

# CONEXIÓN DE UNA PANTALLA LCD 16x2 A LA RASPBERRY PI PARA LA PRESENTACIÓN DE MENSAJES

Bryan V. Alvarado <sup>1\*</sup>, Evelin E. Hidalgo <sup>1</sup>, Carlos E. Orellana <sup>1</sup>, Carlos S. Veloz <sup>1</sup>

*Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Sangolguí, Ecuador*

bvalvarado@espe.edu.ec

## I. 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:

- ¿Cuál es el sistema operativo que utiliza la Raspberry Pi?
- ¿Cuál es el lenguaje de programación utilizado para controlar la pantalla LCD 16x2 para presentación de mensajes?
- ¿Cómo funciona la Pantalla LCD 16x2?

## II. ABSTRACT

The Raspberry Pi is a small computer with ARM architecture which is mounted in the form of an integrated circuit board, designed to occupy a minimum space of low cost and low consumption whose first models were launched in April 2012. A fulfilled the expectations that deposited in her beginners and experts. Exploiting Raspberry Pi's functionalities as well as possible makes it necessary to choose a programming language among the ocean of existing technologies.

The option that has been taken is Python to establish the relationship and connection with the 16x2 LCD screen, device used to display content or information in a graphic way, using characters, symbols or small drawings depending on the model, governed by a microcontroller which directs all its operation and thus be able to present a message on said screen.

## III. INTRODUCCIÓN

La 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. Generalmente este tipo de mini ordenadores ejecutan sistemas operativos basados en Linux y está íntimamente relacionados con el Open Software. No obstante, el desarrollo de Raspberry Pi no es Open Hardware.

A cumplido con las expectativas que depositaban en ella principiantes y expertos. Explotar lo mejor posible las funcionalidades de la Raspberry Pi, hace necesario elegir un lenguaje de programación entre el océano de tecnologías existentes.

La opción que se ha tomado es Python para poder establecer la relación y conexión con la pantalla LCD 16x2, dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo, gobernada por un microcontrolador el cual dirige todo su funcionamiento y así poder presentar un mensaje en dicha pantalla.

## IV. ESTADO DEL ARTE

En 2018, Dr. IS Akila, Akshaya, Deepthi y Sivadarshini del Department of ECE, Coimbatore Institute of Technology, desarrollaron un lector de texto para discapacitados visuales utilizando Raspberry Pi, basado en un prototipo que ayuda al usuario a escuchar. Esto implica la extracción de texto de la imagen y convertir el texto a la salida de audio traducida al idioma inglés. Esto se hace usando una Raspberry Pi 3 modelo B y un módulo de cámara con el conceptos de Tesseract OCR [carácter óptico Motor de reconocimiento] y API de Google Speech [Application Program Interface] que es el textual entrada al motor del habla. El modelo esta programado utilizando el Lenguaje Python (Akshaya, 2018, p.1) [1].

Mario Pattichis, Silvia Celedon-Pattichis y Carlos Lopez del Department of Electrical and Computer Engineering y Department of Language de la University of New Mexico en 2017, desarrollaron un proyecto de enseñanza de imágenes y procesamiento de video utilizando las matemáticas de la escuela media y la Raspberry Pi. El proyecto de ingeniería AOLME utiliza un currículum integrado que se basa en el uso de bibliotecas de Python que se ejecutan en la Raspberry Pi (Pattichis, 2017, p.1) [2].

En general se demostró que el aprendizaje se facilita cuando se emplea una representación de imágenes y video digitales mientras se trabaja en un entorno de programación real trabajando en Linux y no requirió el uso de interfaces gráficas de usuario para aprender a programar (Pattichis, 2017, p.1) [2].

En 2015, Manisha Sharma, Nidhi Agarwal, Dr. SRN Reddy realizaron el diseño y desarrollo de una placa hija para comunicación USB-UART entre Raspberry Pi y PC. Desarrollo de aplicaciones en tiempo real utilizando plataformas como Raspberry Pi que a menudo requieren Interfaz de diferentes sensores y comunicación. Pueden interconectarse fácilmente usando un puerto RS232, pero hoy en día se sustituye por USB. Este trabajo propone el diseño y desarrollo de una placa para comunicación USB a UART a la Raspberry Pi que proporciona una comunicación con diferentes dispositivos mediante el uso de USB a UART, con la ayuda de estos convertidores, se logra la transmisión de datos bidireccional FT232RL los IC CP2102 se utilizan en las juntas secundarias propuestas (Sharma, 2015, p.1) [3].

En 2016, Ing. Felix Palacios desarrolló un Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. El presente trabajo muestra la implementación de un Sensor Biométrico GT511C3, un Arduino Nano, una Raspberry Pi 2.0, una pantalla táctil LCD 3.5" y un módulo USB Wifi, el entorno de programación para el desarrollo del aplicativo Web, se hizo mediante los módulos Django y Adafuit sobre la base de Python. El sistema tiene la capacidad de registrar huellas de personas, las cuales se les

asigna un identificador entero de entre 0 a 199 y almacenarlo en una base de datos instalada como es MySQL, además tiene la capacidad de almacenar nombres asociados al identificador generado, en la base de datos con el cual se puede habilitar el acceso hacia algún lugar específico (Palacios,2016,p.1) [4].

Holger Romero, Alonso Pacheco y Hernan Ramos estudiantes de la Universidad Católica de Santa María, Perú en 2017 implementaron un sistema de monitoreo para supervisar el consumo de energía eléctrica doméstica; esta propuesta requirió la utilización de la placa Arduino Mega, shield Ethernet, el sensor de corriente y la pantalla LCD. Para el monitoreo del consumo eléctrico local se utilizó la pantalla LCD y para un monitoreo remoto se desarrolló una página web alojada en el Shield Ethernet que permite analizar en tiempo real los eventos y alertas mediante cualquier navegador web. El proyecto brindó la suficiente información a los usuarios para controlar su consumo de energía y así tomar la mejor decisión para generar un ahorro continuo y fortalecer una conciencia amigable con el buen uso de la energía eléctrica (Romero,2017,p.1) [5].

Para el trabajo de investigación presente, se utilizó el entorno de programación del módulo Adafruit dirigido a una Raspberry Pi 3 modelo B y una pantalla LCD 16x2 (Palacios Felix, 2016), el entorno de programación para el aplicativo se hizo sobre la base de Python (IS Akila, Akshaya, Deepthi, Sivadarshini, 2018). El sistema tiene la capacidad de proyectar un mensaje en la pantalla LCD y borrarlo después de una previa instrucción (Manisha Sharma, Nidhi Agarwal, SRN Reddy, 2015).

## V. MARCO TEÓRICO

Raspberry Pi 3 – modelo B Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo desarrollado en el Reino Unido. Aunque no se indica expresamente si es hardware libre o con derechos de marca, es compatible con varios Sistemas operativos de libre acceso, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, promueve principalmente el aprendizaje del lenguaje de programación Python – IDLE, otros lenguajes también soportados son: Tiny BASIC,11 C, Perl4 y Ruby.12 [6]

La Raspberry Pi 3 está construida alrededor del nuevo procesador BCM2837 ARMv8 de 64bits con 1,2GHz de velocidad, mucho más rápido y con mayor capacidad de procesamiento que sus antecesores, integra el chip BCM43143 que la dota con conectividad Wifi b/g/n y Bluetooth 4.1 LE de bajo consumo y cuenta con administración de energía mejorada que permite trabajar con más dispositivos USB externos, permite usar fuentes de hasta 2.5A para proveer más energía a los puertos USB [5].

### V-A. Pines GPIO Raspberry Pi 3

La distribución de los pines GPIO del modelo Raspberry Pi 3 es la siguiente:

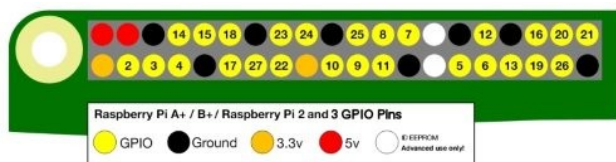


Fig 2.Pines GPIO Raspberry Pi 3

Los pines GPIO tienen funciones específicas (aunque algunos comparten funciones) y se pueden agrupar de la siguiente manera:

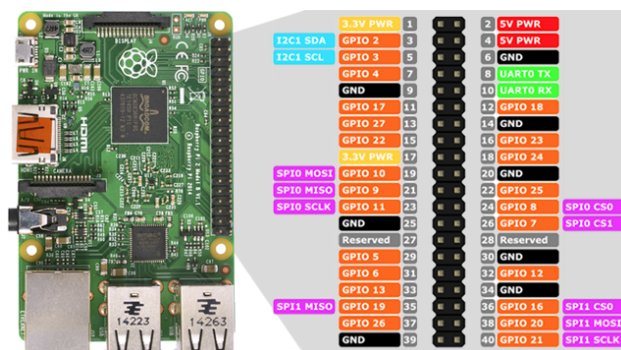


Fig 3.Pines GPIO para Raspberry Pi 2 Modelo B

- Amarillo (2): Alimentación a 3.3V.
- Rojo (2): Alimentación a 5V.
- Naranja (26): Entradas / salidas de propósito general. Pueden configurarse como entradas o salidas. Ten presente que el nivel alto es de 3.3V y no son tolerantes a tensiones de 5V.
- Gris (2): Reservados.
- Negro (8): Conexión a GND o masa.
- Azul (2): Comunicación mediante el protocolo I2C para comunicarse con periféricos que siguen este protocolo.
- Verde (2): Destinados a conexión para UART para puerto serie convencional.
- Morado (5): Comunicación mediante el protocolo SPI para comunicarse con periféricos que siguen este protocolo.

Todos los pines son de tipo "unbuffered", es decir, no disponen de buffers de protección y puedes dañar la placa con un mal uso. Existen 2 formas de numerar los pines de la Raspberry Pi, en modo GPIO o en modo BCM. En el modo GPIO, los pines se numeran de forma física por el lugar que ocupan en la placa (representados por el color gris) viene siendo igual para todas las versiones (comenzamos a contar desde arriba a la izquierda y finalizamos abajo a la derecha) [7].

### V-B. Python – IDLE

IDLE (Integrated Development and Learning Environment) es un entorno de desarrollo integrado (IDE) para Python. El instalador de Python para Windows contiene el módulo IDLE de manera predeterminada. IDLE no está disponible de forma predeterminada en las distribuciones de Python para Linux. Debe instalarse utilizando los respectivos gestores de paquetes [5].

IDLE se puede usar para ejecutar una sola declaración como Python Shell y también para crear, modificar y ejecutar scripts de Python. IDLE proporciona un editor de texto con todas las funciones para crear scripts de Python que incluye funciones como resaltado de sintaxis, autocompletado y sangría inteligente. También tiene un depurador con características de pasos y puntos de interrupción [8].

### V-C. 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 [6].

El LCD 16x2 tiene en total 16 pines hay que tomar en cuenta que la posición correcta del display es con los pines en la parte superior, aunque existen modelos en los que la posición correcta

es con los pines en la parte inferior . La datasheet (pdf) muestra 14 pines, los dos pines adicionales son el ánodo (15) y el cátodo (16) del LED de fondo. Debe notarse que el controlador Hitachi HD44780 se encuentra incorporado al circuito impreso del módulo LCD y sirve de interfaz entre la propia pantalla LCD (donde se muestran los caracteres) y el microcontrolador PIC. Por lo tanto, de todos los pines del HD44780 únicamente se tiene acceso a aquellos necesarios para la conexión al PIC y para el control de contraste [8].

.50.6.5  
Fig 8.Pines LCD 16x2

Pin No	Name	Function	Description
1	Vss	Power	GND
2	Vdd	Power	+ 5 V
3	Vee	Contrast Adj.	(-2) 0 - 5 V
4	RS	Command	Register Select
5	R/W	Command	Read / Write
6	E	Command	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

Fig 6.Pines LCD 16X2

Pines de alimentación:

Vss: Gnd

Vdd: +5 voltios

Vee: corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.

Pines de control:

RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos(1). Es decir el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter [7].

RW: Corresponde al pin de Escritura(0) o de Lectura(1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.

E: Corresponde al pin Enable o de habilitación. Si E(0) esto quiere decir que el LCD no esta activado para recibir datos, pero si E(1) se encuentra activo y podemos escribir o leer desde el LCD.

Pines de Bus de datos:

El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD se puede hacer utilizando 8 bits del bus de datos(D0 a D7) o empleando los 4 bits mas significativos del bus de datos(D4 a D7) [7].

Se realizó una conexión eléctrica que nos brinda la posibilidad de controlar un punto de luz desde dos lugares distintos en este caso se simuló la planta baja y alta de una vivienda que consta de dos focos conectados en paralelo y dos conmutadores en serie.

## VI. PROPUESTA

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



Fig 2.Página principal de Raspberry pi

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.



Fig 2.Página de descarga NOOBS

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/formatter4/eula\\_w10/windows/](https://www.sdcard.org/downloads/formatter4/eula_w10/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.

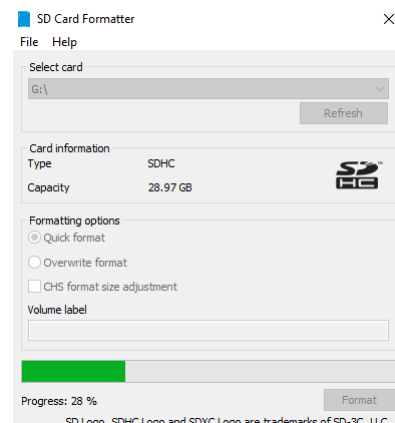


Fig 2.Programa para formatear unidades de almacenamiento

Una vez formateada se debe copiar el archivo NOOBS\_v301.zip en la tarjeta y descomprimirlo

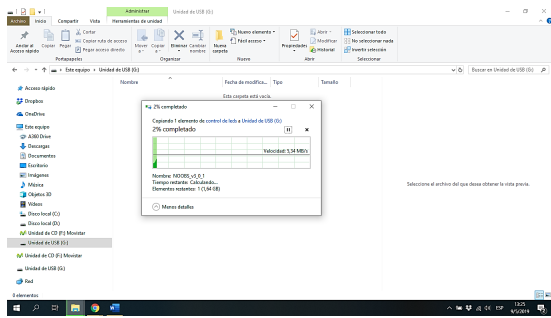


Fig 2. Interfaz de NOOBS en la tarjeta Micro SD

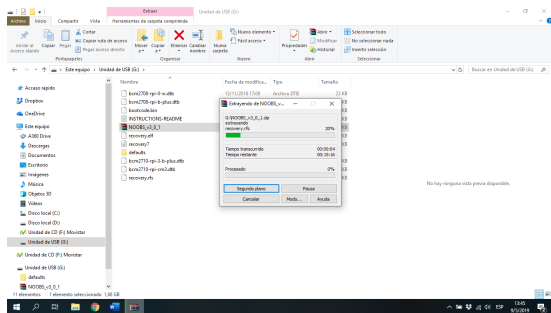


Fig 2. Interfaz de NOOBS en la tarjeta Micro SD

Para finalizar conectamos la Raspberry pi a una fuente de poder de 5v-3A y un cable HDMI hacia un periférico que admita este tipo de conexión, al momento que Raspberry ejecuta la interfaz, automáticamente se empieza a expandir el sistema operativo en la Micro SD, seleccionamos instalar Raspbian, al cabo de unos minutos el dispositivo está listo para ser utilizado, Raspbian por defecto tiene instalado Python 3 IDLE [8].

Una vez diseñado el circuito en el proto, se procede a conectar....  
Proyectar un mensaje en una pantalla LCD 16x2 usando el lenguaje de programación Python con el siguiente código:

```
#-----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)
```

Fig 1. Código Fuente

```
#-----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)

# 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 proyectó 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 ('Digite 2 para borrar la pantalla LCD') # En esta línea se imprime un mensaje
                                                # que indica al usuario quede 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 ('Opcion incorrecta') # En el caso que se digite un número distinto a 2 se imprime
                                # el siguiente mensaje
```

Fig 2. Código Fuente

## LIBRERIAS

- *import board*

Esta librería 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*
- *lcd\_rows = 2*

Estas líneas de código definen el tamaño de caracteres de la 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)	pin4	pin15
lcd_en = digitalio.DigitalInOut(board.D17)	pin6	pin11
lcd_d4 = digitalio.DigitalInOut(board.D25)	pin 11	pin22
lcd_d5 = digitalio.DigitalInOut(board.D24)	pin12	pin18
lcd_d6 = digitalio.DigitalInOut(board.D23)	pin 13	pin16
lcd_d7 = digitalio.DigitalInOut(board.D18)	pin 14	pin12



## INICIALIZAMOS LA CLASE LCD

- `lcd = characterlcd.CharacterLCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows)`
- 15

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')`

En esta línea se imprime un mensaje que indique que el texto ya se proyectó en el LCD

- `lcd.message = "HERRAMIENTAS DESOFTWARE 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 ""

- `print('Digite 2 para borrar la pantalla LCD')`

En esta línea se imprime un mensaje que indica al usuario quede 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`. "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.  
Conectar los pines de la LCD 16x2 a los pines correspondientes de la Raspberry pi 3, mediante cables macho-hembra, teniendo en cuenta lo siguiente:

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	V <sub>0</sub> /V <sub>E</sub>	PIN 3
GND	PROTO -		
5V	PROTO +		

Fig 2.Distribución de pines en la Raspberry pi-LCD

Finalmente al abrir el código de programación de Python, la interfaz identifica tanto los pines de entrada y de salida de la Raspberry Pi como los pines de la LCD 16x2, al momento de ejecutar el programar podemos observar que imprime el mensaje en la pantalla LCD 16x2.

## VII. RESULTADOS

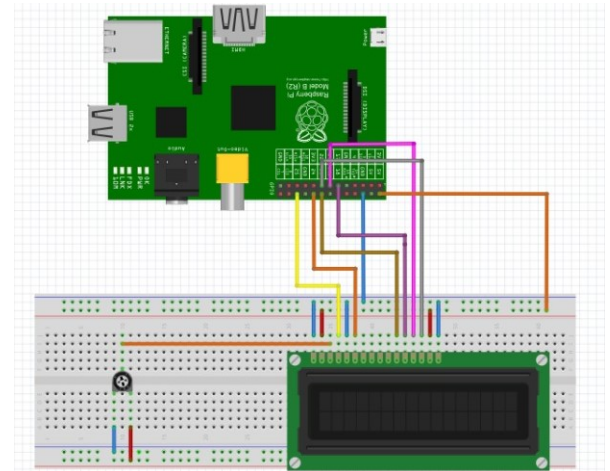


Fig 7.Diagrama electrónico



Fig 7.Mensaje impreso

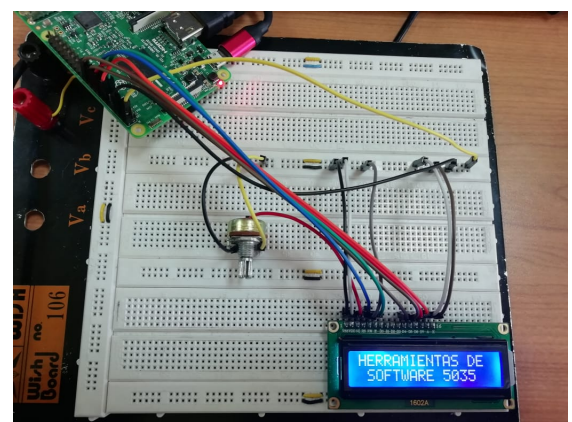


Fig 7.Conexión de la pantalla LCD 16x2-Raspberry Pi

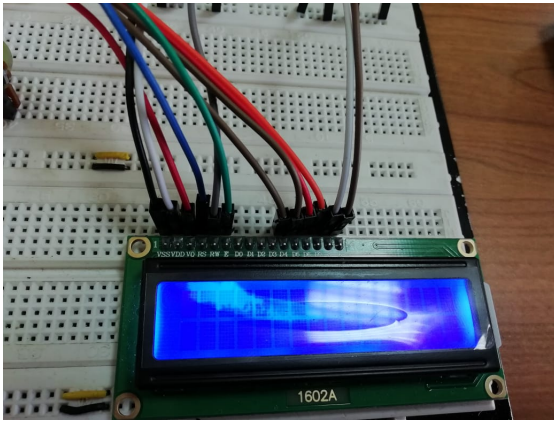


Fig 7. Pantalla LCD 16x2 activa

## VIII. 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.

## REFERENCIAS

- [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 / iccmc.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
- [3] Sharma, M., Agarwal, N. y Reddy, SRN (2015). Diseño y desarrollo de placa hija para comunicación USB-UART entre Raspberry Pi y PC. Conferencia internacional sobre computación, comunicación y automatización. doi: 10.1109 / ccaa.2015.7148532
- [4] Palacios, F. (2016). Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. ISBN-13: 978-0-13-235613-8
- [5] Montes, H., Pacheco, A., Ramos, H. (2017). Monitoreo del Consumo de Energía Eléctrica Doméstica con Arduino. 15th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Global Partnerships for Development and Engineering Education", 19-21 July 2017, Boca Raton FL, United States.
- [6] Tutorials Teacher. (2018). Python-IDLE. Blog de Tecnologías de la Información. Recuperado el 7 de mayo del 2019 de: <https://www.tutorialsteacher.com/python/python-idle>
- [7] Shead, S. (2012). Raspberry Pi delivery delays leave buyers hungry (and angry). Hardware. Recuperado el 8 de mayo del 2019 de: <https://www.zdnet.com/article/raspberry-pi-delivery-delays-leave-buyers-hungry-and-angry/>

- [8] Raspberry Shop. (2019). Hardware y Accesorios. Raspberry Pi 3 modelo B. Recuperado el 8 de mayo del 2019 de: <https://www.raspberrishop.es/raspberry-pi-3.php>