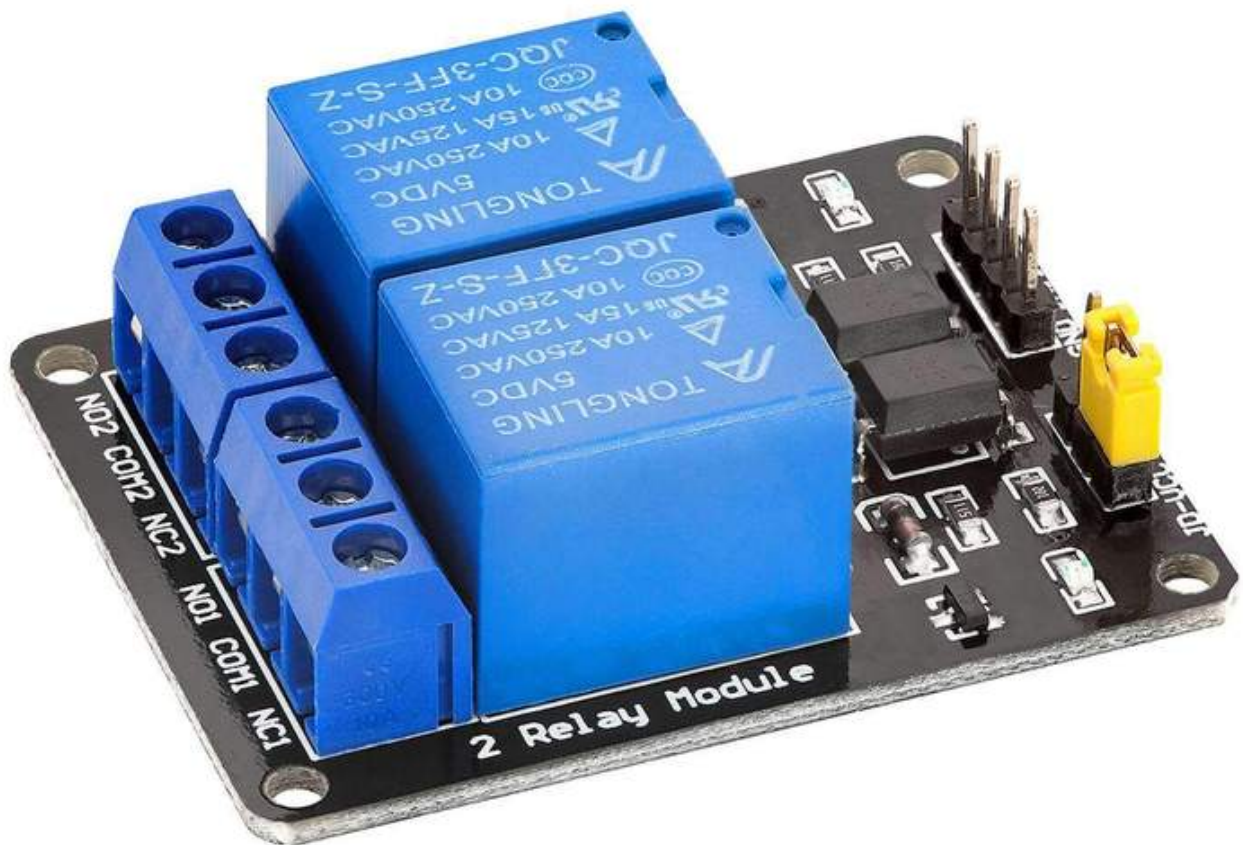


AZ-Delivery

¡Bienvenido!

Muchas gracias por comprar nuestro AZ-Delivery Módulo de 2 Relés. En las siguientes páginas, se presentará como utilizar y configurar este práctico dispositivo.

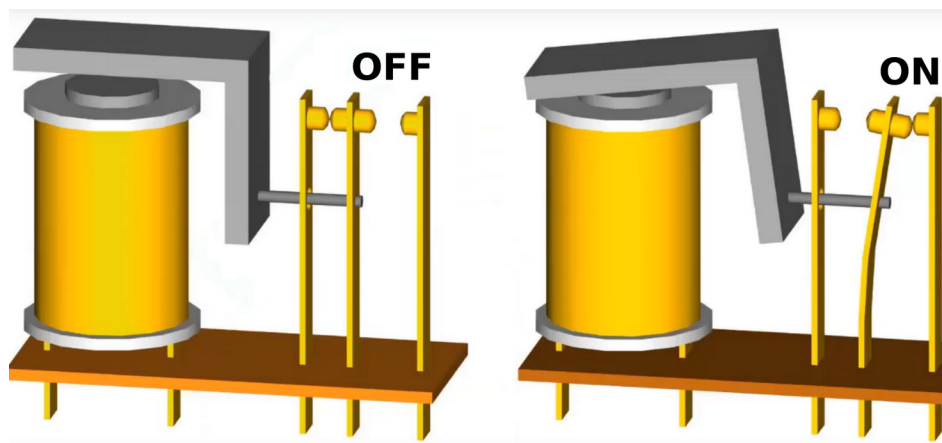
¡Diviértase!



Az-Delivery

Los relés se utilizan para controlar los circuitos de CA, encendiéndolos o apagándolos. El relé es uno de los elementos de control más importantes. Es un interruptor eléctrico que responde a una señal recibida del microcontrolador (como en Arduino o Raspberry Pi). Los relés se utilizan ampliamente en el control remoto, las comunicaciones, los dispositivos mecatrónicos, electrónicos de potencia, etc. También se pueden utilizar para separar el potente voltaje/corriente electrónico (como los motores de CA o CC, o cualquier dispositivo de CA, etc.), de los microelectrónicos (como los microcontroladores, sensores, etc.).

En el interior del relé, hay un interruptor mecánico (tres varillas metálicas amarillas, con una en el centro que está doblada hacia un lado y que es móvil), el cual es controlado por el segundo elemento del electroimán (cilindro amarillo), como se observa en la imagen inferior:



En el estado no activo, el interruptor está en el estado apagado, el pin NC está conectado con el pin común, y el pin NO está desconectado. Cuando se está conectando la energía al electroimán (a través del transistor y el diodo rectificador), éste mueve el interruptor al estado activo, conectando así el pin común al pin NO.



SAFETY WARNING!

When doing projects that are connected to mains voltage, misuse may lead to serious electrical shock!

For the sake of your own safety, be 100% sure what you are doing! Otherwise, ask someone who knows!

Según la normativa vigente, el trabajo con la tensión de la red está reservado a los electricistas cualificados.

El módulo de 2 relés consiste en dos LEDs, cuatro resistencias, dos transistores NPN, dos diodos rectificadores y dos relés capaces de manejar hasta 5A 50V AC.

En el lado de CC del tablero hay cuatro pines, dos pines de entrada para dos relés, uno para la fuente de alimentación (VCC) y otro para tierra (GND). También hay un puente de tres pines para seleccionar la fuente de alimentación (externa o VCC). En el lado de la CA hay un cabezal de terminal de tres pines, donde los pines están etiquetados como: Normalmente Cerrado - pin NC, pin Común y Normalmente Abierto - pin NO.

Especificaciones:

Señal de Control TTL: 3.3V a 12V DC

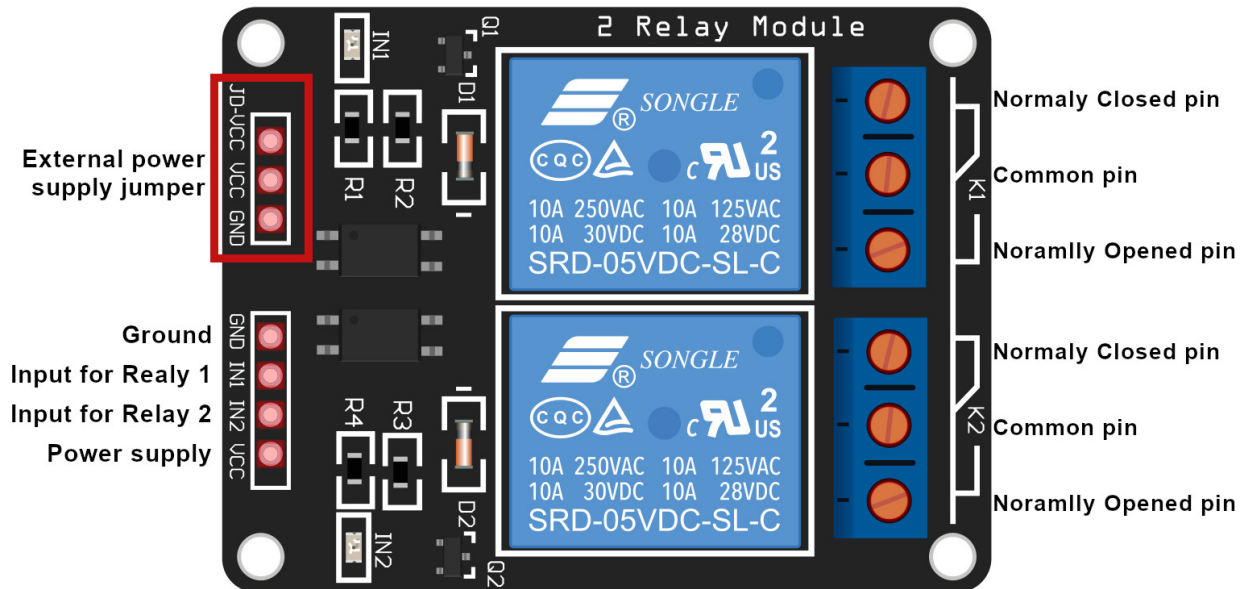
Máximo CA: 5A 50V

Máximo CC: 5A 30V

Tipo de Contacto: Ambos: Normalmente Cerrado – NC,
Normalmente Abierto - NO

Dimensiones: 49 x 51mm [1.9 x 2pulgadas]

La disposición de pines



El puente de la fuente de alimentación externa se utiliza para seleccionar la entrada de la fuente de alimentación. Si se mantiene desconectada, los relés no se encenderán, pero los LEDs del módulo seguirán parpadeando. Si conecta el pin JD-VCC y el pin VCC juntos (con un puente de dos pines), el módulo se encenderá desde el pin VCC.

Si desea utilizar una fuente de alimentación externa, retire el puente de dos pines y conecte el lado positivo de la fuente de alimentación externa al pin JD-VCC y el pin de tierra de la fuente de alimentación externa con el pin de tierra del módulo.



Fuente de alimentación externa

¿Por qué se necesita una fuente de alimentación externa?

En primer lugar, se necesita porque en ocasiones el regulador de voltaje incorporado de Arduino no es lo suficientemente potente para conducir el Arduino y el módulo.

Y en segundo lugar, es mejor que los relés y las fuentes de alimentación de Arduino estén separados. Debido a que los relés se utilizan para controlar la CA o los potentes dispositivos de CC, y el tener que controlar los circuitos electrónicos y los potentes circuitos electrónicos separados es una medida de precaución de seguridad. Una de las ventajas de utilizar relés es que protegen los circuitos microelectrónicos de los circuitos electrónicos potentes.

Configuración del Arduino IDE

Si el Arduino IDE no está instalado, aquí se presenta como hacerlo. Seguir el siguiente link: <https://www.arduino.cc/en/Main/Software> y descargar el archivo de instalación para su plataforma de sistema operativo.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle with a white infinity symbol containing a minus and a plus sign. To its right, the text reads: **ARDUINO 1.8.9**, followed by a description of the IDE as open-source software written in Java, and a note that it can be used with any Arduino board. On the right side, there is a teal sidebar with links for different operating systems: Windows (installer and ZIP file), Windows app (with a 'Get' button), Mac OS X, and Linux (32 bits, 64 bits, and ARM 32/64 bits). At the bottom of the sidebar are links for Release Notes, Source Code, and Checksums.

ARDUINO 1.8.9
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.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Para los usuarios de *Windows*, hacer doble clic en el archivo descargado .exe y seguir las instrucciones de la ventana de instalación.

Az-Delivery

Para los usuarios de *Linux*, descargar un archivo con la extensión *.tar.xz*, que se debe extraer. Cuando se la extrae, ir al directorio extraído y abrir el terminal en ese directorio. Se deben ejecutar dos scripts *.sh*, el primero es *arduino-linux-setup.sh* y el segundo es *install.sh*.

Para ejecutar el primer script en el terminal, ejecutar el siguiente comando:

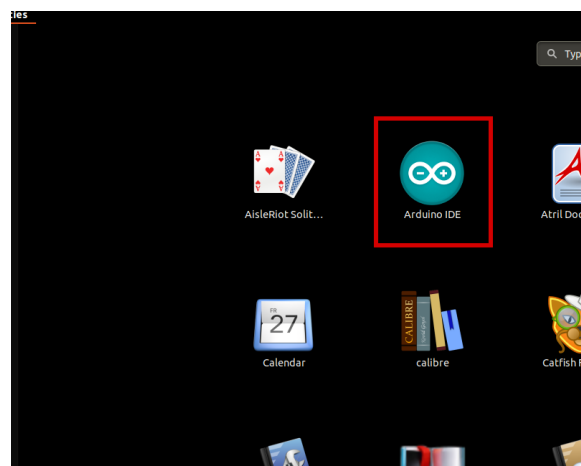
```
sh arduino-linux-setup.sh user_name
```

user_name - es el nombre de un superusuario en el sistema operativo Linux. Una contraseña para el superusuario se debe introducir cuando se inicia el comando. Se debe esperar unos minutos para que el script complete todo.

El segundo script *install.sh* se debe utilizar después de la instalación del primer script. Se debe ejecutar el siguiente comando en el terminal:

```
sh install.sh
```

Después de la instalación de estos scripts, ir a la sección *All Apps*, donde *Arduino IDE* está instalado.

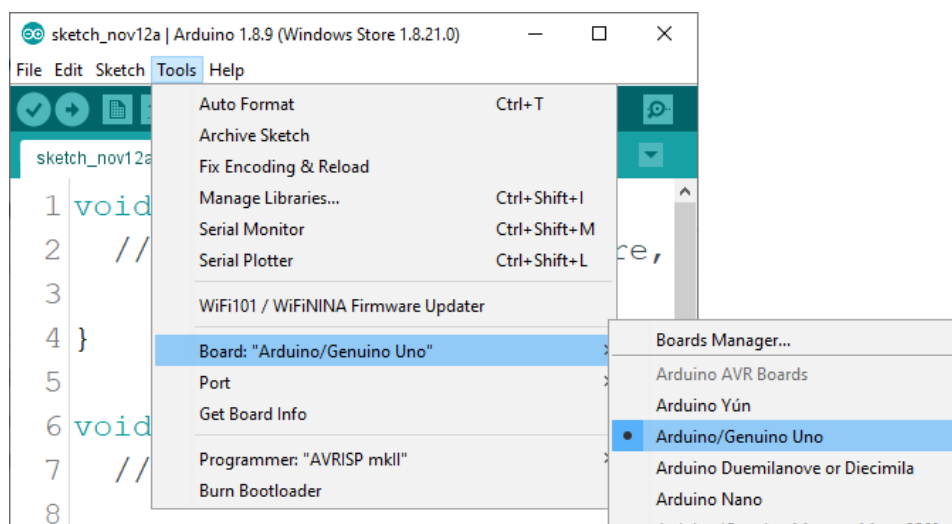


Az-Delivery

Lo siguiente es comprobar si el PC puede detectar una placa Arduino. Para esto, se debe abrir el Arduino IDE recién instalado, e ir a:

Tools > Board > {your board name here}

{your board name here} debe ser el *Arduino/Genuino Uno*, como se puede observar en la imagen a continuación:

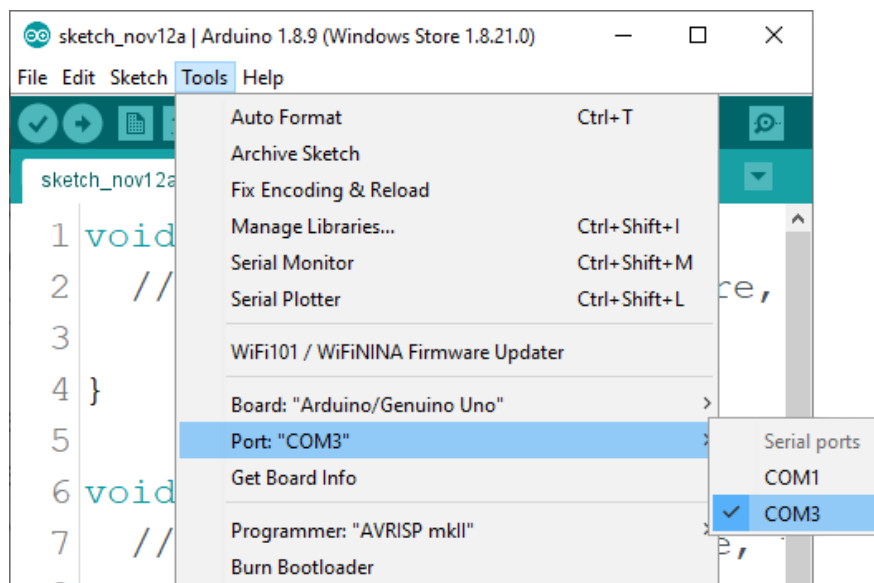


Az-Delivery

Se debe seleccionar el puerto al que está conectado la placa de Arduino. Ir

a: *Tools* > *Port* > {port name goes here}

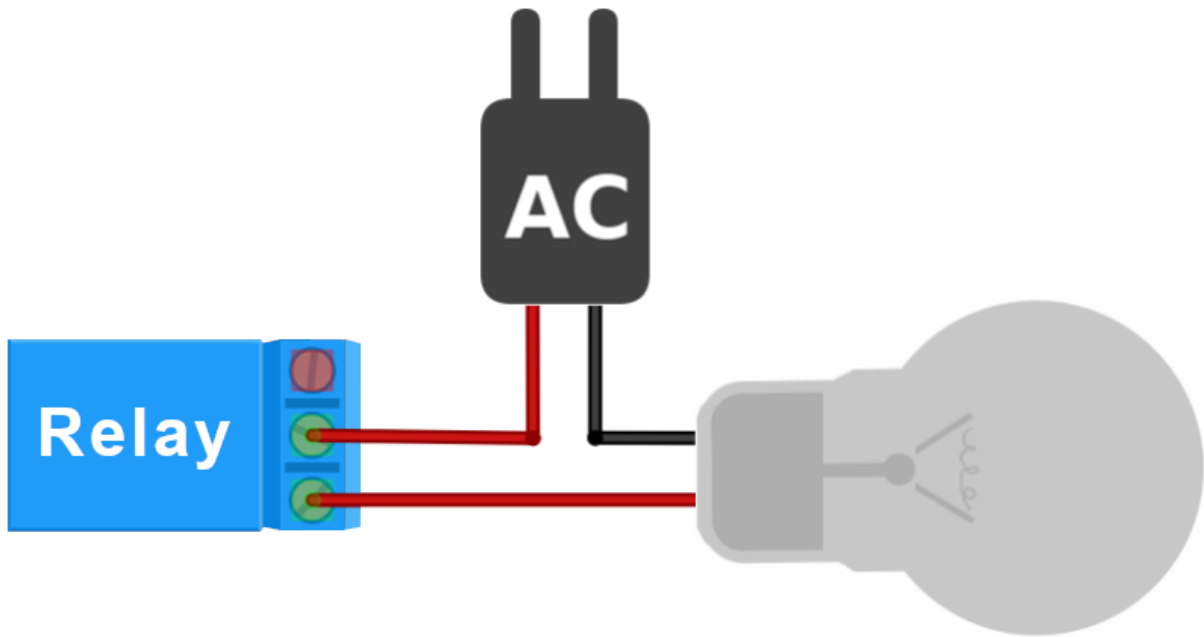
Si conectó la placa Uno en el puerto usb, se deberían observar varios nombres de puertos. Debido a que se está utilizando *Arduino IDE* en *Windows*, los nombres de los puertos son los que se observan en la imagen inferior.



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

Conectando el lado CA del relé

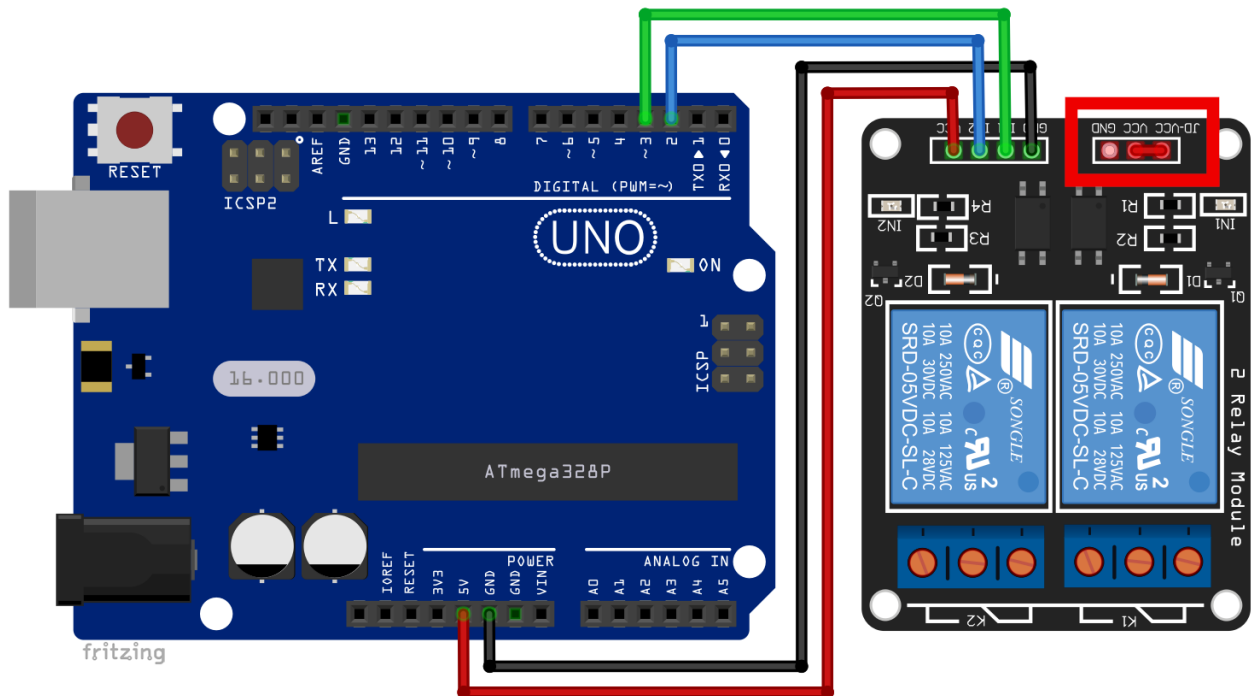
Para esto se utilizará una bombilla, un cable de dos alambres y un enchufe de corriente alterna. La parte de la CA del diagrama de conexión es la misma para ambos relés del módulo de 2 relés. Conecte el relé con la bombilla y el enchufe de energía como se observa en el siguiente diagrama de conexión:



Módulo Pin	> Enchufe de energía, bombilla	
Pin Común	> Un lado del enchufe de energía	Cable rojo
Pin Normalmente Abierto	> Un lado de la bombilla	Cable rojo
Bombilla	> Enchufe de energía	
El otro lado de la bombilla	> El otro lado del enchufe	Cable negro

Conectando el módulo con Uno

Conectar el módulo de 2 relés con el Uno como se observa en el siguiente diagrama de conexión:



Módulo Pin > Pin de Uno

IN1 > D2

IN2 > D3

GND > GND

VCC > 5V

Cable azul

Cable verde

Cable negro

Cable rojo

NOTA: Como puede observar en el rectángulo rojo del diagrama de conexión, un puente de energía externa está conectado, conectando el pin JD-VCC con el pin VCC. Esto representa que el módulo de 2 relés se alimentará desde la placa Arduino a través del pin VCC.

Az-Delivery

Sketch de ejemplo:

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
}  
void loop() {  
  digitalWrite(2, HIGH);  
  digitalWrite(3, HIGH);  
  delay(1000);  
  digitalWrite(2, LOW);  
  digitalWrite(3, LOW);  
  delay(1000);  
}
```

Cuando cargue el sketch en el Uno, se escucharán los clics de los relés. Cuando el relé cambia de estado de activo a reposo y viceversa, se pueden escuchar los clics de conmutación.

Las bombillas conectadas a los relés deberían parpadear cada segundo. Se pueden cambiar los estados de los pines NO/NC por estas líneas del código:

`digitalWrite(2, HIGH);` - **NC** pin no está conectado al pin común
NO pin está conectado al pin común

`digitalWrite(2, LOW);` - **NC** pin está conectado al pin común
NO pin no está conectado al pin común



Configuración del Raspberry Pi y Python

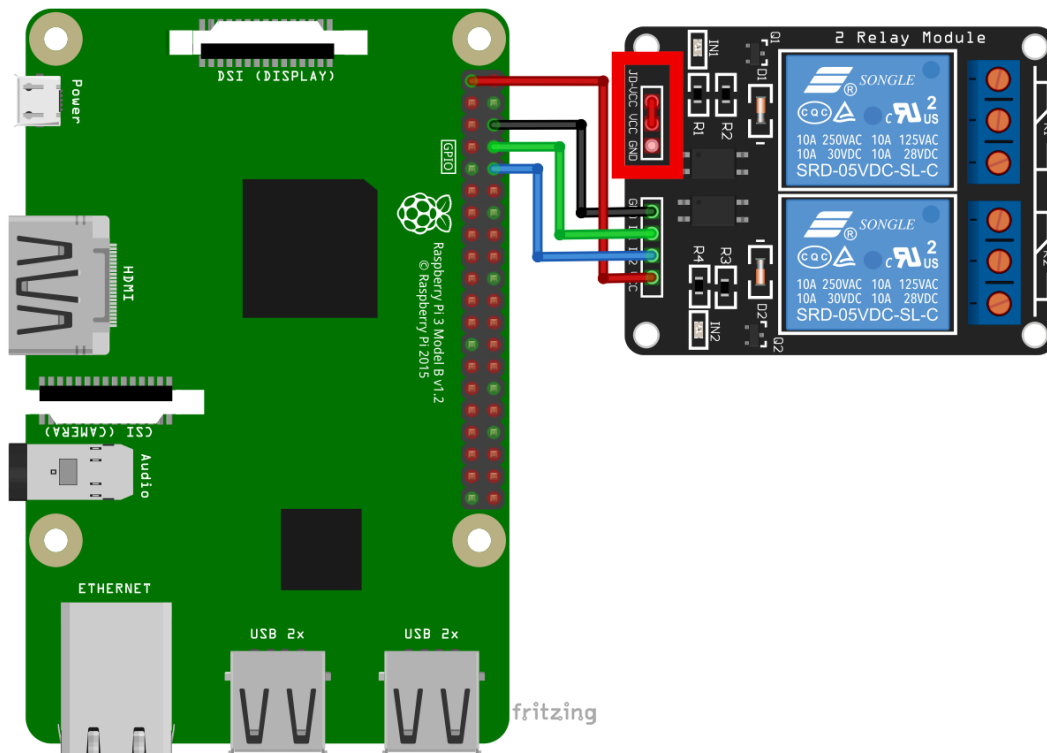
Para la configuración del Raspberry Pi, primero se debe instalar el sistema operativo, después se debe configurar todo para que se pueda utilizar en el modo *Headless*. El modo *Headless* permite la conexión remota al Raspberry Pi, sin la necesidad de un monitor de pantalla de PC, ratón o teclado. Puede encontrar una explicación detallada en el E-book gratuito "*Raspberry Pi Quick Startup Guide*", que se puede encontrar en nuestro sitio:

<https://www.az-delivery.de/products/raspberry-pi-kostenfreies-e-book?ls=en>

El sistema operativo *Raspbian* incluye *Python* preinstalado.

Conectando el módulo con Raspberry Pi

Conecte el módulo de 2 relés con el Raspberry Pi como se observa en el siguiente diagrama de conexión:



Módulo Pin > Pin de Raspberry Pi

VCC > 3V3 [pin 1]

Cable rojo

GND > GND [pin 6]

Cable negro

IN1 > GPIO14 [pin 8]

Cable verde

IN2 > GPIO15 [pin 10]

Cable azul

NOTA: Como puede observar en el rectángulo rojo del diagrama de conexión, un puente de energía externa está conectado, conectando el pin JD-VCC con el pin VCC. Esto representa que el módulo de 2 relés se alimentará desde la placa Raspberry Pi a través del pin VCC.



Script Python:

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

Relay1_PIN = 14
Relay2_PIN = 15
GPIO.setup(Relay1_PIN, GPIO.OUT)
GPIO.setup(Relay2_PIN, GPIO.OUT)

print('[press ctrl+c to end the script]')
try: # Main program loop
    while True:
        GPIO.output(Relay1_PIN, GPIO.HIGH)
        GPIO.output(Relay2_PIN, GPIO.HIGH)
        print('Normally opened pin is HIGH')
        sleep(1) # Waitmode for 1 second
        GPIO.output(Relay1_PIN, GPIO.LOW)
        GPIO.output(Relay2_PIN, GPIO.LOW)
        print('Normally opened pin is LOW')
        sleep(1) # Waitmode for 1 second

# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('Script end!')

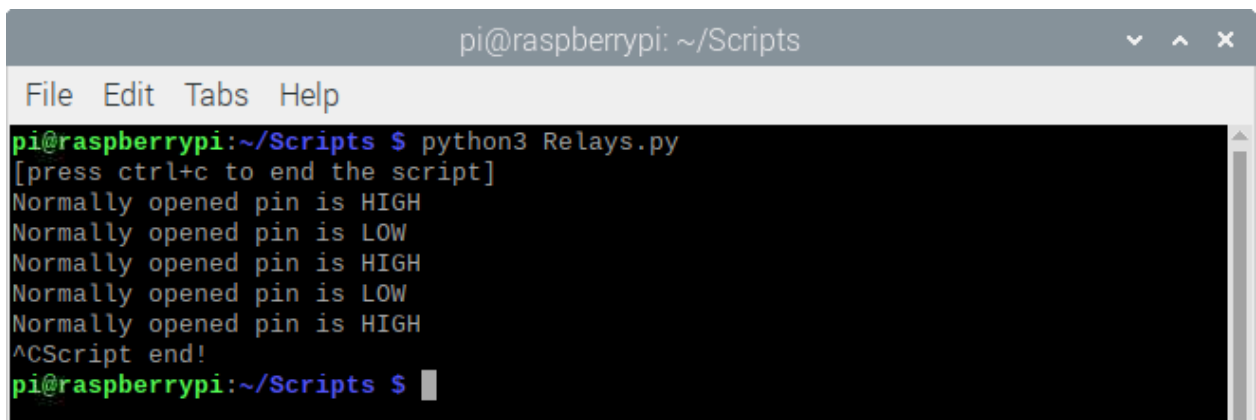
finally:
    GPIO.cleanup()
```

Az-Delivery

Guardar el script con el nombre "*Relays.py*" en el directorio de scripts predeterminado. Para ejecutar el script, abra el terminal en el directorio donde guardó el script y ejecute el siguiente comando:

python3 Relays.py

La salida se debe observar como la de la imagen inferior:

A screenshot of a terminal window titled "pi@raspberrypi: ~/Scripts". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal output shows the command "python3 Relays.py" being executed. The script prints several lines: "[press ctrl+c to end the script]", "Normally opened pin is HIGH", "Normally opened pin is LOW", "Normally opened pin is HIGH", "Normally opened pin is LOW", "Normally opened pin is HIGH", and finally "^CScript end!". The prompt "pi@raspberrypi:~/Scripts \$" is visible at the bottom of the terminal.

```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Relays.py
[press ctrl+c to end the script]
Normally opened pin is HIGH
Normally opened pin is LOW
Normally opened pin is HIGH
Normally opened pin is LOW
Normally opened pin is HIGH
^CScript end!
pi@raspberrypi:~/Scripts $
```

Para terminar el script presione "*CTRL + C*".

El script es auto-explicativo.

¡Lo ha logrado!

Ahora puede usar su módulo para varios proyectos.



Ahora es el momento de aprender y hacer sus propios proyectos. Puede hacerlo con la ayuda de muchos scripts de ejemplo y otros tutoriales, que puede encontrar fácilmente en Internet.

Si está buscando productos de alta calidad para Arduino y Raspberry Pi, AZ-Delivery Vertriebs GmbH es la compañía adecuada donde podrá obtenerlos. Se le proporcionarán numerosos ejemplos de aplicación, guías completas de instalación, libros electrónicos, librerías y la asistencia de nuestros expertos técnicos.

<https://az-delivery.de>

¡Disfrute!

Impressum

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