¿Qué es Arduino?

¿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

¿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

¿Qué se puede hacer con Arduino? ¿Algún ejemplo?

Realmente el límite lo marca tu imaginación pero por dar alguna pista, podrías diseñar un sistema para la apertura y cierre de la puerta de un garaje, hacer un robot móvil que detecte objetos o que siga una línea negra, crear un detector de luz y oscuridad, implementar un termómetro, controlar un cilindro neumático, etc…

En este manual tienes múltiples ejemplos de pequeños proyectos para el aula, aunque Arduino es una herramienta que también se utiliza en el ámbito profesional para monitorización de sensores y automatización a pequeña escala por su flexibilidad, fiabilidad y precio.

Copyright 2026

¿Qué son las entradas y salidas?

Mediante los conectores de Arduino correspondientes a las entradas y salidas podemos comunicar nuestros programas con el "mundo exterior". Si queremos leer el valor de la magnitud física medida por un sensor, por ejemplo una LDR que detecta el nivel de luminosidad, lo tendremos que hacer conectando el sensor a uno de los pines de entrada (en este caso analógicas) de la tarjeta.

De esta forma con una simple instrucción de lectura en el programa, podremos obtener el valor de la magnitud física. Si nuestra intención es actuar o "hacer algo" una vez leído el valor del sensor, por ejemplo encender un led si el sensor de luminosidad detecta oscuridad, tendremos que conectar el actuador (en este caso el led) a un pin de salida que proporcionará la corriente necesaria para activarlo.

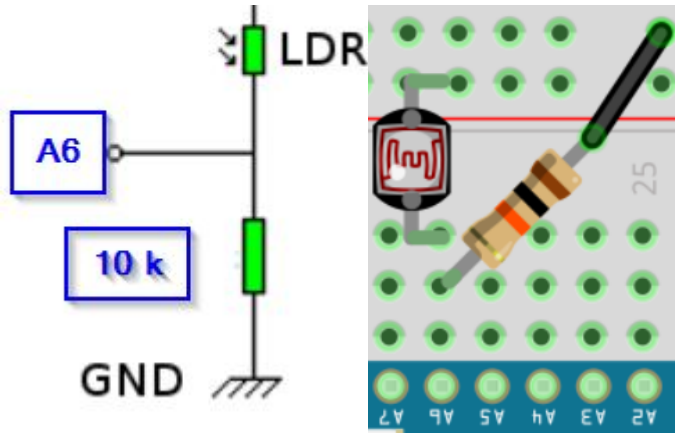
En Arduino las entradas pueden ser analógicas o digitales y las salidas sólo digitales. Cada pin digital tiene doble función entrada o salida. En la zona de configuración del programa hay que indicar explícitamente mediante una instrucción cuál es función desempeña un determinado pin.

¿Dónde se conectan los sensores? ¿A las entradas analógicas o digitales?

La mayoría de sensores miden señales analógicas y proporcionan una variación de voltaje dentro de un rango (normalmente de 0 a +5V) dependiendo de lo que varíe la magnitud física medida. Muchos sensores son resistivos a algo (luz, temperatura, humedad,...), es decir que varían su resistencia eléctrica con la magnitud física, pero mediante un sencillo montaje de divisor de tensión conseguimos una variación de voltaje apta para Arduino. Estos montajes los veremos en las prácticas.

Veamos este ejemplo:
El sensor LDR es una resistencia que cambia según la intensidad de la luz. La estrategia es colocar el LDR en *un divisor de tensión* con otra resistencia de valor parecido al promedio del que da el LDR (por ej 10k), y el valor del punto de unión proporciona una tensión entre 0 y 5V. Como es una señal analógica, la conectamos a una entrada analógica (en la figura al A6)

Copyright 2026



Una vez realizadas las conexiones, si midieramos la salida del sensor con un voltímetro nos daría un valor decimal, por ejemplo un nivel de luz “intermedio” (rango de 0 a 5V) de un sensor de luz podría dar 3,3 voltios. Este tipo de información el microcontrolador **no la entiende tal cual**, sólo es capaz de interpretar números binarios (“0” ó “1”) por lo que para traducir los valores analógicos dispone internamente de un conversor analógico – digital que hará la conversión entre los dos sistemas, de forma que podremos tener valores discretos de la medida de los sensores analógicos. En el Arduino **las entradas analógicas** leen valores analógicos entre 0V y la alimentación (normalmente 5V) y los convierten en números entre **0 y 1023** (porque lo codifica en 10 dígitos binarios proporcionan $2^{10} = 1024$ combinaciones).

> Por ejemplo, si la entrada analógica lee un valor de 3,3V y la fuente de alimentación es 5V, la señal analógica que lee Arduino, haciendo una regla de 3, tiene un valor de 3,3 * 1023 / 5 = 675,18 = **675**

> **Mapeo**
> Para convertir estos valores 0 -1023 a valores más legibles, por ejemplo 0 - 100 para representarlo en % o 0-5 para que represente la medida en voltios ... veremos en programación la función **mapear**

La mayoría de los sensores nos lo venden ya preparados montados en una pequeña placa electrónica y con circuitos integrados auxiliares para no tener que estar haciendo divisores de tensión. Pueden tener salida analógica o **salida digital**, que en este caso lo tenemos que conectar a cualquier entrada digital D0 hasta D13.

Veamos el mismo ejemplo del LDR: Podemos comprar este módulo:

Copyright 2026



Estos módulos proporcionan 3 pines: dos que son la alimentación, (0V, GND o -) y (+5V V+o Vcc) y el pin que proporciona la lectura (Vout o D0 o I/O). En el caso de que sea un sensor que mida una magnitud analógica como en este caso la luz, suelen proporcionar un potenciómetro para determinar qué luminosidad se considera un 0 o un 1.

¿Hay sensores digitales que no estén en una placa electrónica?

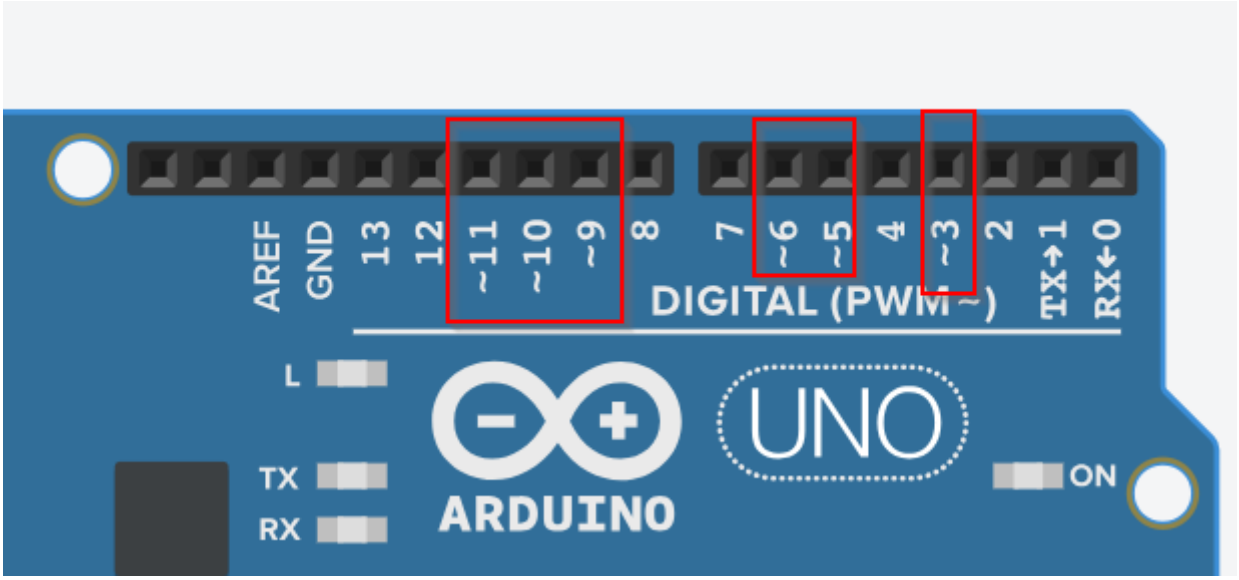
Las entradas digitales sin una placa electrónica son cuando las señales a leer son valores discretos. Por ejemplo queremos poner un pulsador o un interruptor que encienda un led. Hacemos un montaje que cuando se pulse, entren 5 voltios en el pin digital de entrada y cuando no se pulse que “entren” 0 voltios. De esta manera la lectura del pin digital de entrada será “HIGH” con 5 voltios o “LOW” con 0 voltios.

Veremos más adelante que un interruptor no es un simple cable que conectamos a +5V o a 0V pues ¿Qué valor lee Arduino mientras levantamos el cable de un sitio a otro?, para ello veremos configuraciones Pull-up o Pull-down que se repiten en muchos sensores.

¿Qué son las salidas digitales etiquetadas con PWM (~)?

Son salidas digitales que **simulan una salida analógica**. Las siglas significan *Modulación por Ancho de Pulso (Pulse Width Modulation)* es decir, proporcionan una onda cuadrada con un nivel alto (+5V) de “cierta” duración.

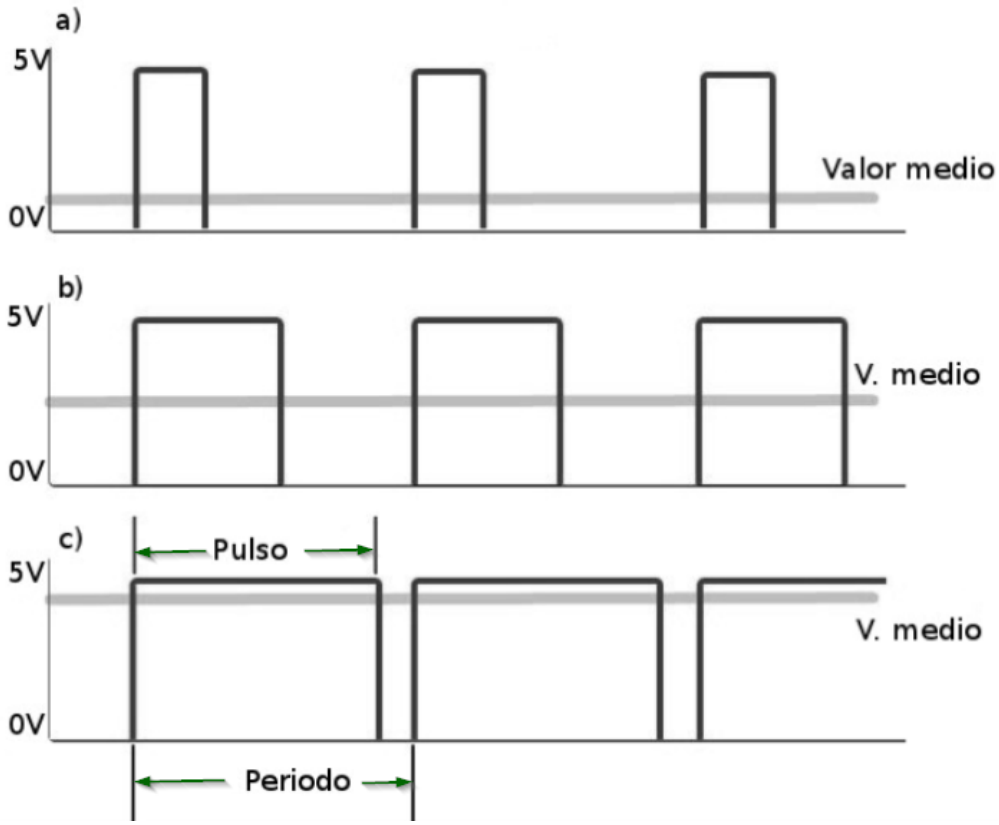
Copyright 2026



Los valores PWM que podemos proporcionar pueden ir desde **0 a 255.**

- Un valor bajo es una señal cuadrada donde el pulso es pequeño comparado con el periodo (ver figura primera señal) por lo tanto el Valor medio es bajo.
- Un valor mitad, es decir 255/2 = 127 o 128, corresponde a una señal cuadrada perfecta (ver figura segunda señal) por lo que el Valor medio es la mitad, si 5V es la tensión de alimentación, sería 2,5V.
- Un valor alto es una señal cuadrada donde el pulso es grande comparado con el periodo (ver figura la tercera señal) por lo tanto el Valor medio es alto
- Un valor 0 corresponde a un 0V analógico y sería una señal sin pulso
- Un valor 255 corresponde a la máxima tensión (la de alimentación (normalmente 5V).
- La frecuencia es de 490Hz para los pines 3, 9, 10, 11 y de 980Hz para los pines 5 y 6 en un Arduino UNO o Nano

Copyright 2026



De esta manera podemos simular señales analógicas, esto es muy útil para activar servomotores y llevarlos a una posición determinada o variar la luminosidad de un led o en los motores de los robots que vayan más deprisa o más despacio

¿Puedo accionar motores DC con Arduino?

Si son motores muy, muy pequeños sí sería posible aunque no es recomendable, pueden dañar la placa. Los motores necesitan un consumo alto de corriente, sobre todo si tienen que mover cierta carga, por lo que se recomienda o bien utilizar una tarjeta Shield o extensión de Arduino que dispone de circuitería apta para proporcionar dicha corriente (transistores).

- En el curso Arduino con código utiliza una Shield llamada *Edubásica* que dispone de un transistor y un circuito integrado LM293 para realizar esta función, además de otras ventajas para el aprendizaje de Arduino.
- En el curso Rover con Arduino utiliza Shield motor para NodeMCU

Revision #3
Created 2022-04-21 17:01:50 CEST by Juanjo Mejías
Updated 2022-11-02 20:00:07 CET by Juanjo Mejías

Copyright 2026