

Conexión de una pantalla LCD 16x2 a la Raspberry Pi para la presentación de mensajes en pantalla.

Bryan V. Alvarado-Evelin E. Hidalgo-Carlos E. Orellana- Carlos S. Veloz

Departamento de Eléctrica y electrónica, Universidad de las Fuerzas Armadas "ESPE", Sangolquí, Ecuador

May 28, 2019



PLANTEAMIENTO DEL PROBLEMA

Se desconoce el funcionamiento, control y conexión de una Raspberry Pi a una pantalla LCD 16x2 para la presentación de mensajes, para lo cual se formularon las siguientes preguntas:

¿Cómo funciona una Raspberry Pi?

¿Cuál es el lenguaje de programación utilizado para controlar la pantalla LCD 16x2 para presentación de mensajes?

¿Cómo se conecta una pantalla LCD 16x2 a una Raspberry Pi?

Conectar una pantalla LCD 16x2 a una Raspberry Pi 3 - Modelo B para proyectar un mensaje en la LCD.

- Comprender la estructura de control básica de Python para controlar una pantalla LCD 16X2 mediante una Raspberry Pi.
- Conocer la función y estructura de cada pin que integra una LCD 16x2.
- Identificar los pines de configuración y las conexiones necesarias para establecer comunicación entre la pantalla LCD 16x2 y la Raspberry Pi.

OBJETIVOS

OBJETIVO GENERAL

Conectar una pantalla LCD 16x2 a una Raspberry Pi 3 - Modelo B para proyectar un mensaje en la LCD.

OBJETIVOS ESPECÍFICOS

- ▶ Comprender la estructura de control básica de Python para controlar una pantalla LCD 16X2 mediante una Raspberry Pi.
- ▶ Conocer la función y estructura de cada pin que integra una LCD 16x2.
- ▶ Identificar los pines de configuración y las conexiones necesarias para establecer comunicación entre la pantalla LCD 16x2 y la Raspberry Pi.

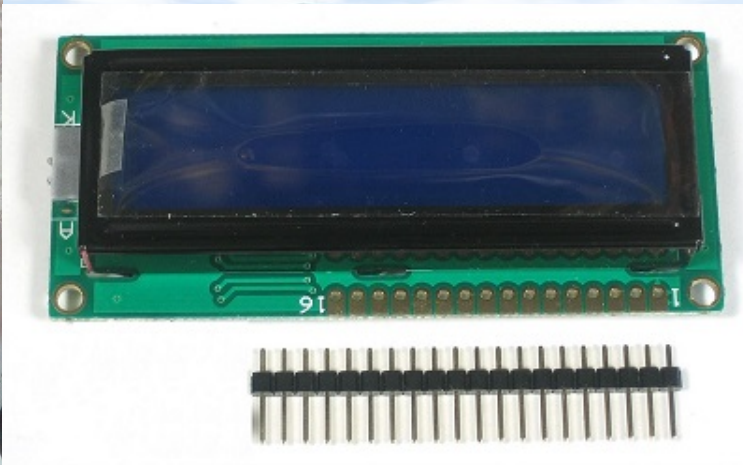
MARCO TEÓRICO

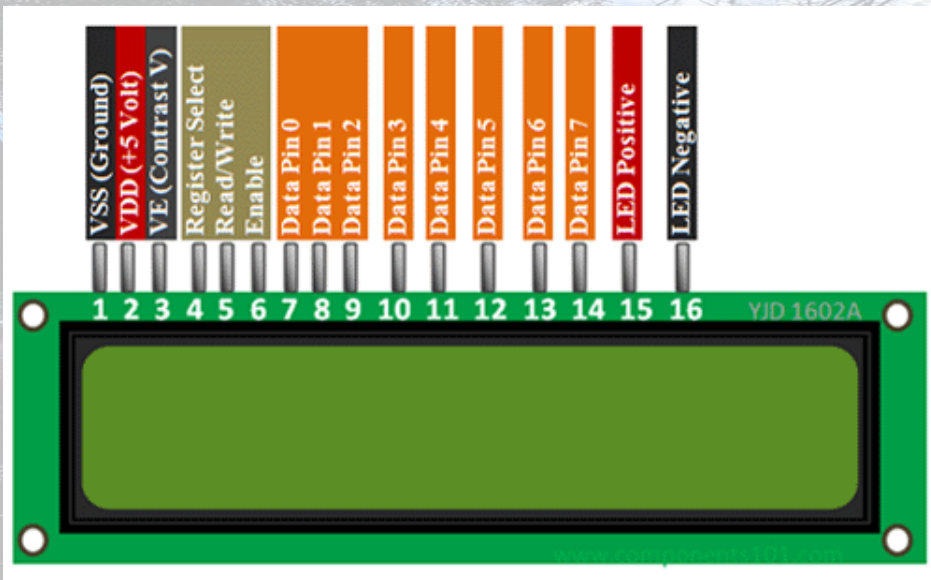
Raspberry Pi

Es un pequeño ordenador con arquitectura ARM el cual está montado en forma de placa de circuito integrado, pensado para ocupar un mínimo espacio de bajo coste y bajo consumo cuyos primeros modelos fueron lanzados en abril de 2012.

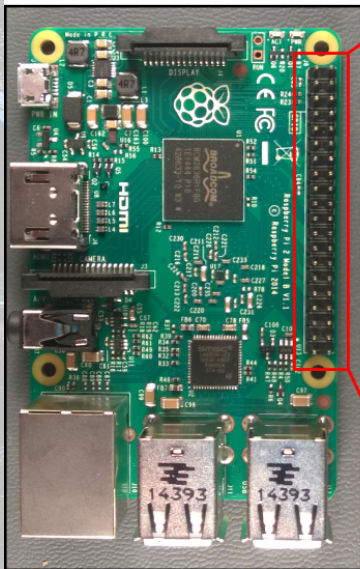
Pantalla LCD 16x2

El LCD(Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento.



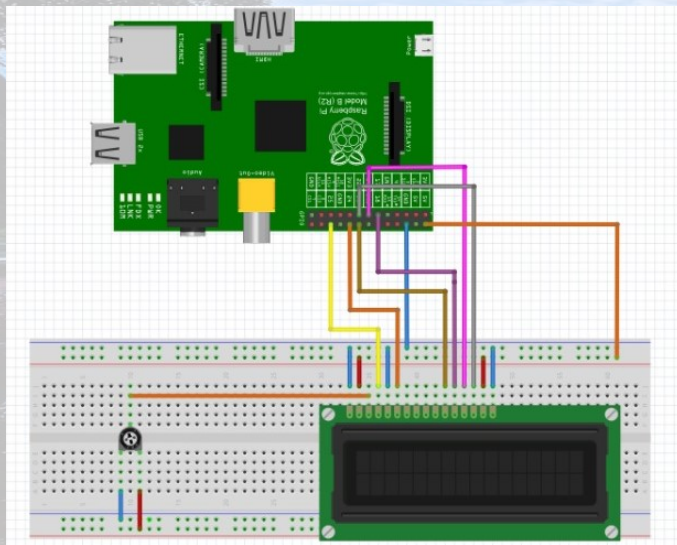


Pin	Nombre del Pin	Función del Pin
1	Vss	Masa
2	VDD	+ 5 V
3	Vee	Ajuste de contraste
4	RS	Selección de modo
5	RW	Lectura / Escritura
6	EN	Habilitación (1) / Deshabilitación (0)
7	DB0	Línea de datos (LSB)
8	DB1	Línea de datos
9	DB2	Línea de datos
10	DB3	Línea de datos
11	DB4	Línea de datos
12	DB5	Línea de datos
13	DB6	Línea de datos
14	DB7	Línea de datos (MSB)
15	LED+	LED (ánodo)
16	LED-	LED (cátodo)



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21
					SPI0 CS0
					SPI0 CS1
					SPI1 CS0
					SPI1 MOSI
					SPI1 SCLK

DIAGRAMA ELECTRÓNICO



LISTA DE COMPONENTES

- ▶ Raspberry Pi
- ▶ raspbian
- ▶ Python 3IDLE
- ▶ 1 Cable HDMI a VGA
- ▶ Potenciómetro 10K
- ▶ Cables conectores
- ▶ Pantalla LCD 16x2
- ▶ Proto board
- ▶ Micro SD de 8 GB

EJECUCIÓN DEL TRABAJO DE INVESTIGACIÓN

Para ejecutar el código en Python en la Raspberry pi 3 – modelo B, primero se debe instalar el sistema operativo en el micrordenador, la descarga del software se la realiza directamente de la página oficial de Raspberry: <https://www.raspberrypi.org>



Descargar la interfaz NOOBS, por defecto tiene precargado el sistema operativo RASBIAN, en esta interfaz se va a implementar el código fuente de Python para realizar el control de leds, NOOBS se descarga como un archivo comprimido



era Raspberri: X +

https://www.raspberrypi.org/downloads/noobs/

Puede comprar una tarjeta NOOBS SD preinstalada en muchos minoristas, como [Pimoroni](#), [Adafruit](#) y [The Pi Hut](#), o descargar NOOBS a continuación y seguir la [guía de configuración del software](#) y el [video de la guía de configuración de NOOBS](#) en nuestras páginas de ayuda.

NOOBS es un instalador de sistema operativo fácil que contiene [Raspbian](#) y [LibreFLEC](#). También proporciona una selección de sistemas operativos alternativos que luego se descargan de Internet y se instalan.

NOOBS Lite contiene el mismo instalador de sistema operativo sin Raspbian precargado. Proporciona el mismo menú de selección de sistema operativo que permite descargar e instalar Raspbian y otras imágenes.



NOOBS
Desconectado e instalación en red
Versión: 3.0.1
Fecha de lanzamiento: 2019-04-08
[Descargar Torrent](#) [Descargar ZIP](#)

SHA-256: 551d8d9352dc02a6fee2010de0217aeefee11f08c44b391b7c428043 SHA-256: 7c2386814201af017be8fe6b27737011aff61008d9d6d3e87e013c17b318db4
36355422f



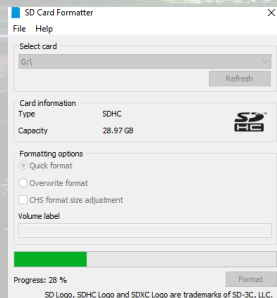
NOOBS Lite
Solo instalación de red
Versión: 3.0
Fecha de lanzamiento: 2018-11-16
[Descargar Torrent](#) [Descargar ZIP](#)

Nota: Raspbian y NOOBS contienen productos de la plataforma Java SE, licenciados bajo el Acuerdo de licencia de código binario de Oracle disponible [aquí](#). Mathematica y Wolfram Language se incluyen en esta [versión de NOOBS](#).

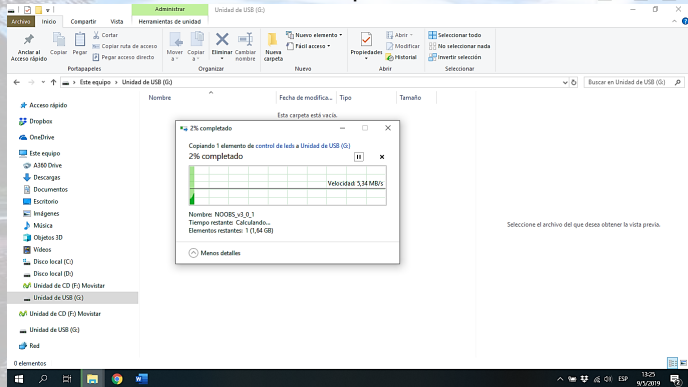
pi.org/NOOBS_latest

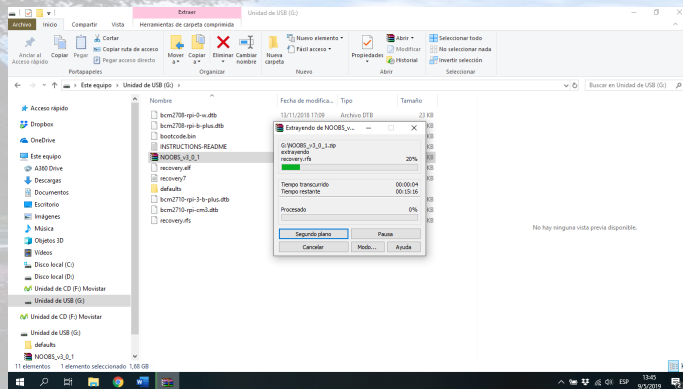
Para guardar el sistema operativo se utiliza una tarjeta Micro SD, pero antes se la debe formatear, el programa que se utiliza para formatear la tarjeta es de libre acceso en el siguiente link: https://www.sdcard.org/downloads/formatter_4/eula_windows/

Según el equipo se realiza la descarga para 32 o 64 bits, en este caso es para 64 bits. Al ejecutar el programa debemos seleccionar la unidad del disco donde se encuentra la Micro SD y el formato deseado.



Una vez formateada se debe copiar el archivo NOOBS_v3_0_1.zip en la tarjeta y descomprimirlo





EXPLICACIÓN DEL CÓDIGO FUENTE

```
*untitled*
File Edit Format Run Options Window Help

#-----LIBRERIAS-----

import board      # Esta libreria sirve para asignar constantes fijas a los pines
                  # del tablero, esto hace que usar el módulo de la placa sea
                  # más seguro y confiable

import digitalio  # Este módulo sirve utilizar la pantalla LCD con una
                  # retroalimentación de un solo color, en el caso que se
                  # desee una retroalimentación RGB (varios colores) se debe
                  # definir los pines de salida según su ubicación en el board
                  # y el color

import adafruit_character_lcd.character_lcd as characterlcd
|
    # Este módulo permite escribir fácilmente el código de Python
    # que controla una LCD de caracteres (ya sea con luz de fondo
    # individual o con luz de fondo RGB)

#----- TAMAÑO DE LA PANTALLA LCD -----

lcd_columns = 16  # Estas líneas de código definen el tamaño de caracteres de
lcd_rows = 2      # nuestra LCD, en este caso la pantalla LCD es de 16 columnas
                  # y 2 filas, si se tiene una LCD de caracteres de diferente
                  # tamaño se debe modificar estos valores

#---- ASIGNACIÓN DE LOS PINES DE CONFIGURACIÓN DE LA RASPBERRY PI Y LCD -----

#Según la ubicación en la placa :      -LCD-      -RASPBERRY PI 3-MODELO B-
lcd_rs = digitalio.DigitalInOut(board.D22) # pin 4          pin 15
lcd_en = digitalio.DigitalInOut(board.D17) # pin 6          pin 11
lcd_d4 = digitalio.DigitalInOut(board.D25) # pin 11         pin 22
lcd_d5 = digitalio.DigitalInOut(board.D24) # pin 12         pin 18
lcd_d6 = digitalio.DigitalInOut(board.D23) # pin 13         pin 16
lcd_d7 = digitalio.DigitalInOut(board.D18) # pin 14         pin 12

# INICIALIZAMOS LA CLASE LCD

lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                     lcd_d7, lcd_columns, lcd_rows)
```

```
File Edit Format Run Options Window Help
lcd.py: C:\Users\user\Desktop\lcd.py: pan 21 pan 22

# INICIALIZAMOS LA CLASE LCD

lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                       lcd_d7, lcd_columns, lcd_rows)

# Esta línea de código especifica el tipo de retroalimentación de la pantalla
# reconociendo la función de cada pin de la LCD, la retroalimentación
# es de un solo color, esto queda especificado en el argumento "Mono", para varios
# colores el argumento sería "RGB"

# CUERPO (VOID)

print (' Mensaje Proyectado \n') # En esta línea se imprime un mensaje que
                                # indique que el texto ya se proyecta en la LCD

lcd.message = "HERRAMIENTAS DE\n SOFTWARE 5035" # Esta línea sirve para ESCRIBIR
                                                  # el mensaje que se va a proyectar
                                                  # en la pantalla LCD, tomando en cuenta
                                                  # que solo se pueden escribir 16 caracteres
                                                  # por línea de texto, para escribir
                                                  # en la siguiente línea se utiliza
                                                  # el argumento "\n"

print ('Dígame 2 para borrar la pantalla LCD') # En esta línea se imprime un mensaje
                                                # que indica al usuario que debe digitar el
                                                # número 2 para borrar la pantalla del LCD

a= int(input()) # Se guarda en una variable el valor ingresado por el usuario, el valor es
                # de tipo entero "int" e "input" debido a que es un dato de entrada

if a==2: # Primera condición a cumplir según el dato que ingrese el usuario

    lcd.clear() # Esta línea sirve para BORRAR la pantalla LCD

else: # Contra punto de la primera condición

    print ('Opción incorrecta') # En el caso que se digite un número distinto a 2 se imprime
                                # el siguiente mensaje
```


Asignación de pines Raspberry Pi-LCD16x2

RASPBERRY PI 3		LCD	
GND	PIN 6	VSS	PIN 1
		RW	PIN 5
		K(-)	PIN 16
5V PWR	PIN 2	VDD	PIN 2
		A(+)	PIN 15
GPIO 25	PIN 22	D4	PIN 11
GPIO 24	PIN 18	D5	PIN 12
GPIO 23	PIN 16	D6	PIN 13
GPIO 18	PIN 12	D7	PIN 14
GPIO 22	PIN 15	RS	PIN 4
GPIO 17	PIN 11	E	PIN 6
POTENCIOMETRO		LCD	
TERMINAL	SEÑAL	Vo/VE	PIN 3
GND	PROTO -		
5V	PROTO +		

RESULTADOS



APORTACIONES

Thonny - /home/pi/Desktop/Prueba 1.py @ 33 : 43

New Load Save Run Debug Over Into Out Resume Stop

Prueba 1.py ✕

```
22 # initialize the LCD class
23 lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
24                                       lcd_d7, lcd_columns, lcd_rows)
25
26
27 print ('Digite 1 para proyectar el mensaje ')
28 a= int(input())
29 if a!=1:
30     print ('opcion incorrecta')
31 else:
32     if a==1:
33         lcd.message = "INTERFAZ \n  AMIGABLE"
34         print ('Digite 2 para borrar la pantalla')
35         b= int(input())
36         if b==2:
37             lcd.clear()
38             print ('pantalla borrada')
39         else:
40             print ('opcion incorrecta')
41
```

Variables

Name	Value
a	1
b	2
board	<module 'board' from '/usr/l
characterlcd	<module 'adafruit_characte
digitalio	<module 'digitalio' from '/us
lcd	<adafruit_character_lcd.che
lcd_columns	16
lcd_d4	<digitalio.DigitalInOut objec
lcd_d5	<digitalio.DigitalInOut objec
lcd_d6	<digitalio.DigitalInOut objec
lcd_d7	<digitalio.DigitalInOut objec
lcd_en	<digitalio.DigitalInOut objec
lcd_rows	2
lcd_rs	<digitalio.DigitalInOut objec
os	<module 'os' from '/usr/lib/l
sleep	<built-in function sleep>
time	<module 'time' (built-in)>

Shell

```
>>> %Run 'Prueba 1.py'
Digite 1 para proyectar el mensaje
1
Digite 2 para borrar la pantalla
2
pantalla borrada
>>> |
```

Th Thonny - /home/pi/_ [pi]

00:05

CONCLUSIONES

En conclusión:

- ▶ La mejor forma de establecer la comunicación entre la Raspberry Pi y la pantalla LCD, es mediante la utilización de la interfaz de Python, simplemente se debe importar la librería Adafruit y definir los pines de salidas de datos, los dispositivos actuarán según las instrucciones que se indiquen en el código fuente.
- ▶ Los pines de la pantalla LCD cumplen con una función específica para la transmisión y respuesta de datos, esto facilita la interacción entre el usuario y la consola, resultado de dicha interacción es la proyección del mensaje en la LCD.
- ▶ Tanto la Raspberry Pi como la pantalla LCD son dispositivos fácilmente adaptables a un entorno de programación con Python siempre y cuando se limite de forma correcta sus variables y se identifique tanto sus pines de control de datos como los pines de configuración y de alimentación.

BIBLIOGRAFÍA

[1]Akila, IS, Akshaya, B., Deepthi, S., y Sivadharshini, P. (2018). Un lector de texto para discapacitados visuales utilizando Raspberry Pi. 2018 Segunda Conferencia Internacional sobre Metodologías de Computación y Comunicación (ICCMC). doi: 10.1109 / iccmmc.2018.8487513

[2]Pattichis, MS, Celedon-Pattichis, S., y LopezLeiva, C. (2017). Enseñanza de imágenes y procesamiento de video utilizando las matemáticas de la escuela media y la Raspberry Pi. Conferencia Internacional IEEE 2017 sobre Acústica, Procesamiento de Voz y Señal (ICASSP). doi: 10.1109 / icassp.2017.7953378