

ACTUADORES y otras salidas

Esta página es un repaso rápido de las posibles salidas del Arduino

¿Qué es un actuador?

“Un actuador es el nombre genérico que empleamos para cualquier **dispositivo capaz de realizar acciones en el mundo físico** y que podremos controlar desde un autómata o procesador como Arduino. En particular, usamos el nombre actuador para los dispositivos que son capaces de generar movimiento.

Luis LLamas CC-BY-NC-SA

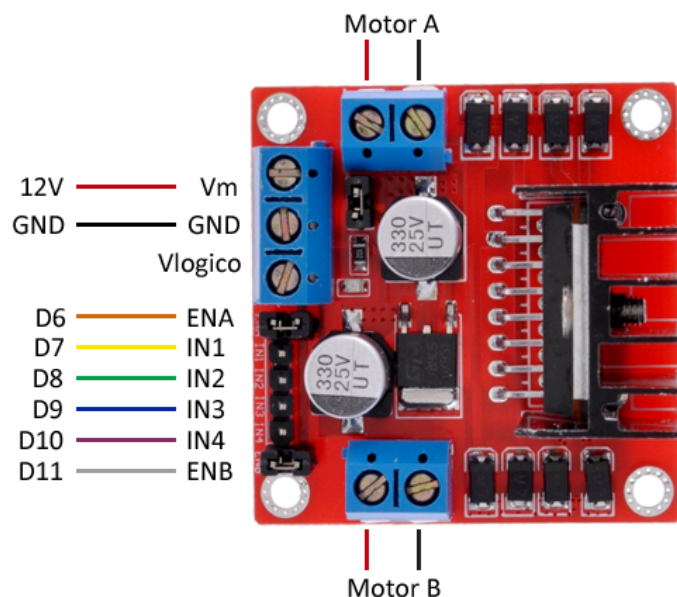
ACTUADORES

Motores

Con el Arduino podemos usar motores de corriente continua de juguete como en la figura, aptos para poner un ventilador y afines por su alta velocidad de rotación, pero lo normal en la robótica con Arduino es utilizar motores con reductoras para bajar las revoluciones como el motor amarillo :



Si quieres usar un motor, no se puede conectar directamente al Arduino (al menos de que sea de muy baja potencia) necesita una alimentación extra, como el LM298N y las pilas., o el B6612FNG o una shield específica.



CC-BY-NC [Luis Llamas](#)

Ejemplos de uso lo puedes ver en

- [Curso mClon con nanoArduino con el B6612FNG](#)
- [Curso Rover con Arduino con la shield L293D ESP-12E para el NODE MCU que internamente tiene el LM298N](#)
- [Curso Arduino con código con LM298N](#) y también con un transistor

Los motores pueden (y deben) de estar conectados a **salidas PWM** de tal manera que se pueda regular la potencia y por lo tanto la velocidad.

Servos

Un servo convencional es un motor donde fijamos el ángulo desde 0º a 180º, pero si queremos una rotación, existen servos rotatorios que simplemente tienen su velocidad de rotación controlada



Ejemplos de uso de servos:

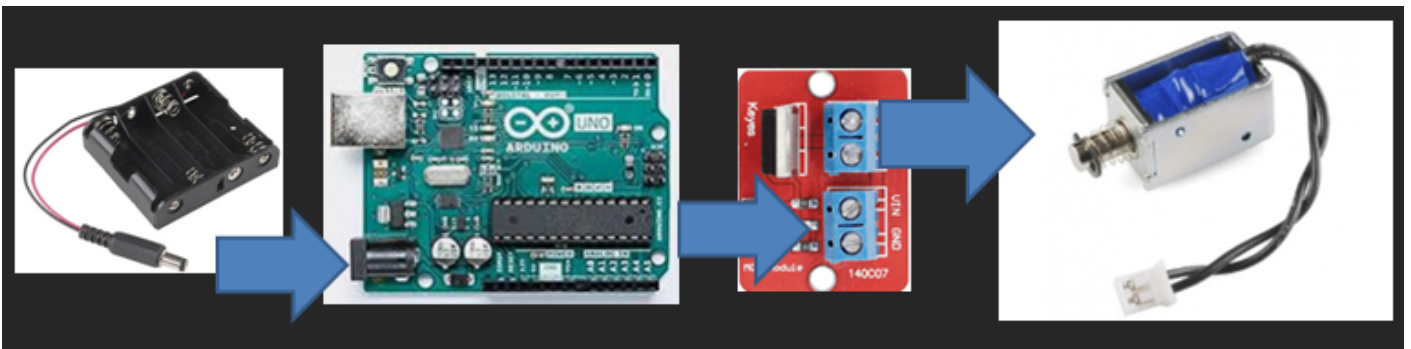
- [Curso Arduino con código](#)

- Curso brazo robótico del mClon con nanoArduino
- Apertura de barrera por ultrasonidos en curso Arduino con ArduinoBlocks
- Tractor entrando en el corral Arduino con EchidnaShield
- Apertura de puerta Domótica con Arduino

Electroimán

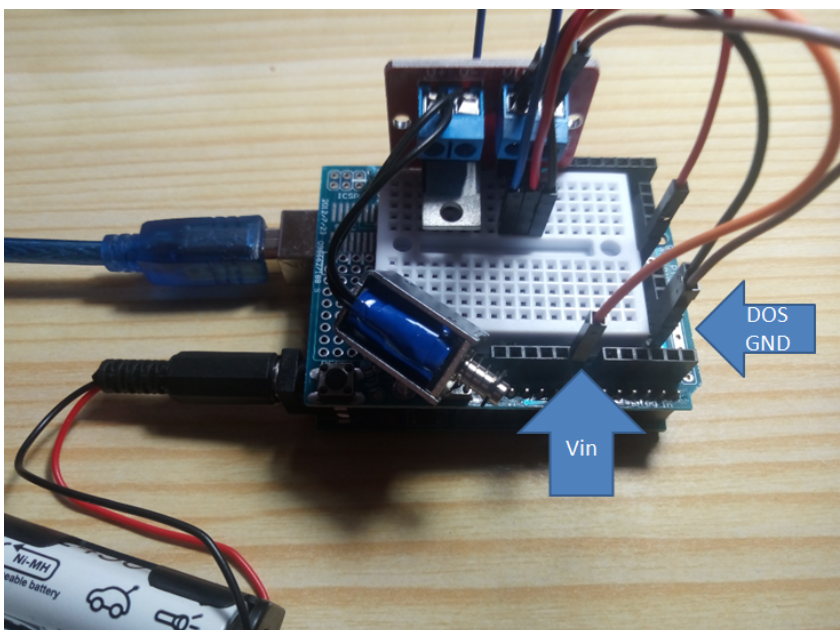
El electroimán no se puede conectar directamente, utilizaremos una etapa de potencia, por ejemplo el transistor el IRF520N que amplifique la señal del Arduino, pues las salidas de Arduino no tienen potencia para mover el electroimán

Para dar esa potencia utilizaremos otra fuente externa, unas pilas:



Las conexiones son :

- SIG del IRF520N a una salida digital por ejemplo D13
- VCC del IRF520N al 5V del ARDUINO
- Los dos GND del IRF520N a GND del ARDUINO
- V+ y V- del IRF520N al solenoide, da igual qué cable pues no tiene polaridad.
- VIN del IRF520N al VIN del ARDUINO (son los voltios de la pilas)



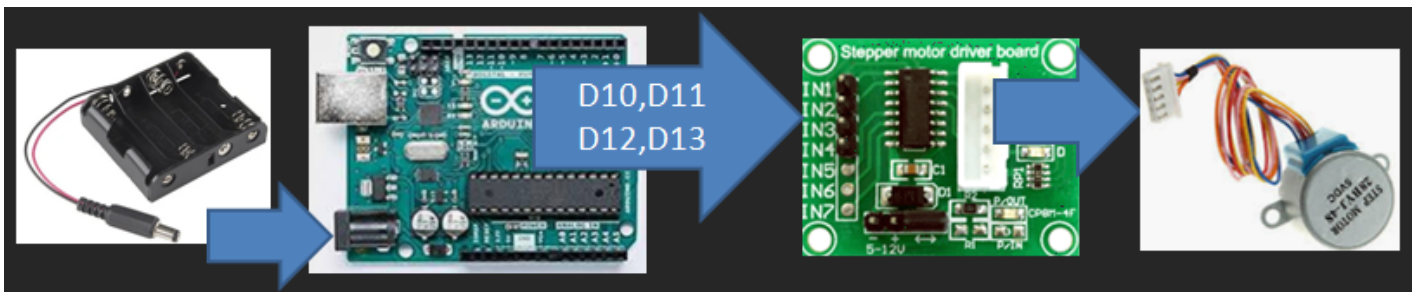
Motor paso a paso

Igual que el electroimán, necesitamos un controlador que nos de la potencia necesaria para mover el motor, el ULN2003



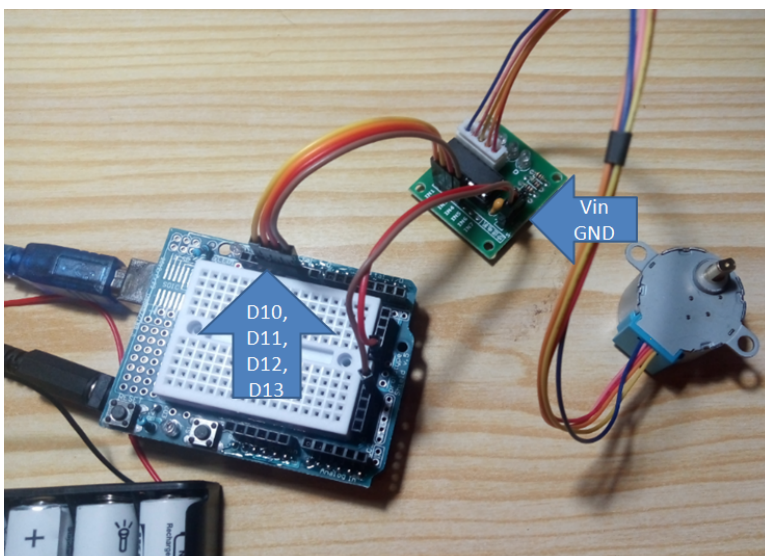
Imagen BY-NC-SA de Luis Llamas

También igual que el electroimán, necesitamos una potencia extra con las pilas:



Conexión :

- Cuatro pines digitales del ARDUINO al IN1,IN2,IN3,IN4 del ULN2003 por ejemplo D10,D11,D12,D13
- El conector blanco del ULN2003 al motor paso a paso
- El (+) del ULN2003 al Vin del ARDUINO
- El (-) del ULN2003 al GND del ARDUINO



La configuración más sencilla es la rotación simple en sentido horario (llamada fase1) :

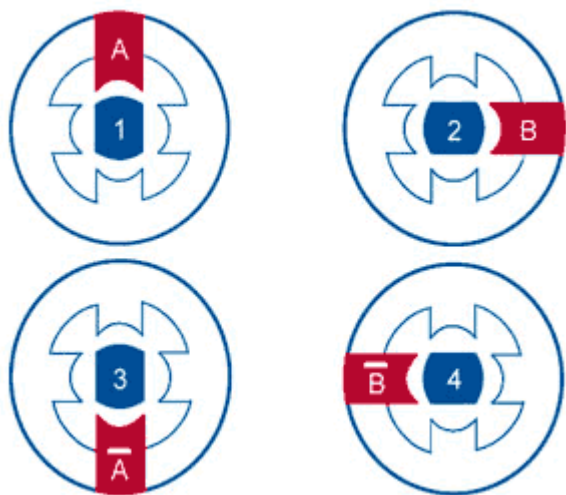


Imagen BY-NC-SA de [Luis Llamas](#)

Es decir:

Paso	IN1=D10	IN2=D11	IN3=D12	IN4=D13
Paso 1	ON	OFF	OFF	OFF
Paso 2	OFF	ON	OFF	OFF
Paso 3	OFF	OFF	ON	OFF
Paso 4	OFF	OFF	OFF	ON

Si has entendido este ejemplo serás capaz de realizar las configuraciones que desees.

OTRAS SALIDAS

No son actuadores pues no representan movimiento pero son también **salidas** del Arduino como los actuadores, luego lo vemos en esta página

Buzzer activo

Reproduce un tono de una frecuencia determinada por el fabricante cuando recibe un '1' digital. Su conexión es muy simple:

- **GND** a GND y **VCC** a 5V (ojo que están a los dos extremos, marcados en rojo y en negro)
- **OUT** o también señalado como I/O a un pin digital, por ejemplo D13



Ejemplos de uso:

- Domótica con Arduino: [Apertura puerta con mBlock](#)
- Domótica con Arduino [Apertura puerta con código](#)

Buzzer pasivo

La diferencia de un buzzer activo con el pasivo es que el pasivo hay que mandar la onda que se tiene que reproducir, como Arduino no puede reproducir onda puras (senoidales analógicas), se le envía ondas cuadradas con la frecuencia que se pretende reproducir. Como se puede ver en la figura, no tiene elementos auxiliares para reproducir un tono, es simplemente un altavoz.



Ejemplos de uso

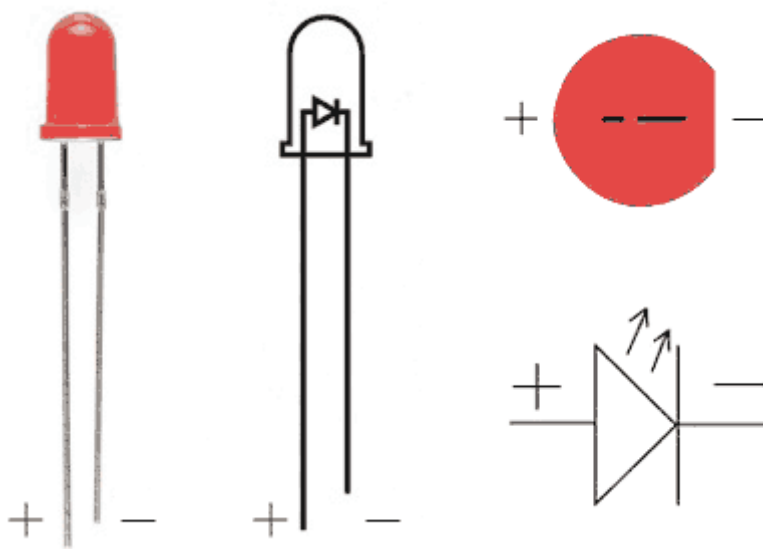
- [Timbre Arduino con Echidna](#)
- [Arduino blocks en el aula](#)

Leds y otros elementos visualizadores

Led normal

Un diodo es la unión de dos semiconductores P-N que sólo permite la corriente en un sentido. Un diodo LED es un diodo, que al pasar esta corriente emite una luz. Hay de 3mm y de 5mm de grosor, transparentes, traslúcidos y de colores (realmente son traslúcidos tintados)

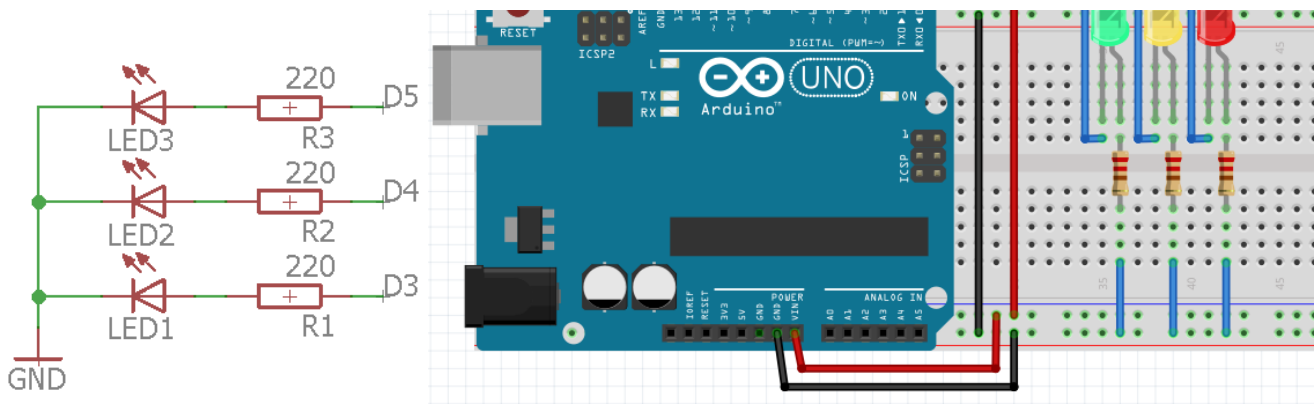
Tenemos pues que respetar su polaridad para que funcione:



Fuente Luis LLamas CC-BY-NC [Encender LED con Arduino](#)

Si alimentamos el LED con una tensión inferior a su tensión de polarización directa $V_d \pm 1,4V-3,8V$ el led no luce. si alimentamos con una tensión superior, la corriente que circula se dispara por lo que se rompe. Conclusión: Hay que poner una resistencia limitante. Para calcularla te recomiendo la página de Luis Llamas [Encender LED con Arduino](#)

Aquí puedes ver un ejemplo de semáforo [en el curso Arduino con código](#)



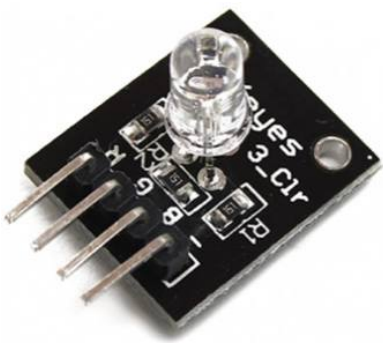
Se puede conectar directamente pues las salidas de Arduino tienen una limitación interna de 20mA como en este ejemplo <https://www.youtube.com/watch?v=EFFSLvIF9rY>

<https://www.youtube.com/embed/EFFSLvIF9rY>

Los LEDs igual que los motores pueden ser encendidos o todo o nada con salidas digitales 0 y 1 o salidas PWM de esta manera conseguimos luminosidades medias, en esta práctica con el potenciómetro se regula la intensidad de la luz de un LED

Led RGB

Se trata de un encapsulado que integra 3 leds de los tres colores básicos. Cada color básico se puede regular desde 0 a 255 de intensidad, luego tenemos 16 millones de tonalidades diferentes.



Un ejemplo de su uso lo puedes ver en

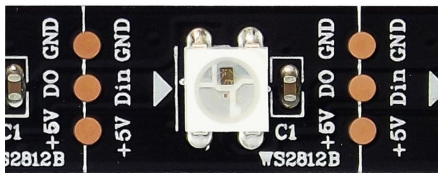
- Domótica con Arduino, interruptor crepuscular
- Intermitente RGB en Arduino con ArduinoBlocks

Existen LEDS RGB que se comunican unos a otros, ([ver este ejemplo en el curso mClon](#)) como las tiras de leds RGB que cambian de color todos simultáneamente.

Neopixel

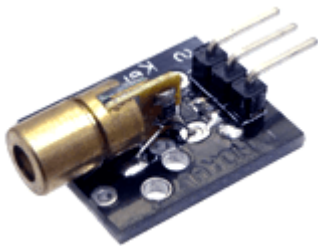
La cinta LED RGB se basa en el chip controlador WS2812B de Neopixel con el cual podemos tener el control de cada pixel y de cada color, ya sea rojo, verde y azul a través de un solo cable de datos.

Ve [cómo conectar Arduino con tiras led](#)



Láser

El diodo láser es un elemento motivador, barato y fácil de usar con el Arduino pues se activa digitalmente, si enviamos un 1 emite un rayo láser. Un ejemplo de uso lo puedes ver en la [Alarma por láser en Domótica con Arduino](#)



Si quieres saber más de este componente, te recomendamos [esta página de Luis Llamas](#).

Si tienes que comprar uno, es importante que no sea superior a 5mW, pues puede dañar permanentemente la retina del ojo [[+info](#)].

El modelo que proponemos es de **1mW**, no obstante, **EVITA SIEMPRE QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

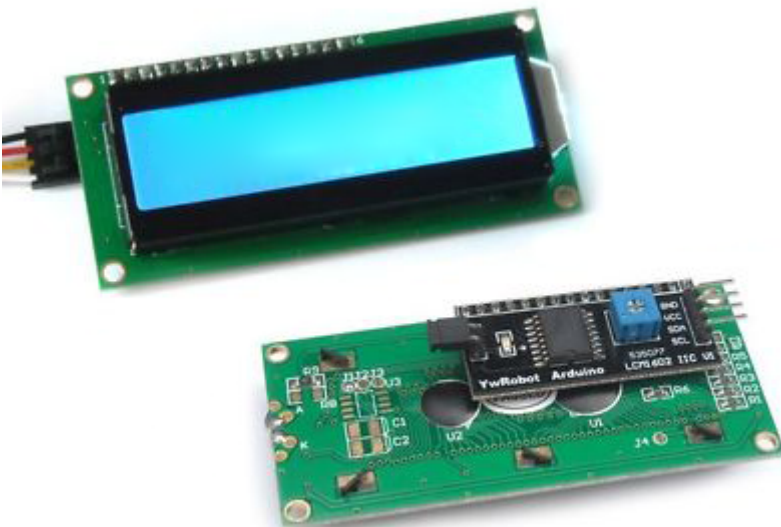


Display LCD

Si queremos visualizar datos de forma independiente del ordenador, existen matrices de led que puedes ver en los cursos de mBot o mClon pero lo mejor es una pantalla Liquid Crystal Display y la mejor forma de conectar esta interfaz es en serie (utilizando el **protocolo I2C**.)

Ejemplo de uso :

- en Arduino con código
- Arduinoblocks en el aula



No confundas Display LCD con matriz de LEDs , o una OLED

Aclarando conceptos: Lógica invertida

Igual que vimos en sensores, con las configuraciones Pull up y Pull down, las salidas de cualquier actuador **PUEDEN SER LÓGICA INVERTIDA**, mira en este NodeMCU (básicamente es un Arduino con wifi, aquí está conectado con el servidor Blynk) como funciona al revés, cuando apretamos, por lo tanto enviamos un uno a la salida, **se apaga la salida !!**

<https://www.youtube.com/embed/GLvTBdcfKAI>

Revision #12

Created 5 December 2023 18:45:50 by Javier Quintana

Updated 8 December 2024 01:39:45 by Javier Quintana