

# Introducción

- [Objetivos y Contenidos](#)
- [Kit ESP32 en el Aula](#)
- [Pensamiento computacional](#)
- [¿Qué es ESP32?](#)
- [Hardware ESP32 Plus STEAMakers](#)
- [Sensores](#)
- [Actuadores y otras salidas](#)
- [Sensores y actuadores de la placa Imagina TDR STEAM](#)
- [Sensores en la placa ESP32 plus STEAMakers](#)
- [Software](#)

# 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
  - Retos **STEAM** de nivel medio adentrándose en las comunicaciones Bluetooth y wifi y almacenamiento de datos
  - Retos **STEAM** de nivel avanzado con prácticas IoT
- Realizar retos STEAM en un entorno **ARDUINOBLOCKS** amigable, sin complicaciones técnicas, multiplataforma y con lenguaje gráfico por bloques
  - Conocer **ARDUINOBLOCKS** una plataforma web que permite programar por bloques gráficos
  - Conocer las diferentes posibilidades de la programación por bloques de **ARDUINOBLOCKS**
  - Conocer las posibilidades didácticas de **ARDUINOBLOCKS** y su gestión de proyectos en el aula
- Conocer el kit robótico **TDR STEAM ESP32** que se propone, con un coste económico bajo pero de calidad para que un centro educativo lo pueda asumir para su aplicación en el aula (mínimo 12 por aula)
  - Conocer la placa **TDR STEAM ESP32**, sus sensores y actuadores
  - Aplicaciones prácticas del **TDR STEAM ESP32**

## Contenidos

- **PLACA ESP32**
  - Placa ESP32, hardware y software
  - Sensores
  - Actuadores
  - Shield TDR Steam
- **ArduinoBlocks**
  - **Retos**
  - **Telecomunicaciones**
    - **Bluetooth**
    - **IOT -WIFI**
      - MQTT
      - BLYNK IOT
      - BLYNK LEGACY



- o Blynk en una raspberry local

# Kit ESP32 en el Aula

Se ha buscado un kit entre todos los comerciales que cumpla los siguientes objetivos :

- Que sea fácil de utilizar en el aula, sin necesidad de utilizar excesivo cableado, con el objetivo de realizar el máximo número de retos.
- Utilizar un lenguaje de programación por bloques fácil de desarrollar y sin complejidades técnicas multiplataforma.
- Con buena relación calidad/precio.
- Con un amplio abanico de prácticas y niveles educativos, desde primaria hasta secundaria.
- Facilidad de crear actividades en entorno IoT

El equipo pedagógico de CATEDU ha elegido para este curso el [Kit TDR STEAM ESP32 de Innovadidactic](#), con el objetivo de alcanzar prácticas más avanzadas dentro del mundo del Internet de las cosas IoT.

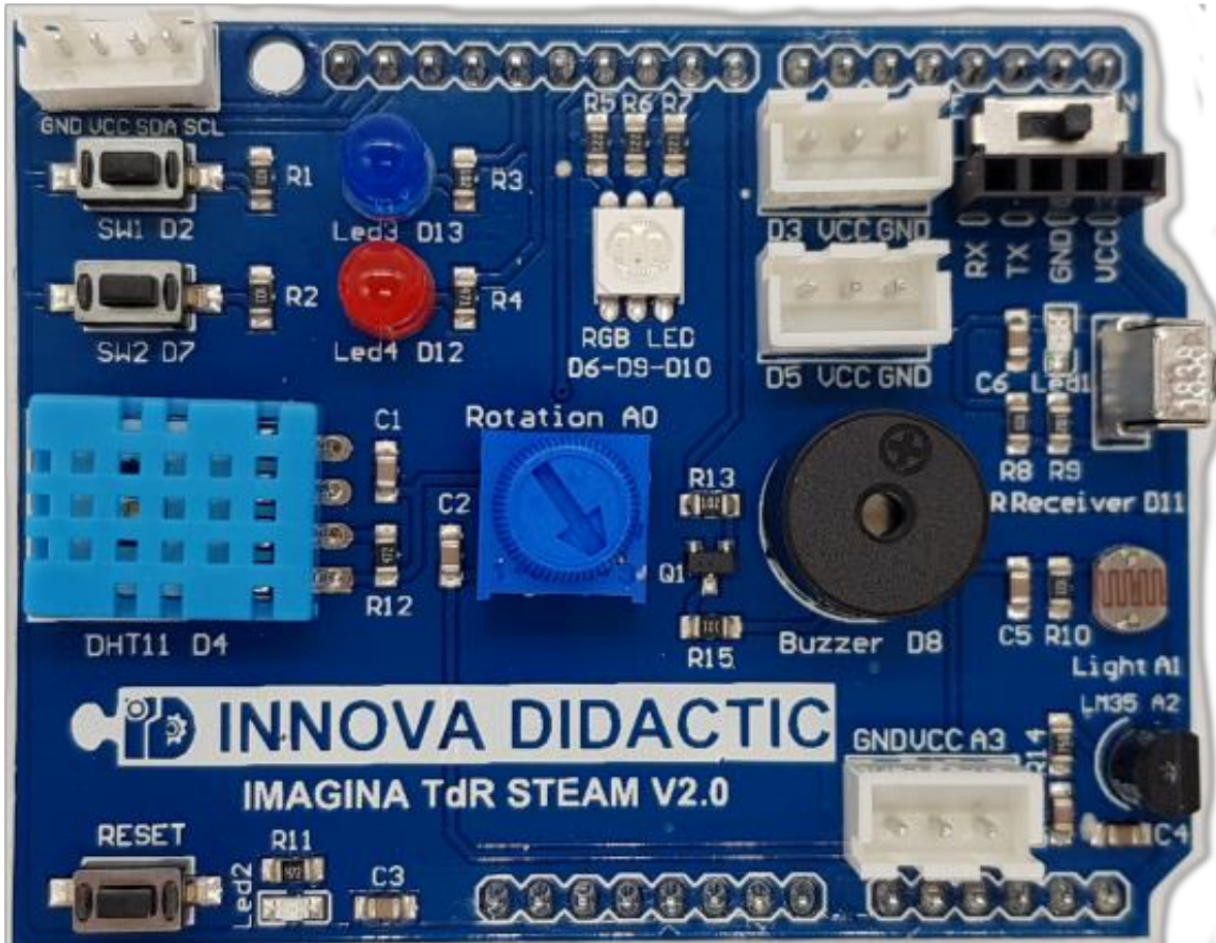
El kit esta pensado **para utilizar dentro del aula** y se ha establecido un número de 12 grupos, consideramos que es un número óptimo para llegar a la ratio del aula y que el profesor llegue a atender correctamente a todos los grupos.



## Partes del kit Imagina TdR STEAM

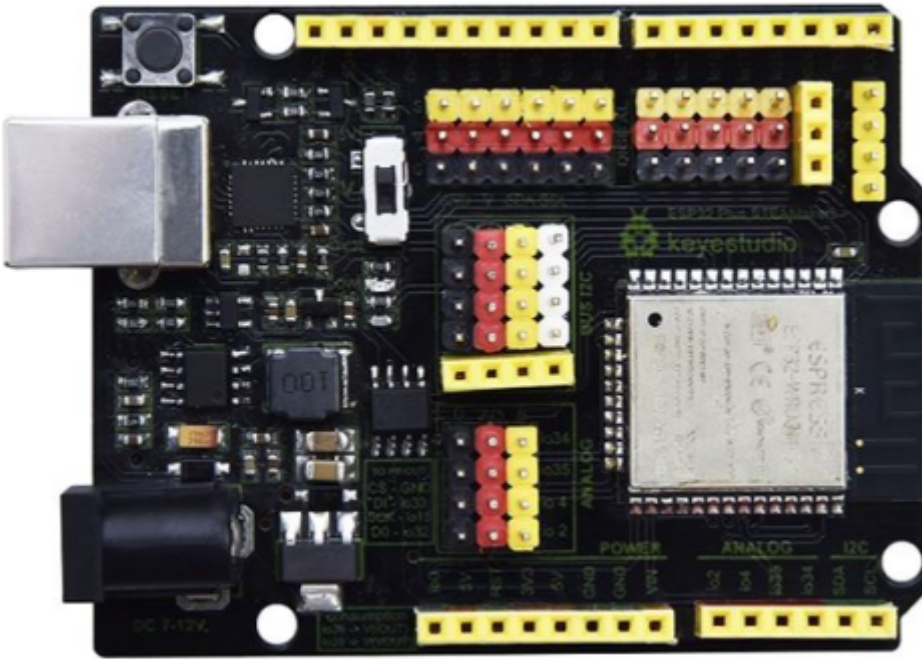
Este apartado está extraído de Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Una placa Imagina TdR STEAM versión 2.0 como la de la imagen siguiente:



Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Una placa ESP32 Plus STEAMakers.



Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Mando de control remoto por infrarrojos para utilizar en conjunto con el sensor de infrarrojos integrado en la placa TdR-STEAM.



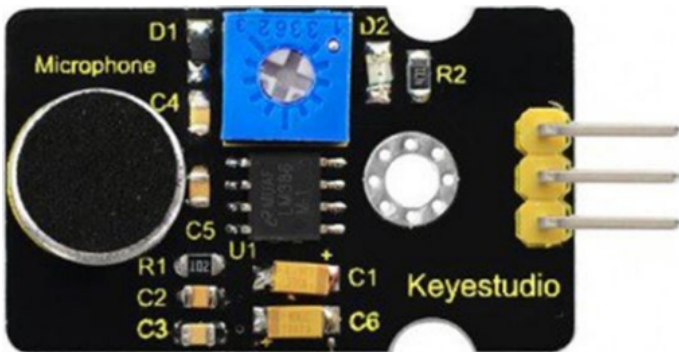
Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Pantalla LCD 1602 IIC (I2C) de 2 línea de 16 caracteres. Nos va a permitir mostrar mensajes de texto e irá conectada al conector I2C de la placa TdR-STEAM.



Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Sensor de sonido analógico (con potenciómetro). Se utiliza básicamente para detectar el nivel sonoro ambiental. El potenciómetro permite ajustar el nivel de ganancia.



Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

Todo el kit se presenta en una caja de plástico, pero una opción práctica para no abrir y cerrar tantas veces la caja, es utilizar una caja impresa 3D, las partes impresas e instrucciones se pueden conseguir en <https://fgcoca.github.io/ESP32-STEAMakers/necesidades/> el resultado es :



Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

# 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 **ESP32 EN EL AULA** tiene la ventaja de tener un precio razonable, y dentro del rango de programación en bloques y de prácticas avanzadas IoT con la ventaja que es un kit muy amigable sin apenas conexiones y con muchas posibilidades.

Guía orientativa

[https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco\\_/embed?start=true&loop=true&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco_/embed?start=true&loop=true&delayms=3000)

Tenemos un **grupo Telegram Robótica Educativa en Aragón,**

<https://t.me/roboticaeducativaaragon>





# ¿Qué es ESP32?

## Primero : ¿Qué es Arduino?

Arduino es una tarjeta electrónica que integra básicamente a un microcontrolador y un conjunto de pines de conexión de entradas y salidas que permiten, mediante un determinado programa, interaccionar con el medio físico mediante sensores y actuadores electrónicos. De esta forma podrás crear tus propios proyectos tecnológicos, dotarlos de sensores que detecten magnitudes físicas como luz, calor, fuerza, etc... y en base a esa información, escribiendo un programa, activar otros dispositivos (actuadores) como pequeñas bombillas, ledes, servomotores, pequeños motores DC, relés, etc... Los sensores se conectan a los pines de entrada y los actuadores a los de salida.

**¿Sabías que.... ?** Uno de los co-creadores de Arduino es Español, de Zaragoza: **David Cuartielles** [+info](#)

## Segundo ¿Qué es un microcontrolador?

Es un circuito integrado que se puede programar, o sea que puede ejecutar las órdenes que tenga almacenadas en su memoria. Tiene las tres funciones principales de un computador: la unidad central de proceso, memoria y entradas y salidas.

Arduino utiliza la marca ATMEL, y el modelo de microcontrolador depende del tipo de tarjeta, por ejemplo la tarjeta Arduino Uno utiliza el micro ATMEL MEGA 328P. Si quieres saber las entrañas de esta placa [aquí](#)

## Tercero ¿Arduino tiene wifi?

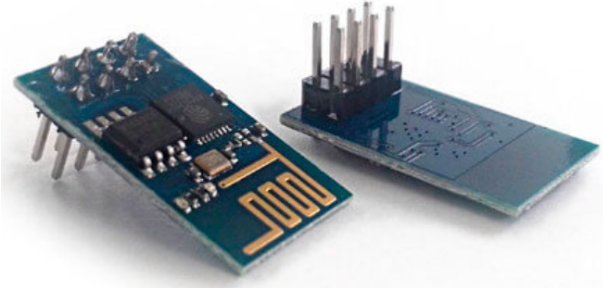
Arduino (el clásico) no tiene wifi, y es importante esto para conseguir hacer prácticas IoT. Hay shields que permiten una conexión Ethernet o Wifi pero resultan caras. Otra opción es utilizar la versión del Arduino MKR1000 pero también resulta cara. A partir del 2023 lanzaron Arduino R4 Wifi pero realmente era un ESP32. Lo mejor es utilizar el ESP8266 para que a través de él nuestro Arduino pueda volcar o recibir datos a través de una Wifi.

Resumiendo brevemente, el **ESP8266** es un chip microcontrolador, es decir, no es un sensor, no es un actuador, no es una entrada/salida del Arduino sino que es, igual que el Arduino, es una placa electrónica montado en un módulo que tiene un microcontrolador (Tensilica L106 de 32bits) capaz de hacer cosas pero que tiene una característica importante: **Que tiene Wifi**, pero no lo veas como un módulo Wifi para Arduino, sino como una placa electrónica completa, como el



Arduino, incluso es su competencia.

El chip **ESP8266** se vende montado en un módulo, el más vendido es el **ESP01** y es el que se proporciona en el kit del curso **Arduinoblocks en el aula** de CATEDU.



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

Por eso se habla indistintamente ESP8266 o el ESP01

Su bajo precio y su software libre permitió al mundo maker utilizar este chip. No sólo se puede utilizar para que el Arduino tenga acceso a Internet, sino también se han desarrollado multitud de módulos con el ESP8266, como veremos más adelante, el más interesante en su evolución es el módulo ESP-12E o el ESP32.

**Pero sigamos con el ESP8266 montado en el módulo ESP01. Tiene unas pegas...** no se diseñó para montarlo en el Arduino : **PRIMERO** La alimentación es 3V a 3.6V con picos de 200mA por lo que no puede conectarse directamente a la alimentación 3.3V y 50mA de Arduino.

**SEGUNDO** Consecuencia del máximo de 3.6V es que las entradas y salidas del ESP8266 no conviene conectarlas directamente a las entradas y salidas del Arduino que van a 5V.

### Cuarto: ¿El ESP32 es ....?

Es una placa sucesora del ESP8266 de bajo coste y consumo que tiene el microprocesador Tensilica 32bits (variante el Xtensa LX6) que tiene Wifi y Bluetooth integrados. Como dice Luis Llamas en <https://www.luisllamas.es/esp32/> el ESP32 es el hermano mayor del ESP8266 con Wifi y Bluetooth.



Módulo ESP32 Fuente Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/esp32/>

Es posible programarlo usando código IDE de Arduino, pero también se puede programar con MicroPython.

Su principal característica es su potencial de uso en aplicaciones IoT

- Para saber más
  - Consideraciones sobre ESP32 <https://fgcoca.github.io/ESP32-STEAMakers/conesp32/>

“ Evolucion de búsquedas en Google de Arduino, Raspberry PI, ESP32 y MicroBit

- Arduino y Rpi tuvieron máximo en 2017-18. Desde entonces bajan.
- Arduino aguanta tipo
- Rpi cuesta abajo sin frenos
- ESP32 despegó
- Microbit no, pese a que tuvo buen arranque (inmerecido?)

[pic.twitter.com/A2PI18qecZ](https://pic.twitter.com/A2PI18qecZ)

— Luis Llamas (@LuisLlamas) [August 9, 2023](#)

# Hardware ESP32 Plus STEAMakers

Esta placa está basado en el ESP32 que hemos visto que tiene :

- Wifi integrado
- Bluetooth integrado
- Puede trabajar con el código IDE de Arduino

Pero, esta [versión de Innova Didactic](#) además incorpora :

- Zócalo para tarjetas microSD para almacenamiento de datos.
- Conexiones iguales que cualquier tarjeta Arduino UNO

Esto permite usar cualquier Shield de Arduino, como la TDR STEAM Imagina de este curso.  
Para saber las diferentes Shields de Arduino mira esta página

<https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/hardware>

En la figura ¿Puedes localizar el ESP32 famoso?

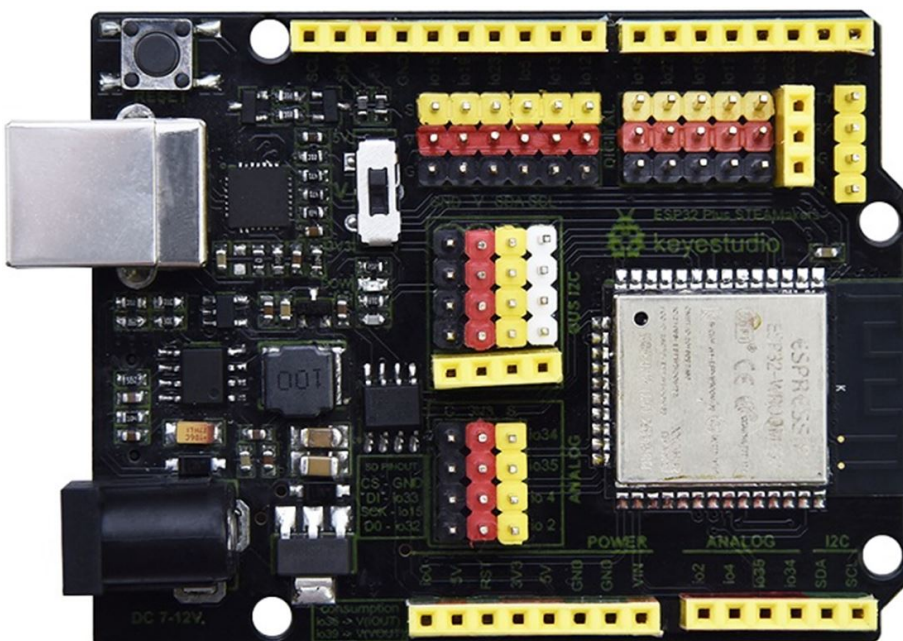


Imagen Fuente Innovadidactic

El contenido mostrado a continuación es de [la web de Federico Coca](#) Licencia CC-BY-SA

La placa ESP32 Plus STEAMakers nos ofrece una gran cantidad de prestaciones al estar basada en un microcontrolador de 32 bits con conectividad WiFi y Bluetooth integradas en la propia placa y también un zócalo para tarjetas  $\mu$ SD para el almacenamiento de datos. También dispone de conexiones para todas las entradas y salidas con posibilidad de tener la alimentación adjunta y puertos de expansión I2C para poder conectar diferentes dispositivos directamente en la placa.

La placa está basada en el microcontrolador ESP32-WROOM-32 y sus principales especificaciones técnicas son:

- Microcontrolador Tensilica Xtensa 32-bit LX6 a 160MHz.
- Conectividad WiFi 802.11 b/g/n/e/i.
- Conectividad Bluetooth 4.2 y modo BLE.
- Zócalo para tarjetas  $\mu$ SD.
- 14 entradas y salidas digitales con alimentación.
- Conector serie hembra con alimentación.
- Conector I2C para conectar hasta 5 dispositivos a la vez sobre la misma placa.
- Conector hembra I2C para conexión de una pantalla OLED.
- Botón de Reset.
- Conector de 5V
- Conector de 3.3V
- Interruptor 3.3-5V para cambiar entre estas dos tensiones en algunos pines de alimentación.
- Entradas y salidas analógicas.
- Sensor Hall y de temperatura integrado.
- 2 convertidores Digital-Analógico (DAC) de 8 bits.
- 16 convertidores Analógico-Digital (ADC) de 12 bits.
- 16 canales PWM.
- 2 UART.
- 2 canales I2C.
- 4 canales SPI.
- 448Kb ROM.
- 520 KB SRAM.
- 8KB+8KB SRAM en RTC.
- 1kbit eFUSE.
- 512 bytes Memoria Flash (EEPROM).
- 10 sensores táctiles.

- 4 temporizadores internos de 64 bits.

No están disponibles todas las características del controlador ESP-WROOM-32, ya que algunos pines tienen funciones dobles y se utilizan en la placa de forma específica (como, por ejemplo, para controlar la tarjeta SD). Pero la mayoría de funciones se pueden utilizar, además de disponer la placa ESP32 Plus STEAMakers de una mejor conexión de elementos debido a los pines para conectores tipo Dupont de entrada y salida, de I2C y de alimentación. Además, algunos pines de alimentación pueden cambiar su valor (3,3V o 5V) mediante un interruptor en función de nuestras necesidades.

<https://www.youtube.com/embed/SEAhjQx-e4A?list=PL1pKD-Bz2QBAgyf580m8OaQ2Z60v6DOhCindex=2>

A continuación vemos una imagen en la que se compara el potencial de la placa ESP32 Plus STEAMakers En la figura siguiente vemos los elementos que componen la placa ESP32 Plus STEAMakers:

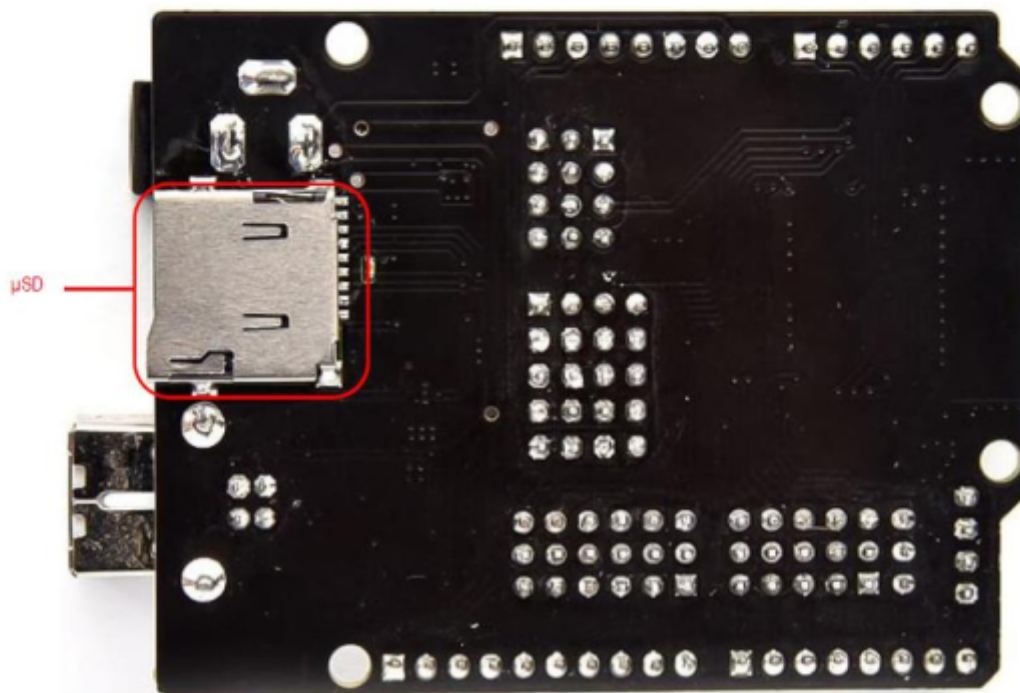
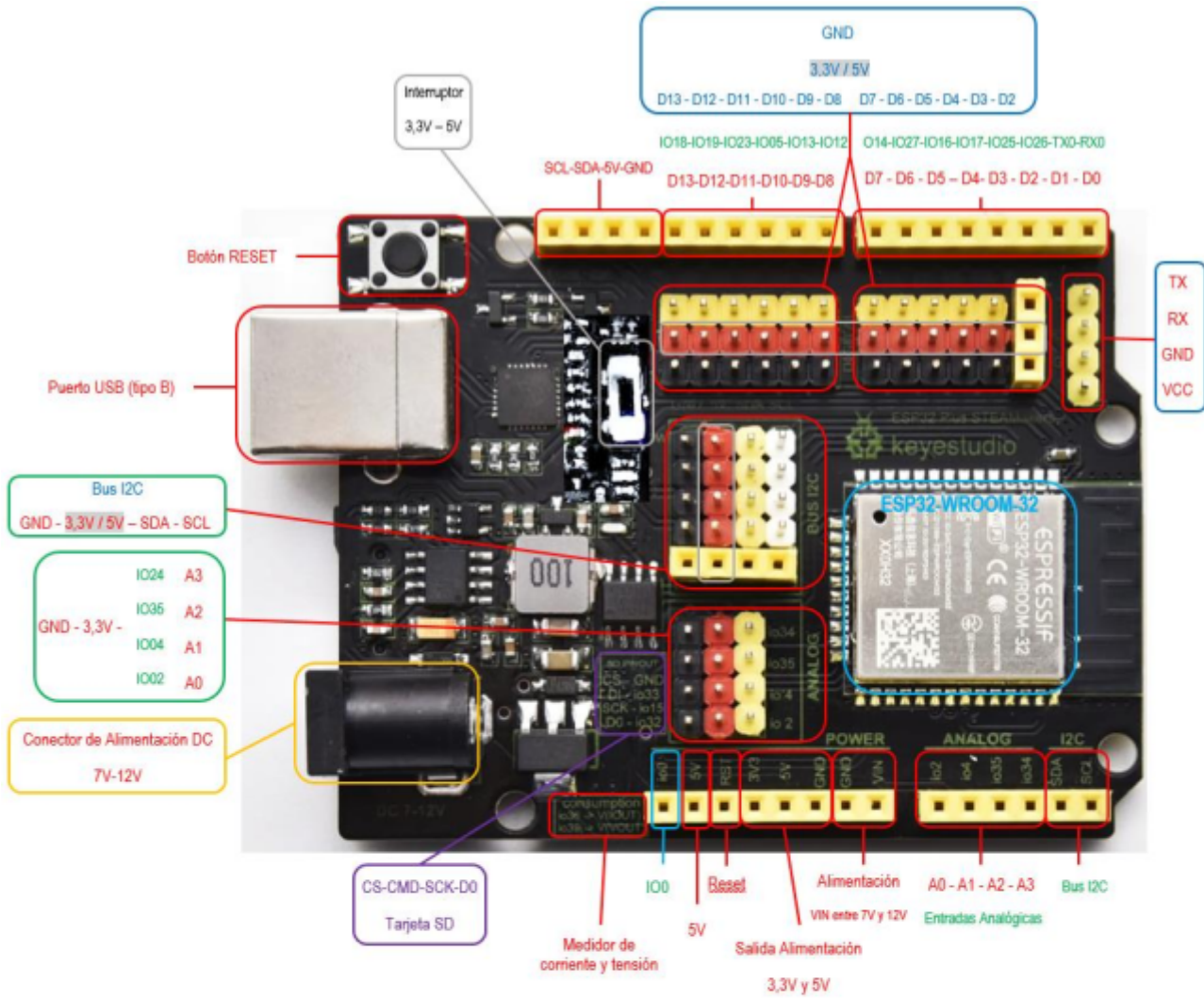


Imagen [Federico Coca](#) CC-BY-SA

Las conexiones de la placa Imagina TDR STEAM con la placa ESP32 Plus STEAMakers son las mismas que si utilizamos cualquier placa compatible con Arduino UNO.

**Importante:** Todos los pines IOxx son entradas y salidas digitales, algunas con más funciones. Utilizando la comunicación WiFi no funciona el ADC2.

En la tabla siguiente tenemos relacionados todos los pines entre los tipos de placas UNO, Imagina TdR STEAM y ESP32 STEAMakers.

UNO	TdR STEAM	ESP32		
Pin	Función	Pin	Función	Ampliación
D0	Rx	IO03	Rx	UART 0 RX
D1	Tx	IO01	Tx	UART 0 TX
D2	Pulsador SW1	IO26	ADC2 CH9	DAC2
D3	Libre	IO25	ADC2 CH8	DAC1
D4	DHT11	IO17		UART 2 TX
D5	Libre	IO16		UART 2 RX
D6	Color rojo del LED RGB	IO27	ADC2 CH7	ADC2-7 / TOUCH7
D7	Pulsador SW2	IO14	ADC2 CH6	ADC2-6 / TOUCH6
D8	Zumbador o buzzer	IO12	ADC2 CH5	ADC2-5 / TOUCH5
D9	Color verde del LED RGB	IO13	ADC2 CH4	ADC2-4 / TOUCH4
D10	Color azul del LED RGB	IO05		VSPI CS0
D11	Sensor IR	IO23		VSPI MOSI
D12	LED rojo	IO19		VSPI MISO
D13	LED azul	IO18		VSPI CLK
GND		GND		
AREF		Reset		
SDA	I2C	IO21		

UNO	TdR STEAM	ESP32		
SCL	I2C	IO22		
A0	Potenciómetro	IO02	ADC2 CH2	
A1	Sensor de luz (LDR)	IO04	ADC2 CH0	
A2	Sensor de temperatura (LM35)	IO36	ADC1 CH0	
A3	Libre	IO34	ADC1 CH6	
A4	I2C	IO38		
A5	I2C	IO39	ADC1 CH3	
VIN		VIN		
GND		GND		
GND		GND		
5V		5V		
3.3V		3.3V		
RST		Reset		
5V		5V		
		IO00	¡ No conectar !	
-		IO32	D0 - uSD	
-		IO15	CLK - uSD	
-		IO33	CMD - uSD	
-		IO35	IOUT	Medidor de corriente
-		IO37	VOUT	Medidor de tensión

- Para saber más :
  - Notas sobre ESP32 STEAMakers <https://fgcoca.github.io/ESP32-STEAMakers/hsteamakers/>

# 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,

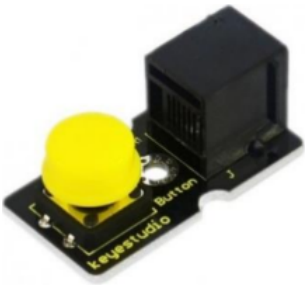


giroscopios, GPS,... pasando por detectores de gases, de pulsos cardiacos, sensores de efecto HALL,...

## 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  $1023 = 5\text{V.}$

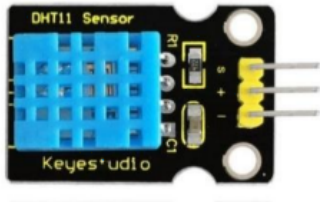


- **DATOS:** el sensor ofrece su información a través de una interfaz de comunicación. La forma de comunicación puede ser por sistemas estándar como **I2C** o **SPI** o algunos sensores usan su propio protocolo para codificar la información y debemos realizar desde el software la decodificación correcta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

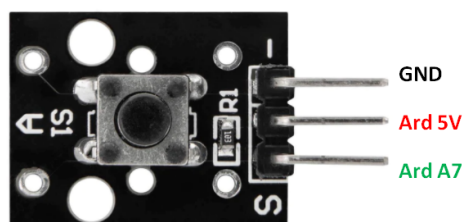
En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

### Sensor pulsador

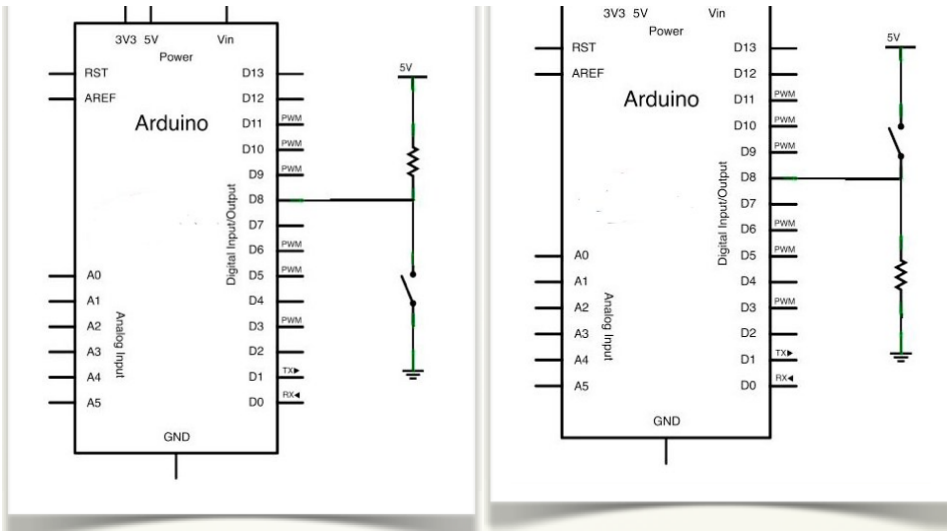
Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso

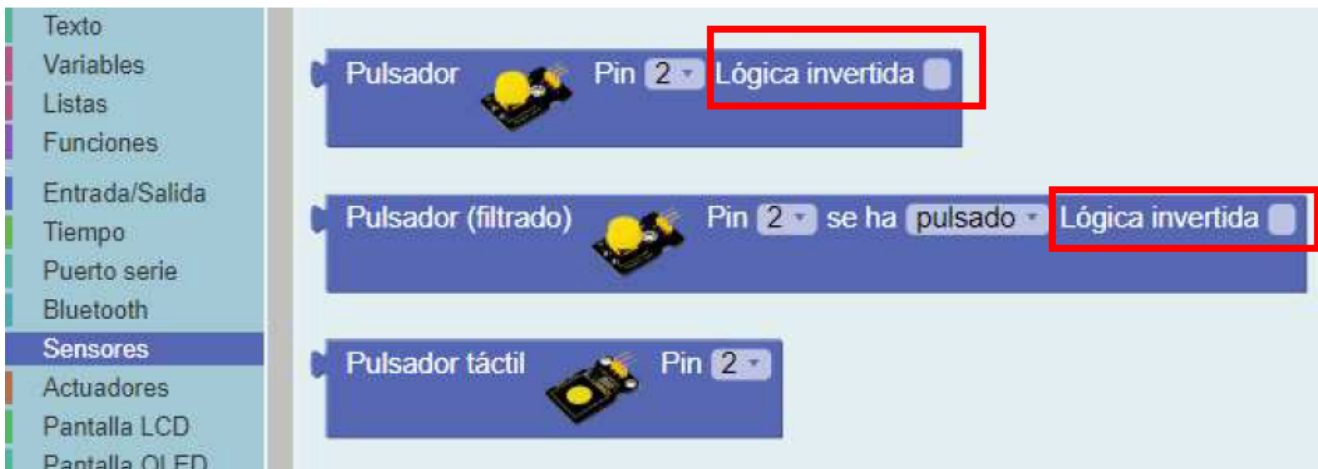
• en el robot mClop



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.

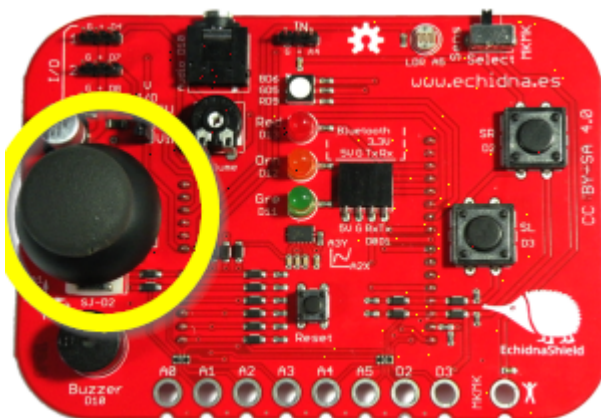
Un ejemplo de uso en

- [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 Focélula LDR.

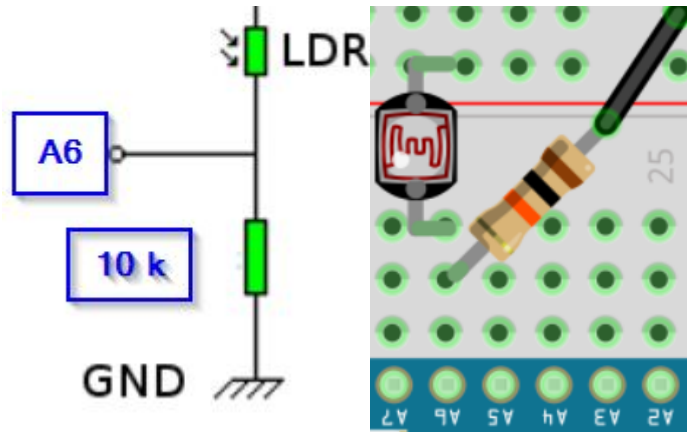
El uso de fotocélulas es muy común en nuestras vidas, las encontramos en el encendido automático de farolas, apertura de puertas,... La fotocélula es un semiconductor. Es ampliamente utilizado en campos de interruptores de control automático como cámaras, luces solares de jardín, lámparas de césped, 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:



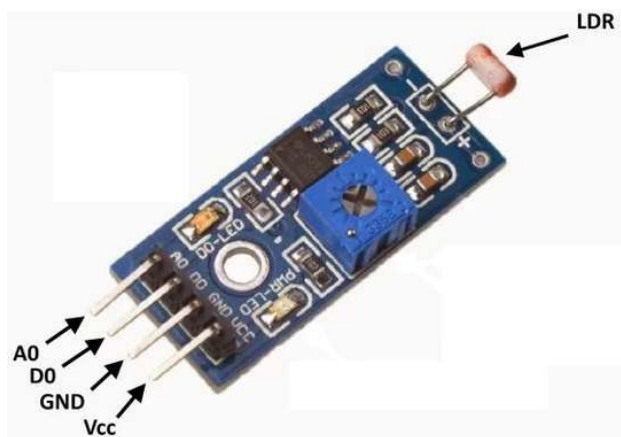
La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.



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:

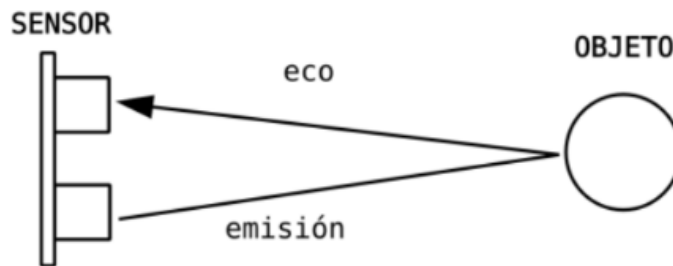
**Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) \* (V.Sonido 34000 en cm/s)} / 2**

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.

bloques, 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

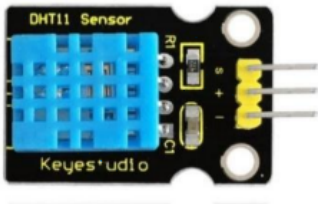
**Checksum** (en binario) como la letra del DNI. Por ejemplo **0100 0111 0000 0011 0001 1001 0000 0000 0001 1000** es **0100 0111 0000 0011** = **47.03%** de humedad y **0001 1001 0000 0000** = **19.00°C** y la comprobación es la suma de  $4+7+0+3+1+9+0+0=24=$ **11000**

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](#)

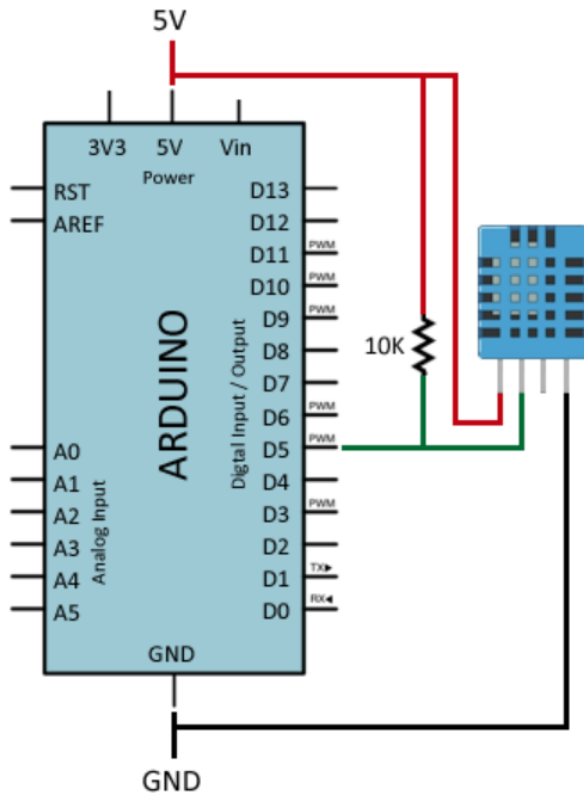


No es un sensor con gran sensibilidad, pero para propósitos educativos cumple sus funciones. Por dentro tiene una resistencia NTC que decrecienta su resistencia si aumenta la temperatura. Hay otros que van al revés, los PTC. Tanto los NTC como los PTC se llaman **thermistores**. Para la humedad, mide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

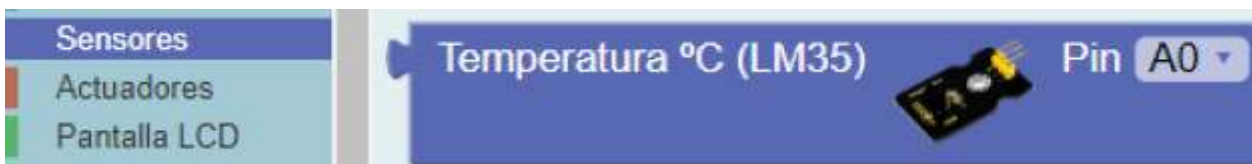


Ejemplo de uso de un DHT11 sin encapsular:



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 $\lambda$
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.

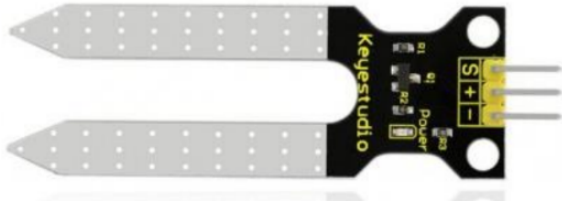
“ Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene 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 :

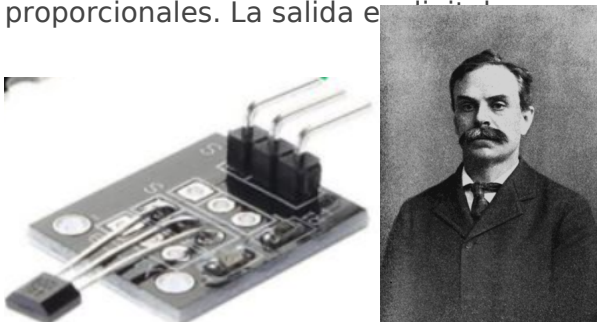
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

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.

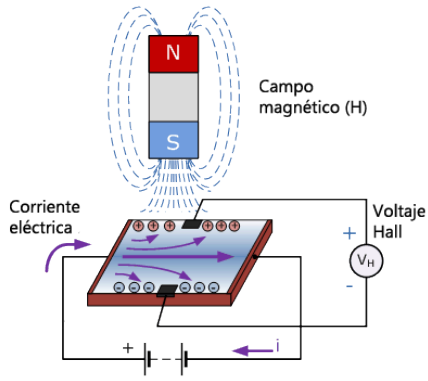


Sensor Hall.

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](#)

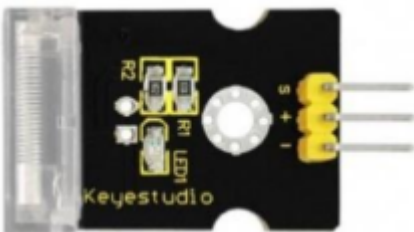
## Sensor inclinación

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

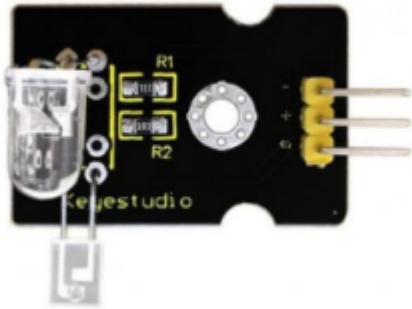
Es un sensor digital que al ser golpeado este sensor envía una señal momentánea.



## Sensor de pulso cardíaco.

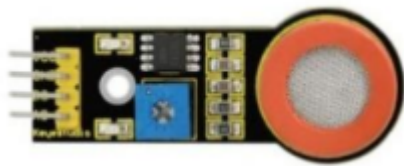


Este módulo utiliza un LED infrarrojo (IR) ultrabrillante y un fototransistor para detectar el pulso en el dedo. Principio de funcionamiento: Se debe colocar el dedo entre el LED infrarrojo ultrabrillante (parte superior) mientras que el fototransistor, que queda en el otro lado, recoge la cantidad de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



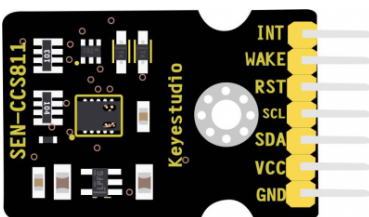
## Sensor de Alcohol.

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

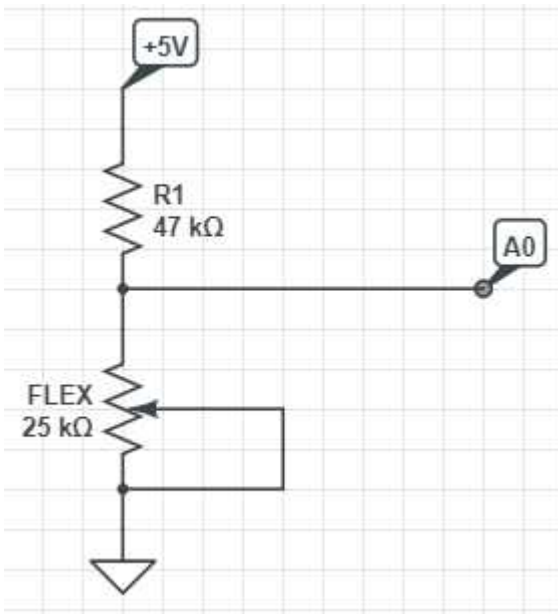


## Resistencia Flex

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

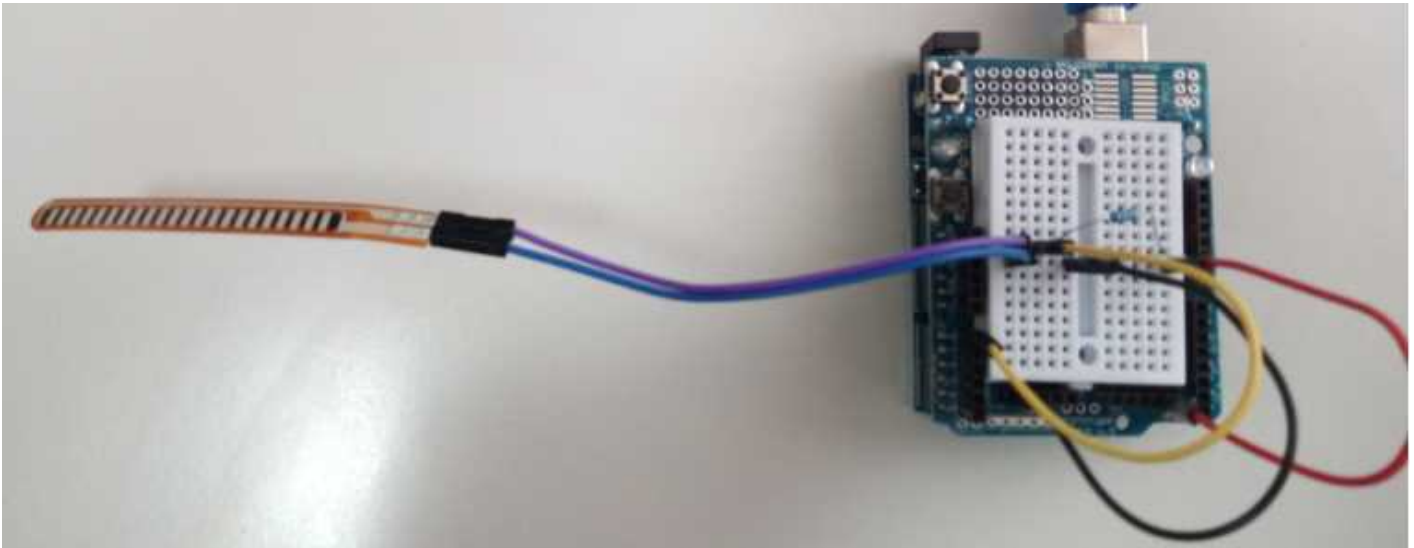


Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable,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



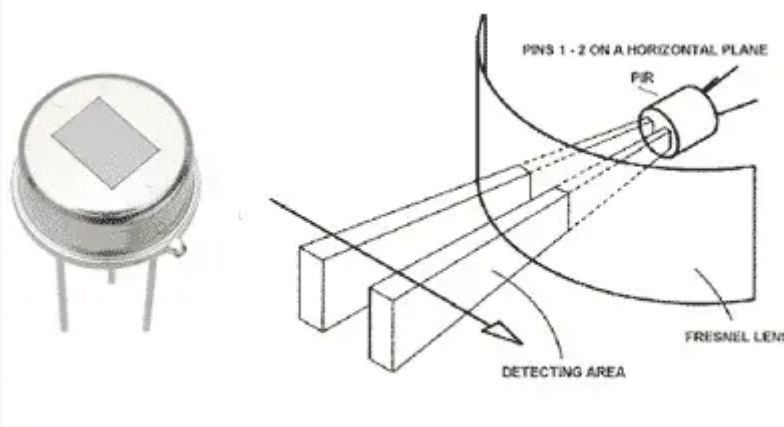
Este sensor tiene posibilidades para usarlo en "ropa inteligente".

## 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.

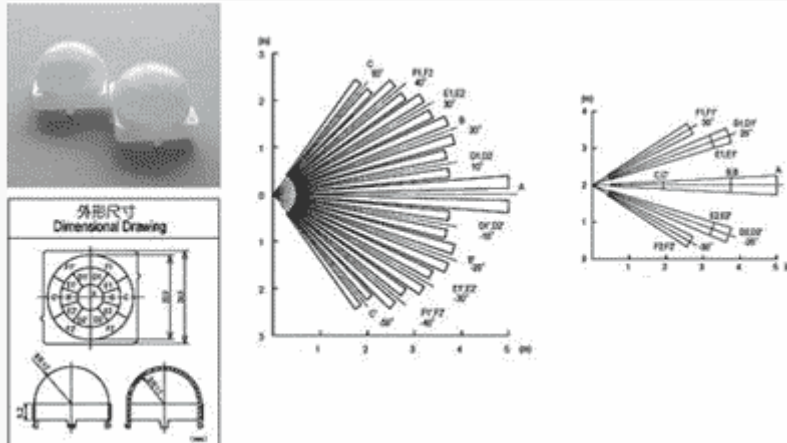
Los sensores PIR **se basan en la medición de la radiación infrarroja**. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una 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 :

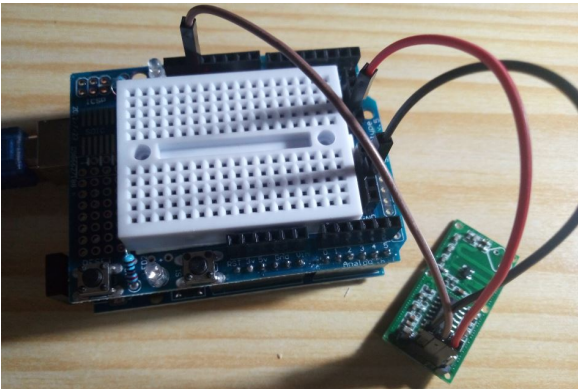
- [Smart Home para Microbit](#)
- [Smart Agriculture Kit para Microbit](#)



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 y otras salidas

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 autómeta 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](#)

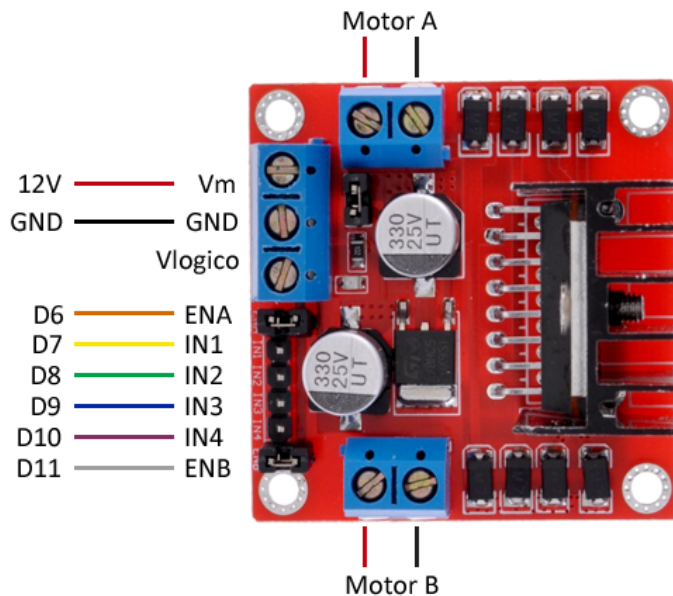


### Motores con algo de potencia

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



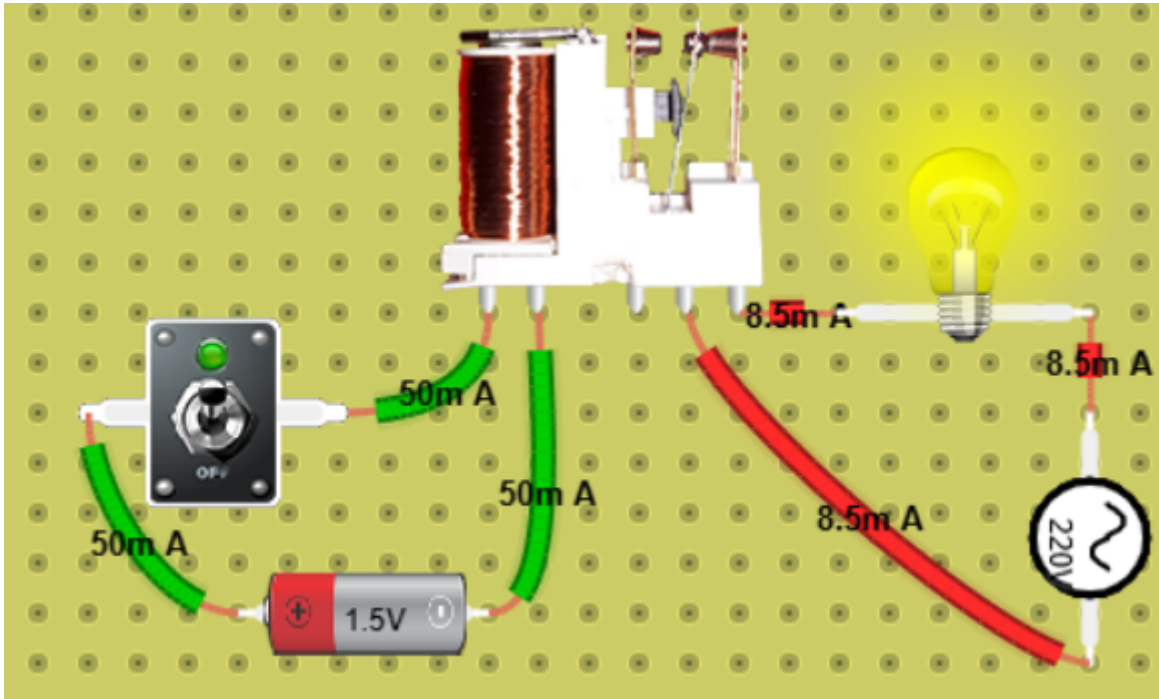
Ejemplos de uso de servos:

- [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

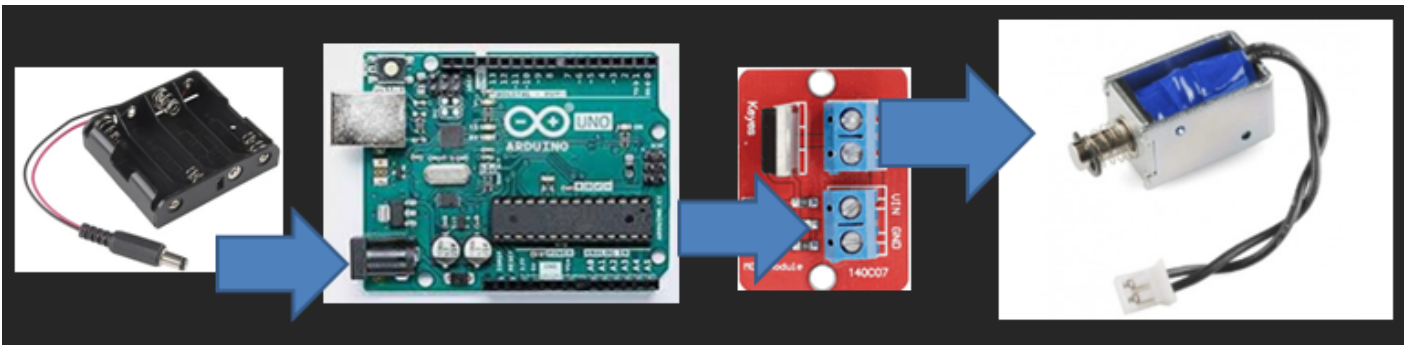
Un relé es un interruptor activado por un electroimán, lo que permite independizar los circuitos. En el dibujo se ve que el circuito rojo de 220V está separado del verde, de sólo 1.5V. Pero es el circuito verde que al funcionar, hace que el electroimán mueva el interruptor del relé y encienda la bombilla. 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](#)



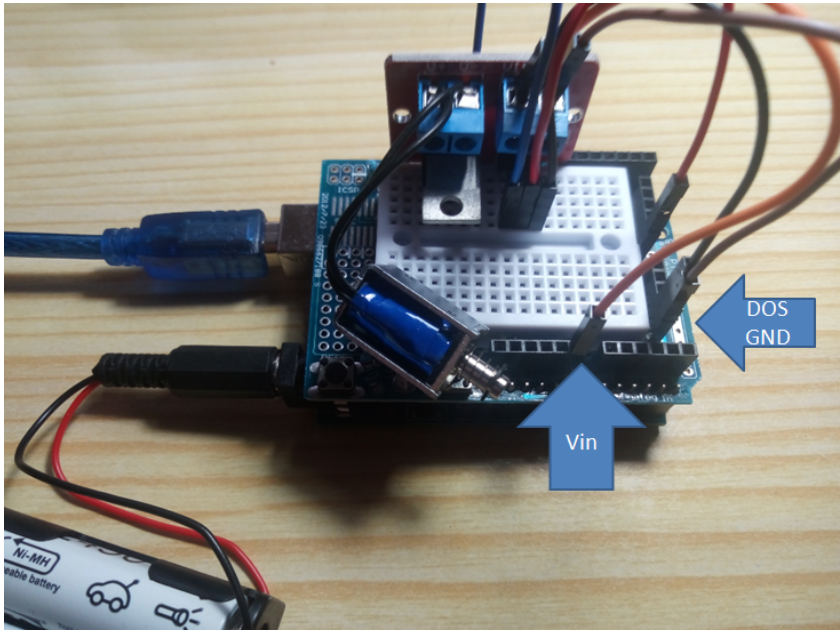
## Electroimán

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)



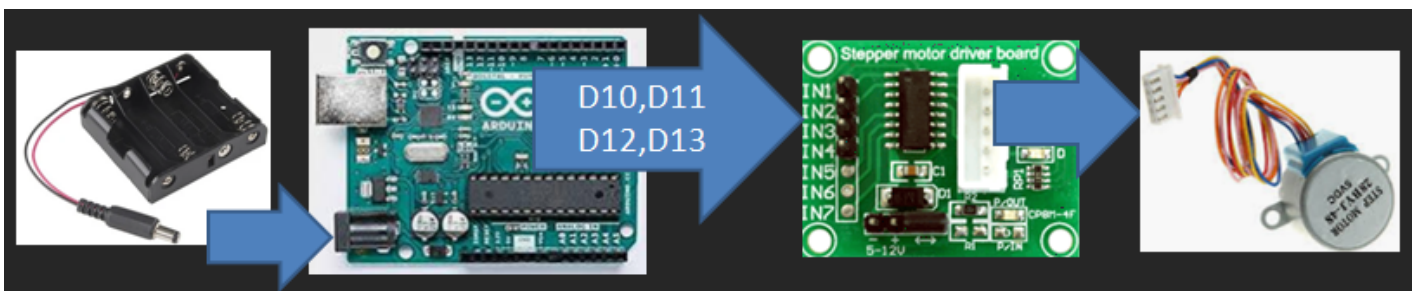
## Motor paso a paso

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](#)

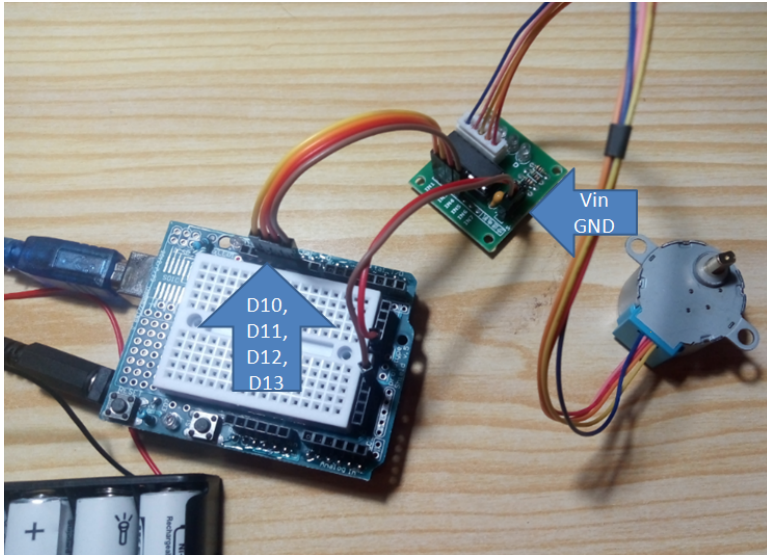
También igual que el electroimán, necesitamos una potencia extra con las pilas:



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) :

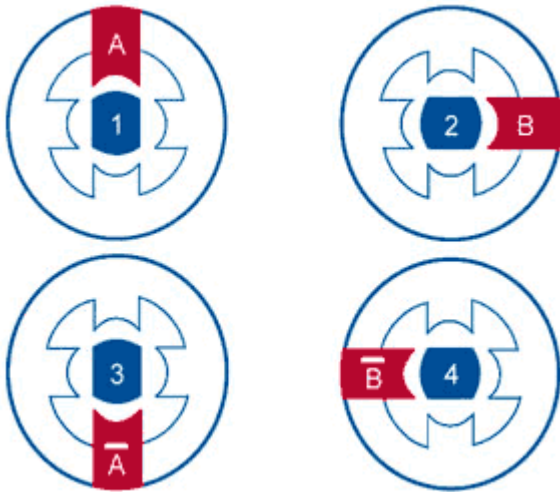


Imagen BY-NC-SA de [Luis Llamas](#)

Es decir:

Paso	IN1=D10	IN2=D11	IN3=D12	IN4=D13
<b>Paso 1</b>	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.

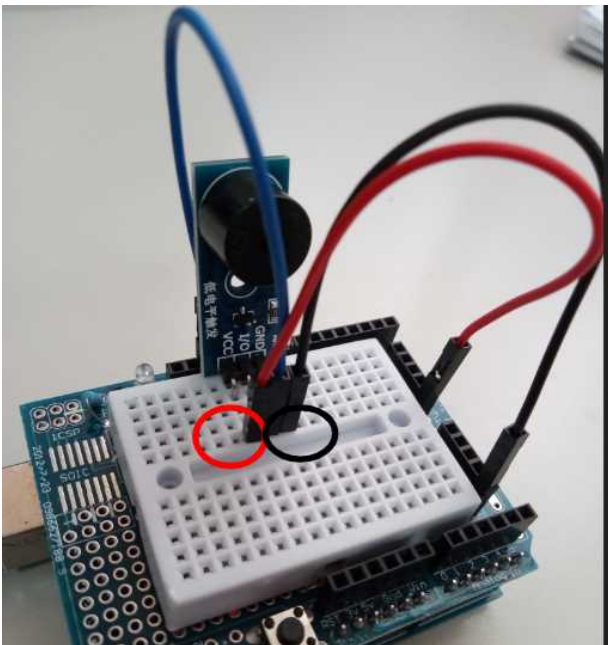
## OTRAS SALIDAS

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

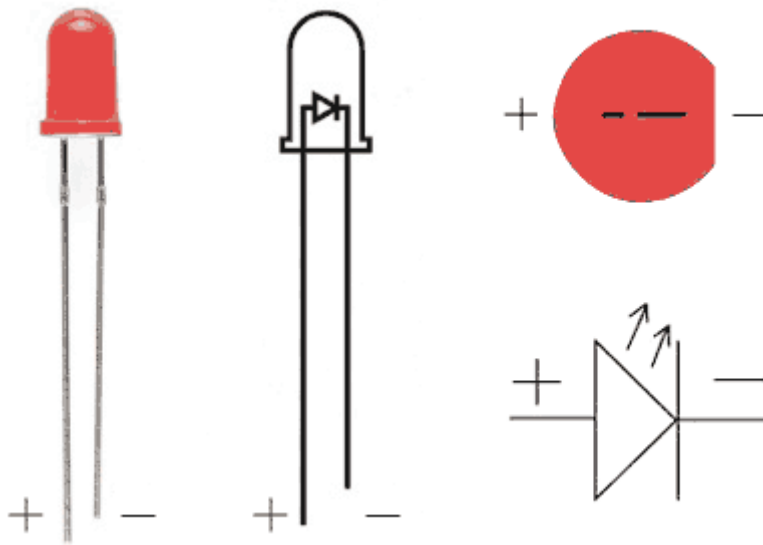
- [Timbre Arduino con Echidna](#)
- [Arduino blocks en el aula](#)

## 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)

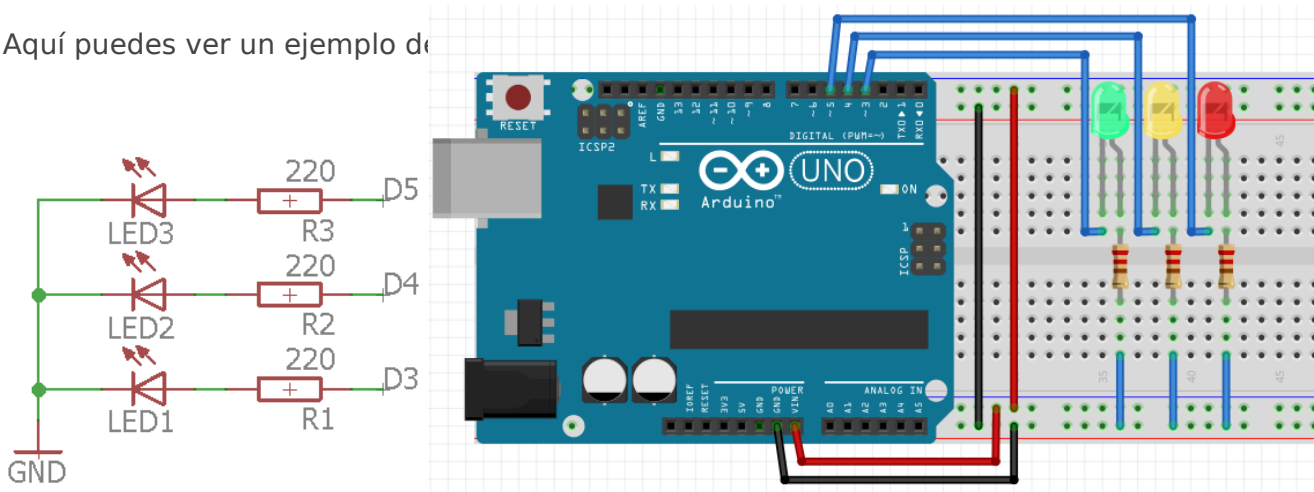
Tenemos pues que respetar su polaridad para que funcione:



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



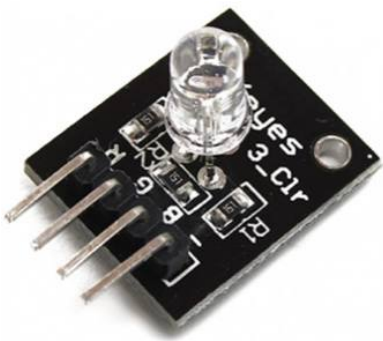
Se puede conectar directamente pues las salidas de Arduino tienen una limitación interna de 20mA como en este ejemplo <https://www.youtube.com/watch?v=EFFSLvIF9rY>

<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

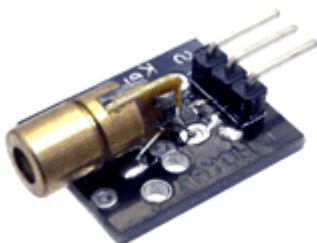
Luis Llamas CC-BY-NS-SA <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>



- [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.

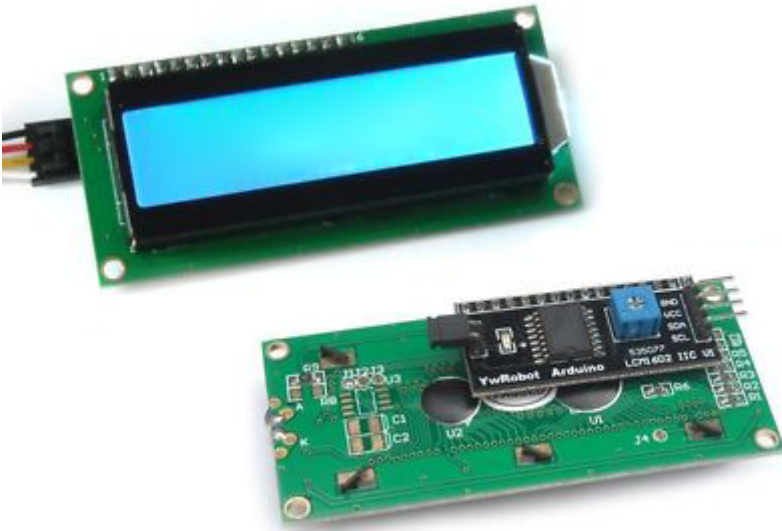


## Display LCD

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](#)

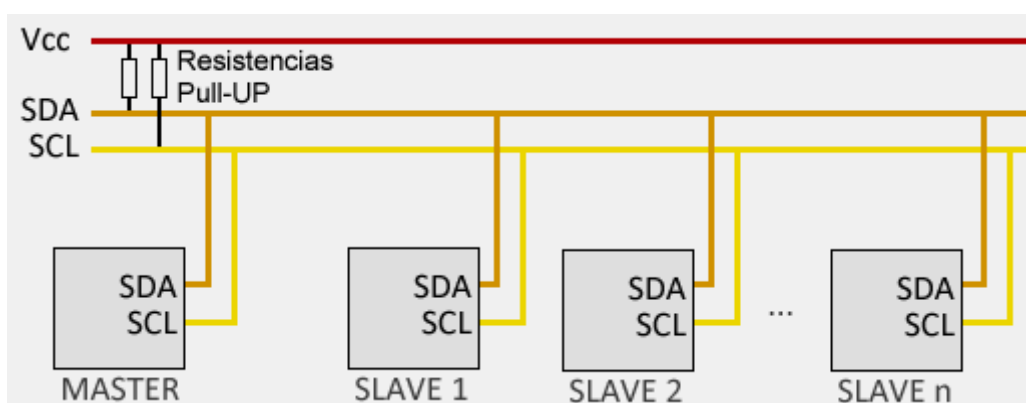


CC-BY-SA [Luis Llamas](#)

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



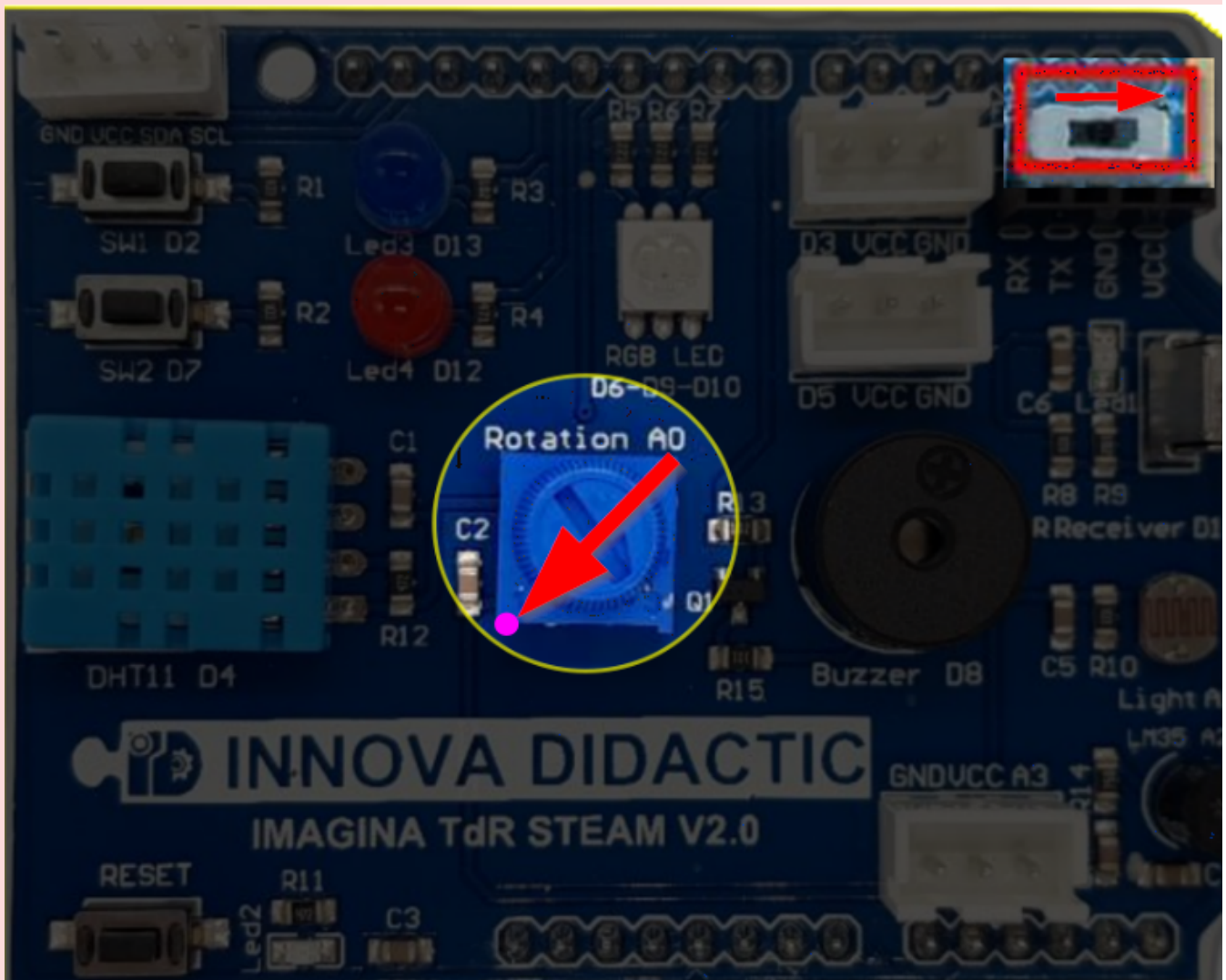
Igual que vimos en sensores, con las configuraciones Pull up y Pull down, las salidas de cualquier actuador **PUEDEN SER LÓGICA INVERTIDA**, mira en este NodeMCU (básicamente es un Arduino con wifi, aquí está conectado con el servidor Blynk) como funciona al revés, cuando apretamos, por lo tanto enviamos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

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

# Sensores y actuadores de la placa Imagina TDR STEAM

**ATENCIÓN**, ANTES DE SUBIR UN PROGRAMA A TU PLACA TIENES QUE

- TENER EL POTENCIÓMETRO (8) A CERO, DE LO CONTRARIO PUEDE DAR ERROR AL SUBIRLA **FLECHA DEL POTENCIÓMETRO QUE APUNTE AL PUNTO ROJO**
- El interruptor del Bluetooth tiene que estar **hacia al borde**



Adaptado de Federico Coca

### Notas sobre ESP32 STEAMakers CC-BY-SA

- El motivo es que el potenciómetro comparte la conexión A0 (GPIO02) con el sistema de grabación del programa y si el mismo no está en su posición de cero resistencia se producirá un error en el envío del programa a la placa porque se entenderá que los pines de transmisión están ocupados con otra tarea. (Federico Coca Notas sobre ESP32 STEAMakers CC-BY-SA)
- El interruptor del Bluetooth tiene que estar **hacia al borde** de lo contrario no se comunica con el puerto COM

¿Y si no lo hago qué pasa? Pues esto:

🔍 arduinoblocks :: Info

✖ Error compilando o subiendo

(", None, 'Error during Upload: uploading error: uploading error: exit status 2')

Esta página es extraída de Actividades con Imagina TDR STEAM y ArduinoBlocks

<https://github.com/arduinoblocks/libros>

Autoría: Equipo de Innova Didàctic y Robolot Team. Licencia **CC-BY-NC-ND**

La placa Imagina TDR STEAM es una placa didáctica desarrollada por el equipo ROBOLOT que presenta la gran ventaja de tener una gran cantidad de sensores, actuadores y conexiones de expansión incorporados directamente en ella. Únicamente hay que conectar esta placa a una placa Arduino UNO (en nuestro caso, una placa compatible llamada Keyestudio UNO) y ya está todo listo para empezar a programar.

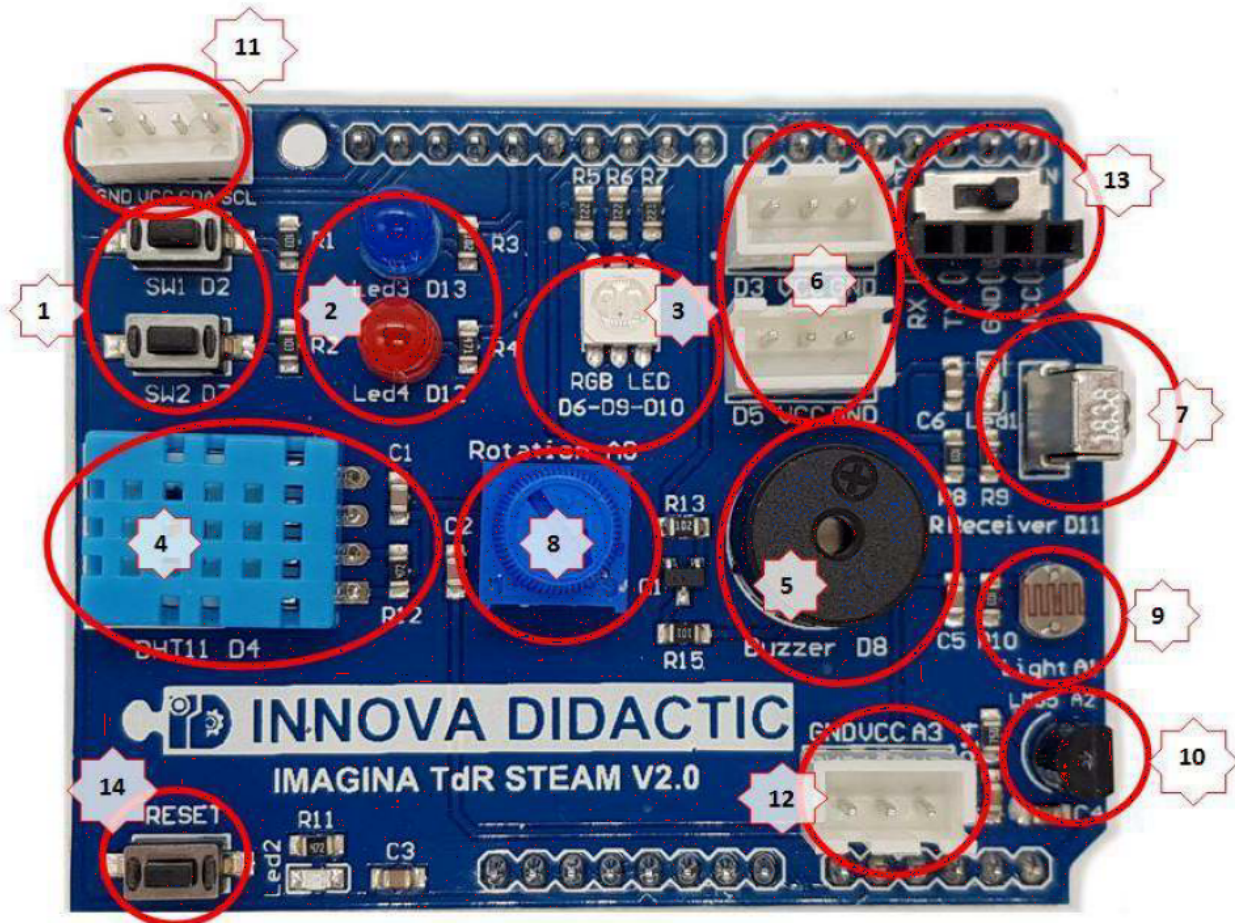


Tabla con la relación de elementos que hay en la placa Imagina TDR STEAM y sus conexiones:

	<b>Sensor/ Actuador/ Módulo</b>	<b>Pin de conexión</b>
1	Dos pulsadores (SW1, SW2)	D2 y D7
2	Dos leds (Azul Led3 y Rojo Led4)	D13 y D12
3	Led RGB	D6-D9-D10
4	Módulo DHT11 Sensor de Temperatura y Humedad	D4
5	Zumbador o Piezoeléctrico	D8
6	Dos puertos (Entradas/Salidas) digitales	D3 y D5
7	Módulo receptor de infrarrojos (IR)	D11
8	Módulo potenciómetro giratorio	A0
9	Sensor de luminosidad (LDR)	A1
10	Sensor de temperatura (LM35)	A2
11	Interface I2C compatible con sensores y módulos Keyestudio	SDA-A4 SCL-A5
12	Puerto Entrada Analógico	A3
13	Conexión de comunicaciones Bluetooth y Wifi (Swich On/Off)	Rx - Tx
14	Botón de reinicio.	-

Esta página es extraída de Actividades con Imagina TDR STEAM y ArduinoBlocks  
<https://github.com/arduinoblocks/libros>

Autoría: Equipo de Innova Didàctic y Robolot Team. Licencia **CC-BY-NC-ND**

**ATENCIÓN: EN LA PLACA ESP32, SI SE UTILIZA LA COMUNICACIÓN WIFI DEJA DE FUNCIONAR A0 Y A1 ES DECIR EL POTENCIÓMETRO Y EL LDR**

<https://view.genial.ly/5ec3b33a8ba5700d1e8078ed>

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



# Sensores en la placa ESP32 plus STEAMakers

Página extraída de Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

La placa ESP32 Plus STEAMakers se basa en el microcontrolador ESP32-WROOM-32 que dispone de dos sensores internos. Se trata de un **sensor de efecto hall** y un **sensor de temperatura**.

La placa ESP32 Plus STEAMakers lleva implementado un **medidor de tensión e intensidad** con el que podemos saber la tensión de alimentación en todo momento. Si la tensión de alimentación baja de aproximadamente 4,8V la placa no funcionará correctamente (sobre todo la transmisión WiFi). Es recomendable realizar una verificación de la tensión que está entregando el puerto USB del ordenador. Si la tensión no es próxima a 5V deberemos cambiar el cable USB, alimentar el puerto USB de forma externa o alimentar la placa con una fuente de alimentación. Es decir, dispone de un sistema para poder medir el consumo de energía.

Los bloques para trabajar con estos sensores están en 'Sensores', entrada 'Integrados', y son los que vemos en la figura siguiente:



Lógica	 Sensor de campo magnético
Control	
Matemáticas	
Texto	 Temperatura °C
Variables	
Listas	
Funciones	 Medidor de energía <input type="text" value="Amps (A)"/>
TDR STEAM	 Medidor de energía > Calibrar I=0
ESP	
▶ Tiempo	 Medidor de energía > Reset Wh=0
▶ Entrada/Salida	
▼ Sensores	
Integrados	 Medidor de energía > Offset V= <input type="text" value="0"/>
Receptor IR	

# Software

Hemos visto que el ESP32 acepta código IDE Arduino, por lo tanto puede programarse como un Arduino, (acepta microPython, mBlock, etc... ) pero nuestra recomendación es programarlo con **bloques gráfico** con dos opciones :

- **STEAMAKERSBLOCKS** (antes ARDUINOBLOCKS) es el programa que se usa en este tutorial
  - es **lento** cargando el programa y hay que tener bien instalado Connector
  - sólo online
  - permite programas **IoT**
  - tiene una buena **gestión de alumnado** ideal para envío de proyectos y corregirlos
  - tiene un canal de Telegram de apoyo al docente  
[https://t.me/innovadidactic\\_comunidad](https://t.me/innovadidactic_comunidad)
- **MICROBLOCKS** El tutorial recomendado es:  
<https://pedroruizf.github.io/steamakersmicroblocks>
  - no permite muchos programas IoT
  - es muy rápido al ejecutar y no da problemas de configuración
  - local o online

[https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWIGvlyVJr83Yk56kAoYD7bXLnYDm\\_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWIGvlyVJr83Yk56kAoYD7bXLnYDm_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000)

Si quieres saber más sobre las diferencias entre programar con código, con bloques gráficos, en vivo, en carga... [aquí](#)