

ARDUINO

1. ¿Qué es Arduino?.....	2
2. Arduino Uno.....	3
2.1 Características técnicas de Arduino Uno.....	4
2.2 Conexión de componentes.....	6
2.3 Programación de Arduino.....	8
2.3.1 Programas de ejemplo.....	10
3. Enlaces de interés:.....	12

1. ¿Qué es Arduino?

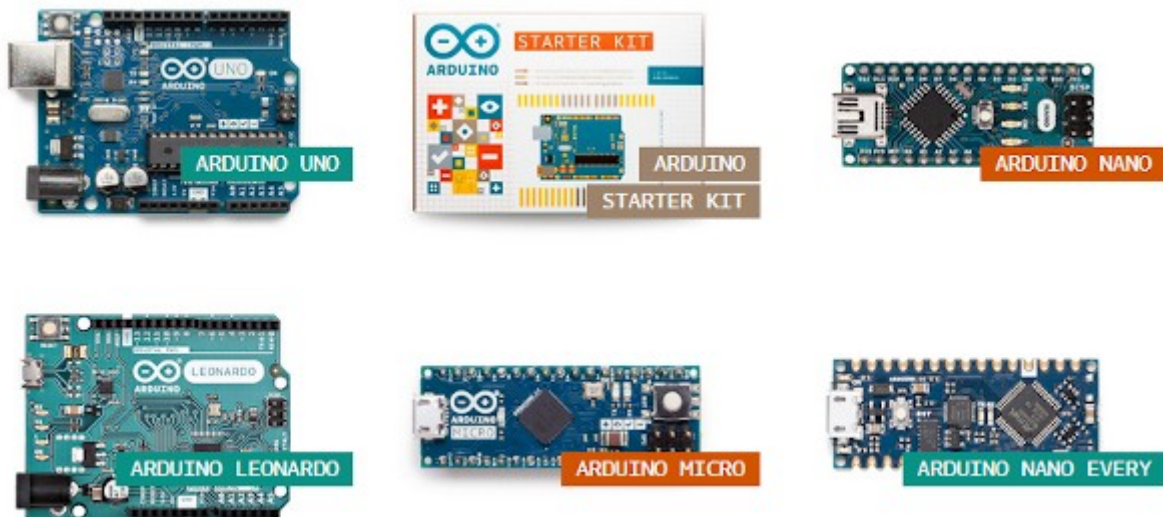
Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activando un motor, encendiendo un LED, publicando algo en línea.

A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto; sus contribuciones se han sumado a una increíble cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos.

Arduino es multiplataforma: el software Arduino (IDE) se ejecuta en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.

El entorno de programación es simple y claro: el software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible para que los usuarios avanzados también lo aprovechen.

<https://www.arduino.cc/en/Guide/Introduction>



<https://www.arduino.cc/en/Main/Products>

2. Arduino Uno

Dentro de las placas Arduino, hay muchos tipos de placas con diferentes características y potenciales. Hay placas de pequeño tamaño para integrarlas en proyectos que necesiten poco espacio como “**Arduino Nano**”, otras tienen integrados diferentes funciones como el “**Arduino Uno wifi**” donde tenemos integrado un emisor y receptor **wifi**, “**Arduino Ethernet**” que tiene **conexión con Internet**. Hay un largo etcétera de **Arduinos** con complementos integrados.

<https://www.arduino.cc/en/Main/Products>

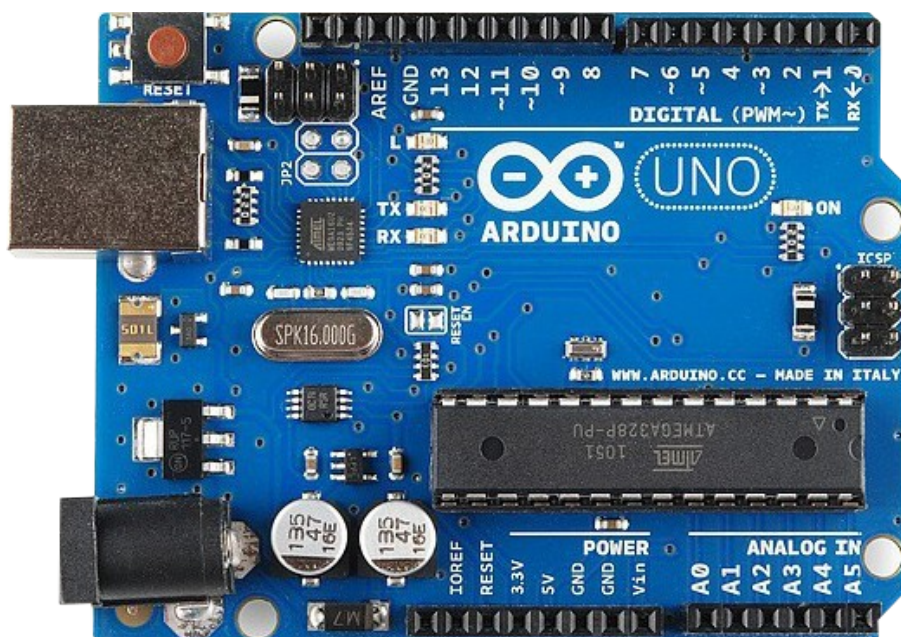
Dentro de **Arduino** como es Software y Hardware que lo puede usar cualquiera al ser libre y abierto, han salido muchas versiones de otros fabricantes, sobre todo las versiones CHINAS, mucho más baratas y que no funcionan mal.

También destacan otras por su compatibilidad con Arduino como WEMOS, que integra compatibilidad con wifi muy funcionales.

<https://www.wemos.cc/en/latest/>

De las placas de Arduino uno de los más populares es “**Arduino Uno**”, que tiene unas cuantas conexiones con el exterior, tanto de entrada como de salidas tanto digitales como analógicas. Por estas entradas y salidas nos conectamos a otra circuitería que hay en el exterior para hacer diferentes cosas: encender y apagar diodos, conectarse a sensores de temperatura, conexión WIFI o BlueTooth, Internet y a todo lo que la imaginación nos permita. De hecho hay muchos módulos de Arduino que nos permiten no tener que diseñarlos nosotros. Puede ser programado desde el ordenador y luego enviar la información por un puerto USB (como todas la demás).

ARDUINO UNO



2.1 Características técnicas de Arduino Uno

Todos los componentes electrónicos tienen unas hojas de características, que describen los datos técnicos, que hay que conocer antes de usarlos, describen valores técnicos y esquemas de circuitos que puedes montar, etc.

Estos son algunos valores técnicos de la placa “**Arduino Uno**”.

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V (Alimentación)
Input Voltage (limit)	6-20V (Alimentación)
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

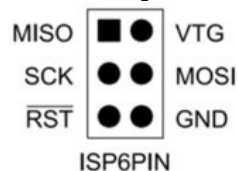
De los pines destacan:

Vin: voltaje de entrada, por este pin también se puede alimentar la placa. De 5 a 12 V.

2 salidas, una de **5V** y otra de **3.3 V**.

3 salidas de **GND**.

ICSP (In circuit serial programming) 6 PINES macho. ICSP es un conector consistente en 6 señales: MOSI, MISO, SCK, RESET, VCC, GND y además de ser un puerto para programar Arduino, también es el conector de expansión del bus SPI mediante el que también podemos comunicar periféricos y es usado en algunos casos para comunicar Arduino con los shields.



De las 14 entradas digitales:

PWM = modulación de ancho de pulso. De las entradas digitales son las 3,5,6,10 y 11.

Permite una modulación de 8 bits usando la función **analogWrite ()**. Esto quiere decir que emulan una señal analógica con una salida digital modulada.

Transmisión serie: Rx Tx. Las patillas 0 y 1. Permite a Arduino convertirse en un transmisor y receptor de datos.

Modo de transmisión SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Otro sistema de transmisión diferente al serie.

Activación de interrupción: las patillas 2 y 3. Detecta una de las patillas un suceso y se pone en marcha una rutina. Se activa con la función `attachInterrupt()`.

LED 13: es un LED integrado (built-in) conectado a la entrada digital 13. cuando la entrada digital de la patilla 13 es 1 el diodo está encendido y viceversa.

Entradas analógicas: son 6.

10 bits de resolución. Esto quiere decir 1024 valores diferentes.

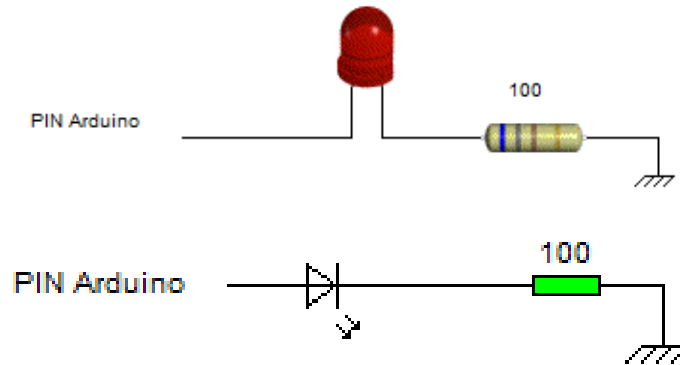
Comunicación I^2C : 4 (SDA) y 5 (SCL). A través de la librería Wire.

La medición a tierra es de 5 V, aunque es posible cambiarla usando el pin AREF y la función `analogReference()`.

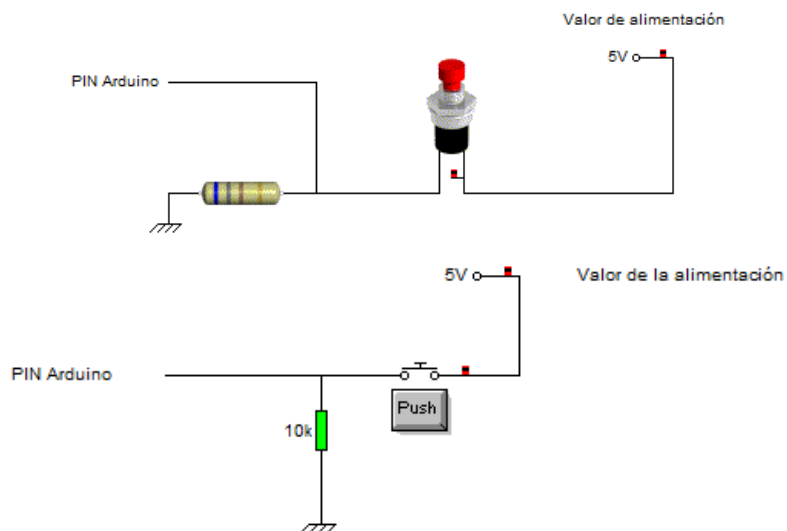
2.2 Conexión de componentes

Para conectar los componentes electrónicos a los pines de Arduino hay que hacerlo de la forma adecuada.

Por ejemplo un **diodo** se conecta de la siguiente forma:

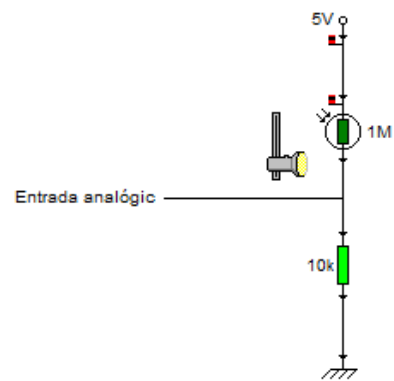


Para un **pulsador** la conexión se hará de la siguiente forma:



Captador analógico de señal luminosa con una **LDR**:

7
Arduino



2.3 Programación de Arduino.

Para hacer que Arduino funcione tenemos que usar 2 funciones:

```
void setup() {  
  
}  
  
void loop(){  
  
}
```

La función primera "**void setup()**" sirve para configurar el dispositivo. Este se programará con instrucciones incluidas entre las llaves.

La función "**void loop ()**" es donde se incluirá un programa para realizar una determinada tarea, según la configuración de la función anterior.

Funciones esenciales.

De configuración:

- "**pinMode(PIN,INPUT/OUTPUT);**". Se configura uno de los pines digitales (del 0 al 13) como salida o como entrada de. Como salida sacamos unos 5 V, y como entrada valoramos la tensión que haya en la circuitería de salida.

De programación:

- "**analogRead(PINanalógico);**". Leerá el valor analógico del puerto analógico (del 0 al 5). Ejemplo: **analogRead(3);**
- "**digitalWrite(PIN,LOW/HIGH);**". Se saca por la salida del PIN (del 0 al 13) un valor 0 (LOW) o 1 (HIGH). Por ejemplo para encender (5V) o apagar un diodo (0V). Ejemplo: **digitalWrite(3,HIGH);**
- "**digitalRead(PIN);**". Lee el valor que hay en la entrada digital. Ejemplo:
digitalRead(10);

Generales:

if: es una función condicional y significa en castellano SI. Sirve para hacer una determinada acción SI ocurre un suceso. Por ejemplo: si se pulsa el pulsador del circuito se enciende una bombilla:

```
if (digitalRead (10)==1) digitalWrite (3,HIGH);
```

Significa que si el valor que hay en la patilla 10 es 5 voltios (1 lógico) se enciende la bombilla que hay en la patilla 3.

delay();. Retraso de un tiempo en milisegundos. 1000ms es un segundo.

Ejemplo: **delay(2000);**

Pasan 2 segundos (2000ms) sin que se haga nada

for: es un bucle para hacer una acción varias veces.

Ejemplo:

```
int i;
```

```
for (i=1;i<=5;i++)  
{  
  digitalWrite(3,HIGH);  
  delay(1000);  
  digitalWrite(3,LOW);  
  delay(1000);  
}
```

Dentro de for:

i=1 significa que el valor inicial del conteo es 1.

i<=5 significa que el valor máximo de i será 5.

i++ significa que el valor de i en cada paso se incrementa 1. Primero 1, luego 2 , luego 3, luego 4 y al final 5. Podría se i-- y disminuiría el valor de 1 en 1.

En este ejemplo se enciende y se apaga una luz 1 segundo 5 veces. Antes se declara la variable **i**.

2.3.1 Programas de ejemplo.

1) Como encender un diodo con un pulsador y que se quede activa la luz 2 segundos y luego se apague:

Vamos a suponer que el diodo es la **salida digital 3**

Vamos a suponer que el pulsador es la **entrada digital 13**

```
void setup() {  
    pinMode (13,INPUT); // Programo a patilla 13 entrada pulsador.  
    pinMode (3,OUTPUT); // Programa la patilla 3 para el diodo  
    digitalWrite (3,LOW); // Valor de salida del diodo 0 (apagado)  
}  
  
void loop(){  
    if (digitalRead (13)==1) // Si se pulsa el botón  
    {  
        digitalWrite (3,HIGH); // El diodo se enciende  
        delay (2000); // Se para el proceso 2 segundos  
        digitalWrite (3,LOW);  
    }  
}
```

2) Al pulsar el botón de la entrada digital 12 se encienda el diodo de la entrada digital 4 y se encienda y se apague 10 veces. Se enciende 1 segundo y se apaga 1 segundo.

```
void setup() {  
    pinMode (12,INPUT); // Programo a patilla 12 entrada pulsador.  
    pinMode (4,OUTPUT); // Programa la patilla 4 para el diodo  
    digitalWrite (4,LOW); // Valor de salida del diodo 0 (apagado)  
}  
  
void loop(){  
    int i;  
    if (digitalRead (12)==1) for (i=1;i<=10;i++)  
        // Si se pulsa el botón que indicado 12 empieza el bucle  
        {  
            digitalWrite(4,HIGH);  
            delay(1000);  
            digitalWrite(4,LOW);  
            delay(1000);  
        }  
}
```

3) Como programar un display:

En este ejercicio se va a usar por primera vez una librería, esto es un pequeño programa con funciones para manejar un display:

Para incluir la librería se hace pinchando en: **Programa --> Include library --> Display4056A**

Aparecerá un texto en la parte superior del programa que pone: **#include <TM1637.h>**

Incluiremos un pulsador (entrada), y al presionar sobre el activaremos una rutina para que cuente de 0 a 10 con un tiempo de 0,5 segundos cada paso. El brillo del display será 3.

```
#include <TM1637.h> // Librería que contiene las funciones para manejar un display.
#define CLK 3        // El pin digital 3 es CLK
#define DIO 2        // El pin digital 2 es DIO
TM1637 disp(CLK,DIO);
void setup ()        // Función de configuración.
{
  pinMode (13,INPUT); // Configura la patilla 13 como entrada de un pulsador.
  disp.set(3);        // Configuración del brillo del Display. Brillo mínimo 0, máximo 7.
  disp.init(D4056A);  //D4056A es el tipo de Display.
}

void loop ()         // Rutina o secuencia de instrucciones.
{
  int i;
  if (digitalRead (13)==1)
  {
    for (i=1;i<=10;i++)
    {
      disp.display(i);
      delay (500);
    }
    disp.init(D4056A); // Se apaga el display cuando termine el conteo.
  }
}
```

4) Conteo hacia atrás: se hará de la misma forma que el apartado anterior, pero en el bucle **for** el valor inicial será el valor máximo, por ejemplo 18, y en vez de incrementarse, se decrementa de 1 en 1 usando el parámetro **i--**. Ejemplo: **for (i=18; i>=0;i--)**
i=18 : significa que el valor inicial del conteo es 18.
i>=0 : significa que en todo el proceso el valor de i es mayor o igual que 0.
i-- : el valor se decrementa de 1 en 1. 18,17,16....., 0.

3. Enlaces de interés:

En inglés. Es la más completa. Página oficial.

<https://www.arduino.cc/>

En español

<http://arduino.cl/>

Funciones de programación

[https://www.arduino.cc/reference/en/?](https://www.arduino.cc/reference/en/?_gl=1*1ts9l0s*_ga*MTMyNjgzODE1OS4xNjUyNjA1MzUw*_ga_NEXN8H46L5*MTY1MjYzMDM5MS4yLjEuMTY1MjYzMDQ3OS40MA..)

[_gl=1*1ts9l0s*_ga*MTMyNjgzODE1OS4xNjUyNjA1MzUw*_ga_NEXN8H46L5*MTY1MjYzMDM5MS4yLjEuMTY1MjYzMDQ3OS40MA..](https://www.arduino.cc/reference/en/?_gl=1*1ts9l0s*_ga*MTMyNjgzODE1OS4xNjUyNjA1MzUw*_ga_NEXN8H46L5*MTY1MjYzMDM5MS4yLjEuMTY1MjYzMDQ3OS40MA..)

Programación

<https://arduinobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>

Fuentes:

<http://digital.csic.es/bitstream/10261/127788/7/D-c-%20Arduino%20uno.pdf>

<https://aprendiendoarduino.wordpress.com/2016/11/06/icsp/>

<https://www.arduino.cc/>

Mis conocimientos.