

AZ-Delivery

Bienvenido.

Gracias por elegir nuestra pantalla LCD de AZ-Delivery. Estos están disponibles individualmente o en paquete con convertidor I2C en verde o azul, y en dos tamaños: 16x02 y 20x04. En las siguientes páginas le explicaremos cómo configurar y utilizar el dispositivo.

¡Que te diviertas!



Áreas de aplicación

Educación y enseñanza: uso en escuelas, universidades e instituciones de formación para enseñar los conceptos básicos de electrónica, programación y sistemas integrados. Investigación y desarrollo: Uso en proyectos de investigación y desarrollo para crear prototipos y experimentos en los campos de la electrónica y la informática. Desarrollo de prototipos: Uso en el desarrollo y prueba de nuevos circuitos y dispositivos electrónicos. Proyectos Hobby and Maker: utilizado por entusiastas y aficionados a la electrónica para desarrollar e implementar proyectos de bricolaje.

Conocimientos y habilidades requeridos.

Conocimientos básicos de electrónica e ingeniería eléctrica. Conocimientos de programación, especialmente en el lenguaje de programación C/C++. Capacidad para leer esquemas y diseñar circuitos simples. Experiencia trabajando con componentes electrónicos y soldadura.

Condiciones de operación

El producto sólo puede funcionar con los voltajes especificados en la hoja de datos para evitar daños. Se requiere una fuente de alimentación CC estabilizada para su funcionamiento. Al realizar la conexión a otros componentes y circuitos electrónicos, se deben observar los límites máximos de corriente y voltaje para evitar sobrecargas y daños.

Condiciones ambientales

El producto debe utilizarse en un ambiente limpio y seco para evitar daños causados por la humedad o el polvo. Proteja el producto de la luz solar directa (UV), ya que esto puede afectar negativamente la vida útil de la pantalla.

Uso previsto

El producto está diseñado para su uso en entornos educativos, de investigación y desarrollo. Se utiliza para desarrollar, programar y crear prototipos de proyectos y aplicaciones electrónicos. El producto no pretende ser un producto de consumo terminado, sino más bien una herramienta para usuarios con conocimientos técnicos, incluidos ingenieros, desarrolladores, investigadores y estudiantes.

Uso inadecuado previsible

El producto no es adecuado para uso industrial o aplicaciones relevantes para la seguridad. No se permite el uso del producto en dispositivos médicos o para fines de aviación y viajes espaciales.

desecho

¡No lo deseches con la basura doméstica! Su producto es acorde al europeo. Directiva sobre residuos de aparatos eléctricos y electrónicos que deben eliminarse de forma respetuosa con el medio ambiente. Las valiosas materias primas contenidas en ellos se pueden reciclar. La aplicación de esta directiva contribuye a la protección del medio ambiente y la salud. Utilice el punto de recogida habilitado por su municipio para devolver y Reciclaje de aparatos eléctricos y electrónicos viejos. N.º registro RAEE: DE 62624346

descarga electrostática

La pantalla es sensible a las descargas electrostáticas (ESD), que pueden dañar o destruir los componentes electrónicos. Tenga en cuenta las siguientes instrucciones de seguridad para evitar riesgos de ESD: Atención: Las cargas electrostáticas en su cuerpo pueden dañar la pantalla. Nota: Conéctese a tierra usando una muñequera antiestática conectada a una superficie con conexión a tierra o tocando una superficie metálica con conexión a tierra antes de manipular la pantalla. Atención: Utilice bolsas y tapetes antiestáticos para proteger la pantalla. Nota: Coloque la pantalla sobre una alfombra de trabajo antiestática y guárdela en bolsas antiestáticas cuando no esté en uso. Nota: Un lugar de trabajo limpio y conectado a tierra minimiza el riesgo de ESD. Acción: Mantenga su lugar de trabajo limpio y libre de materiales que puedan generar cargas electrostáticas. Asegúrese de que todas las superficies utilizadas estén conectadas a tierra.

instrucciones de seguridad

Aunque la pantalla cumple con los requisitos de la Directiva RoHS (2011/65/UE) y no contiene sustancias peligrosas en cantidades superiores a los límites permitidos, aún pueden existir riesgos químicos residuales. Tenga en cuenta las siguientes instrucciones de seguridad: Atención: La parte posterior de la pantalla y la placa de circuito pueden liberar residuos químicos de fabricación o durante el funcionamiento. Nota: Use guantes protectores cuando manipule o instale la pantalla durante mucho tiempo para evitar irritación de la piel. Precaución: Los componentes electrónicos pueden emitir pequeñas cantidades de compuestos orgánicos volátiles (COV), especialmente si la pantalla es nueva. Nota: asegúrese de trabajar en un área

bien ventilada para minimizar la concentración de vapores en el aire. Precaución: No utilice productos químicos ni disolventes agresivos para limpiar la pantalla, ya que pueden dañar la capa protectora o los componentes electrónicos. Nota: Utilice un paño de limpieza antiestático o un limpiador especial para componentes electrónicos para limpiar cuidadosamente la pantalla. Aunque la pantalla cumple con los requisitos de la Directiva RoHS (2011/65/UE) y no contiene sustancias peligrosas en cantidades superiores a los límites permitidos, aún pueden existir riesgos químicos residuales. Tenga en cuenta las siguientes instrucciones de seguridad: Atención: La parte posterior de la pantalla y la placa de circuito pueden liberar residuos químicos de fabricación o durante el funcionamiento. Nota: Use guantes protectores cuando manipule o instale la pantalla durante mucho tiempo para evitar irritación de la piel. Precaución: Los componentes electrónicos pueden emitir pequeñas cantidades de compuestos orgánicos volátiles (COV), especialmente si la pantalla es nueva. Nota: asegúrese de trabajar en un área bien ventilada para minimizar la concentración de vapores en el aire. Precaución: No utilice productos químicos ni disolventes agresivos para limpiar la pantalla, ya que pueden dañar la capa protectora o los componentes electrónicos. Nota: Utilice un paño de limpieza antiestático o un limpiador especial para componentes electrónicos para limpiar cuidadosamente la pantalla. La pantalla contiene componentes electrónicos sensibles y una capa superior. Un manejo inadecuado o una presión excesiva pueden causar daños a la pantalla o lesiones. Observe las siguientes instrucciones de seguridad para evitar riesgos mecánicos: Atención: La tapa del display es frágil y puede romperse si se manipula incorrectamente. Nota: Evite aplicar una presión fuerte o doblar la pantalla. Manipule la pantalla con cuidado y sólo por la placa de circuito para evitar roturas. Precaución: Las caídas o los impactos pueden agrietar la superficie de la pantalla y dañar los componentes electrónicos de la parte posterior. Nota: Evite dejar caer la pantalla y protéjala de impactos. Utilice una superficie suave cuando trabaje para evitar rayones. Atención: Si la pantalla se rompe, los trozos de vidrio afilados pueden provocar lesiones. Nota: Si la pantalla se rompe, manipule los fragmentos con cuidado y use guantes protectores para evitar cortes. Deseche las piezas de vidrio de forma segura. Nota: Una fijación incorrecta puede provocar tensión mecánica y rotura de la pantalla. Acción: Coloque la pantalla de forma segura y sin presión excesiva. Utilice soportes o carcasas adecuados para montar la pantalla de forma estable. Precaución: Los métodos de limpieza inadecuados pueden rayar o dañar la superficie. Nota: Utilice únicamente paños suaves y antiestáticos para limpiar la pantalla. Evite agentes de limpieza agresivos y fricciones fuertes. La pantalla funciona con voltajes y corrientes eléctricas que, si se usan incorrectamente, pueden provocar descargas eléctricas, cortocircuitos o incendios. Tenga en cuenta las siguientes instrucciones de seguridad: Atención: Utilice el producto únicamente con los voltajes especificados. Nota: Los límites de rendimiento del producto se pueden encontrar en la hoja de datos asociada. Nota: Las fuentes de voltaje inadecuadas pueden dañar la pantalla o provocar situaciones peligrosas. Acción: Utilice únicamente fuentes de alimentación o baterías probadas y adecuadas para alimentar sus circuitos. Asegúrese de que la fuente de voltaje cumpla con los requisitos de la pantalla. Precaución: Evite cortocircuitos entre los conectores y componentes del producto. Nota: Asegúrese de que ningún objeto conductor toque o puentee la placa de circuito. Utilice herramientas aisladas y preste atención a la disposición de las conexiones. Precaución: No realice ningún trabajo en el producto cuando esté conectado a una fuente de alimentación. Nota: Desconecte el producto de la alimentación antes de realizar cambios en el circuito o conectar o quitar componentes. Nota: busque señales de daños eléctricos, como humo, olores inusuales o decoloración. Acción: Si ocurren tales signos, apague la alimentación inmediatamente e inspeccione minuciosamente el circuito para detectar errores. La pantalla puede generar calor durante el funcionamiento, lo que podría provocar sobrecalentamiento, quemaduras o incendios si se manipula incorrectamente. Tenga en cuenta las siguientes instrucciones de seguridad: Atención: Algunos componentes de la pantalla pueden calentarse durante el funcionamiento o en caso de error. Medida: Después de apagarla, deje que la pantalla se enfríe lo suficiente antes de tocar directamente los componentes individuales de la parte posterior. Evite el contacto directo con componentes calientes. Precaución: La sobrecarga puede provocar un calentamiento excesivo de los componentes electrónicos. Nota: Asegúrese de que la fuente de alimentación y voltaje cumpla con las especificaciones de la pantalla y no cause sobrecarga.

Índice

Introducción.....	3
Datos técnicos.....	6
Pinout.....	8
Cómo configurar el IDE de Arduino	10
Cómo configurar la Raspberry Pi y Python	15
Conexión del módulo con el microcontrolador	16
Ejemplo de croquis.....	18
Conexión del módulo con el microcontrolador mediante adaptador I2C.....	19
Biblioteca Arduino IDE	20
Ejemplo de croquis	21
Conexión de la pantalla a la Raspberry Pi	24
Python-Skript.....	26
Conexión de la pantalla a la Raspberry Pi mediante un adaptador I2C.....	34
Bibliotecas y herramientas para Python	36
Python-Skript.....	38



Introducción

Una pantalla de cristal líquido, o LCD, es un dispositivo que utiliza cristales líquidos para mostrar caracteres visuales.

Los cristales líquidos no emiten luz por sí mismos. Las pantallas LCD utilizan una retroiluminación y cristales líquidos para bloquear la luz. Cuando no circula corriente por los cristales líquidos, éstos se encuentran en un estado caótico. La luz de la retroiluminación atraviesa los cristales líquidos sin esfuerzo. Cuando la corriente fluye a través de los cristales líquidos, éstos se organizan en un estado uniforme. Esto crea una sombra en la pantalla que crea objetos.

En pocas palabras, los LCD utilizan cristales líquidos para pasar o bloquear la luz procedente de la retroiluminación. Así se crean los píxeles. Se combinan para mostrar cualquier objeto 2D visible en la pantalla. Cada área de píxeles puede encenderse y apagarse suministrando electricidad a través de un chip controlador integrado.

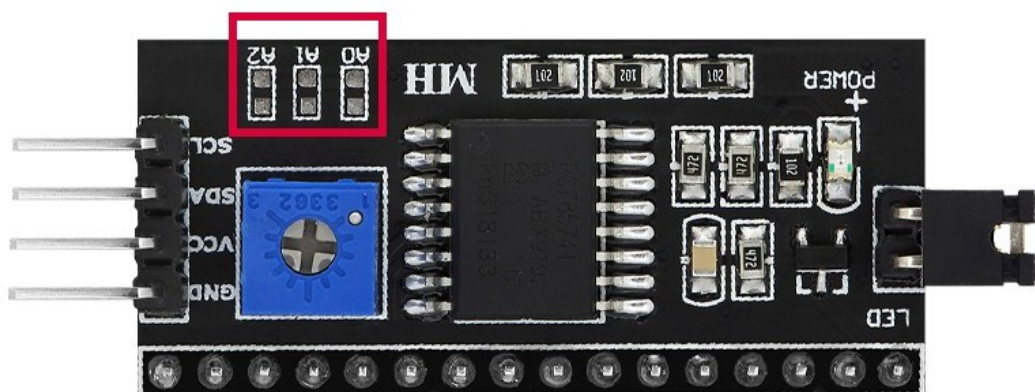
Cada segmento de la pantalla de cristal líquido representa un píxel y debe controlarse por separado. Por esta razón, se requiere un circuito integrado especial llamado chip controlador. El LCD 16x02(20x04) tiene un chip controlador llamado "HD44780". Para controlar la pantalla, el microcontrolador debe comunicarse con el chip controlador. El chip controlador utiliza un tipo de interfaz SPI para comunicarse con el microcontrolador.

El adaptador I2C

El adaptador I2C es un dispositivo que simplifica la conexión entre las pantallas LCD y los microcontroladores. Utiliza la interfaz I2C para comunicarse con el microcontrolador. Se pueden conectar varios adaptadores a la misma interfaz I2C. El adaptador es compatible con pantallas LCD, con chips HD44780 integrados. El adaptador I2C es compatible tanto con paneles LCD 16x02 como con paneles LCD 20x04 ofrecidos por AZ-Delivery.

El adaptador I2C viene con una dirección I2C predefinida, que es 0x27. Sin embargo, se puede cambiar soldando los pads etiquetados A0, A1 y A2 en el adaptador.

SOLDERING PADS (A0-A1)





En la siguiente t a b l a se muestra cómo configurar una dirección I2C específica del adaptador:

PADS			DIRECCIÓN I2C
A2	A1	A0	
C	C	C	0x20
C	C	O	0x21
C	O	C	0x22
C	O	O	0x23
O	C	C	0x24
O	C	C	0x25
O	O	C	0x26
O	O	O	0x27
O - Abierto			C - Cerrado



Datos técnicos

" Tensión de funcionamiento: de	3,3 V a 5 V
" Área de visualización:	12 x 56 mm
" Tipo LCD:	STN, positivo, transreflectivo, verde/azul
" Retroiluminación:	ED, blanco
" Punto de vista:	180°
" Modos:	paralelo (8 bits y 4 bits)
" Temperatura de funcionamiento:	-10 °C a 60 °C
"Dimensiones:	36x 80 x 12,5 mm [1,4 x 3,1 x 0,5 pulgadas].
" Interfaz:	I2C/paralelo
" Dirección I2C:	0x20 - 0x27
" Ajuste del contraste :	Potenciómetro
" Ajuste del fondo.:	Puente

La tensión de la retroiluminación es de 5 V CC. Para las funciones lógicas el consumo de corriente en funcionamiento es de 1,5mA y para la retroiluminación de 30mA.

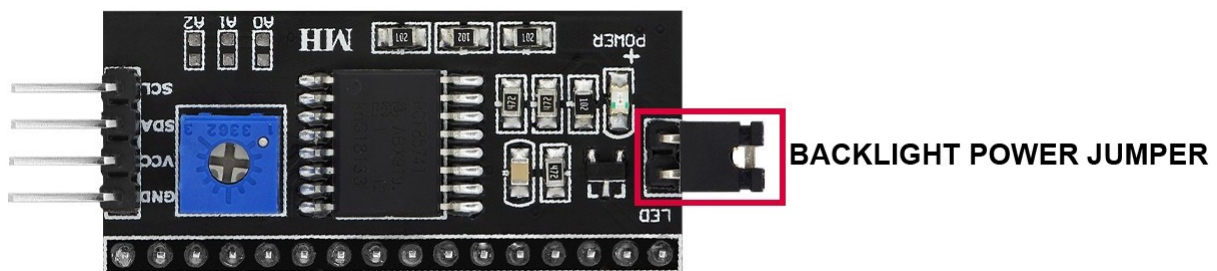
La resolución de la pantalla es 16x02, lo que significa que los caracteres pueden mostrarse en 2 líneas con 16 caracteres por línea. Cada carácter consta de 5x7 píxeles.

Para ajustar el contraste de la pantalla, el adaptador I2C tiene un potenciómetro incorporado. Por lo tanto, se necesita un pequeño

Az-Delivery

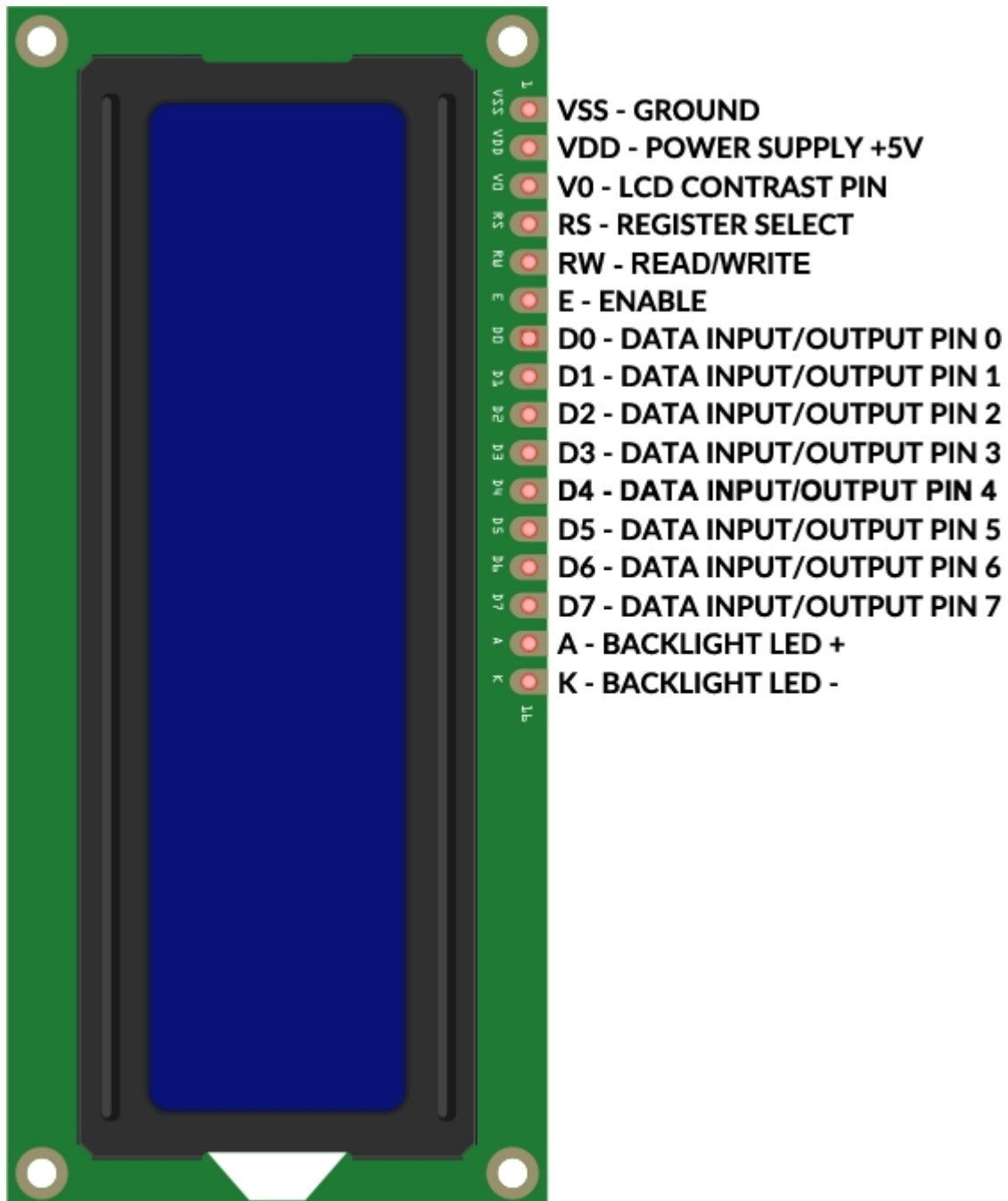
destornillador para el ajuste.

Para controlar la retroiluminación de la pantalla LCD, el adaptador I2C tiene un puente de alimentación para la retroiluminación. El puente se utiliza para encender/apagar la luz de fondo. Cuando el puente está conectado, se utiliza para conectar la fuente de alimentación para la luz de fondo. Cuando el jumper está desconectado, la fuente de alimentación de la retroiluminación está desconectada.



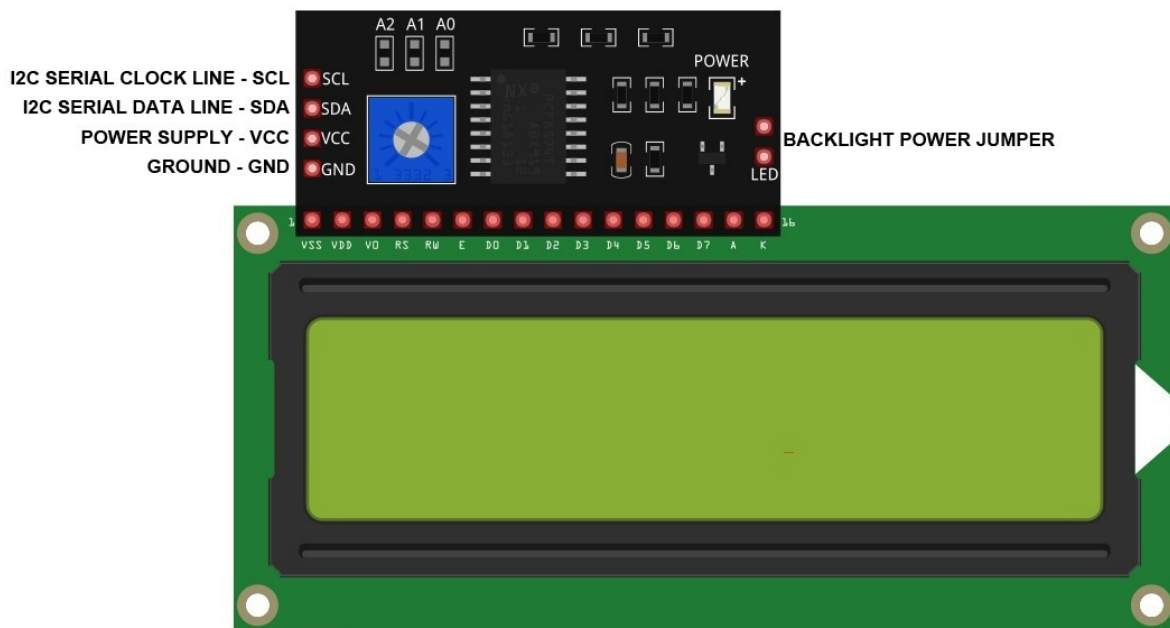
Pinout sin adaptador I2C

La pantalla LCD 16x02 tiene 16 pines. La asignación de pines es la siguiente (también se aplica al 20x04):



Pinout con adaptador I2C

El LCD 16x02 tiene 16 pines y el adaptador I2C tiene 20. El adaptador se conecta al LCD de la siguiente manera (esto también se aplica al 20x04):



Nota: Es necesario conectar el adaptador I2C y la pantalla LCD exactamente como se muestra arriba. Si se conectan de forma diferente, los dispositivos pueden dañarse.



Nota para la Raspberry Pi: El voltaje del nivel lógico TTL de los pines de E/S es de 5V. Para utilizar el panel LCD y el adaptador I2C con la Raspberry Pi, es necesario utilizar un conversor de nivel lógico. De lo contrario, la alimentación de la señal a través de los pines de E/S del módulo a los pines GPIO de la Raspberry Pi puede causar daños. Por lo tanto, utilice el [TXS0108E 8ch Logic Level Converter](#). ofrecido por AZ-Delivery.

Cómo configurar el IDE de Arduino

Si el IDE de Arduino no está instalado, siga el [enlace](#) y descargue el archivo de instalación para el sistema operativo de su elección.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: **ARDUINO 1.8.9**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with the following options: "Windows Installer, for Windows XP and up" and "Windows ZIP file for non admin install". Below these, it says "Windows app Requires Win 8.1 or 10" with a "Get" button. Further down, it lists "Mac OS X 10.8 Mountain Lion or newer". At the bottom, it lists "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", and "Linux ARM 64 bits". At the very bottom of the sidebar, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Para usuarios de Windows: Haga doble clic en el archivo .exe descargado y siga las instrucciones de la ventana de instalación.

Az-Delivery

Para los usuarios de Linux, descargue un archivo con la extensión .tar.xz, que deberá extraer. Una vez extraído, hay que ir al directorio extraído y abrir el terminal en ese directorio. Hay que ejecutar dos scripts .sh, el primero llamado **arduino-linux-setup.sh** y el segundo llamado install.sh.

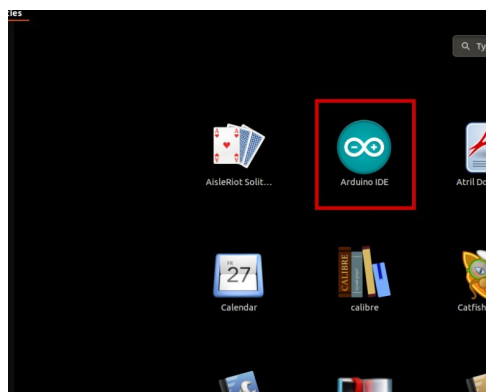
Para ejecutar el primer script en el terminal, abra el terminal en la carpeta extraída y ejecute el siguiente comando:

sh arduino-linux-setup.sh nombre_usuario

nombre_usuario - es el nombre de un superusuario en el sistema operativo Linux. Debe introducirse una contraseña para el superusuario al iniciar el comando. Espere unos minutos a que se complete el script.

El segundo script, denominado install.sh script, debe utilizarse una vez instalado el primero. Ejecute el siguiente comando en el terminal (directorio extraído): **sh install.sh**

Después de instalar estos scripts, vaya a Todas las aplicaciones donde está instalado el IDE Arduino.



Az-Delivery

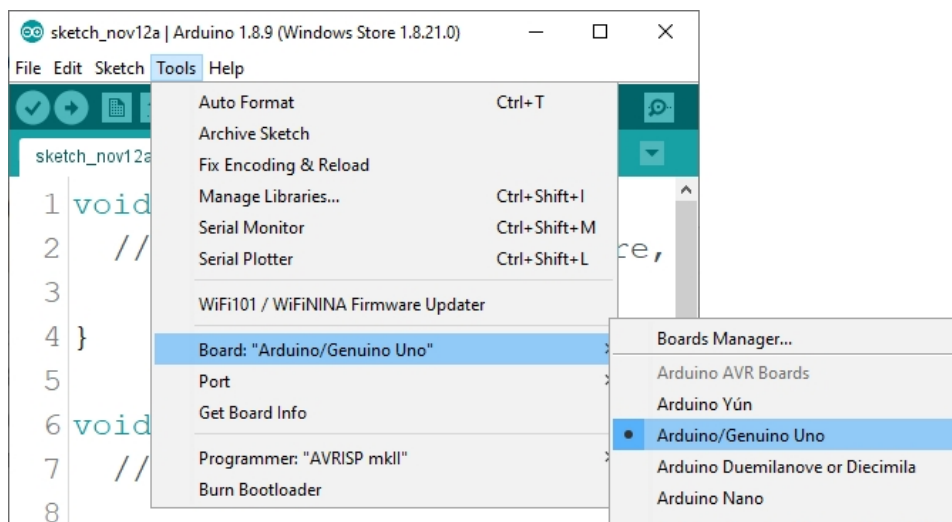
Casi todos los sistemas operativos vienen con un editor de texto preinstalado (por ejemplo, Windows con Notepad, Linux Ubuntu con Gedit, Linux Raspbian con Leafpad, etc.). Todos estos editores de texto son perfectamente adecuados para el propósito del eBook.

Lo primero que debe comprobar es si su PC puede reconocer un microcontrolador

placa. Abra el IDE Arduino recién instalado, y vaya a:

Herramientas > Placa > {nombre de tu placa aquí}

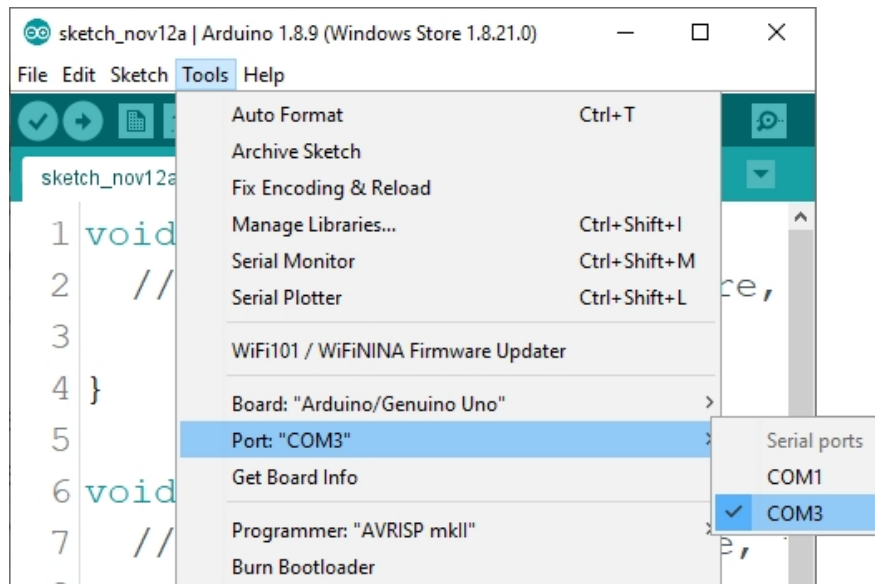
{el nombre de tu placa aquí} debería ser el Arduino/Genuino Uno, como se puede ver en la siguiente imagen:



Se debe seleccionar el puerto donde está conectada la placa microcontroladora. Vaya a: Herramientas > Puerto > {el nombre del puerto va aquí} y si la placa microcontroladora está conectada al puerto USB, el nombre del puerto se puede ver en el menú desplegable de la imagen anterior.

Az-Delivery

Si se utiliza el IDE Arduino en Windows, los nombres de los puertos son los siguientes:



Para los usuarios de Linux, por ejemplo, el nombre del puerto es /dev/ttyUSBx, donde x es un número entero entre 0 y 9.



Cómo configurar la Raspberry Pi y Python

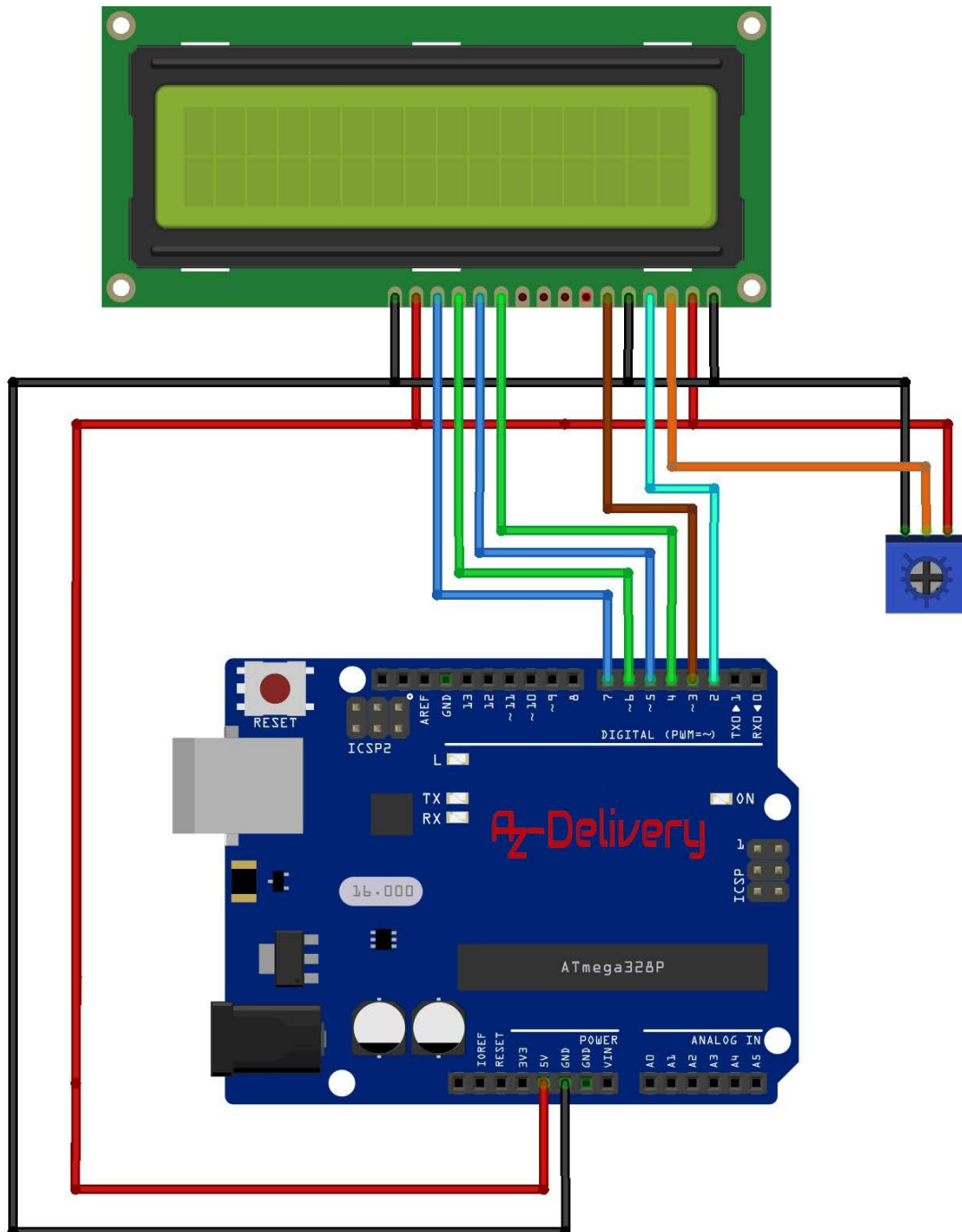
En el caso de la Raspberry Pi, primero hay que instalar el sistema operativo y, a continuación, configurar todo para que pueda utilizarse en modo headless. El modo Headless le permite conectarse a la Raspberry Pi de forma remota sin necesidad de una pantalla de PC, ratón o teclado. Las únicas cosas que se utilizan en este modo son la propia Raspberry Pi, la fuente de alimentación y la conexión a Internet. Todo esto se explica en detalle en el eBook gratuito:

[Guía de inicio rápido de Raspberry Pi](#)

El sistema operativo Raspbian viene con Python preinstalado.

Conexión del módulo con el microcontrolador

Conecte la pantalla al microcontrolador como se muestra a continuación (se aplica también al 20x04):



fritzing

Clavija LCD	Clavija MC	Color del dibujo
VSS	GND	Schwarzer Draht
VDD	5V	Roter Draht
V0	Poti	Oranger Draht
RS	D2	Cyan Draht
RW	GND	Schwarzer Draht
E	D3	Brauner Draht
D4	D4	Grüner Draht
D5	D5	Blauer Draht
D6	D6	Grüner Draht
D7	D7	Blauer Draht
K	GND	Schwarzer Draht
A	5V	Roter Draht



Ejemplo sketch

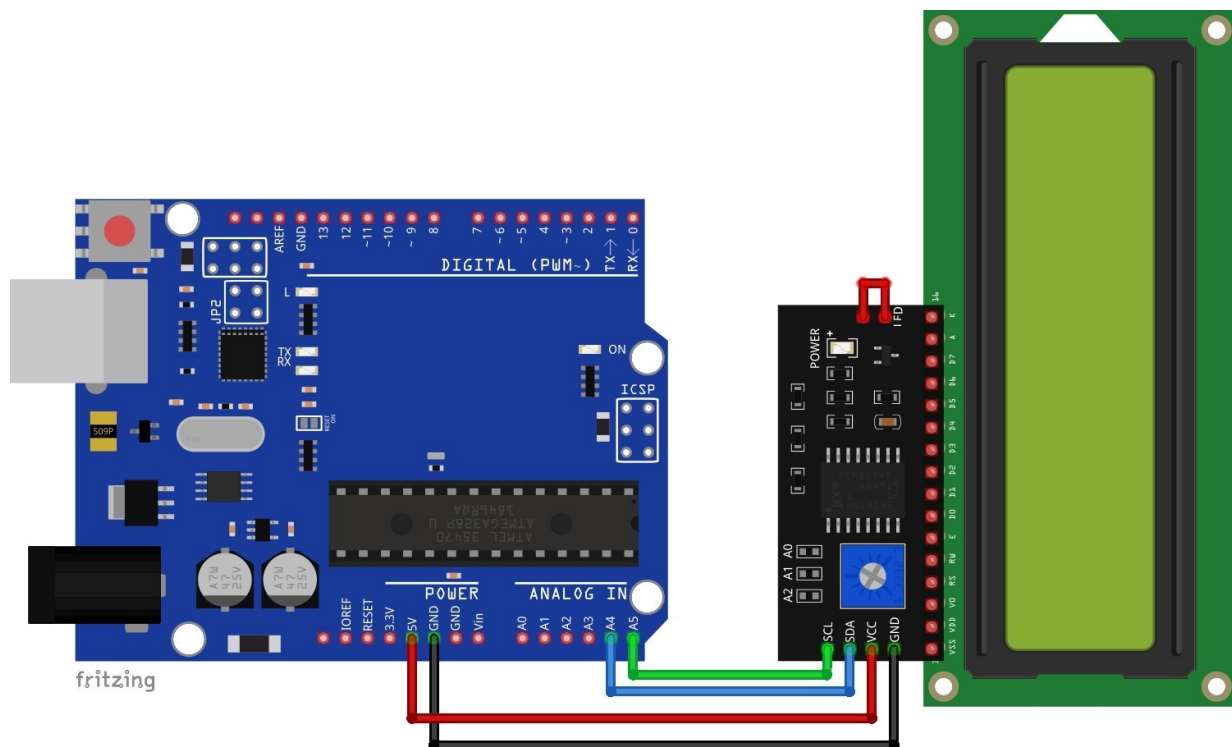
El siguiente ejemplo de sketch es un sketch modificado del IDE de Arduino:

Archivo > Ejemplos > LiquidCrystal > HelloWorld

```
#include <LiquidCrystal.h>
const uint8_t rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
  lcd.begin(16, 2); // cambiar a "20, 4" si se utiliza 20x04
  lcd.clear();
}
void loop() {
  lcd.setCursor(0, 0);
  lcd.print("AZ-Delivery");
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}
```

Conexión del módulo al microcontrolador con adaptador I2C

Conecte la pantalla 16x02 y el adaptador I2C al microcontrolador como se muestra a continuación (se aplica también a la 20x04):



Pin adaptador I2C	Clavija MC	Color
SCL	A5	cable verde
SDA	A4	cable azul
VCC	5V	cable rojo
GND	GND	cable negro



Biblioteca para Arduino IDE

Para utilizar el módulo con el IDE de Arduino, se recomienda descargar una librería externa para el mismo. La librería utilizada en este eBook se llama LiquidCrystal_I2C. Para descargarla, haga clic en este [enlace](#) y descargue el archivo .zip.

Para incluir la biblioteca, vaya al IDE de Arduino y vaya a:

Sketch > Incluir biblioteca > Añadir biblioteca .ZIP

y cuando se abra una nueva ventana, localice y seleccione el archivo .zip descargado.



Ejemplo sketch

El siguiente ejemplo de sketch es un sketch modificado del IDE de Arduino:

Archivo > Ejemplos > LiquidCrystal > HelloWorld

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
//cambiar a "20, 4" si se utiliza
20x04 void setup() {
  lcd.init();
  lcd.backlight();
  delay(250);
  lcd.noBacklight();
  delay(1000);
  lcd.backlight();
  delay(1000);
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print("AZ-Delivery");
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
  delay(100);
}
```


Az-Delivery

El sitio boceto comienza con el sitio inclusión de
bibliotecas denominadas Wire y
LiquidCrystal_I2C.

A continuación, se crea un objeto llamado lcd. El objeto representa la pantalla en sí, y para crear este objeto, se utiliza la siguiente línea de código:
`LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);`

Donde 0x27 es la dirección I2C del adaptador I2C. 16 es el número de caracteres por línea y 2 es el número de líneas.

En la función setup(), el objeto lcd se inicializa con la siguiente línea de código : `lcd.init();`

Al final de la función setup(), se prueba la retroiluminación apagándola y encendiéndola con un retardo de 1000ms (1s).

En la función loop() se utilizan dos funciones predefinidas de la librería LiquidCrystal_I2C.

AZ-Delivery

a primera función se llama `setCursor()`. La función tiene dos argumentos y no devuelve ningún valor. Los valores de los argumentos son enteros. El primer número representa la posición Y del cursor, con valores en el rango de 0 a 1, donde 0 representa la primera línea y 1 representa la segunda línea de la pantalla. El segundo argumento representa la posición X del cursor, con valores comprendidos entre 0 y 15, donde 0 refleja la primera columna y 15 la última columna de la pantalla.

La función debe utilizarse antes de la función `print()`. Para indicar a la función `print()` dónde mostrar el texto. Si no utiliza la función `setCursor()`, la función `print()` mostrará el texto en la posición (0, 0).

La función `print()` tiene un argumento y no devuelve ningún valor. El argumento representa el texto, un valor de cadena, que se mostrará en la pantalla.

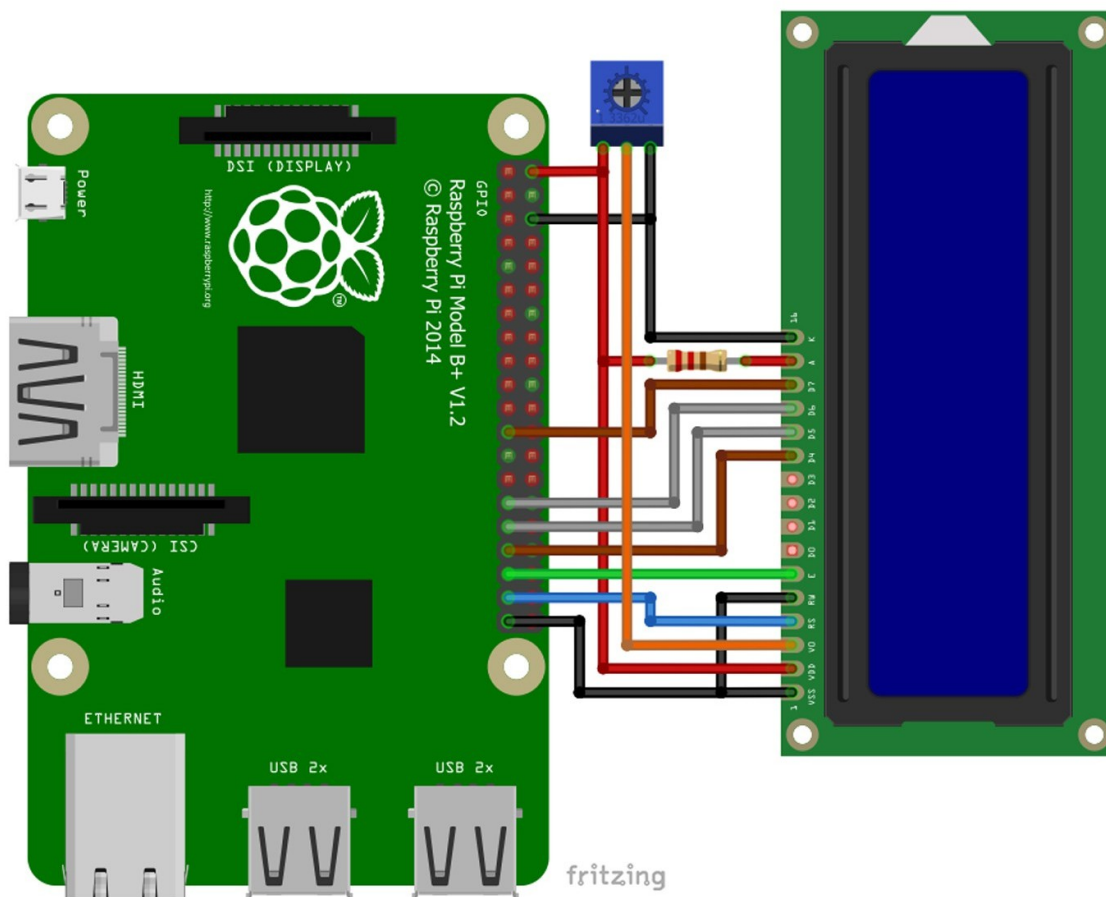
En la función `loop()`, primero se coloca el cursor en la primera línea y luego la función `print()` muestra el mensaje AZ-Delivery. A continuación, el cursor se coloca en la segunda línea y se muestra el número de segundos transcurridos desde que el microcontrolador se encendió o reinició por última vez.

NOTA: Si su pantalla no muestra nada, tiene que girar el potenciómetro del adaptador I2C.

Conexión de la pantalla a la Raspberry Pi

Conecte el módulo a la Raspberry Pi como se muestra a continuación (se aplica también al 20x04):

16X02 Blue LCD Screen
Connection diagram with Raspberry Pi



Az-Delivery

Pantalla Pin	Pin de Raspberry Pi	Pin físico	color del cable
VSS	GND	6	cable negro
VDD	5V	2	cable rojo
RS	GPIO26	37	cable azul
RW	GND	39	cable negro
E	GPIO19	35	cable verde
D4	GPIO13	33	cable marrón
D5	GPIO6	31	cable gris
D6	GPIO5	29	cable gris
D7	GPIO11	23	cable marrón
K	GND	6	cable rojo
A	5V, mediante resistencia de 220Ω	2	cable naranja
V0	clavija central del potenciómetro		

Potenciómetro			
GND	derecha Pin	6	cable negro
5V	izquierda Pin	2	cable rojo



Python- Skript

Se crean dos scripts, uno para todas las funciones y otro para utilizar estas funciones. A continuación encontrará el código para el primer script:

```
import RPi.GPIO as GPIO
import tiempo

LCD_RS = 26
LCD_E  = 19
LCD_D4 = 13
LCD_D5 = 6
LCD_D6 = 5
LCD_D7 = 11

# Define algunas constantes del dispositivo
LCD_WIDTH = 16#    Caracteres máximos por línea, cambiar a "20 "si
20x04
LCD_CHR = Verdadero
LCD_CMD = Falso

LCD_LINE_1 = 0x80 # Dirección LCD RAM para la 1ª línea
LCD_LINE_2 = 0xC0 # Dirección LCD RAM para la 2ª línea

# Constantes de
tiempo E_PULSE =
0.0005
E_RETRASO = 0,0005
```

Az-Delivery

```
# una
# pestaña
def lcd_init(RS, E, D4, D5, D6, D7):
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    global LCD_RS, LCD_E, LCD_D4, LCD_D5, LCD_D6, LCD_D7
    LCD_RS = RS
    LCD_E = E
    LCD_D4 = D4
    LCD_D5 = D5
    LCD_D6 = D6
    LCD_D7 = D7
    GPIO.setup(LCD_E, GPIO.OUT) # E
    GPIO.setup(LCD_RS, GPIO.OUT) # RS
    GPIO.setup(LCD_D4, GPIO.OUT) # DB4
    GPIO.setup(LCD_D5, GPIO.OUT) # DB5
    GPIO.setup(LCD_D6, GPIO.OUT) # DB6
    GPIO.setup(LCD_D7, GPIO.OUT) # DB7
    # Inicializar pantalla
    lcd_byte(0x33, LCD_CMD) # 110011 Inicializar
    # 110010 Inicializar
    lcd_byte(0x32, LCD_CMD)
    # 000110 Dirección de movimiento del cursor
    lcd_byte(0x06, LCD_CMD)
    # 001100 Pantalla Encendida, Cursor Apagado,
    # Parpadeo Apagado lcd_byte(0x0C, LCD_CMD)
    # 101000 Longitud de datos, número de líneas, tamaño de
    # fuente lcd_byte(0x28, LCD_CMD)
    # 000001 Borrar pantalla
    lcd_byte(0x01, LCD_CMD)
    time.sleep(E_DELAY)
```

Az-Delivery

```
# una
# pestaña
def lcd_byte(bits, modo):
    GPIO.output(LCD_RS, mode) # RS
    GPIO.output(LCD_D4, False)
    GPIO.output(LCD_D5, False)
    GPIO.output(LCD_D6, False)
    GPIO.output(LCD_D7, False)
    if bits & 0x10 == 0x10:
        GPIO.output(LCD_D4, True)
    if bits & 0x20 == 0x20:
        GPIO.output(LCD_D5, True)
    if bits & 0x40 == 0x40:
        GPIO.output(LCD_D6, True)
    if bits & 0x80 == 0x80:
        GPIO.output(LCD_D7, True)
    lcd_toggle_enable() # Alternar pin 'Enable'
    GPIO.output(LCD_D4, False)
    GPIO.output(LCD_D5, False)
    GPIO.output(LCD_D6, False)
    GPIO.output(LCD_D7, False)
    if bits & 0x01 == 0x01:
        GPIO.output(LCD_D4, True)
    if bits & 0x02 == 0x02:
        GPIO.output(LCD_D5, True)
    if bits & 0x04 == 0x04:
        GPIO.output(LCD_D6, True)
    if bits & 0x08 == 0x08:
        GPIO.output(LCD_D7, True)
    # Alternar pin 'Enable'
    lcd_toggle_enable()
```

Az-Delivery

```
# una
pestaña
def lcd_toggle_enable():
    # Activar
    time.sleep(E_DELAY)
    GPIO.output(LCD_E, True)
    time.sleep(E_PULSE)
    GPIO.output(LCD_E, False)
    time.sleep(E_DELAY)

def lcd_string(mensaje, línea):
    # Enviar cadena a la
    pantalla LCD_LINE_1 = 0x80
    LCD_LINE_2 = 0xC0
    message = message.ljust(LCD_WIDTH, " ")
    if line == 0:
        lcd_byte(LCD_LINE_1, LCD_CMD)
    elif line == 1:
        lcd_byte(LCD_LINE_2, LCD_CMD)
    else:
        print('¡Esta lcd tiene dos líneas, línea 0 y línea 1!')
    for i in range(LCD_WIDTH):
        lcd_byte(ord(mensaje[i]), LCD_CHR)

def lcd_clear():
    lcd_byte(0x01, LCD_CMD)
```

Guarda el script con el nombre *lcd16x02.py*.

El código de la secuencia de comandos es un código modificado de la secuencia de comandos en el [enlace](#):

Az-Delivery

A continuación encontrará el código del script principal:

```
importar lcd16x02
from tiempo import sueño

LCD_RS = 26
LCD_E  = 19
LCD_D4 = 13
LCD_D5 = 6
LCD_D6 = 5
LCD_D7 = 11

# Inicializar pantalla
lcd16x02.lcd_init(LCD_RS, LCD_E, LCD_D4, LCD_D5, LCD_D6, LCD_D7)
i = 0

print('¡Pulsa CTRL + C para finalizar el
script!') try:
    lcd16x02.lcd_string('AZ-Entrega', 0)
    print('AZ-Entrega')
    print('Imprimiendo variable en el LCD...')
    while True:
        lcd16x02.lcd_string('{}'.format(i), 1)
        i+=1
        sleep(0.001) # 1 milisegundo de retraso

except KeyboardInterrupt:
    print('¡Fin del script!')

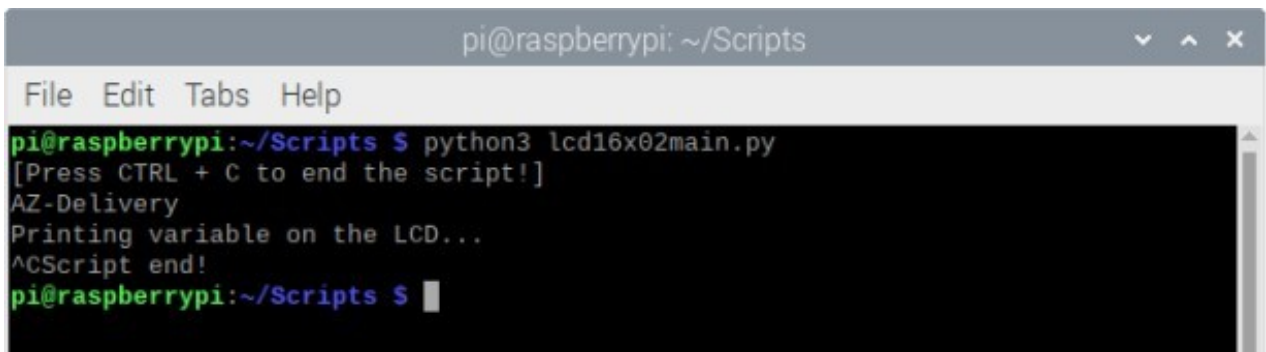
finally:
    lcd16x02.lcd_clear()
```

AZ-Delivery

Guarde el script con el nombre `lcd16x02main.py` en el mismo directorio que el script anterior. Para ejecutar el script, abra el terminal en el directorio donde está guardado el script y ejecute el siguiente comando:

```
python3 lcd16x02main.py
```

El resultado debería ser el siguiente:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 lcd16x02main.py
[Press CTRL + C to end the script!]
AZ-Delivery
Printing variable on the LCD...
^CScript end!
pi@raspberrypi:~/Scripts $
```

Para detener el script, pulse 'Ctrl + C' en el teclado.

AZ-Delivery

El primer script se utiliza para crear todas las funciones para controlar la pantalla LCD, que no se trata en este eBook. Sólo se explica la función principal del script.

El script principal comienza importando el primer script e importando la función sleep de la librería time.

A continuación, se definen seis variables para representar los pines de la pantalla que están conectados a los pines de la Raspberry Pi.

A continuación, la pantalla se inicializa con la siguiente línea de código: `lcd16x02.lcd_init(LCD_RS, LCD_E, LCD_D4, LCD_D5, LCD_D6, LCD_D7).`

A continuación se crea la variable "i" y se inicializa con el valor cero. Esto se utiliza para indicar datos cambiantes en la pantalla.

A continuación, se crea un bloque de código try-except-finally. En el bloque try, primero se muestra el mensaje AZ-Delivery en la primera línea de la pantalla y, a continuación, se crea un bloque de bucle indefinido (while True:). En él, la variable i se temporiza en la segunda línea de la pantalla y el valor de la variable i se incrementa en 1. Entre cada repetición del bloque de bucle indefinido hay una pausa de un milisegundo (sleep(0.001)).

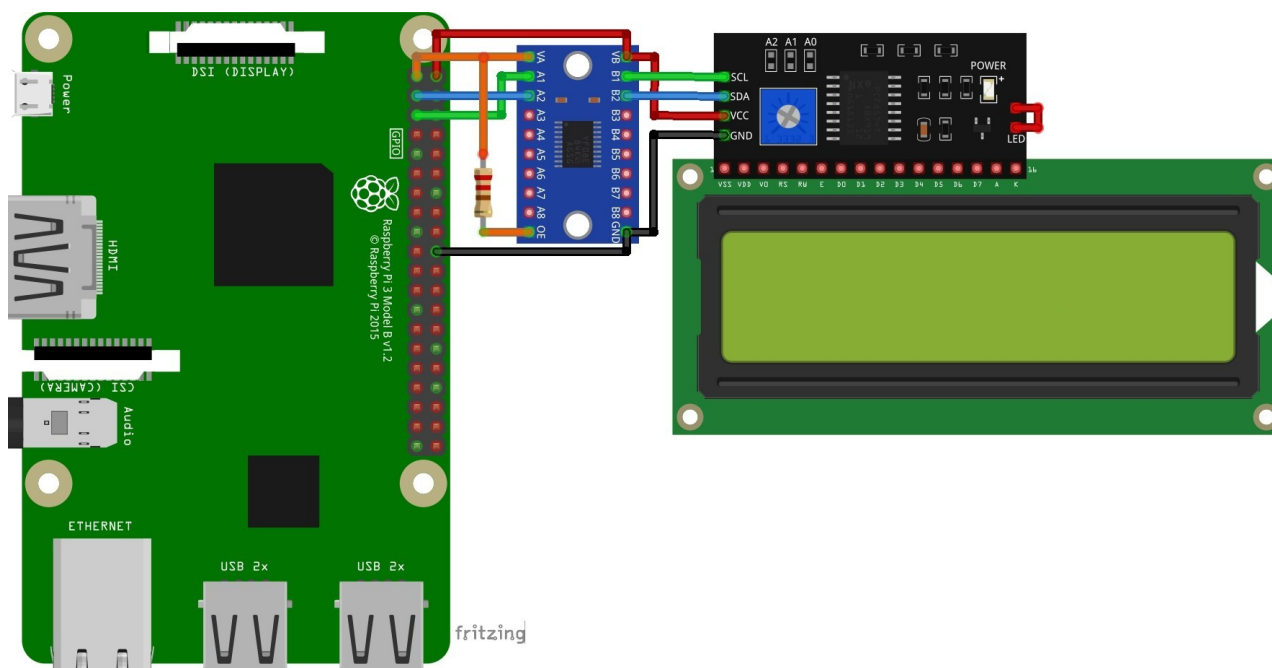
Az-Delivery

El bloque de código `except` se ejecuta con `Ctrl+C`. Esto se denomina `KeyboardInterrupt` Cuando se ha ejecutado este bloque de código, aparece en el terminal el mensaje ¡Fin de script!

El bloque de código `finally` se ejecuta después del script. Cuando se ha ejecutado el bloque de código `finally`, se llama a la función `lcd_clear()`, que borra el búfer de datos de la pantalla.

Conexión de la pantalla a la Raspberry Pi con el adaptador I2C

Conecte la pantalla 16x02 y el adaptador I2C a la Raspberry Pi como se muestra a continuación(se aplica lo mismo a 20x04):



aquí se debe utilizar el convertidor de nivel lógico, porque el adaptador I2C sólo funciona en el rango de 5V. El convertidor de nivel lógico utilizado en este eBook se llama *TXS0108E Logic Level Converter*.

Conecte el adaptador I2C al panel LCD como se muestra en el diagrama de conexión. Asegúrese de que el puente de retroiluminación integrado está conectado (**cable rojo**, el lado derecho del adaptador I2C en el diagrama de conexión).

Pin adaptador I2C	Pin LLC	Color
SCL	B1	cable verde
SDA	B2	cable azul
VCC	VB	cable rojo
GND	GND	cable negro

Pin LLC	Pin de Raspberry Pi	Clavija física	color
VA	3.3V	1	cable naranja
A1	GPIO3	5	cable verde
A2	GPIO2	3	cable azul
OE	3,3 V (mediante resistencia)	1	cable naranja
GND	GND	20	cable negro
VB	5V	2	cable rojo



Bibliotecas y herramientas para Python

Para utilizar la pantalla con la Raspberry Pi, se recomienda descargar una librería externa. Para descargarla, ve al siguiente [enlace](#) y descarga el script `lcd_class_i2c.py`. Guarda el script en el mismo directorio donde está guardado el script del siguiente capítulo.

El script `lcd_class_i2c.py` utiliza la librería `smbus` para Python. Si aún no está instalada, abre el terminal y ejecuta los siguientes comandos uno tras otro:

```
sudo apt-get update
```

```
sudo apt-get install python3-smbus python3-dev
```

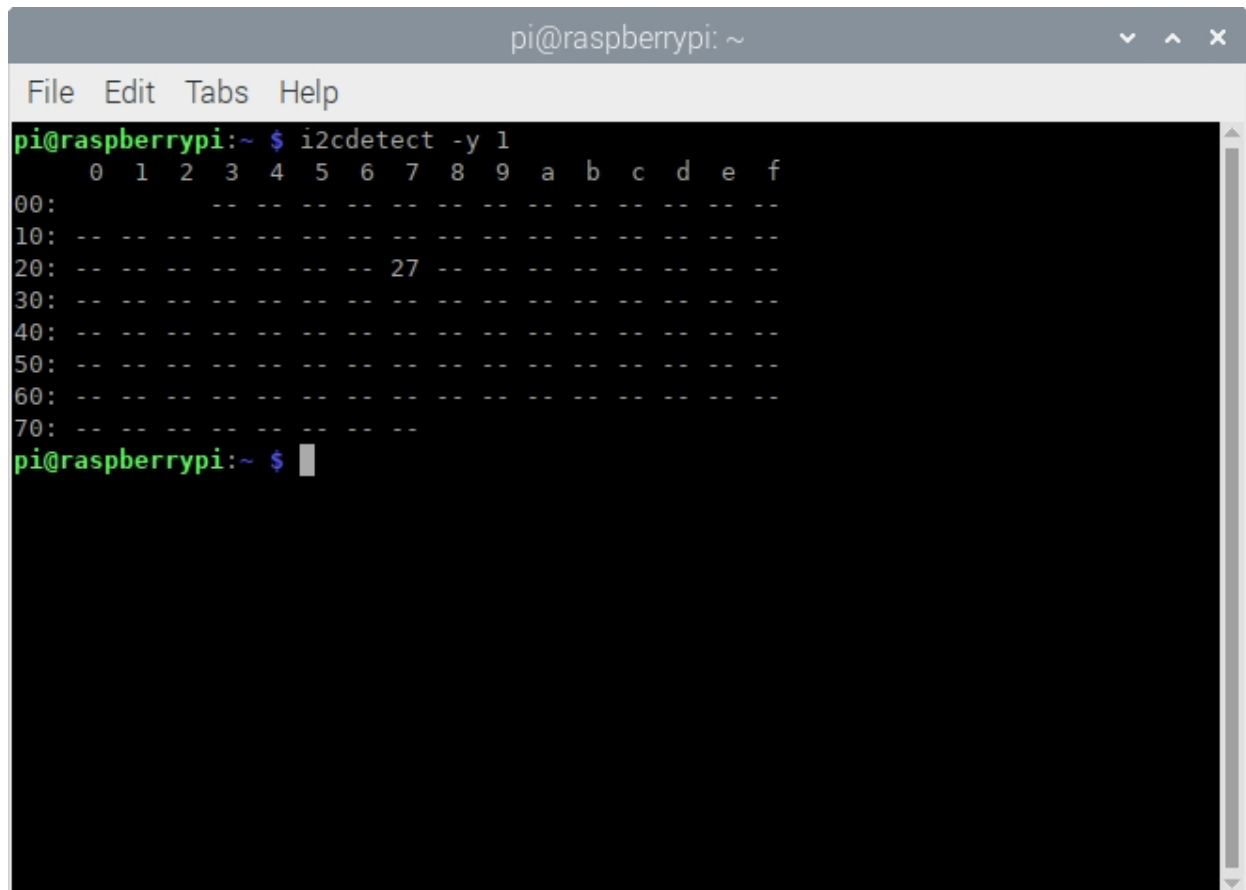
Para determinar la dirección I2C del adaptador I2C, debe estar instalado `i2c-tools`. Si aún no está instalado, ejecute el siguiente comando en el terminal:

```
sudo apt-get install i2c-tools
```

Az-Delivery

Para determinar la dirección I2C del adaptador I2C, ejecute el siguiente comando:

```
i2cdetect -y 1
```



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ i2cdetect -y 1  
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  -- 27  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
pi@raspberrypi:~ $
```

donde 0x27 es la dirección I2C del adaptador I2C.

Az-Delivery

Python-Skript

A continuación encontrará el código del script principal:

```
import lcd_class_i2c as LCD
import time

I2C_ADDR = 0x27
LINE_WIDTH = 16

pantalla = LCD.lcd(line_width=LINE_WIDTH, i2c_address=I2C_ADDR)

print('[¡Pulsa CTRL + C para finalizar el
script!]) try:
    mientras sea verdad:
        print('Imprimiendo mensajes en pantalla')
        screen.lcd_print('AZ-DELIVERY', 'LINE_1', 'CENTER')
        time.sleep(1) # 3 segundos de retraso

    print('Imprimiendo variable en pantalla')
    for i in range(100):
        si i < 10:
            screen.lcd_print('0{}'.format(i), 'LINE_2', 'CENTER')
        else:
            screen.lcd_print('{}'.format(i), 'LINEA_2', 'CENTRO')
        time.sleep(0.00001)
```

Az-Delivery

```
# una pestaña
    print('Probando
    retroiluminación')
    time.sleep(1)
    screen.backlight('OFF')
    time.sleep(1)
    screen.backlight('ON')
    time.sleep(1)
    print('Borrando la
    pantalla\n') # Pantalla en
    blanco screen.clear_screen()
    time.sleep(1)

except KeyboardInterrupt:
    print('\n¡Fin del
    script!')

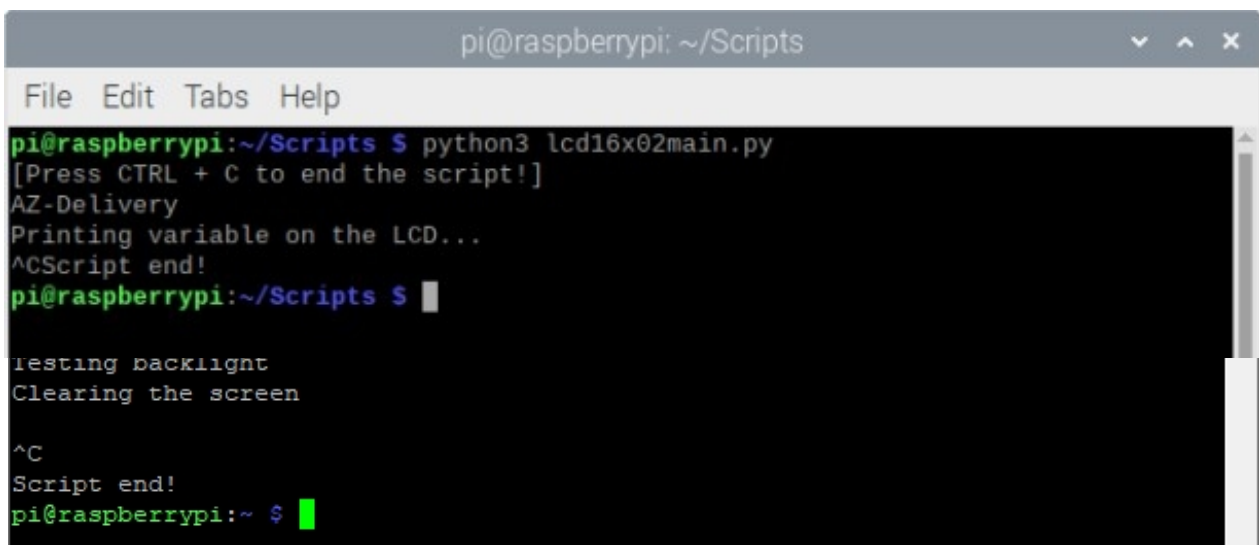
finalmente:
    screen.clear_screen()
```

Az-Delivery

Guarde el script con el nombre `lcd16x02i2c.py` en el mismo directorio que el script `lcd_class_i2c.py`. Para ejecutar el script, abra el terminal en el directorio donde están guardados los scripts y ejecute el siguiente comando:

```
python3 lcd16x02main.py
```

El resultado debería ser el siguiente:

A screenshot of a terminal window titled 'pi@raspberrypi: ~/Scripts'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal shows the command 'python3 lcd16x02main.py' being executed. The output of the script is displayed on the screen: 'AZ-Delivery', 'Printing variable on the LCD...', '^CScript end!', 'resting backlight', 'Clearing the screen', '^C', and 'Script end!'. The prompt 'pi@raspberrypi:~ \$' is visible at the bottom of the terminal.

```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 lcd16x02main.py
[Press CTRL + C to end the script!]
AZ-Delivery
Printing variable on the LCD...
^CScript end!
pi@raspberrypi:~/Scripts $
resting backlight
Clearing the screen
^C
Script end!
pi@raspberrypi:~ $
```

Para detener el script, pulse 'Ctrl + C' en el teclado.

Az-Delivery

La primera secuencia de comandos se utiliza para crear todas las funciones para controlar la pantalla LCD, que no está cubierto en este eBook.

El script principal comienza importando el primer script `lcd_class_i2c.py` y la librería `time`.

A continuación, se crea el objeto pantalla. Se utiliza para controlar la pantalla. Creamos el objeto con la siguiente línea de código:

```
screen=LCD.lcd(line_width=LINE_WIDTH, i2c_address=I2C_ADDR)
```

Donde el constructor `lcd()` tiene dos argumentos. El primer argumento se llama `line_width` y representa el número de caracteres por línea de la pantalla. El script `lcd_class_i2c.py` puede usarse tanto para pantallas LCD de 16x02 como para pantallas de 20x04. Así que los valores pasados al argumento `line_width` son 16 o 20. Cualquier otro valor provocará un error, y el script establece el valor de este argumento en 16. El segundo argumento, llamado `i2c_address` representa la dirección I2C del adaptador I2C, que en este caso es 0x27.

AZ-Delivery

A continuación se crea un bloque de código try-except-finally. En el bloque try, se crea un bloque de bucle Indefinido (while True:). En el bloque de bucle indefinido, primero se muestra el mensaje AZ-Delivery en la primera línea de la pantalla. El mensaje se sitúa en el centro de la línea. Esto se hace con la siguiente línea de código:

```
screen lcd_print('AZ-DELIVERY', 'LINE_1', 'CENTER')
```

Donde se utiliza la función lcd_print(). Esta función muestra un mensaje en la pantalla. Tiene tres argumentos y no devuelve ningún valor. El segundo y tercer argumento son opcionales. El primer argumento es una cadena que representa el mensaje mostrado en la pantalla. El segundo argumento, también una cadena, representa la línea en la que se muestra el mensaje. Los valores de este argumento son: LÍNEA_1 o LÍNEA_2. El valor por defecto es LÍNEA_1, que se selecciona si no se utiliza el argumento. Cualquier otro valor provocará un error y el valor se establecerá en el valor por defecto LÍNEA_1. El tercer argumento, también una cadena, representa la alineación del texto en la línea. Los valores de este argumento son: IZQUIERDA, CENTRO o DERECHA. El valor por defecto es IZQUIERDA, que se selecciona si no se utiliza el argumento. Cualquier otro valor provocará un error y el valor también se establecerá en el valor por defecto IZQUIERDA.

Para mostrar una variable en la pantalla, utilice las siguientes líneas de código:

```
mi_var = 10  
screen lcd_print('{}'.format(mi_var))
```

Az-Delivery

Para controlar la retroiluminación de la pantalla, se utiliza la función `backlight()`. Esta función tiene un argumento y no devuelve ningún valor. El valor del argumento es una cadena que sólo puede tener dos valores: ON u OFF. Para encender la pantalla, se utiliza la siguiente línea de código:

```
screen.backlight('ON')
```

Para desactivarlo, utilice lo siguiente:

```
screen.backlight('OFF')
```

Para borrar la pantalla (búfer de datos de la pantalla), utilice la función `clear_screen()`. Esta función no tiene argumentos y no devuelve ningún valor.

El bloque de código `except` se ejecuta cuando se pulsa CTRL + C en el teclado. Esto se llama `KeyboardInterrupt`. Cuando se ha ejecutado este bloque, aparece en el terminal el mensaje ¡Fin de script!

El bloque de código `finally` se ejecuta después del script. Cuando se ha ejecutado el bloque de código `finally`, se ejecuta la función `clear_screen()`. Esta función borra el buffer de datos de la pantalla.

Lo ha conseguido. Ya puede utilizar nuestro módulo para sus proyectos.



¡Ahora es tu turno! Desarrolla tus propios proyectos e instalaciones domésticas inteligentes. Te mostraremos cómo hacerlo de forma sencilla y comprensible en nuestro blog. Allí te ofrecemos scripts de ejemplo y tutoriales con pequeños proyectos interesantes para iniciarte rápidamente en el mundo de la microelectrónica. Además, Internet también te ofrece innumerables oportunidades para aprender más sobre microelectrónica.

Si busca microelectrónica y accesorios de alta calidad, AZ-Delivery Vertriebs GmbH es el lugar adecuado para usted. Le ofrecemos numerosos ejemplos de aplicación, instrucciones de instalación detalladas, libros electrónicos, bibliotecas y, por supuesto, el apoyo de nuestros expertos técnicos.

<https://az-delivery.de>

¡Diviértete!

Impresionante

<https://az-delivery.de/pages/about-us>