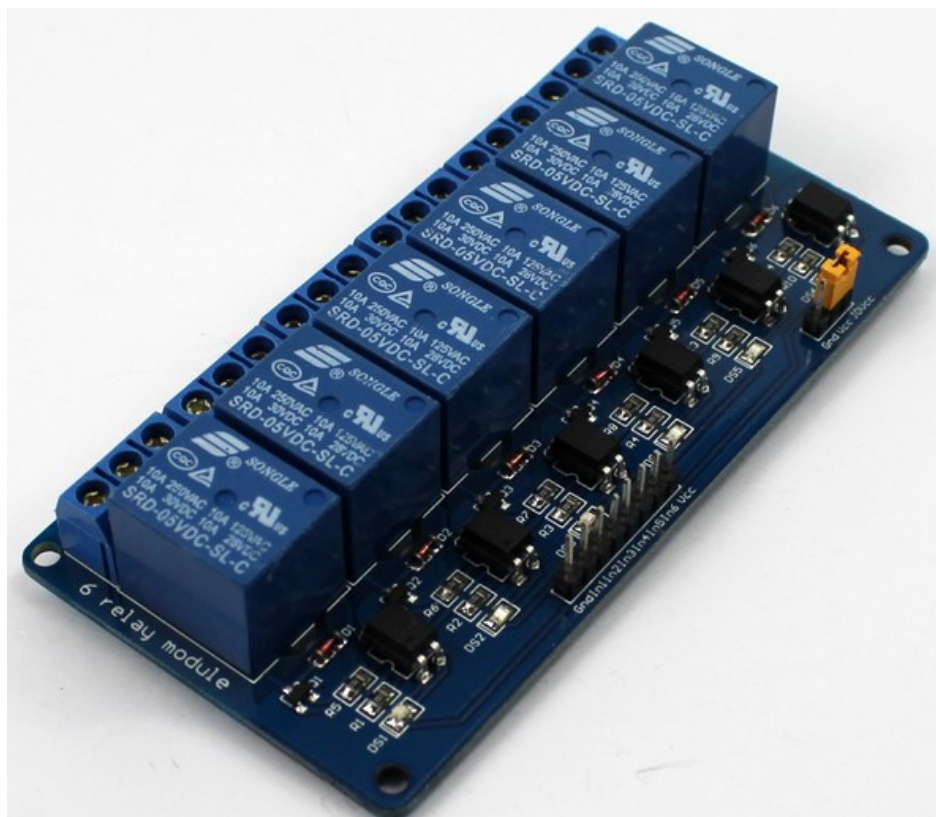
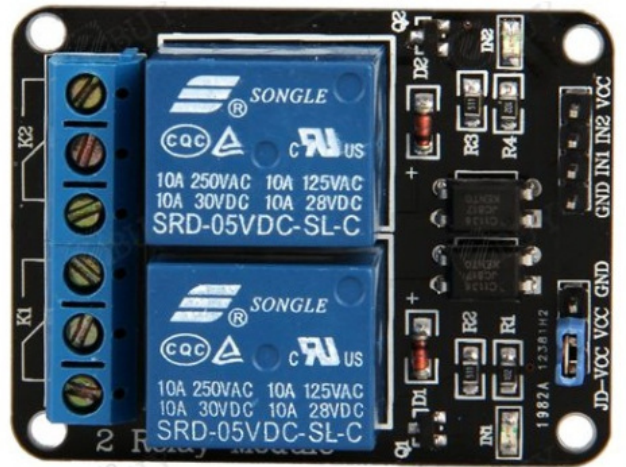


# MÓDULO DE RELÉS

(Versión 03-8-18)

El módulo que vamos a emplear es uno muy típico y a un precio aceptable:

Como podemos ver, en este módulo (que es el que he empleado para el tutorial) trae dos entradas para poder activar dos relés, pero podemos encontrar módulos con más relés e incluso de un solo relé, lo que necesitemos para nuestros proyectos.



## Descripción del Producto

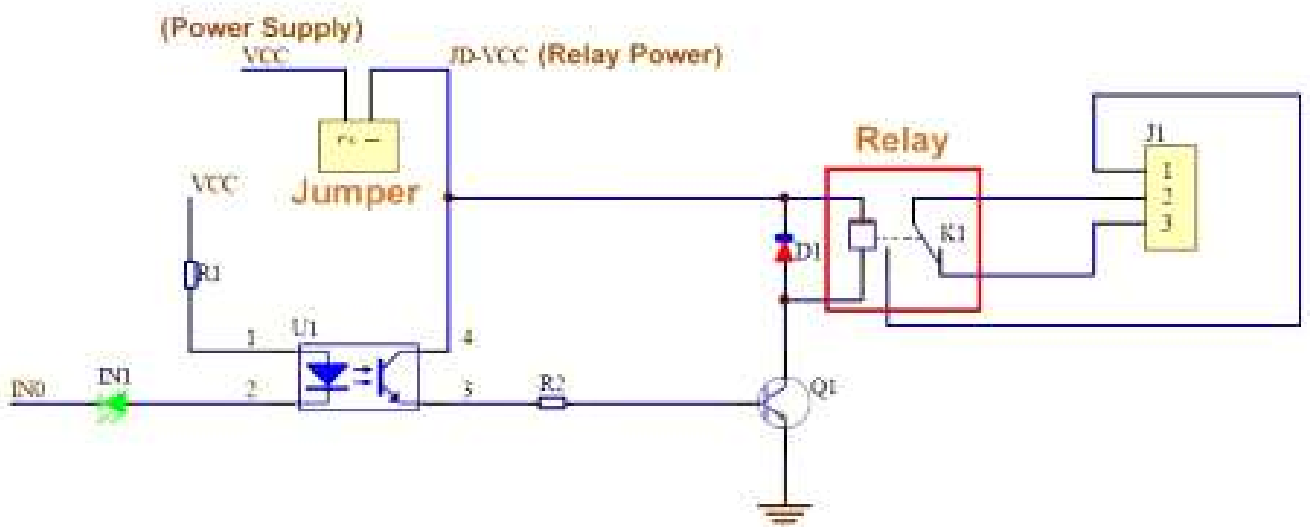
Plataforma: Arduino 8051 AVR PIC DSP ARM MSP430 TTL. (Son distintas plataformas)  
Control Dispositivos varios/cargas  
Voltaje de operación 250VAC/30VDC  
Voltaje de la bobina (relé) 5V  
Corriente de operación 10A. *Algunos fabricantes aclaran:*

<b>Corriente máx: 10A (NO), 5A (NC)</b> <b>Tiempo de acción: 10 ms / 5 ms</b>
--

**Corriente de activación por relé 15mA~20mA** **(ATENCIÓN: VEA ANEXO1)**  
Canales 1-2 - 4 - 6 -8 (independientes protegidos con Optoacopladores)  
LED indicador Para cada canal (cuando bobina está activa)

Este módulo de relevadores (relés) es para conmutación de cargas de potencia. Los contactos de los relevadores están diseñados para conmutar cargas de hasta 10A y 250VAC (30VDC), aunque se recomienda usar niveles de tensión por debajo de estos límites. Las entradas de control se encuentran aisladas con optoacopladores para minimizar el ruido percibido por el circuito de control mientras se realiza la conmutación de la carga. La señal de control puede provenir de cualquier circuito de control TTL o CMOS como puede ser un microcontrolador. Este módulo es ideal para conmutar cargas de corriente alterna conectadas a la red eléctrica. Soporta todos los microcontroladores, aplicaciones en zonas industriales, control del PLC, entre otros. Este módulo es capaz de controlar varios equipamientos de alta corriente durante un tiempo prolongado. Puede ser controlado por muchos microcontroladores como Arduino, 8051, AVR, PIC, DSP, ARM, MSP430, TTL.

Esquema de conexión para que funcione todo correctamente:



Como podemos ver en la foto del módulo observamos que tenemos dos tiras de pines, una que es en la que esta el jumper, que tiene tres pines y podemos seleccionar alimentar todo el módulo de forma conjunta (es decir seleccionar con el jumper jd-vcc y vcc), o bien alimentar de forma independiente tanto optoacopladores como relés (quitando el jumper), alimentar todo de manera conjunta parece la forma más sencilla de hacerlo, así que he dejado el jumper uniendo jd-vcc y vcc.

Y la otra tira de pines que es donde nos aparece en el siguiente orden:

**GND IN1 IN2 VCC**

Aquí es donde unimos la alimentación con nuestro arduino, es decir:

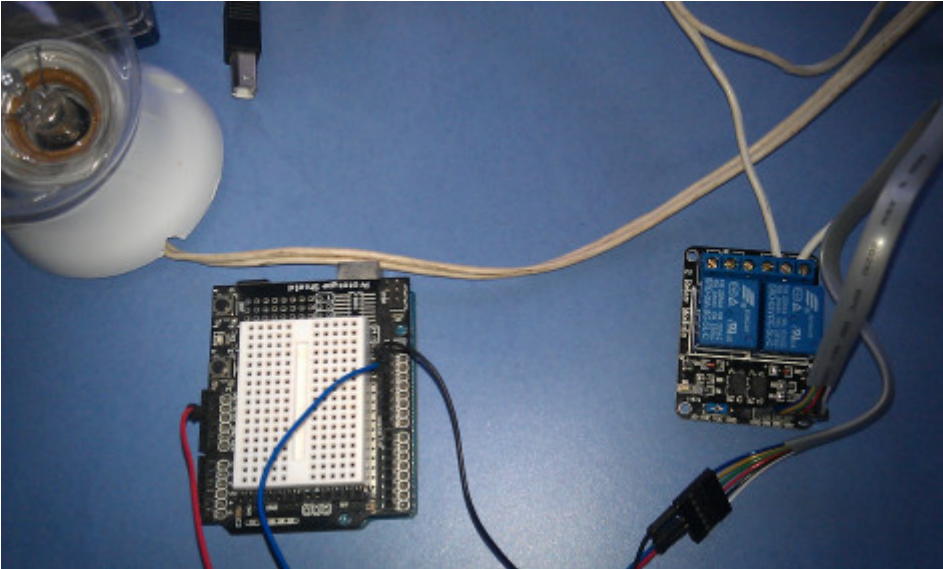
**GND del módulo con GND del Arduino**

**VCC del módulo con 5V del Arduino**



Entradas **IN1 e IN2** (se activan las bobinas de los relés cuando colocamos a **GND** dichas entradas) conectados a los pines que deseemos de nuestro Arduino, y serán los encargados de activar las bobinas de los relés correspondientes.

**NOTA:** En la figura de conexión se observa que se justifica colocar GND en IN1 e IN2, ya que de esa manera activamos el LED del OPTOACOPLADOR.



Por otro lado tenemos los conectores a los que conectaremos los aparatos eléctricos (**Contactos principales**) que queremos activar o desactivar mediante nuestro módulos de relés, como podéis observar en la imagen de arriba, tienen tres conectores cada relé, empezando por abajo observamos en este orden: Conector normalmente cerrado del relé 1 (está activo hasta que se activa la bobina del relé mediante la entrada IN1 poniéndola a LOW), el conector común del relé 1 y el conector normalmente abierto del relé 1 (está desactivado hasta que se activa la bobina del relé mediante la entrada IN1 y permanece activado hasta que se ponga a HIGH dicha entrada).

Ver que en la serigrafía de la placa se muestra cual es el contacto principal normalmente abierto y cual el cerrado.



## EL CASO MAS SIMPLE- UN SOLO RELE

### MÓDULO RELEE/RELAY

El manejo de un módulo relee o relay es muy sencillo y nos permite controlar el encendido y apagado de cualquier aparato que se conecte a una fuente de alimentación eléctrica externa. El relee hace de interruptor y se activa y desactiva mediante una entrada de datos. Gracias a esto podemos controlar el encendido de cualquier aparato.



Aquí podemos ver un relee de 1 canal pero en el mercado hay (como ya se menciono) de varios canales 2, 4, 8, etc. la única diferencia entre ellos es el número de pines de datos de entrada que tienen, 1 por canal.

A parte del pin que controla al relee, el módulo viene con 2 entradas que suelen estar situadas a cada uno de los extremos de la fila de pins de entrada. 1 es GND (Tierra) y otra VCC. Hay varios modelos con distintos voltajes de entrada. Si queremos utilizarlo desde nuestro Arduino sin necesidad de otra fuente de alimentación alternativa debería de ser de 5V pero podría ser de 12V (que son los más comunes).

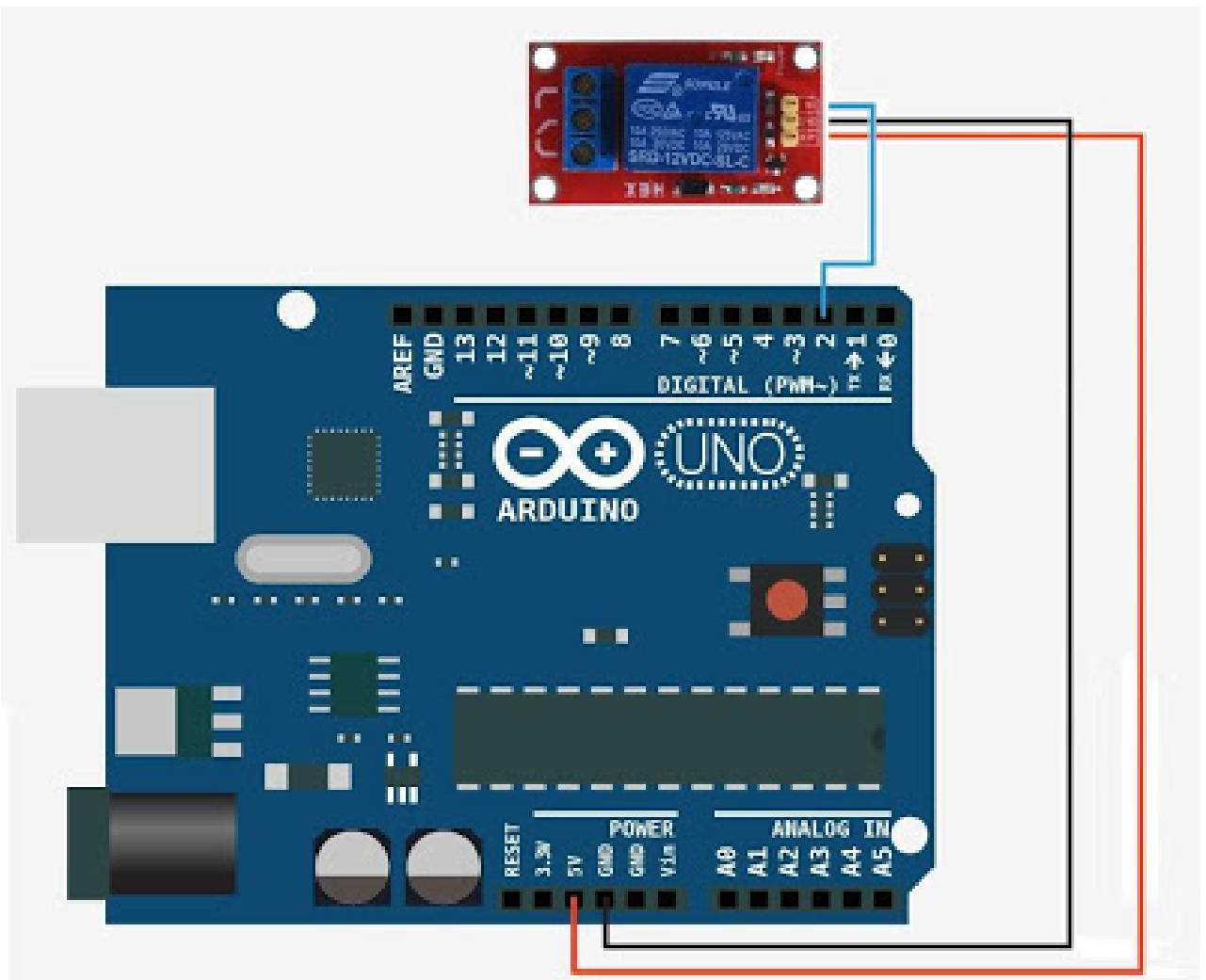
Para este pequeño ejemplo vamos a trabajar con un relee de 5V de esta forma nuestro Arduino va a ser totalmente autónomo.

Normalmente en el lado opuesto de donde se encuentra la fila de pins de entrada se encuentra un conector eléctrico con 3 hilos.

La conexión entre el módulo de relee y el Arduino es bastante simple, ya que lo único que tenemos que hacer es conectar los pines de la siguiente forma:

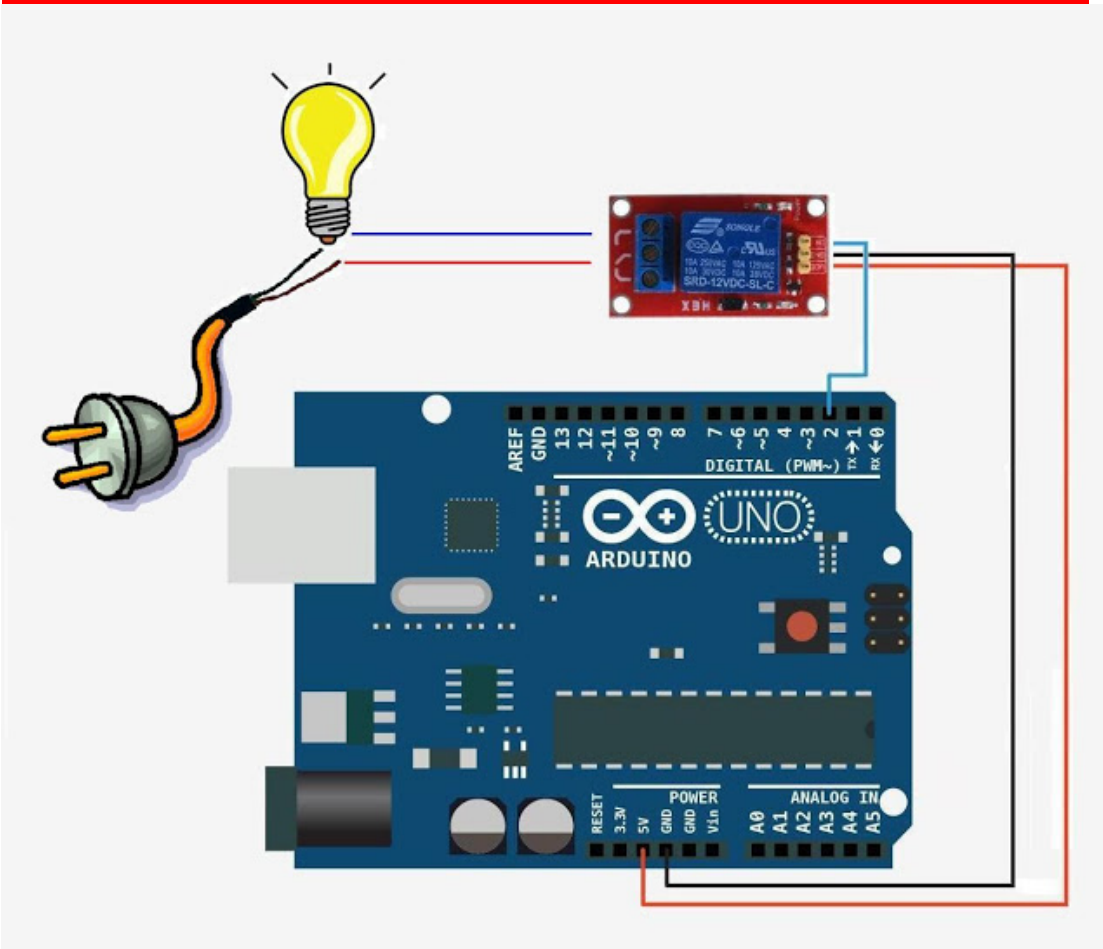
Módulo Relee	Arduino
VCC	5V
IN	D2 (Por ejemplo)
GND	GND





Ahora sólo nos faltaría conectar un aparato eléctrico al otro lado del relee, por ejemplo podemos poner una luz. Uno de los hilos eléctricos van directamente del enchufe al aparato y el otro es cortado por el relee. Como se puede ver aquí:

**LEA LA SERIGRAFIA DE SU MODULO – PUEDE DIFERIR DE LA IMAGEN**



Solo nos falta hacer un pequeño programa y cargarlo al Arduino, por ejemplo algo así:

```
// Modulo Relee
// Conectamos el modulo de la siguiente forma:

// GND -> GND

// VDD -> 5V

// IN -> D2

// Por ejemplo conectamos a la entrada Digital 2

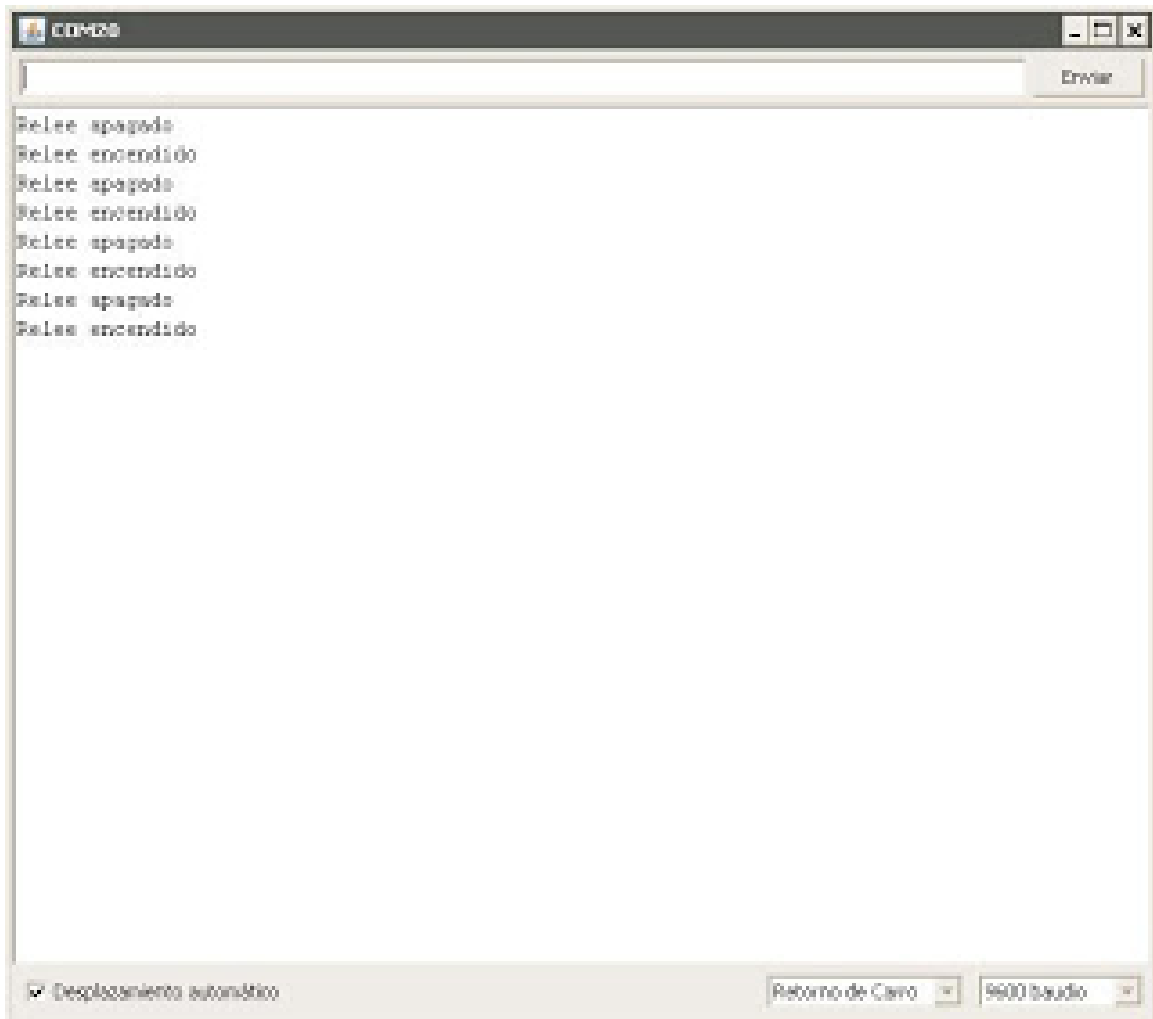
// http://arubia45.blogspot.com.es/

const int myRelee = 2; // Declaramos la salida del relee
                       // LOW encendido HIGH apagado

void setup() {
  // Relee
  pinMode(myRelee, OUTPUT);
  // Declaramos que el pin sera salida
  // Preparamos el puerto serie
  Serial.begin(9600);
}

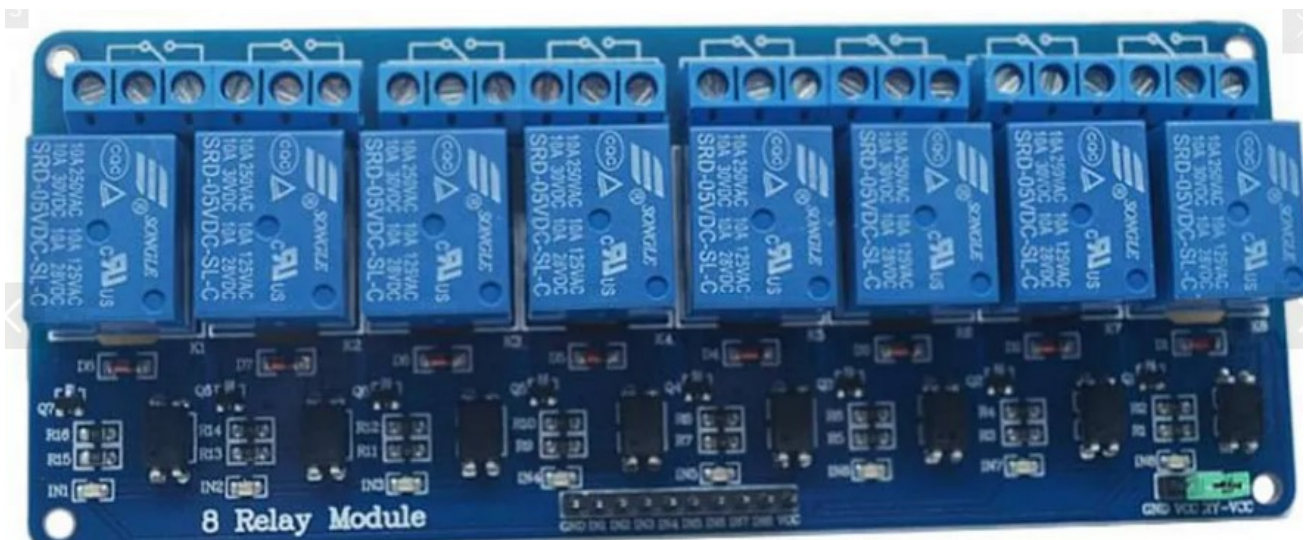
void loop(){
  digitalWrite(myRelee, HIGH);
  Serial.println("Relee apagado");
  delay (2000);
  digitalWrite(myRelee, LOW);
  Serial.println("Relee encendido");
  delay (2000);
}
```

Al cargar el sketch en el Arduino veríamos que el relee empezaría a parpadear así como los mensajes por el puerto serie.



## ANEXO 1:

La siguiente información es provista por el vendedor para el caso de módulos de 8 RELES:



### Descripción del Producto:

#### Características

Permite controlar el encendido/apagado de equipos de alta potencia (electrodomésticos).  
Funciona perfectamente con Arduino, Pic o cualquier otro sistema digital.

Dentro de la gran variedad de proyectos que podemos realizar con Arduino, podemos llegar a desear controlar componentes de alto voltaje o alto amperaje, como bombillas o bombas de agua, los cuales no pueden ser manejados directamente con Arduino. En estos casos es

necesario utilizar Relays o Reles, estos dispositivos permiten controlar cargas de alto voltaje con una señal pequeña.

El modulo posee 8 Relays de alta calidad, capaces de manejar cargas de hasta 250V/10A. Cada canal posee aislamiento eléctrico por medio de un optoacoplador y un led indicador de estado. Su diseño facilita el trabajo con Arduino, al igual que con muchos otros sistemas como Raspberry Pi, ESP8266 (NodeMCU y Wemos), Teensy y Pic. Este modulo Relay activa la salida normalmente abierta (NO: Normally Open) al recibir un "0" lógico (0 Voltios) y desactiva la salida con un "1" lógico (5 voltios). Para la programación de Arduino y Relays se recomienda el uso de timers con la función "millis()" y de esa forma no utilizar la función "delay" que impide que el sistema continúe trabajando mientras se activa/desactiva un relay.

Entre las cargas que se pueden manejar tenemos: bombillas de luz, luminarias, motores AC (220V), motores DC, solenoides, electroválvulas, calentadores de agua y una gran variedad de actuadores más. Se recomienda realizar y verificar las conexiones antes de alimentar el circuito, también es una buena práctica proteger el circuito dentro de un case.

### Datos técnicos

8 canales independientes

8 Relevadores (Relays) de 1 polo 2 tiros

El voltaje de la bobina del relé es de 5 VDC

Led indicador para cada canal (enciende cuando la bobina del relé esta activa)

Activado mediante corriente: el circuito de control debe proveer una corriente de 15 a 20 mA

Puede controlado directamente por circuito lógicos

Terminales de conexión de tornillo (clemas)



Terminales de entrada de señal lógica con headers macho



Puede ser controlado directamente por circuito lógicos

### Alimentación y consumo

La forma mas sencilla de alimentar este módulo es desde Vcc y GND de la placa Arduino, manteniendo el Jumper en su lugar, con lo que JD-Vcc = Vcc. Esta conexión tiene dos limitaciones importantes:

Se pierde la aislación eléctrica que brindan los optoacopladores, lo que aumenta la posibilidad de daño al Arduino si hay algún problema con las cargas de los relés.

La corriente consumida por las bobinas de los relés debe ser provista por la placa Arduino. Cada bobina consume unos 90 mA y las cuatro juntas suman 360 mA. Si a esto le sumamos los consumos que pueden tener otras salidas, **estamos muy cerca de los 500 mA que puede suministrar un puerto USB**. En este caso se debería alimentar al Arduino con una fuente externa, lo que aumenta el limite de corriente a 1 A (en el caso de la Arduino UNO).

La forma mas segura es remover el jumper y alimentar la placa de relés con dos fuentes: la de la placa Arduino conectada a Vcc y una segunda fuente, con el positivo a JD-Vcc y el negativo a GND, sin estar éste unido a la placa Arduino. Esta conexión tiene como ventajas:

Hay completa aislación entre la carga y el Arduino.

Todo el consumo de los relés es tomado de la segunda fuente y no del Arduino o el puerto USB.



## **Entradas**

Las entradas a la placa pueden conectarse directamente a las salidas digitales de la placa Arduino. La única precaución a tener en cuenta es que cuando Arduino arranca al ser alimentado, los pines se configuran como entradas automáticamente y puede ocurrir que, por un brevísimo lapso de tiempo entre el arranque y la correcta configuración de estos pines como salidas, las entradas de control al módulo de relé queden en un estado indeterminado. Esto se puede evitar conectando en cada entrada un pull-up con una resistencia de 10K a Vcc, lo que asegura un estado ALTO durante el arranque.