

Introducción

- [Objetivos y Contenidos](#)
- [Pensamiento computacional](#)
- [Hardware ¿Qué es Microbit?](#)
- [Hardware: Descripción de la placa micro:bit](#)
- [Sensores](#)
- [Actuadores](#)
- [Software para la micro:bit](#)
- [Software para la Smart home](#)

Objetivos y Contenidos

Objetivos

- Capacidad de desarrollar retos **STEAM** con un amplio nivel educativo: primaria hasta secundaria
 - Retos **STEAM** de nivel principiante con manejo de sensores y actuadores con lenguaje de bloques.
 - Retos **STEAM** de nivel medio medio en las comunicaciones Bluetooth.
 - Retos **STEAM** de nivel avanzado con código Python
- La capacidades se adaptan al perfil del docente
 - Para primaria lenguaje por bloques
 - Para secundaria lenguaje por bloques para comunicaciones Bluetooth y lenguaje código Python

Contenidos

- ¿Qué es Smart Home para micro:bit?
- Introducción sobre lenguajes, actuadores y sensores
- Jugando sólo **con la placa micro:bit** con lenguaje por **bloques Makecode**
- La maqueta Smart Home: **Montaje**
- Jugando con la **maqueta** y lenguaje por **bloques Makecode**
- Optativo: Jugando con **Bluetooth**
- Optativo Jugando fuerte con **Python**

Pensamiento computacional

¿Dónde se encaja este robot? ¿Se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta es la hoja de ruta que proponemos, no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta robótica:

Como se puede ver **SMARTHOME CON MICROBIT** tiene la ventaja de TENER LA MICROBIT con el centro de la programación y ampliarlo dentro de un proyecto maker con un amplio rango de prácticas con sensores y actuadores reales

Guía orientativa

https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco_/embed?start=trueloop=true&delayms=3000

Tenemos un **grupo Telegram Robótica Educativa en Aragón**,
<https://t.me/roboticaeducativaaragon>



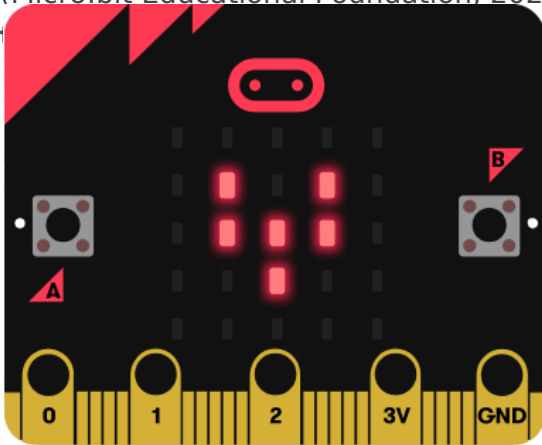


Hardware ¿Qué es Microbit?

La placa micro:bit

Micro:bit es un pequeño ordenador del tamaño de media tarjeta de crédito creado inicialmente por la BBC (British Broadcasting Corporation) en 2015 con el fin de promover el desarrollo de la robótica y el pensamiento computacional entre la población escolar de entre 11 y 15 años del Reino Unido.

Actualmente su uso está extendido entre **39 millones de escolares de más de 60 países** (Micro:bit Educational Foundation, 2023) con más de **5 millones de tarjetas distribuidas por**



Placa BBC micro:bit. Micro:bit Educational Foundation CC BY-

SA 4.0

El hardware es 100% libre, y está gestionado por una fundación sin ánimo de lucro fundada en 2016, la **micro:bit Educational Foundation**.

Micro:bit no es el primer proyecto de alfabetización informática de la BBC. Ya en 1981 produjo el **BBC Micro**, un ordenador de 8 bit programable mediante varios lenguajes de programación, entre ellos un lenguaje BASIC muy avanzado para la época. De este ordenador llegaron a venderse 1,5 millones de unidades, principalmente en el Reino Unido. Fue el ordenador con el que las escuelas británicas comenzaron la enseñanza de la informática.



BBC Micro. Dominio público.

Volviendo a micro:bit, el desarrollo del proyecto ha sido llevado a cabo por **29 socios tecnológicos** de primera línea. Por ejemplo, la implementación del Bluetooth corrió a cargo de la fundación propietaria de la marca, Bluetooth SIG, asociación privada sin ánimo de lucro.

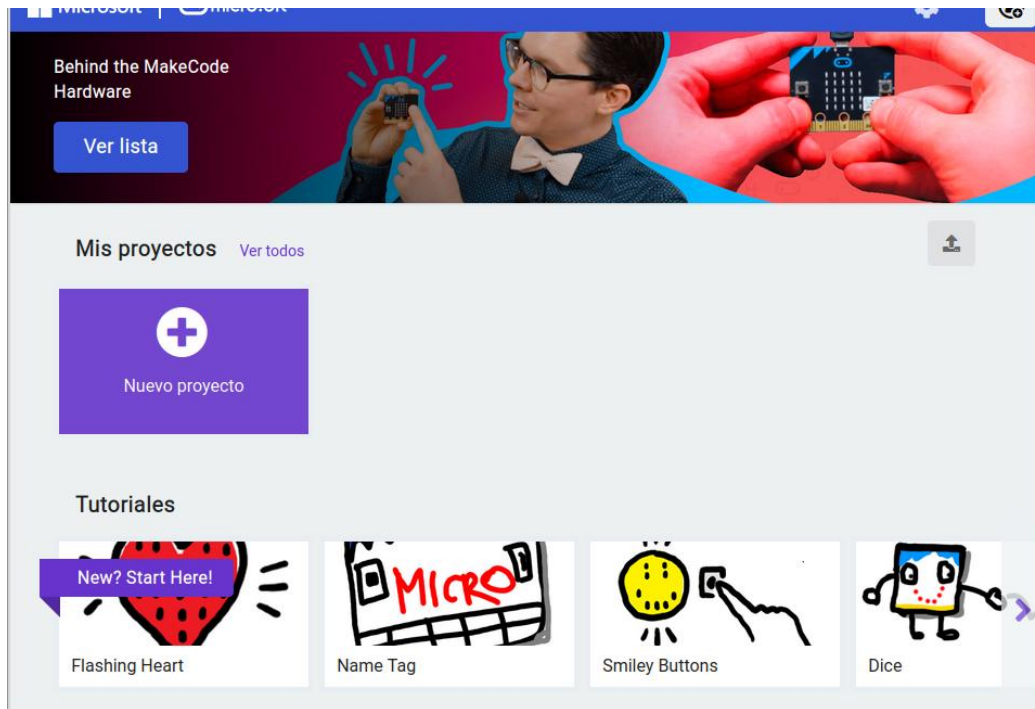
Micro:bit es **económico**; tanto las placas como los accesorios producidos por terceras empresas tienen un precio muy contenido.

El sistema destaca por su **alta integración de software y hardware**: basta un clic de ratón para cargar las librerías necesarias para que funcione cualquier complemento robótico, como sensores, pantallas, tarjetas de Internet de las Cosas, robots, casas domóticas, etc.

Programación de micro:bit

La programación de la placa se realiza desde un ordenador a través de un navegador cualquiera, estando disponibles **12 lenguajes de programación**. Dado su carácter abierto, existen múltiples soluciones de programación, aunque las más común es MakeCode.

El sitio **MakeCode** permite programar con bloques y también en Python y en Java, traduciendo de un lenguaje a otro instantáneamente. **No se necesita ningún registro** en la plataforma para poder programar. Nuestros programas se guardan en el servidor de MakeCode siempre que no utilicemos nuestro navegador de Internet en modo incógnito o que no tengamos activado el borrado automático de cookies.



Sitio web de Make Code.

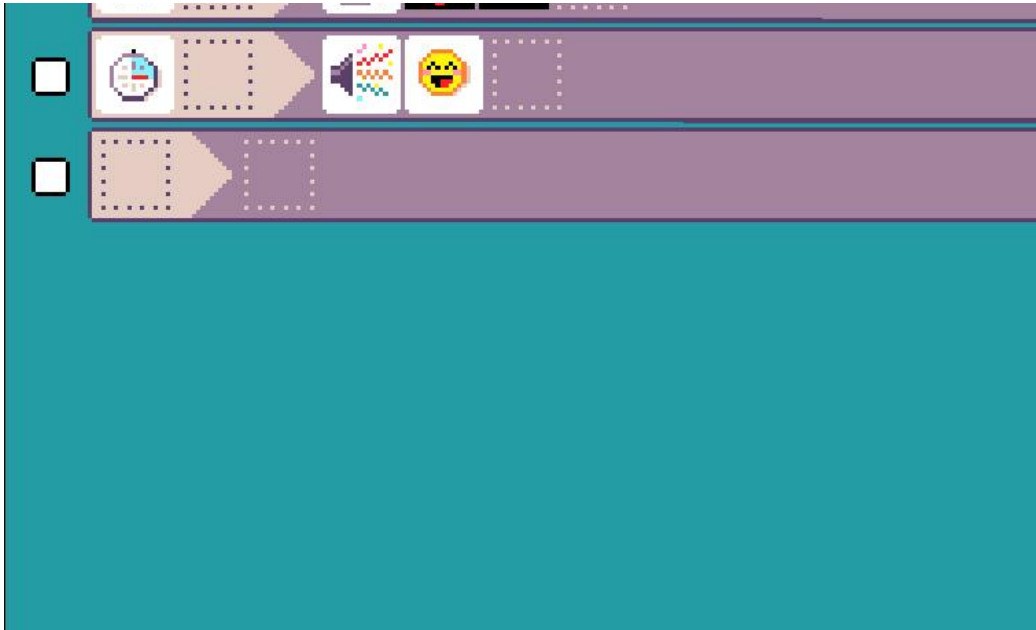
Captura de pantalla

Los **programas también pueden guardarse descargados en el ordenador** compilados en código de máquina. Al subir de nuevo el programa en el editor, se realiza una decompilación automática al lenguaje de bloques, Python o Java. Los programas guardados en código de máquina se pueden cargar directamente en micro:bit, que en el escritorio de un ordenador aparece como una simple unidad USB.

MakeCode contiene además **múltiples recursos** como tutoriales, vídeos, fichas de programación, cursos para el profesorado, ejemplos y propuestas de proyectos y experimentos en varios idiomas.

Otra solución muy usada para programar micro:bit es **MicroPython**, creada por Python Software Foundation, otra organización sin ánimo de lucro.

MicroCode permite que los más pequeños, a partir de los 6 años de edad, programen micro:bit mediante un sistema de fichas dispuestas en líneas de acción. Están disponibles un tutorial introductorio en 20 idiomas, una guía del usuario y muchos ejemplos. El proyecto es de código abierto.



Ejemplo de programa con

MicroCode. Captura de pantalla

Micro:bit también es programable en **Scratch** con sólo añadir una extensión al editor.

Todos los entornos de desarrollo descritos disponen de un **simulador de micro:bit**, por lo que ni siquiera resulta necesario disponer de una tarjeta física para aprender a programar.

Características de micro:bit

Características de **micro:bit V2**:

- Procesador de 64 MHz.
- 512 KB de RAM Flash y 128 KB de RAM.
- Matriz de 5 x 5 LED rojos.
- Dos pulsadores mecánicos y un tercer pulsador de apagado y reset.
- Pulsador táctil en el logotipo.
- Micrófono y altavoz.
- Acelerómetro y compás.
- Sensores de luz y de temperatura.
- Bluetooth de bajo consumo.
- Alimentación a 3 V o por USB.
- 25 pines de entradas y salidas.
- GPIO, PWM, I2C y SPI.



- 200 mA disponibles en las salidas para alimentar accesorios.

La placa micro:bit que se encontraba disponible en el mercado antes de la introducción de la V2 era la denominada **V1.5**. Se distingue de la V2 en que carece de micrófono, de altavoz y de pulsador táctil en el logo. Internamente tiene un procesador menos potente y menos memoria RAM. Tampoco tiene piloto de encendido ni puede apagarse desde el pulsador de reset.

<https://sketchfab.com/models/115e78b4f2b44f798da727888a0ecc6c/embed>

[Microbit 3d](#) by [GmedranoTIC](#) on [Sketchfab](#)

Hardware: Descripción de la placa micro:bit

Presentación de micro:bit

Micro:bit es una **pequeña placa de circuito impreso** de 52 mm x 42 mm, dimensiones que corresponden aproximadamente al tamaño de media tarjeta de crédito. El circuito dispone sus componentes electrónicos al aire, sin ninguna carcasa, y se encuentran soldados por las dos caras de la placa.

La placa **micro:bit es segura** para su uso con el alumnado de 8 a 14 años (Micro:bit Educational Foundation, 2023b). Ha sido ampliamente probada y cumple con todos los requisitos de seguridad y compatibilidad medioambiental exigibles. En el siguiente enlace se muestran todas las normas y estándares que cumple la tarjeta:

<https://microbit.org/compliance/>

El siguiente vídeo, pensado para personas sin conocimientos de robótica, proporciona una primera idea acerca de las posibilidades y del manejo de la placa micro:bit.

<https://www.youtube.com/embed/u2u7UJSRuko>

Kits

Micro:bit se encuentra disponible comercialmente como placa simple (*micro:bit single*) o como parte de un kit. El coste medio en el año 2023 de una placa es de unos 22 €. Por un precio un poco más elevado se puede adquirir un kit más completo, cuyo contenido está pensado como un **equipo de iniciación**.



Contenido de un kit de inicio.

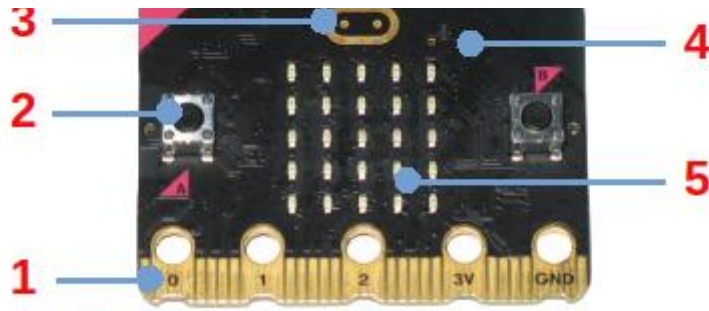
Elaboración propia

El kit presentado en la imagen contiene:

- Una **placa micro:bit** versión 2. El color de la decoración de la placa puede ser magenta, azul, verde o amarillo.
- Una pequeña **guía icónica de inicio rápido**.
- Una hoja con **recomendaciones de seguridad** en varios idiomas.
- Un **cable corto micro USB** para conectar la placa al ordenador.
- Una cajita **portapilas** con conector JST.
- Dos **pilas alcalinas** IEC LR03 (AAA).

Componentes electrónicos integrados en la placa

El anverso de la placa en su versión 2 tiene el aspecto mostrado de la figura.



Anverso de la placa BBC micro:bit.

Elaboración propia

La parte marcada con el número **1** es una **fila de contactos o pines** que permiten conectar la placa a múltiples accesorios, como altavoces, robots, sensores o actuadores. Dispone de 25 pines, 5 de ellos de anillo. Estos últimos están pensados para conectar cables por medio de pinzas o bananas.

El número **2** corresponde al **pulsador A**. En el lado derecho de la placa existe otro **pulsador B**. Ambos pulsadores son programables y permiten al usuario comunicarse con micro:bit.

El número **3** apunta al **pulsador táctil**. Se trata de una novedad de la versión 2 de micro:bit. El pulsador puede distinguir entre pulsaciones rápidas y pulsaciones prolongadas.

El sensor marcado con el número **4** es un **micrófono**. Es otra novedad de la versión 2 de la placa. Aunque puede grabar sonido, las grabaciones resultan de una calidad muy pobre. Es por ello que el micrófono se usa sobre todo para medir el ruido ambiental. Un **diodo LED de color rojo** indica que el micrófono está en funcionamiento.

La **matriz de 25 LED rojos** está marcada con el número **5**. Se usa para representar mensajes de texto e iconos. Su intensidad es ajustable. Sirve también como **sensor del nivel de luz** ambiental.

El reverso de la placa es un poco más complejo. Aquí se encuentran soldados los dispositivos que permiten el funcionamiento de micro:bit. Se trata del procesador, de los sensores y de los circuitos integrados que proporcionan funcionalidades como la comunicación USB o Bluetooth.



Reverso de la placa BBC micro:bit.

Elaboración propia

El circuito número **6** es un **acelerómetro y un compás**. Este dispositivo permite medir aceleraciones y giros, así como campos magnéticos.

El componente **7** es un pequeño **zumbador** que puede reproducir tonos y sonidos. Su potencia es muy baja, pero micro:bit puede también conectarse a un altavoz externo.

El procesador es el circuito marcado con el número **8**. Además contiene un **sensor de temperatura**.

La **antena Bluetooth** se indica con el número **9**. Gracias al Bluetooth incorporado, micro:bit puede enviar y recibir datos por radio a cortas distancias.

Cuando el **LED de color rojo 10** está encendido indica que la placa está conectada a una fuente de alimentación. Micro:bit puede ser alimentado por un ordenador o por una batería externa través de un cable USB o bien por un par de pilas.

La placa puede comunicarse con un ordenador mediante el **conector micro USB 11**. Mediante esta conexión micro:bit puede ser programado y también puede enviar al ordenador los datos que recoja con sus sensores.

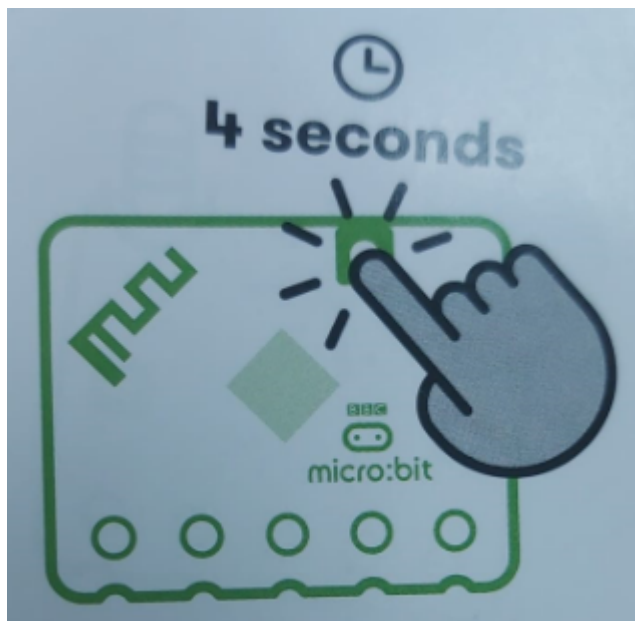
El **LED ámbar 12** parpadea cuando la placa está conectada a un ordenador y se encuentra emitiendo o recibiendo datos.

El **pulsador 13** se usa para **reiniciar** el programa contenido en la placa. Una pulsación prolongada apaga micro:bit, pero sólo si la placa es de la versión 2.



El **conector JST 14** permite conectar el portapilas para alimentar la placa con dos pilas alcalinas de 1,5 V. También se pueden usar pilas recargables; las clásicas pilas recargables de NiMH o níquel-hidruro metálico que se venden en algunos supermercados hacen que tanto micro:bit como sus accesorios funcionen sin problemas.

En la versión 2 existe un botón para apagar



Sensores

Esta sección es una visión rápida de las posibles **entradas** de los robots.

NO LEAS TODOS SINO LOS QUE TIENE TU ROBOT

Un poco de teoría...

Cualquier sistema de control podríamos decir que funciona de una manera similar a un ser humano, salvando las distancias. Nosotros recibimos la información del mundo exterior gracias a nuestros sentidos (oído, olfato, gusto, vista y tacto), nuestro cerebro procesa esa información y a través de nuestros músculos o de nuestra voz realizamos diferentes acciones. Pues lo mismo sucede con los sistemas de control, reciben información del exterior gracias a los diferentes **SENSORES**, procesan esa información en sus PLACAS CONTROLADORAS (sus cerebros) tales como Arduino y dan una respuesta utilizando sus diferentes **ACTUADORES**.



Un sensor es un objeto capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas. Los sensores o periféricos de entrada nos permiten obtener información del mundo real para utilizarla desde el programa de Arduino.

En la actualidad la cantidad de sensores disponibles es tan extensa como las variables que queramos medir, desde sensores de temperatura, humedad, luminosidad,... hasta acelerómetros,



giroscopios, GPS,... pasando por detectores de gases, de pulsos cardiacos, sensores de efecto HALL,...

Tipos de sensores

- **DIGITAL:** un sensor digital sólo tiene dos estados: activado/desactivado, ON/OFF, 1/0, Alto/Bajo, ... En este caso conectaremos el sensor a una de las entradas digitales de Arduino para leer el estado.

Ejemplo: un pulsador es un tipo de sensor sencillo que sólo nos da dos estados, “pulsado o no pulsado”. Conectado a la placa Arduino debe generar 0v en reposo y 5v al pulsarlo. De esta forma desde el programa de Arduino podremos leer el estado del botón.



- **ANALÓGICO:** el sensor nos puede dar un rango de valores, normalmente se traduce en un valor de tensión o de corriente variable en función de la señal captada al sensor. En este caso conectaremos el sensor a una de las entradas analógicas de Arduino (A0,..., A5). El rango de entrada será una tensión entre 0v (GND) y 5v.

Ejemplo: Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Su valor varía entre 0 y 5 v. la cantidad de valores que pueden leer las entradas analógicas de Arduino son de 10 bits es decir 1024 valores. De tal modo que $0 = 0 \text{ v.}$ y $1023 = 5 \text{ V.}$

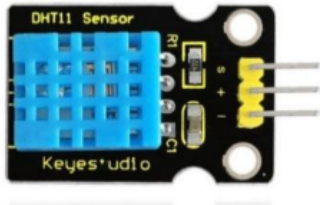


- **DATOS:** el sensor ofrece su información a través de una interfaz de comunicación. La forma de comunicación puede ser por sistemas estándar como **I2C** o SPI o algunos sensores usan su propio protocolo para codificar la información y debemos realizar desde el software la decodificación correcta para interpretar los datos del sensor (normalmente



los desarrolladores de este tipo de sensores ofrecen una librería software para Arduino que hace todo el trabajo).

Ejemplo: el sensor DHT11. Por un solo pin envía los datos de temperatura y humedad.



Sensores modulares.

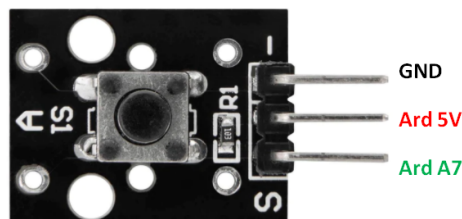
En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

Sensor pulsador

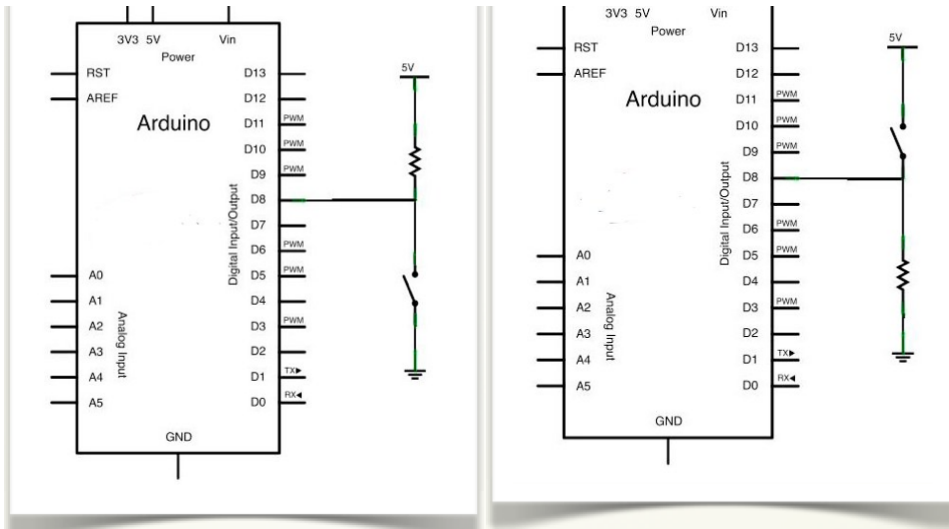
Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso

• en el robot mClen



Otra manera más "barata" de sustituir este módulo pulsador es poner un pulsador normal y una resistencia ($\pm 10k$), al pulsar se produce una entrada en el Arduino, hay dos configuraciones, que al pulsar se emita un 0 lógico (configuración **Pull up**) o que al pulsar emita un 1 lógico (configuración **Pull down**) [¿Por qué hay que poner una resistencia?](#)



Lo "normal" es que al pulsar se emita un '1' configuración **Pull down**, pero hay pulsadores que funcionan **Pull up** y los llaman **lógica invertida**, por eso en la programación por bloques podemos encontrar esto:



Sensor Táctil Capacitivo.

Este pequeño sensor puede "sentir" a las personas y el tacto y la retroalimentación de metales a un nivel de voltaje alto / bajo. Incluso aislado por alguna tela y papel, todavía puede sentir el tacto. Su sensibilidad disminuye a medida que la capa de aislamiento se hace más gruesa. En nuestra opinión lo preferimos frente al *Sensor pulsador* pues es muy económico, duradero y fiable.

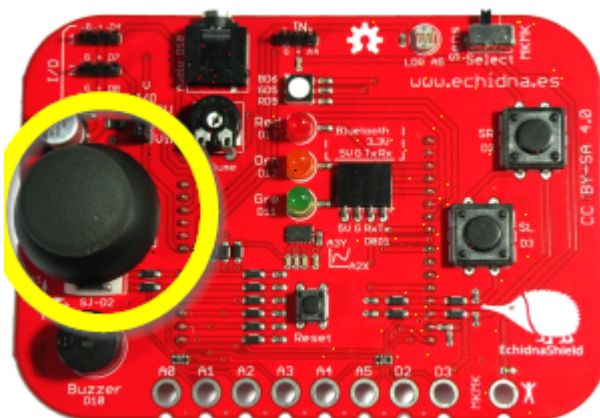
Un ejemplo de uso en

- [Disparo láser en Arduino con ArduinoBlocks](#)
- [Apertura de puerta en Domotica con Arduino](#)



Potenciómetro y joystick

Un potenciómetro es una resistencia variable, es decir, cambia de valor mecánicamente, lo tenemos en multitud de dispositivos. El **joystick** es internamente dos potenciómetros con un pulsador integrado en un solo mando.





Este sensor es **analógico**, su salida puede ser cualquier valor entre Vcc y GND (si está en divisor de tensión como en la placa Edubásica no llega a esos valores extremos), por lo tanto hay que conectarlo a una entrada analógica de Arduino y como cualquier entrada analógica, proporcionará valores entre 0 y 1023.

Ejemplos de uso:

- Arduino con código: [Mapeo del potenciómetro](#)
- Arduino con código: [Regular la luz con potenciómetro](#)
- [Arduinoblocks en el aula](#)
- En [Arduino con Echidna](#), con joystick
- [Domótica con Arduino con joystick](#)

Sensor Fotocélula LDR.

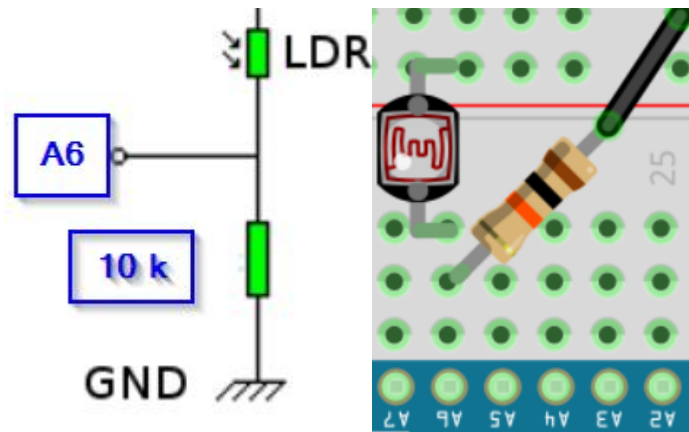
El uso de fotocélulas es muy común en nuestras vidas, las encontramos en el encendido automático de farolas, apertura de puertas,... La fotocélula es un semiconductor. Es ampliamente utilizado en campos de interruptores de control automático como cámaras, luces solares de jardín, lámparas de césped, detectores de dinero, relojes de cuarzo, tazas de música, cajas de regalo, mini luces nocturnas, interruptores de control de luz y sonido, etc. Es un sensor analógico dando valores entre 0 y 5V y como entrada analógica de un Arduino se traduce en un rango de 0 a 1023 valores.

Un ejemplos de uso :

- [el interruptor crepuscular del curso Arduino con ArduinoBlocks](#)
- [Medir la luz en Rover con Arduino](#)
- [Medir la oscuridad en Arduino con mBlock](#)
- [Hinchar un balón en Arduino con mBlock](#)

Una manera más económica de montar este sensor es utilizar una resistencia y un LDR:

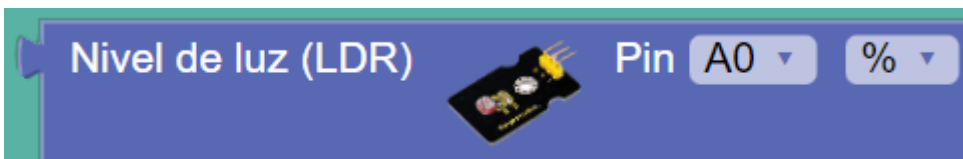
- El LDR cuando más oscuridad, más resistencia
- En una configuración **PULL DOWN**, cuanto **más** luz, la resistencia del LDR baja, por lo tanto **más** tensión en A6



Los módulos LDR que se venden suelen esta configuración Pull down, es decir, cuanto **más** luz, **más** tensión:



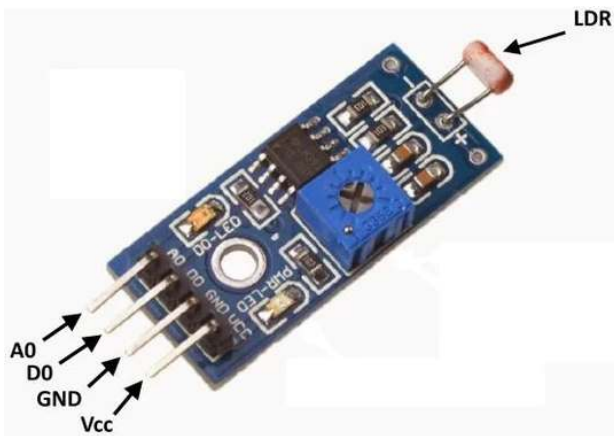
La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.



Hay módulos LDR ya montados, que tienen componentes **activos** es decir, llevan circuitos electrónicos, transistores que amplifican etc... y dan la salida **digital** con un potenciómetro para definir el rango de luz que cambia de estado lógico. Puedes ver en la figura que tiene una salida digital **D0**.



O hay [algunos que tienen 4 pines](#) como en la figura que ofrecen las dos cosas: salida analógica **A0** y digital **D0**.



Nosotros aconsejamos el divisor de tensión por tres razones: más barato, no implica gran circuitería y es visible su funcionamiento frente a estos encapsulados.

Sensor de Ultrasonidos.

Es un sensor digital de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 350 cm. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno.

No es un sensor preciso, con una ligera inclinación de la superficie ya da lecturas erróneas pero es muy barato

El más común es el [HC-SR04](#) que tiene 4 pines de conexión: **VCC** **Trig** (Disparo del ultrasonido)

Echo (Recepción del ultrasonido) y **GND** aunque en algunos modelos como el de [ElecFreaks](#) tiene 3 pines. Integra Trig y Echo en uno sólo.

La distancia se calcula con esta fórmula:

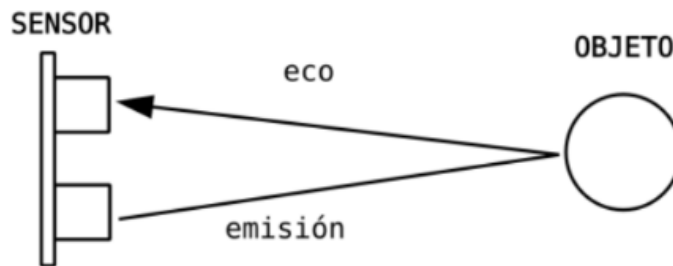
Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) * (V.Sonido 34000 en cm/s)} / 2

Si programas en código, tienes que utilizar la fórmula anterior, previamente tienes que programar



el cálculo del tiempo entre una emisión de un pulso en Trg y la respuesta en Echo.

loques, no es necesario, seguro que hay un bloque que lo hace todo



Ejemplos de uso:

- [Alarma en Domótica con Arduino](#)
- [Piano invisible en Arduino con ArduinoBlocks,](#)
- [Sensor parking en Arduino con ArduinoBlocks](#)
- [Piano invisible en Arduino con mBlock](#)
- [Sensor parking en Arduino con mBlock](#)
- [Sensor de distancia de ultrasonidos con Picobricks](#)

Sensor DHT11 (Temperatura y Humedad).

Este sensor de temperatura y humedad **DHT11** nos permite [determinar las zonas de confort](#) para un rango de temperaturas entre 0°C y 50°C con un error de $\pm 2^\circ\text{C}$ y un rango de humedad entre 20 y 90 % $\pm 5\%$. Una salida digital para dos variables cómo lo hace? Tiene dentro un pequeño microprocesador que lanza por el bit de datos 40 bits en serie, los 16 primeros son la humedad (en BCD) y los 16 restantes es la temperatura (en BCD) los 8 restantes son de comprobación

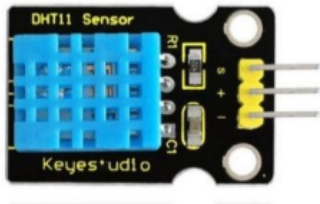
Checksum (en binario) como la letra del DNI. Por ejemplo 0100 0111 0000 0011 0001 1001 0000 0000 0001 1000 es 0100 0111 0000 0011 = 47.03% de humedad y 0001 1001 0000 0000 = 19.00°C y la comprobación es la suma de 4+7+0+3+1+9+0+0=24=11000

Ejemplos de uso:

- [Medir H y T con Blink en Rover con Arduino](#)
- [Estación meteorológica Arduino con Arduinoblocks](#)

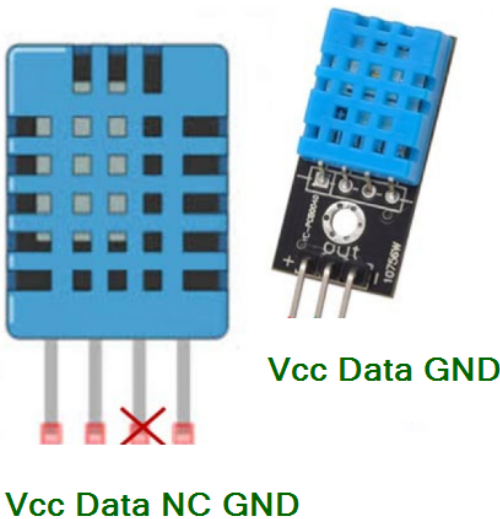


- [Arduinoblocks en el aula](#)
- [SMART HOME con Micro:bit](#)

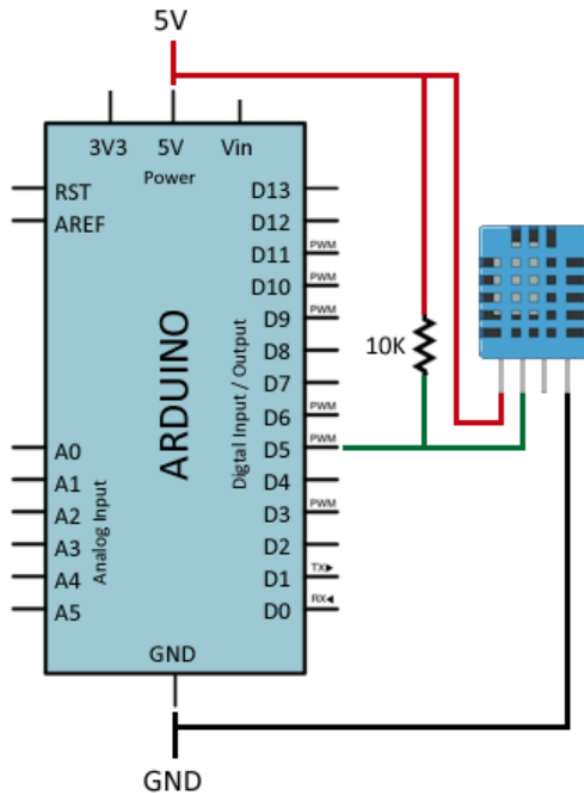


No es un sensor con gran sensibilidad, pero para propósitos educativos cumple sus funciones. Por dentro tiene una resistencia NTC que decrementa su resistencia si aumenta la temperatura. Hay otros que van al revés, los PTC. Tanto los NTC como los PTC se llaman **thermistores**. Para la humedad, mide la capacidad de un condensador que es sensible a la humedad, o sea, un **sensor capacitivo**.

Tenemos dos opciones comerciales: **Encapsulado** que lo tienes preparado para conectar la alimentación y leer por el pin de datos, o **sin encapsular**, que hay que colocar una resistencia de aproximadamente 10k entre Vcc y Data

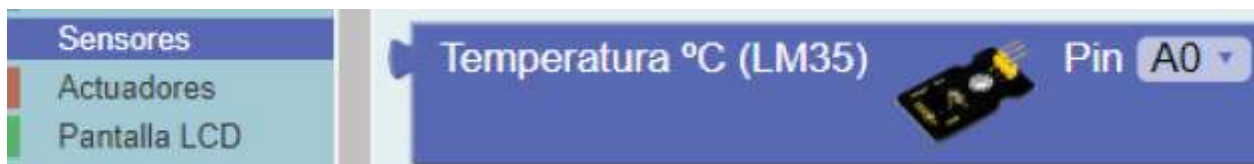


Ejemplo de uso de un DHT11 sin encapsular:



Fuente Luis LLamas CC-BY-NC-SA <https://www.luisllamas.es/arduino-dht11-dht22/>

Si queremos mejorar la sensibilidad, podemos utilizar el **DHT22** que es igual pero de color blanco y más caro. Si lo que queremos es sólo la temperatura es mejor utilizar el **LM35D** que tiene un rango de temperaturas desde 0°C a 100°C con una sensibilidad de 2mV/°C



Es un sensor bastante mediocre, si necesitas una precisión el doble, te recomendamos el DHT22 que funciona exactamente igual pero es de color blanco y más caro ~3€. Ver <https://www.luisllamas.es/arduino-dht11-dht22/>

Sensor IR

Es un sensor para distancias cortas hasta 2cm y no da la distancia, simplemente si hay o no hay obstáculo, pero son muy baratos, unos 0.30€. [Aquí tienes un ejemplo de evita obstáculos en un rover marciano con Raspberry](#) Para saber más te recomendamos [esta página de Luis Llamas](#)



<https://sketchfab.com/models/6ad4f3afb83940fea95cd3846aa68a18/embed>

[IR Sensor Module for Arduino Projects | 3D Model](#) by [Veer AI](#) on [Sketchfab](#)

Sensor Llama

Este sensor de llama se puede utilizar para detectar fuego u otras luces cuya longitud de onda se encuentra entre 760 nm ~ 1100nm.

Un ejemplo de su uso:

- [Alarma por fuego en Domótica con Arduino](#)

Radiación		Longitud de onda λ
Ultravioleta 100-400 nm	ultravioleta C	100 nm – 280 nm
	ultravioleta B	280 nm – 315 nm
	ultravioleta A	315 nm – 400 nm
Visible 400-780 nm	violeta	400 nm – 455 nm
	azul	455 nm – 490 nm
	verde	490 nm – 570 nm
	amarillo	570 nm – 590 nm
	anaranjado	590 nm – 620 nm
	rojo	620 nm – 780 nm
Infrarroja 780nm-1mm	infrarroja A	780 nm – 1400 nm
	infrarroja B	1400 nm – 3000 nm
	infrarroja C	3000 nm – 1 mm



Sensor de Gas (MQ2).

Detecta gases inflamables : GLP, I-butano, propano, metano, alcohol, hidrógeno, humo... con más sensibilidad en algunos que en otros. Siempre detecta el conjunto. Son usados en electrónica de consumo y mercados industriales.

- **Sensibilidad** Tiene alta sensibilidad y se puede ajustar girando el potenciómetro.
- **Tiempo de respuesta:** Internamente posee un calentador para aumentar su temperatura y que estos gases reaccionen con la resistencia interna que tiene, por lo



tanto tardan algo en responder la primera vez que se conectan, incluso horas en algunos modelos. Una vez calentados son rápidos en la respuesta.

- **Tipo de salida:** Analógico pero si tiene 4 pines como el de la figura, incorpora un pin digital.
- Ejemplos de uso:
 - [Smart Home Microbit](#)
 - [Smart Home ESP32](#)



Sensor de humedad de suelo.

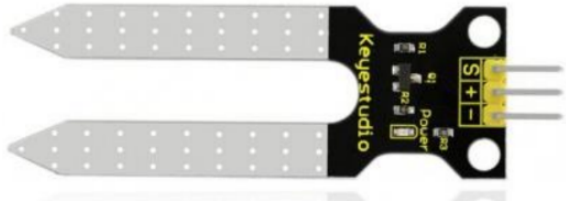
“ Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo, pero tampoco es necesario para controlar un sistema de riego. Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023.

Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

Se puede utilizar este sensor para hacer un dispositivo de riego automático, puede detectar si las plantas “tienen sed” y evitar que se marchiten.

La corriente de trabajo del sensor es menor de 20mA. El voltaje de salida es de 0V (en el aire) a 2,3V (totalmente sumergido en agua).

- [Smart Agriculture Kit micro:bit](#)



Sensor de humedad.

Este sensor **analógico** está diseñado para identificar y detectar la presencia de agua y su cantidad. Puede servir para detectar el nivel de agua, para disparar una alarma en caso de una fuga de agua, también para hacer un limpiapalabrisas automático.... puedes ver un ejemplo de uso en :

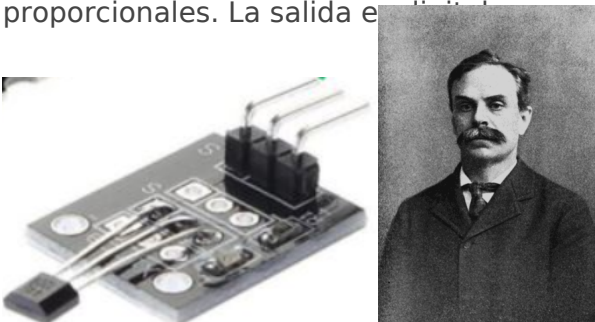
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

Mide el volumen de agua caída a través de una serie de rastros de cables paralelos expuestos.



Sensor de efecto Hall.

Este es un sensor de inducción magnética. Detecta los materiales magnéticos dentro de un rango de detección de hasta 3 cm. El rango de detección y la fuerza del campo magnético son proporcionales. La salida es digital.

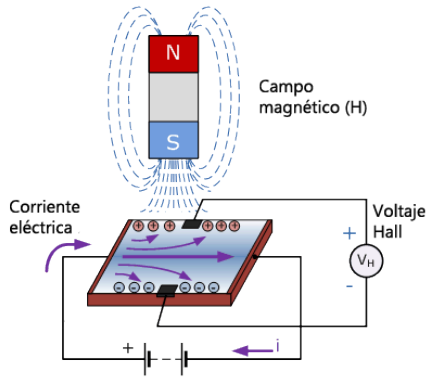


Sensor Hall.

Edwin Helber Hall De Desconocido - [Popular Science Monthly Volume](#)

[64, Dominio público](#)

[Edwin Helbert Hall](#) descubrió en 1879 que en presencia de un campo magnético, un conductor que conduzca una corriente se le producía un campo eléctrico porque las cargas eléctricas se desviaban de su trayectoria principal, nuestro sensor simplemente mide ese campo eléctrico:



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

El sensor tiene un led de color rojo que indica que hay una lectura de campo magnético. Un ejemplo de uso lo puedes ver aquí: [medir rocas magnéticas con el Rover con Arduino](#)

Sensor inclinación

Este sensor funciona al hacerle vibrar, emitiendo una señal digital de todo o nada. El módulo del sensor viene provisto de un potenciómetro para poder regularlo.



Sensor de golpe

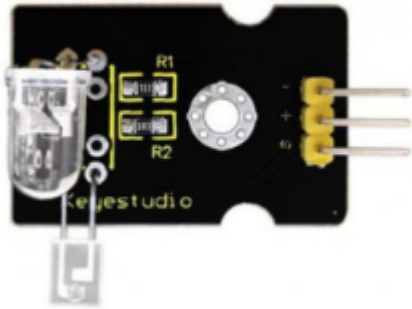
Es un sensor digital que al ser golpeado este sensor envía una señal momentánea.



Sensor de pulso cardíaco.

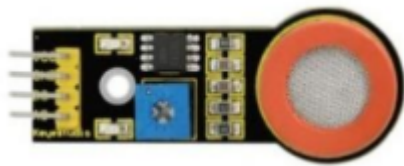


Este módulo utiliza un LED infrarrojo (IR) ultrabrillante y un fototransistor para detectar el pulso en el dedo. Principio de funcionamiento: Se debe colocar el dedo entre el LED infrarrojo ultrabrillante (parte superior) mientras que el fototransistor, que queda en el otro lado, recoge la cantidad de luz transmitida. La resistencia del fototransistor variará levemente a medida que la sangre pase a través de su dedo.



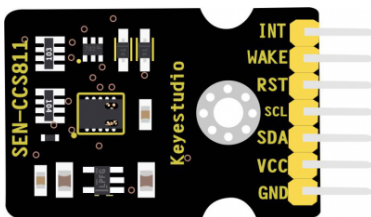
Sensor de Alcohol.

Este sensor de gas analógico MQ-3 es adecuado para detectar alcohol. Se puede usar en un analizador de aliento. También tiene una alta sensibilidad al alcohol y baja sensibilidad a la bencina (éter de petróleo). La sensibilidad se puede ajustar con el potenciómetro.



Sensor de CO2

Hay sensores que utilizan **el protocolo I2C**, este protocolo permite conexiones serie y pueden compartir el mismo cable pues cada elemento tiene una dirección diferente. Esto lo veremos en el Display LCD. Se identifican por los pines SDA y SCL

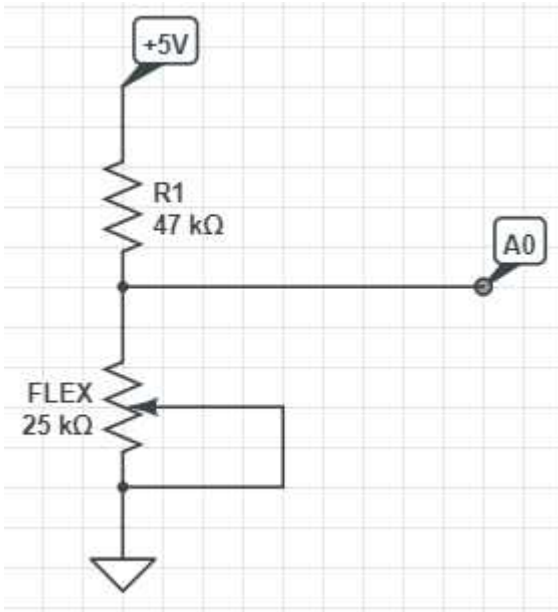


Resistencia Flex

Es una resistencia que cuanto más se dobla más resistencia ofrece, desde 25k hasta 125k

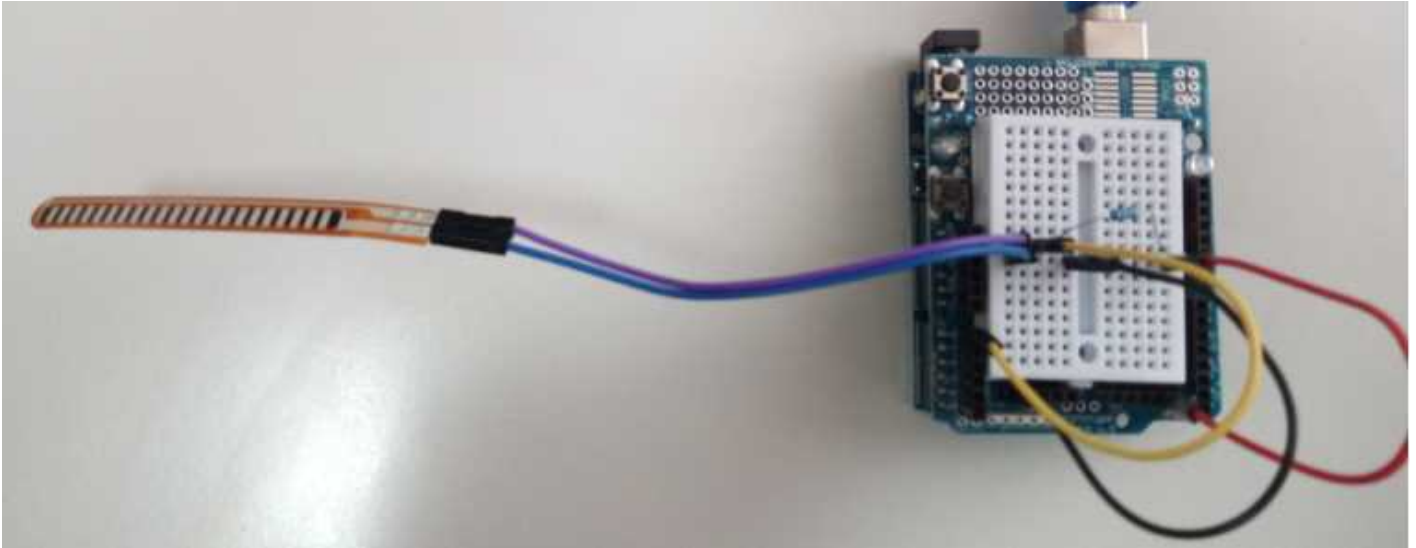


Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable, esa tensión es variable y ya tenemos la entrada **analógica**:



Es decir:

- La resistencia entre masa GND del ARDUINO (cable negro) y un punto en la placa protoboard
- ese punto medio conectarlo a una entrada analógica, por ejemplo A0 (cable amarillo)
- Una resistencia de valor parecida a la Flex de decenas de K entre ese punto y +5V (cable rojo en la foto)



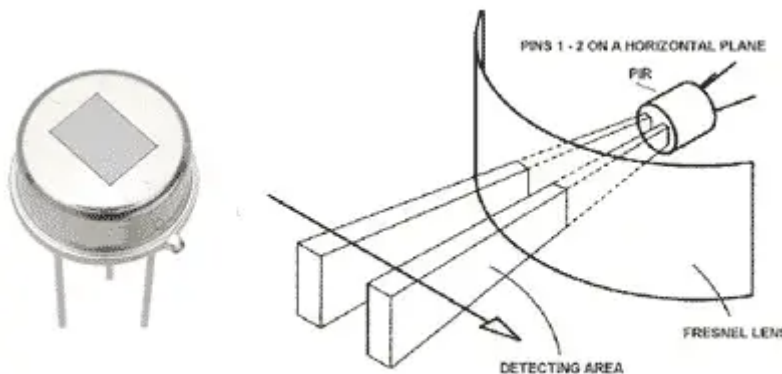
Este sensor tiene posibilidades para usarlo en "ropa inteligente".

Sensor de movimiento

“ Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

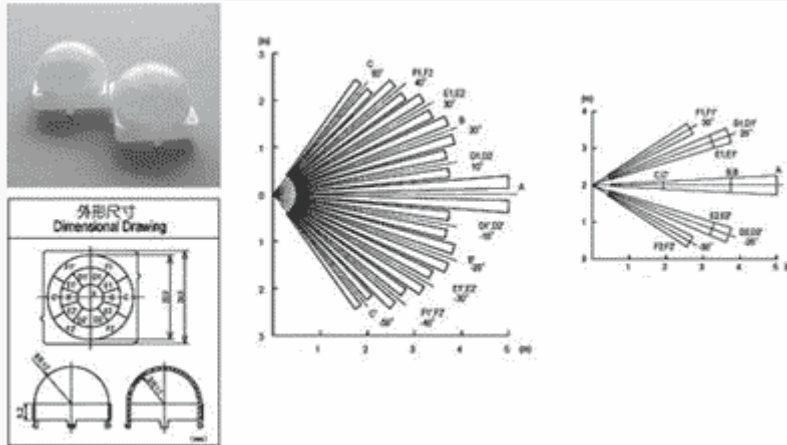
Los sensores PIR **se basan en la medición de la radiación infrarroja**. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica. En realidad **cada sensor está dividido en dos campos** y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.



El otro elemento restante para que todo funcione es **la óptica del sensor**. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.



Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Puedes ver ejemplos de uso en robótica en :

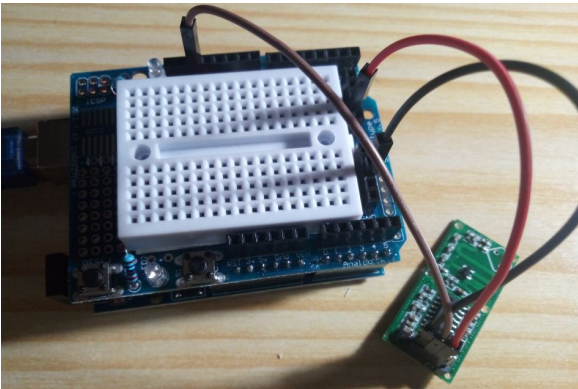
- [Smart Home para Microbit](#)
- [Smart Agriculture Kit para Microbit](#)



Más sensibles son los **sensores de microondas**. Son un radar que por efecto Doppler pueden captar cualquier objeto en movimiento dentro de un alcance de 5-7 metros en cualquier dirección e independiente de su temperatura. Es un buen sensor para alarmas, activación de luz por presencia..... Para saber más [ver la página de Luis Llamas](#)



Su conexión es muy sencilla, es un detector digital que hay que alimentarlo como el resto de sensores.



Curiosamente, la gran ventaja técnica de los de microondas **es un gran inconveniente para usarlo en el aula**, con cualquier movimiento se dispara, luego **para clase es mejor el sensor PIR**

Esta página esta adaptada de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA.

Actuadores

Esta página es un repaso rápido de las posibles **salidas** de los robots
NO LEAS TODOS SÓLO LOS QUE TIENE TU ROBOT

¿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 autómatas 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 baja potencia

Con el Arduino podemos usar motores de corriente continua de juguete como en la figura, aptos para poner un ventilador. Ejemplo de uso [Smart Home para microbit](#)

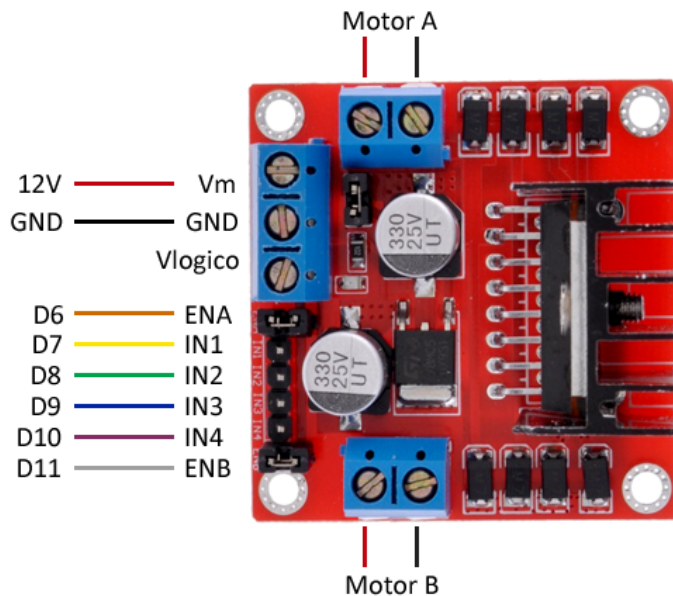


Motores con algo de potencia

Si quieres usar un motor para mover unas ruedas, ya pide más corriente, no se puede conectar directamente a la placa (Arduino, esp32 etc..). El de la figura es típico en robótica y lleva reductores dentro para bajar la velocidad :



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

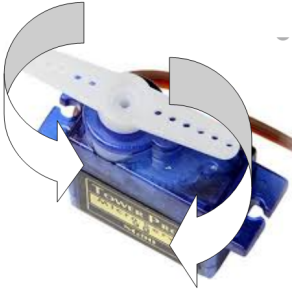
- [Curso mClon con nanoArduino con el B6612FNG](#)
- [Curso Rover con Arduino con la shield L293D ESP-12E para el NODE MCU que internamente tiene el LM298N](#)
- [Curso Arduino con código con LM298N](#) y también con un transistor

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



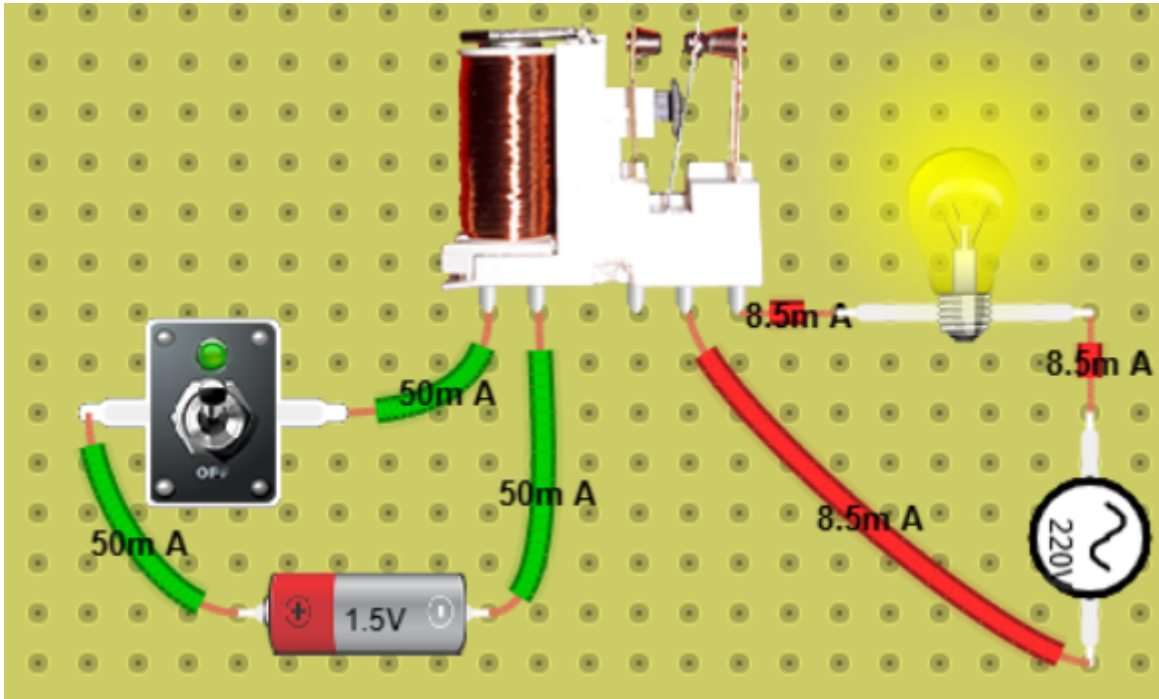
Ejemplos de uso de servos:

- [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](#)
- [Servo con PicoBrick](#)
- [Apertura ventana y puerta en Smarth Home para microbit](#)
- [Smart Agriculture Kit para micro:bit](#)

Relés

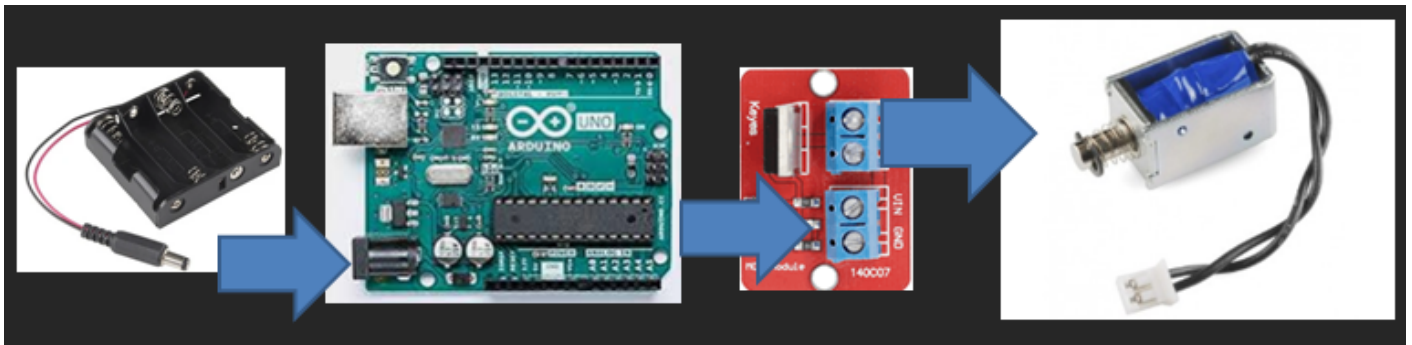
Un relé es un interruptor activado por un electroimán, lo que permite independizar los circuitos. En el dibujo se ve que el circuito rojo de 220V esta separado del verde, de sólo 1.5V. Pero es el circuito verde que al funcionar, hace que el electroimán mueva el interruptor del relé y encienda la bombilla. El objetivo es que he podido encender una bombilla de 220V sin tocar los 220V peligrosos. En el circuito verde, el interruptor puede ser un Arduino. Experimentalo en este [simulador](#).

- Un ejemplo con Arduino en [Luis LLamas](#)
- Un ejemplo de uso en [Picobricks](#)



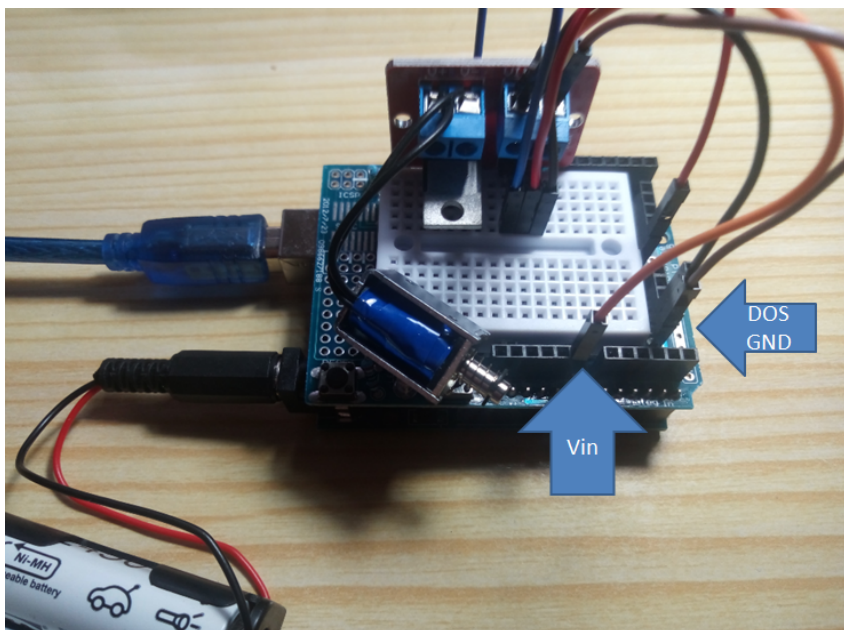
Electroimán

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)



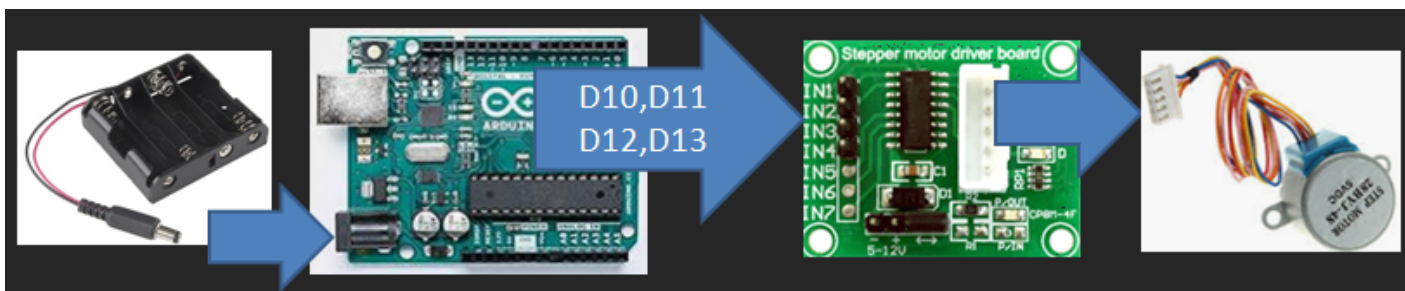
Motor paso a paso

Igual que el [electroimán](#), necesitamos un controlador que nos de la potencia necesaria para mover el motor, el ULN2003



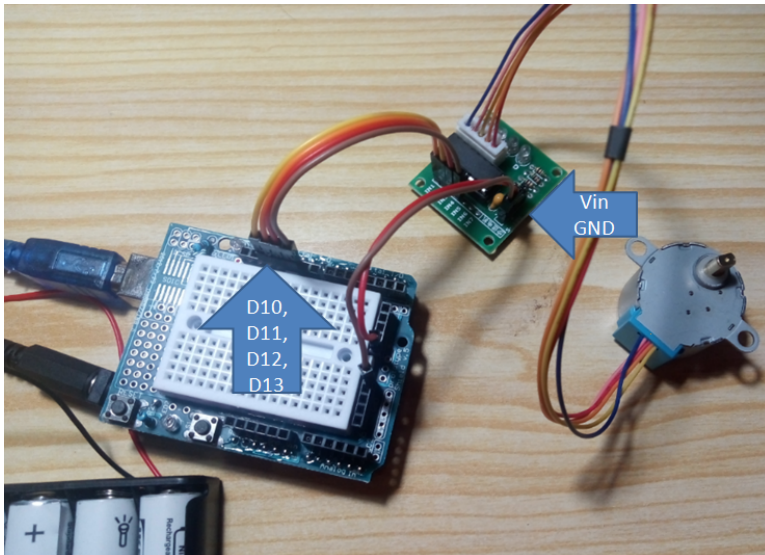
Imagen BY-NC-SA de [Luis Llamas](#)

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) :

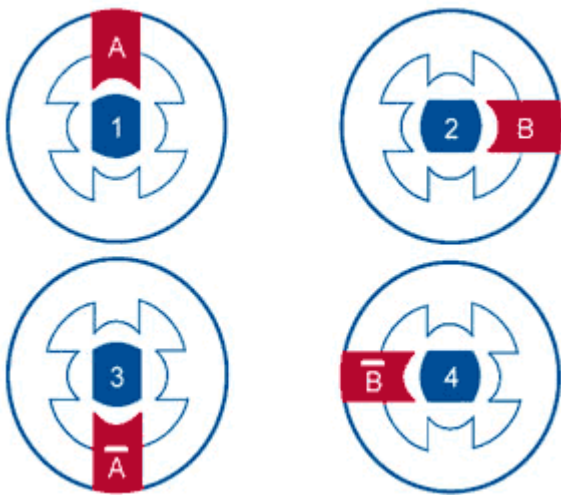


Imagen BY-NC-SA de [Luis Llamas](#)

Es decir:

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

Paso	IN1=D10	IN2=D11	IN3=D12	IN4=D13
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.

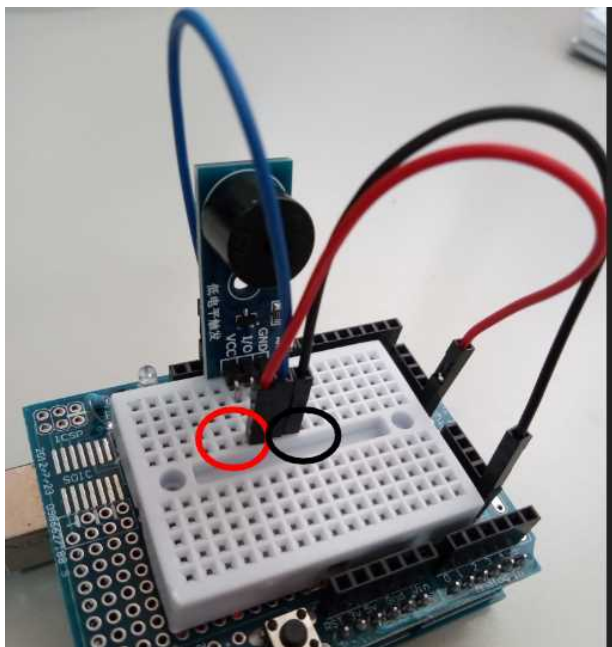
OTRAS SALIDAS

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

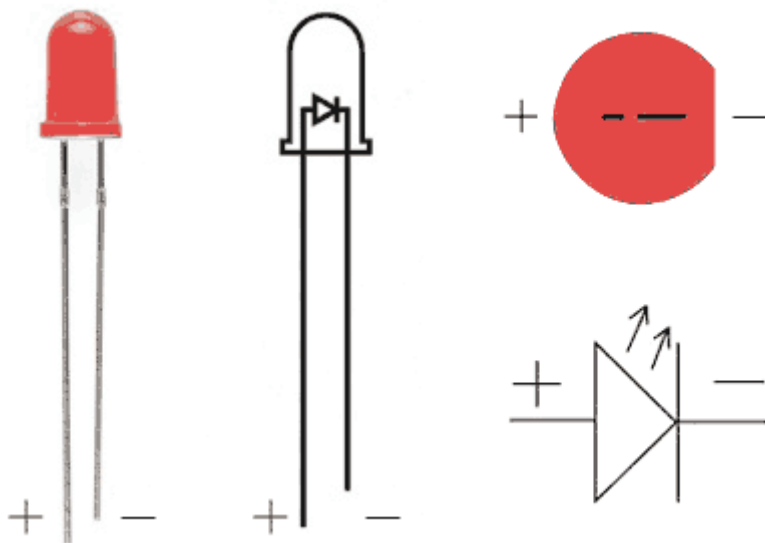
- [Timbre Arduino con Echidna](#)
- [Arduino blocks en el aula](#)

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)

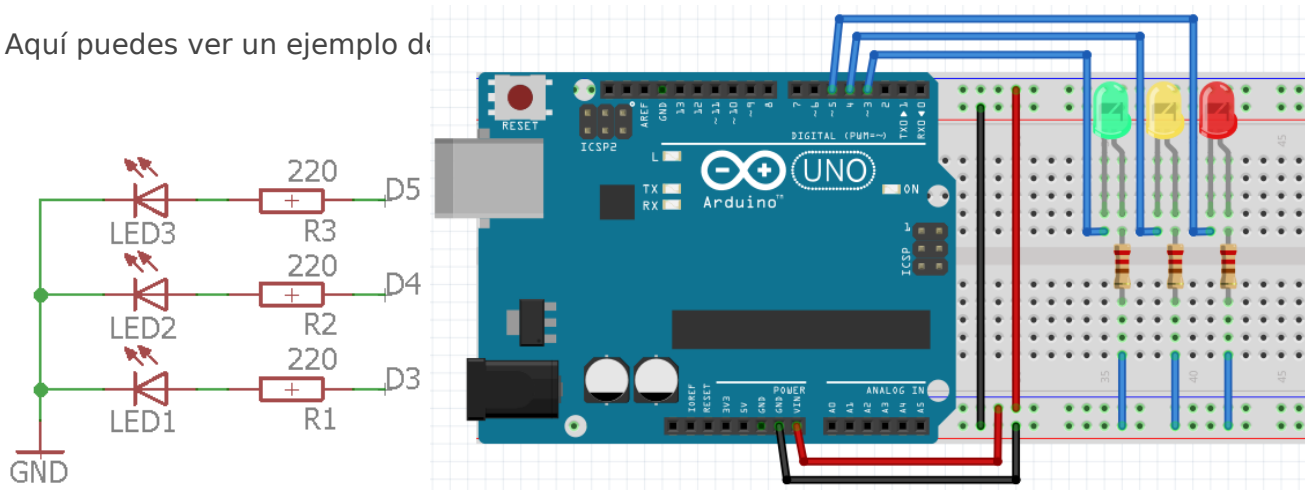
Tenemos pues que respetar su polaridad para que funcione:



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



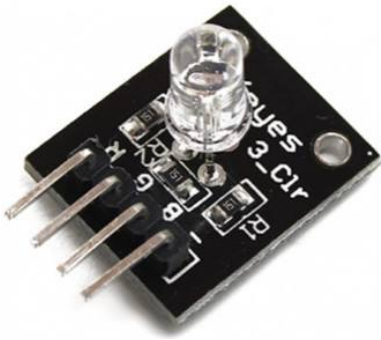
Se puede conectar directamente pues las salidas de Arduino tienen una limitación interna de 20mA como en este ejemplo <https://www.youtube.com/watch?v=EFFSLvIF9rY>

<https://www.youtube.com/embed/EFfSLvIF9rY>

Los LEDS igual que los motores pueden se 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.

“ Los WS2811, WS2812 y WS2812B son LED que disponen de lógica integrada, por lo que es posible variar el color de cada LED de forma individual (a

diferencia de las tiras RGB convencionales en las que todos los LED cambian de color de forma simultánea). Están basados en el LED 5050, llamado así porque tiene un tamaño de 5.0 x 5.0 mm.

Es un LED de bajo consumo y alto brillo, que incorpora en un único encapsulado los 3 colores RGB. La genial novedad del WS2812B (y resto de familia) es añadir un integrado dentro de cada LED, que permite acceder a cada pixel de forma individual. Por este motivo este tipo de LED se denominan “individual addressable”. Esto abre la puerta a un sinfín de aplicaciones y combinaciones, que van desde dotar de iluminaciones distintas zonas con una única tira, animaciones complejas, o incluso generar pantallas enteras de alta luminosidad....

A los LED WS2812B también se les denomina NeoPixel

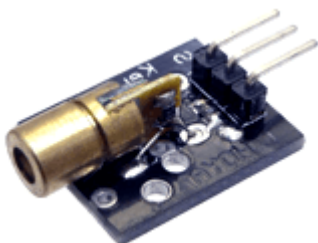
Luis Llamas CC-BY-NS-SA <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>



- [Smart Home para Microbit con 4 leds RGB Neopixel](#)
- [Smart Home para ESP32](#)
- [Smart Agriculture Kit micro:bit](#)

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.

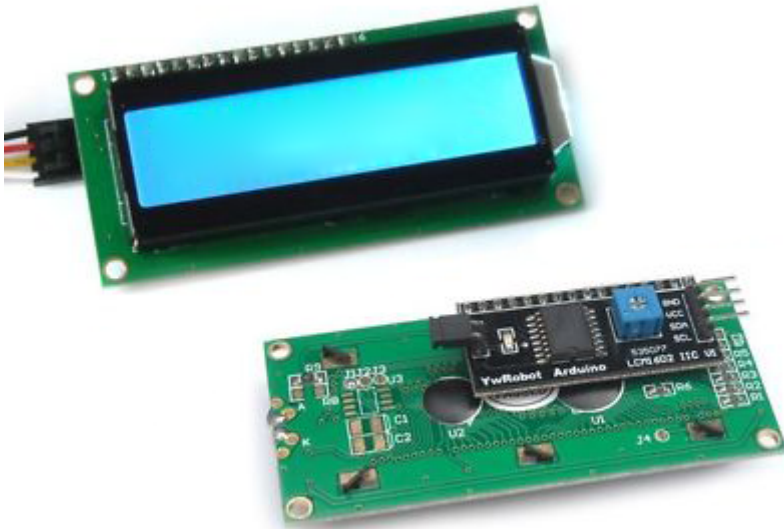


Display LCD

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](#)
- [Smart Home para Microbit](#)

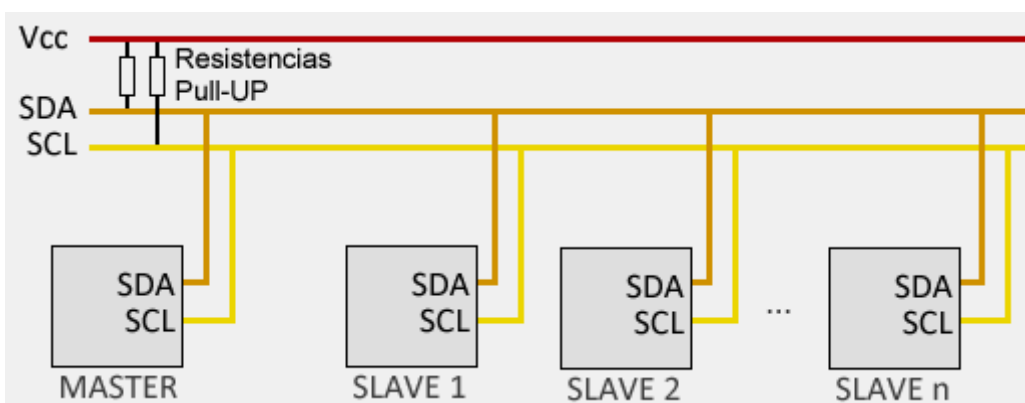


CC-BY-SA [Luis Llamas](#)

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

Aclarando conceptos: Protocolo I2C

Es un protocolo de emitir los datos con sólo dos cables **SDA** (los datos) **y SCL** (reloj) y además pueden compartir varios dispositivos SLAVE) para ello la placa Arduino ESP32, microbit... (MASTER) tiene que saber **qué dirección tiene cada dispositivo IMPORTANTE:**



Fuente [Luis Llamas ArduinoI2C](#) CC-BY-SA

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>

Software para la micro:bit

Micro:bit se puede programar :

Tipo de lenguaje	Lenguaje	Ventajas	Inconvenientes	En AULARAGON
BLOQUES	MAKECODE https://makecode.microbit.org/	-Muy extendido -Existen en Internet muchos recursos y ejemplos -Con CretateAI podemos usar IA		- micro:bit - micro:bit+car - Smart Home - Smart Agriculture
BLOQUES	SCRATCH+Learnin gML https://learningml.org/lml-scratch/	-Se puede usar IA -En vivo		Es el que proponemos en el curso de micro:bit en LearningML
BLOQUES	Classroom Microbit https://classroom.microbit.org/	- Se puede crear una clase, enviar proyectos a los alumnos, corregir....		
BLOQUES	Microblocks https://microblocks.fun/	-Código abierto		
BLOQUES	mBlock	-Permite usar multitud de extensiones	Recomendamos usar la versión descargada no la online	
CÓDIGO	PYTHON https://python.microbit.org/	- Se usa todo el potencial del código	- Curva de aprendizaje alta, no recomendado en primaria.	Micro:bit y Python

Guía de los lenguajes de programación para robots

https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWIGvlyVJr83Yk56kAoYD7bXLnYDm_tCQkeAgg/pubembed?start=false&loop=false

[&delayms=3000](#)

Tenemos un **grupo Telegram Robótica Educativa en Aragón,**
<https://t.me/roboticaeducativaaragon>



Software para la Smart home

¿Qué opciones tenemos para para programar SMART HOME PARA MICRO:BIT?

- **LENGUAJE POR BLOQUES O GRAFICO**

- MAKECODE ([tutorial específico para SmartHome para Micro:bit in English](#)) es el elegido en este curso por estar muy extendido.
- MICROBLOCKS ([tutorial específico para SmartHome para Micro:bit en catalán](#))

- **LENGUAJE CON CÓDIGO**

- PYTHON ([tutorial específico para SmartHome para Micro:bit in English](#))

Por ahora con la maqueta **NO se puede programar con scratch+learningML**, pues no tiene opción de manejar los pines. Sólo se podría utilizar scratch+learningML sólo con la placa micro:bit como se explica aquí

<https://libros.catedu.es/books/microbit/chapter/learning-ml>