generada automáticamente

**Activar la interfaz I2C en la Raspberry**

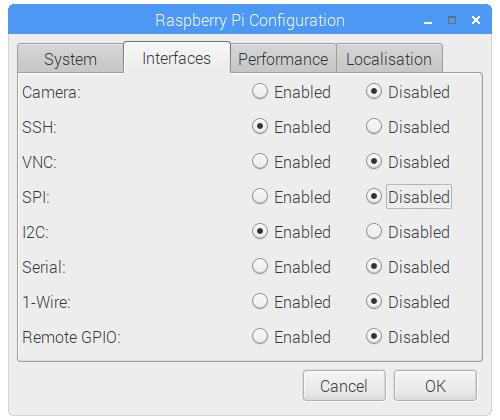
Hay dos maneras:

1. Ejecutar en la terminal Raspberry Pi >>sudo raspi-config

Seleccione Opciones de interfaz> I2C, aparece la siguiente interfaz, confirme y reinicie.



2. Para la versión de escritorio de Raspberry Pi, haga clic en el botón de menú en la esquina superior izquierda del escritorio, luego seleccione "Preferencias / Preferencias"> "Configuración de Raspberry Pi", luego seleccione la pestaña Interfaces en la ventana emergente y seleccione Habilitar en la opción I2C, como se muestra en la siguiente figura.



**Preparación del entorno de Raspberry Pi**

Python 3 puede ejecutar la siguiente serie de comandos para autocomprobación:

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

**Descargue e instale la biblioteca de controladores de pantalla SSD1306, CANVAS y SH1106 para Raspberry Pi**

Hay muchas bibliotecas de controladores de pantalla SSD1306 de código abierto, lo que uso es Adafruit. La biblioteca de Python se puede utilizar para pantallas de 128x32 y 128x64. La dirección es:https://github.com/adafruit/Adafruit\_Python\_SSD1306。

Hay dos formas de descargar e instalar:

Clona directamente desde gitHub a local

git clone https://github.com/adafruit/Adafruit\_Python\_SSD1306.git

cd Adafruit\_Python\_SSD1306

Bajo Python 3: >>sudo python3 setup.py install;

Bajo Python 2:>>sudo python setup.py install

Utilice pip para descargar

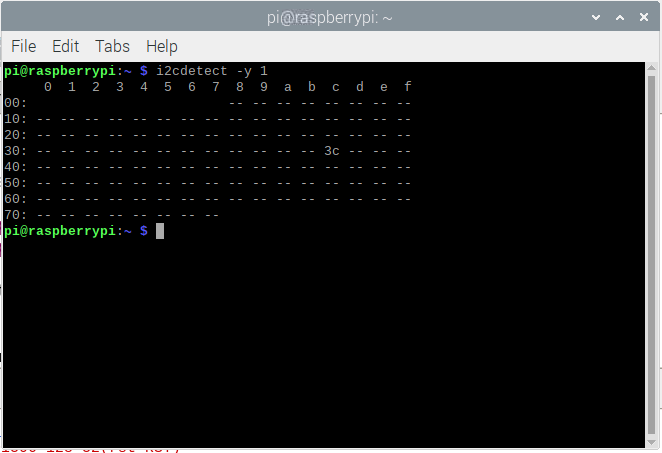
>>sudo pip install Adafruit-SSD1306

Y para instalar la librería SH1106 desde thonny la buscamos en el package manager.

**Búsqueda de 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



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.

**Programa en Python. (Python/Subprogramas/pantalla\_oled.py)**

# Librerias para la pantalla Oled

from luma.core.interface.serial import i2c

from luma.core.render import canvas

from luma.oled.device import sh1106

import socket

from PIL import ImageFont, ImageDraw, Image

#Para la pantalla

serial = i2c(port=1, address=0x3c)

#device = ssd1306(serial, rotate=0)

device = sh1106(serial, width=128, height=64, rotate=0)

#device.capabilities(width=128, height=64, rotate=0)

print("size: " , device.bounding\_box)

device.clear()

with canvas(device) as draw:

draw.rectangle(device.bounding\_box, outline="white", fill="black")

#font = ImageFont.load\_default(size=12)

#font = ImageFont.truetype(, size=12)

draw.text((30, 20), "Código IoT", fill="white")

draw.text((10, 30),"Hola mundo", fill="white")