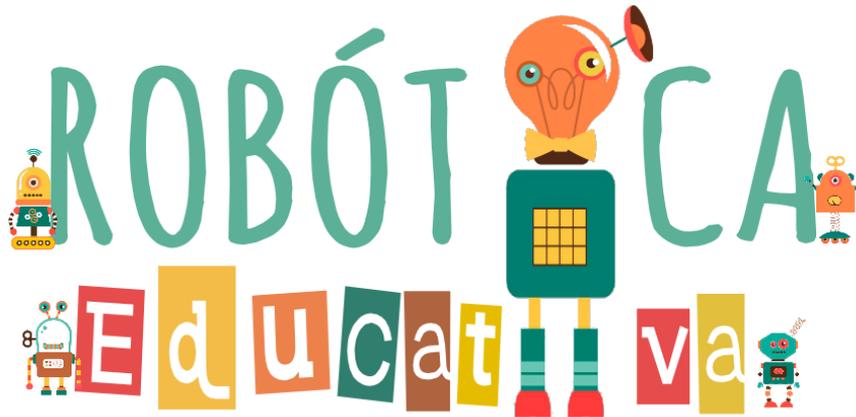
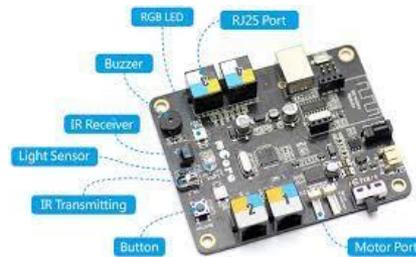




Robótica Educativa



3.4 MICROCONTROLADORES

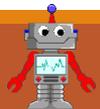


CONTENIDOS

1.- Introducción

2.- Definición y Microcontroladores en la Robotica educativa

3.- Memorias y periféricos



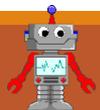
INTRODUCCIÓN

Que es un microcontrolador? , Que es un micro procesador?

Los microprocesadores y los microcontroladores son dispositivos electrónicos que tienen la capacidad de llevar a cabo procesos lógicos.

El microprocesador (CPU), tiene la característica de que sus unidades están físicamente separadas, esto significa que interactúa con una memoria RAM, una memoria ROM y con dispositivos de entrada y salida por medio de buses de comunicación.

El microcontrolador, por su parte, es un solo dispositivo que internamente contiene todo lo necesario para poder llevar a cabo sus acciones. Este contiene su propio CPU, memorias RAM y ROM y dispositivos de entrada y salida. Por lo tanto es superior al microprocesador debido a su reducido tamaño y capacidad de ser implementado en circuitos electrónicos



INTRODUCCIÓN

Conceptos fundamentales :

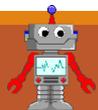
Unidad Central de Proceso (CPU).- Es el "cerebro" de una computadora, de manera más precisa, es la parte de una computadora que se encarga de ordenar y controlar el proceso y la transferencia de información. La CPU interpreta las instrucciones del programa y coordina su ejecución.

Microprocesador (μp).- Es una CPU en un sólo circuito integrado.

Microcomputadora (μc).- Es una computadora cuya CPU es un μp .

Microcontrolador (μcc).- Es una microcomputadora en un sólo circuito integrado.

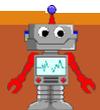
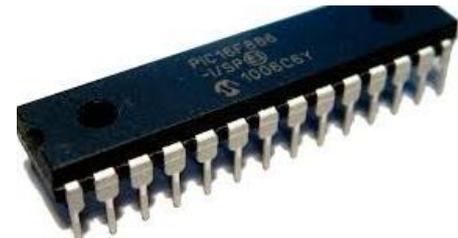
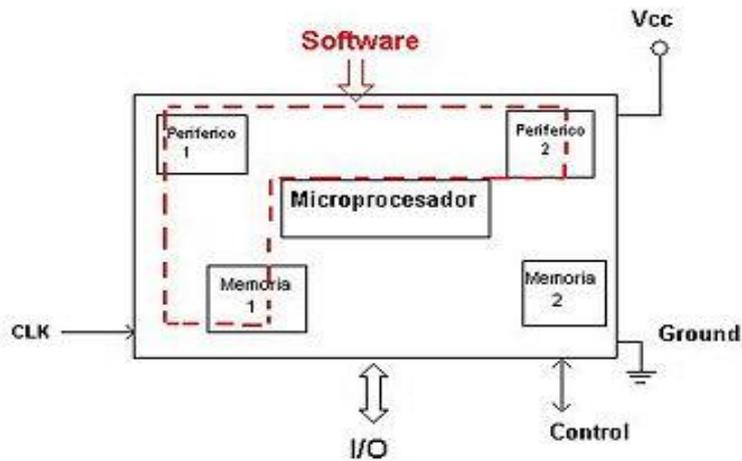
Computadora.- Una computadora es un sistema secuencial síncrono programable, la cual para desempeñar sus funciones debe poseer además de la CPU: - Conductos para el flujo de la información - Dispositivos para almacenar información - Dispositivos para comunicarse con el exterior



MICROCONTROLADOR

Definición

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora.



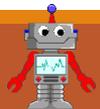
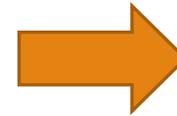
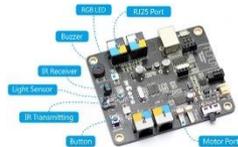
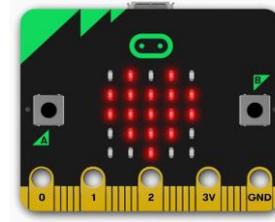
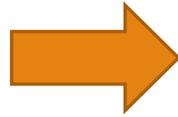
MICROCONTROLADOR

Definición

Se utiliza en diferentes Marcas

Con la finalidad de controlar algo

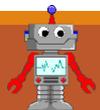
Base del Microcontrolador
PIC



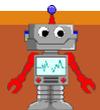
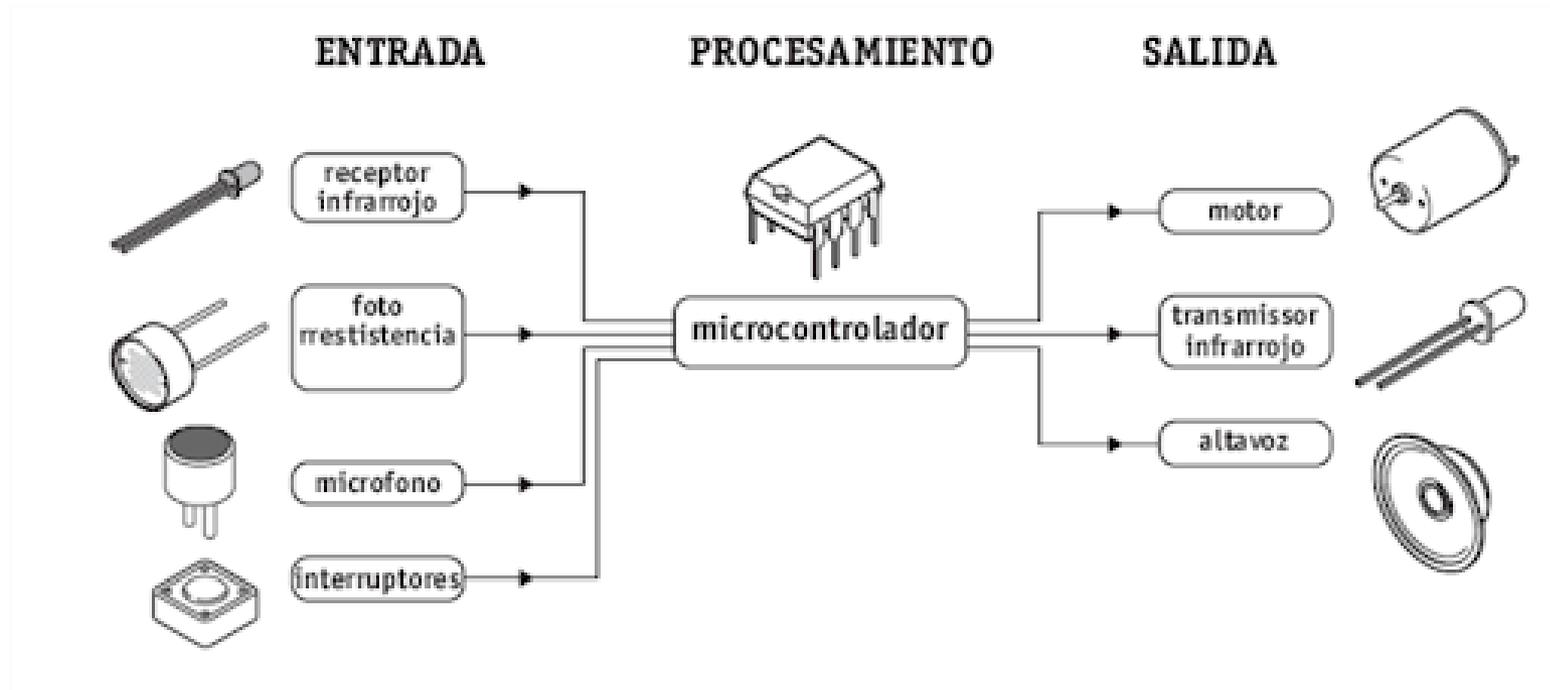
MICROCONTROLADOR

El microcontrolador realiza procesos lógicos.

Estos procesos son programados en un computador utilizando un lenguaje de programación, generalmente lenguaje Assembler, y se insertan o graban en la memoria del microcontrolador mediante un “programador de PIC”.



MICROCONTROLADOR



MICROCONTROLADOR

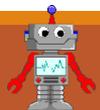
GAMAS DE MICROCONTROLADORES

Se clasifican en tres gamas:

Base, Media y Mejorada

Con un total de casi 300 modelos diferentes.

Con Diferentes capacidades de memoria, periféricos y encapsulados.



MICROCONTROLADOR

GAMAS DE MICROCONTROLADORES

GAMA BASE

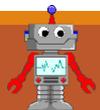
- - Responden a 33 instrucciones de 12 bits de longitud cada una y disponen de una pila con dos niveles de profundidad.
- - Esta compuesta por 14 modelos, de los cuales 6 están encapsulados con 6 pines, por ello se les llama enanos.

GAMA MEDIA

- GAMA MEDIA
- CARACTERÍSTICAS:
 - - Dispone de un repertorio de 35 instrucciones.
 - - 14 bits de longitud cada una.
 - - Pila de 8 niveles.
 - - Un vector de interrupción.
 - - 71 modelos diferentes.
 - - Encapsulados de 8 a 64 pines.

GAMA MEJORADA

- CARACTERÍSTICAS:
 - - Dispone de 77 instrucciones.
 - - 16 bits de longitud cada una.
 - - Pila de 16 niveles.
 - - Dos vectores de interrupción.
 - - Es la gama que más número de dispositivos posee.
 - - Memoria programable que puede alcanzar los 128 KB



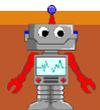
MICROCONTROLADOR

Que es un PIC?

Los PIC son una familia de microcontroladores desarrollados y fabricados por la empresa Microchip Technologies Inc., los cuales cuentan con una tecnología tipo RISC (*Reduced Instruction Set Computer*) y poseen en su arquitectura interna características especiales que varían según el modelo de PIC que deseamos utilizar.

PIC = CONTROL INTEGRADO DE PROSESOS

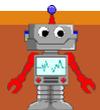
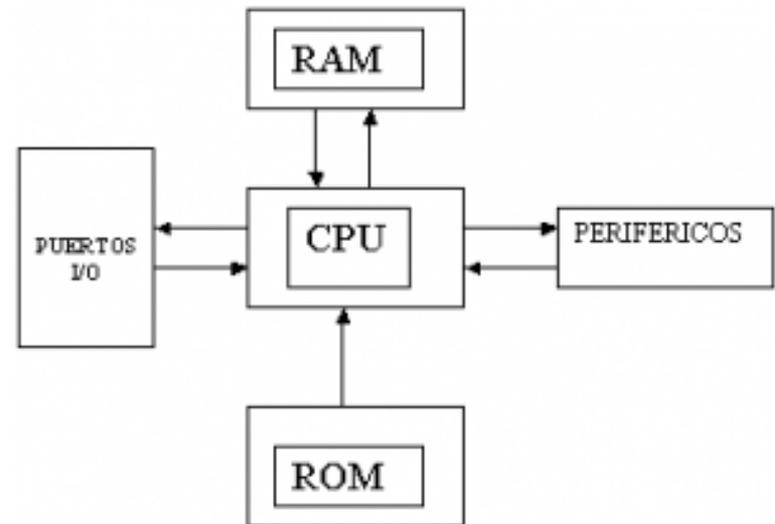
Podríamos decir que estos dispositivos se asemejan a una computadora pero de tamaño muy reducido,



MICROCONTROLADOR

PIC

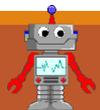
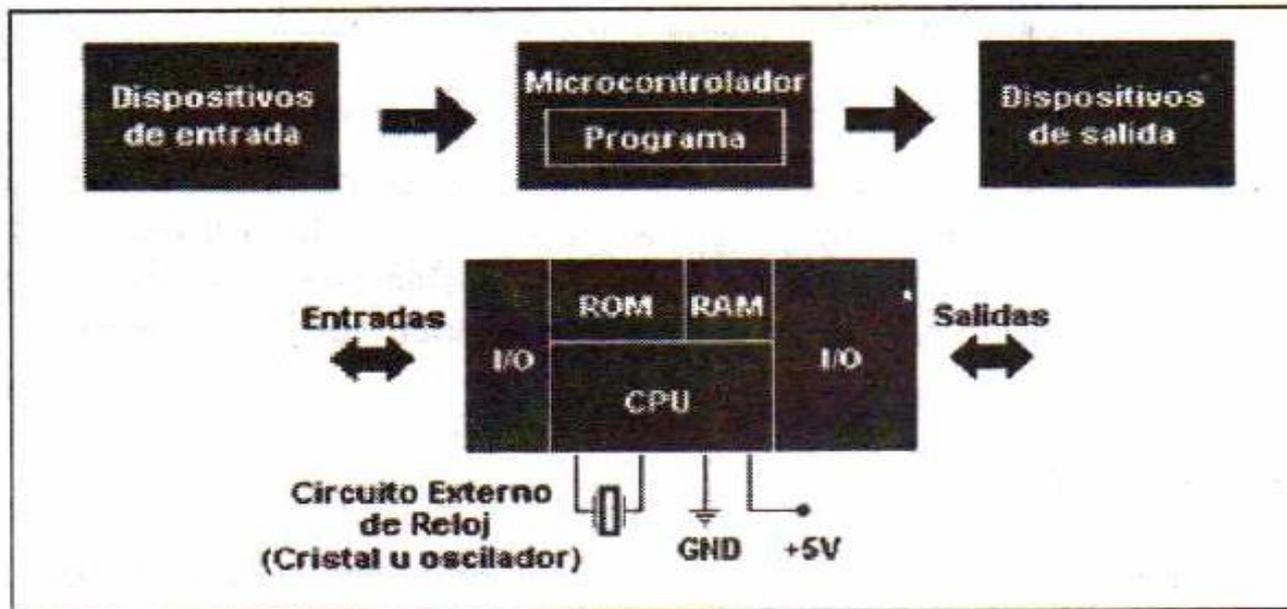
El microcontrolador es el encargado de dirigir todos los procesos de un circuito electrónico, en base a las instrucciones de programa o rutinas que definen funciones específicas de control, donde las mismas serán realizadas en lenguaje Basic para microcontroladores PIC.



MICROCONTROLADOR

MICROCONTROLADOR PIC

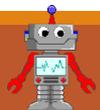
En sí un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica, sus partes o componentes principales son:



MICROCONTROLADOR

MICROCONTROLADOR PIC

- Memoria ROM (Memoria de sólo lectura)
- Memoria RAM (Memoria de acceso aleatorio)
- Líneas de entrada / salida (I / O) También llamados puertos
- Lógica de control Coordina la interacción entre los demás bloques

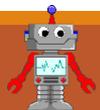


MICROCONTROLADOR

Tipos de arquitecturas de los microcontroladores:

Arquitectura Von Neumann.

Arquitectura Harvard.

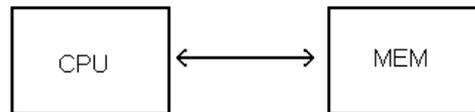


MICROCONTROLADOR

Arquitectura Von Neumann.

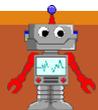
En este tipo de arquitectura, el CPU (Unidad Central de Proceso) está conectado a una única memoria en donde se guardan las instrucciones del programa y los datos. En otras palabras, el CPU está conectado a una ROM y una RAM por medio de un solo bus de comunicación.

El tener un único bus de comunicación, hace que el microcontrolador sea más lento, puesto que no puede acceder a la memoria a buscar una nueva instrucción mientras no finalice la transferencia de datos de la instrucción anterior



Las limitaciones

- 1) Por tener un único bus de datos, hace que el CPU tenga que hacer varios accesos a la memoria para buscar instrucciones complejas.
- 2) A causa del único bus de datos, se reduce el tiempo en realizar las operaciones, impidiendo superponer los tiempos de acceso.

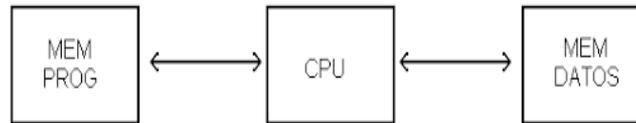


MICROCONTROLADOR

Arquitectura Harvard.

A diferencia de la arquitectura Von Neumann, la arquitectura Harvard tiene el CPU conectado a dos memorias por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa y la otra solo almacena datos.

Ambos buses son diferentes, y pueden ser de distinto ancho.

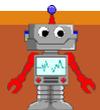


Tiene las siguientes ventajas:

- 1) Mayor velocidad, ya que puede acceder a la memoria de datos para completar la instrucción en curso y al mismo tiempo puede acceder a buscar una nueva instrucción.
- 2) El tiempo de acceso a ambas memorias puede superponerse, logrando mayor velocidad.

Desventaja

Es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas

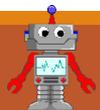


MICROCONTROLADOR

ARDUINO

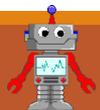
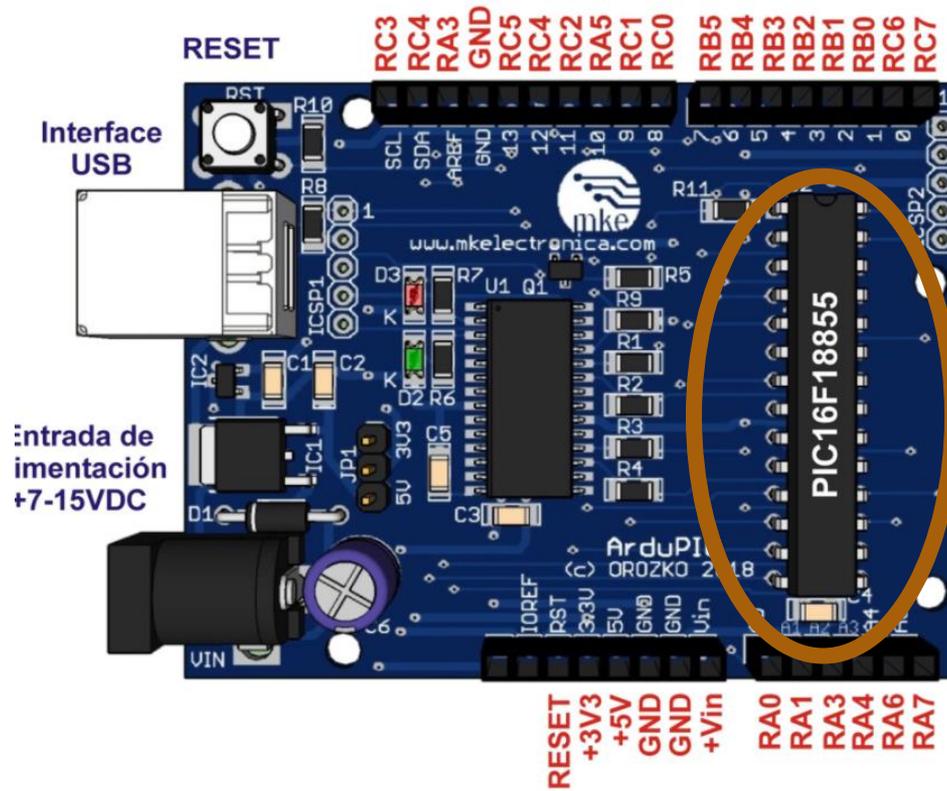
Arduino es una plataforma para prototipado de electrónica basada en hardware y software libre y fácil de utilizar. Podemos construir circuitos electrónicos y programarlos con esta placa.

Realmente lo que estamos haciendo es programar un microcontrolador, estos dispositivos electrónicos programables nos rodean en nuestro día a día, en el coche, nuestra casa, el trabajo, etc...



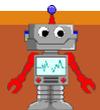
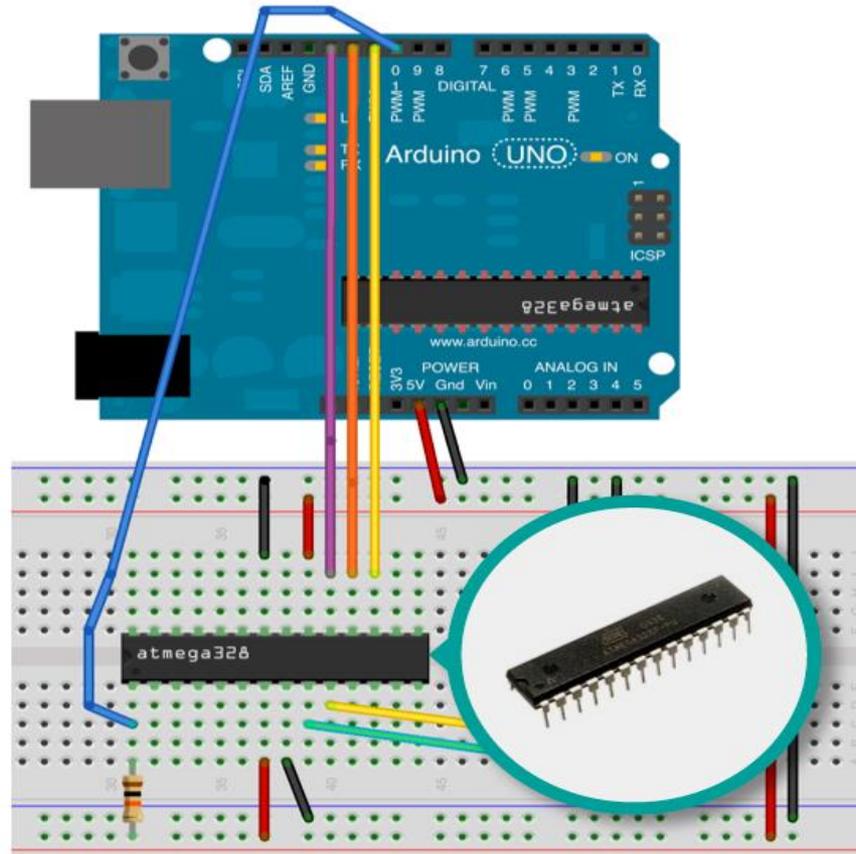
MICROCONTROLADOR

ARDUINO



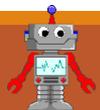
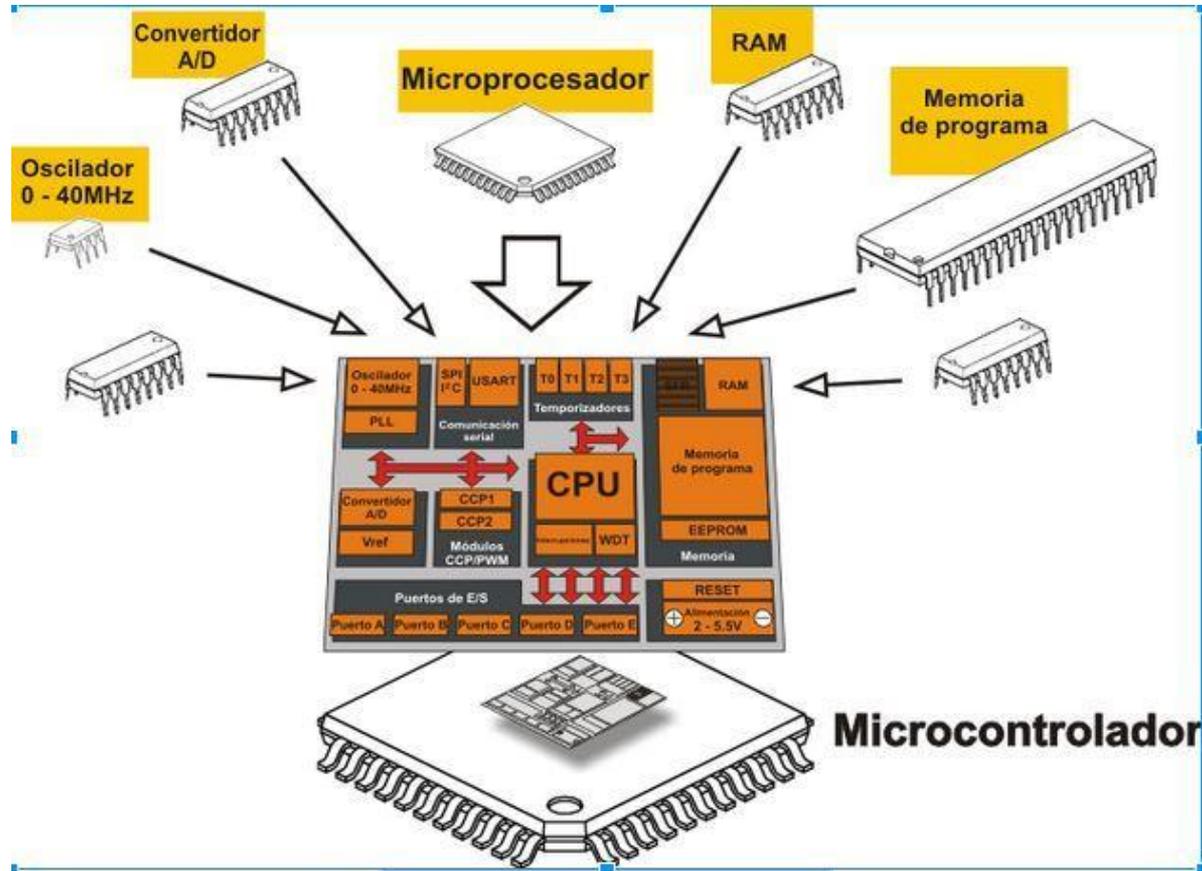
MICROCONTROLADOR

ARDUINO



MICROCONTROLADOR

ARDUINO

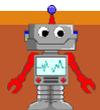


MICROCONTROLADOR

ARDUINO - Características

Las principales características que podemos encontrar en nuestra placa de Arduino UNO son las siguientes:

- El microcontrolador es un circuito integrado programable capaz de realizar operaciones matemáticas complejas a gran velocidad.
- La alimentación de una placa de Arduino es mediante el puerto USB mientras se está programando. Una vez programado podemos desconectarlo del ordenador y que trabaje de forma autónoma y se alimenta Arduino mediante una fuente de alimentación o pila de 9V.
- Tanto las entradas como las salidas dotan al sistema de información y realizan diferentes actuaciones.

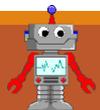


MICROCONTROLADOR

ARDUINO - Pines

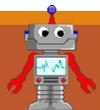
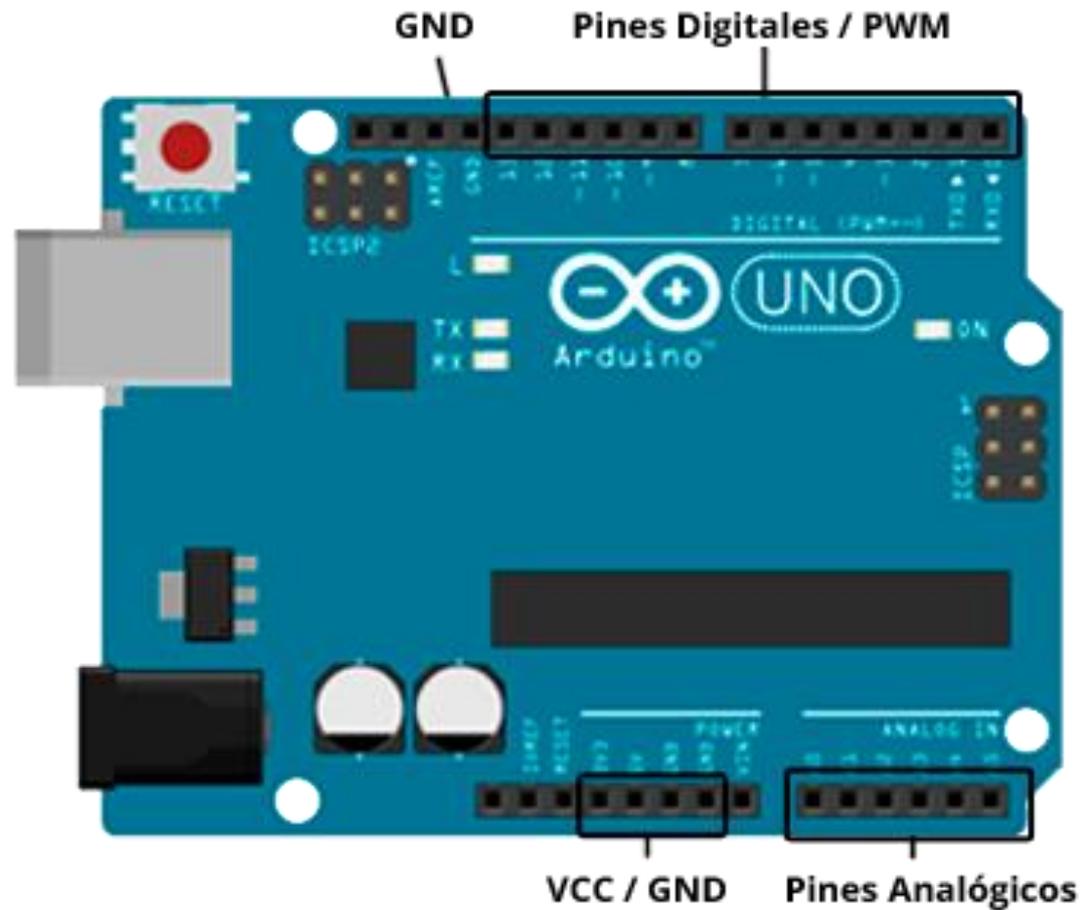
Arduino contiene la siguiente distribución de pines:

- Disponemos de 14 pines digitales que pueden ser configurados como entradas o salidas, de los cuales (serigrafiadas con el símbolo ~) pueden ser utilizados como señales digitales PWM 6 pines.
- Igualmente disponemos de 6 pines analógicos serigrafiadas desde A0 hasta A5 para las entradas analógicas.
- También disponemos de 3 pines GND para conectar a tierra nuestros circuitos.
- Y por último 2 pines de alimentación de 5V y 3.3V respectivamente.



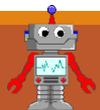
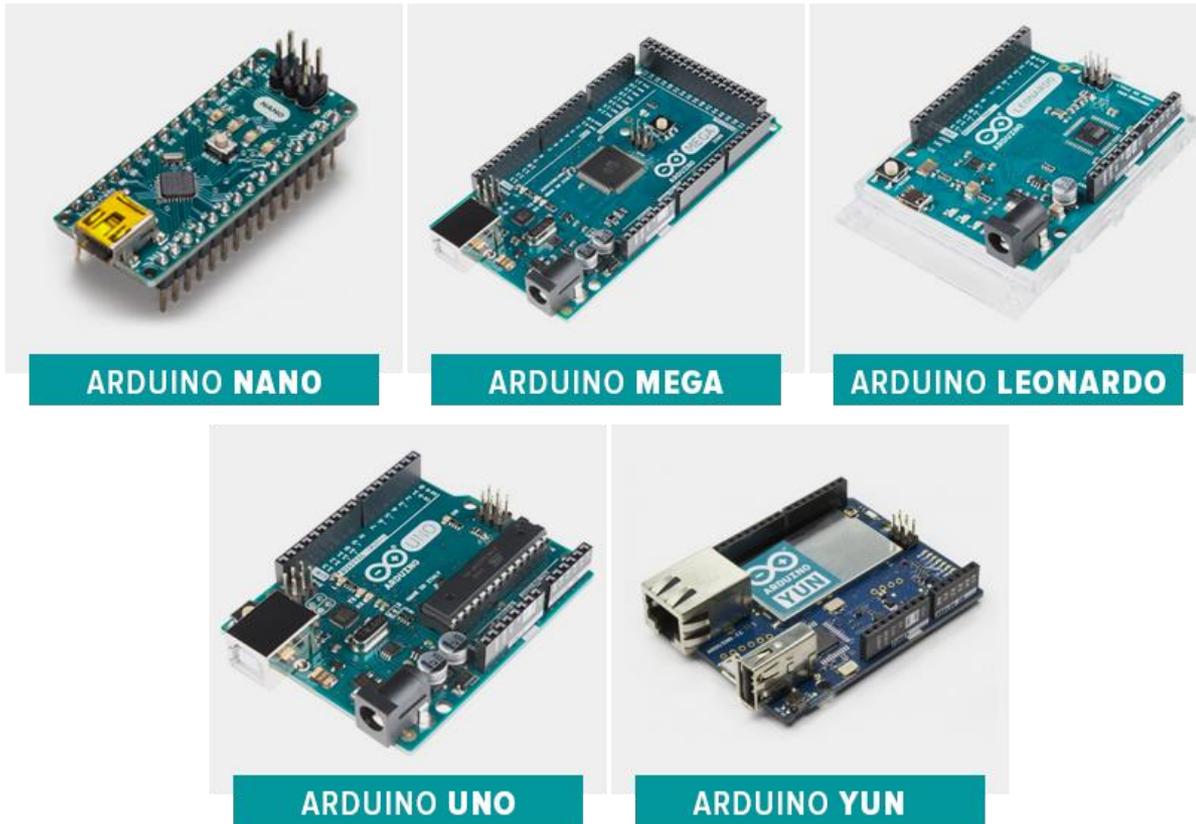
MICROCONTROLADOR

ARDUINO - Pines



MICROCONTROLADOR

ARDUINO - Placas

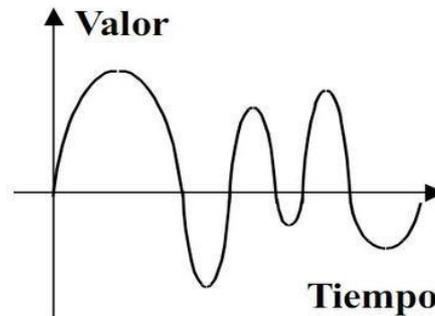


MICROCONTROLADOR

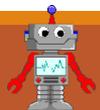
Señales Digitales Y Señales Analógicas

Señales analógicas

Son variables eléctricas que evolucionan en el tiempo en forma análoga a alguna variable física. Estas variables pueden presentarse en la forma de una corriente, una tensión o una carga eléctrica. Varían en forma continua entre un límite inferior y un límite superior



Señal Analógica



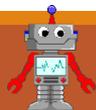
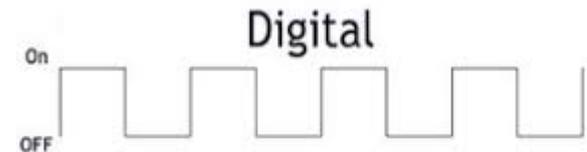
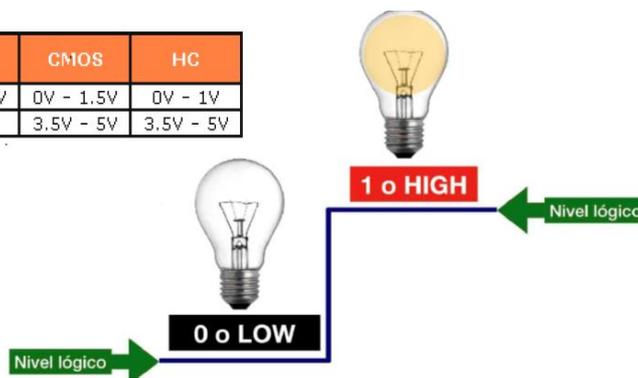
MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Señales digital

Son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 ó 1, V o F, etc. Los niveles específicos dependen del tipo de dispositivos utilizado

Nivel de tensión	TTL	CMOS	HC
Bajo (0)	0V - 0.8V	0V - 1.5V	0V - 1V
Alto (1)	2V - 5V	3.5V - 5V	3.5V - 5V



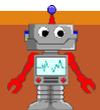
MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Analog Signal

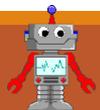
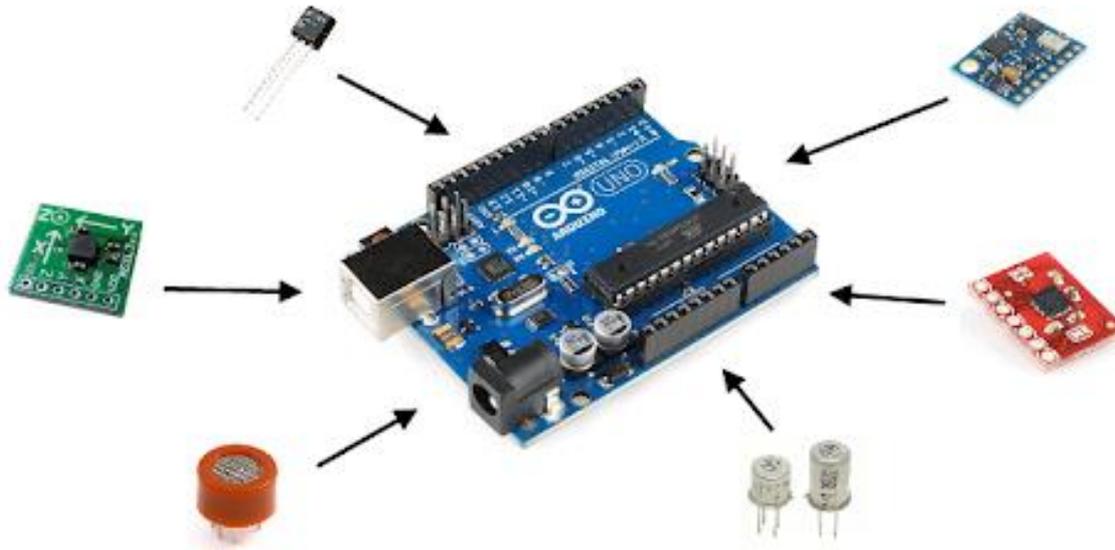


Digital Signal



MICROCONTROLADOR

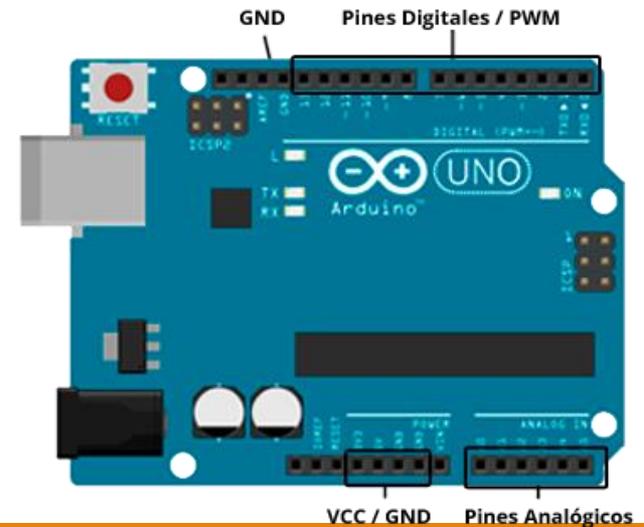
Señales Digitales Y Señales Analógicas



MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

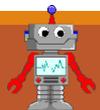
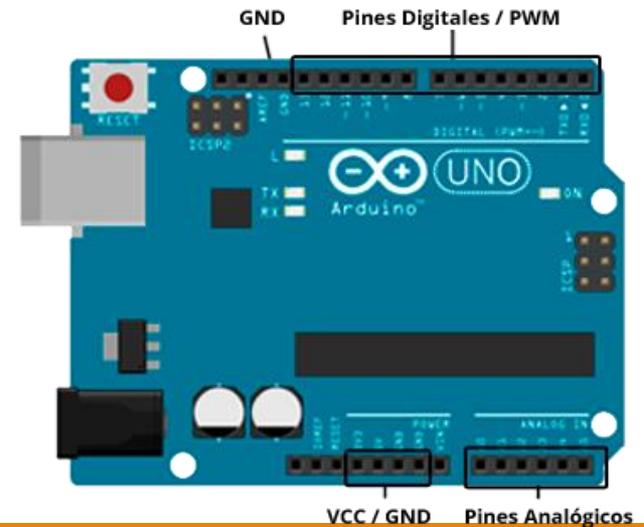
Arduino no sólo puede enviar señales sino que también puede recibirlas con dos propósitos principales como son leer datos de sensores y recibir mensajes de otros dispositivos (shield, otro Arduino, PC, etc.). Las entradas las clasificaremos en analógicas y digitales.



MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Arduino no sólo puede enviar señales sino que también puede recibirlas con dos propósitos principales como son leer datos de sensores y recibir mensajes de otros dispositivos (shield, otro Arduino, PC, etc.). Las entradas las clasificaremos en analógicas y digitales.

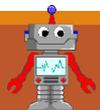
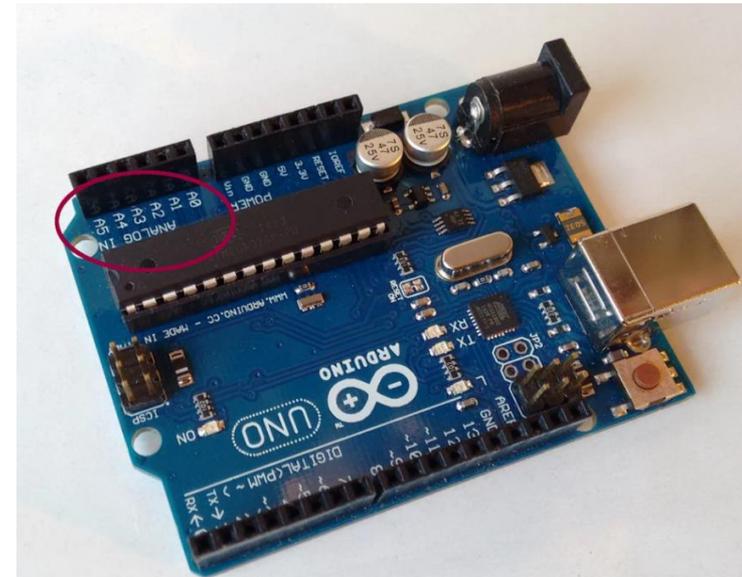


MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Entradas analógicas

Las entradas analógicas del modelo Uno son las correspondientes a los pines de A0 a A5. Se caracterizan por leer valores de tensión de 0 a 5 Voltios con una resolución de 1024 (10 bits). Si dividimos 5 entre 1024 tenemos que ser capaz de detectar variaciones en el nivel de la señal de entrada de casi 5 mV.



MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

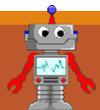
Entradas analógicas

Para hacer la lectura de uno de estos pines escribiremos en nuestro código

lectura = analogRead(pinentrada);

“lectura” lo sustituimos por el nombre de la variable donde queremos almacenar el valor leído y en “pinentrada” tendremos que poner el número del pin analógico que hemos elegido (0,1,...5) o el nombre de la variable que almacena dicho número.

Esta función nos devolverá un valor que va de 0 a 1023 en proporción al nivel de la señal de entrada. Para una entrada nula obtendremos el valor 0, para una entrada de 2.5 Voltios 511 (la mitad de 1023) y para 5 Voltios 1023.

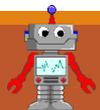
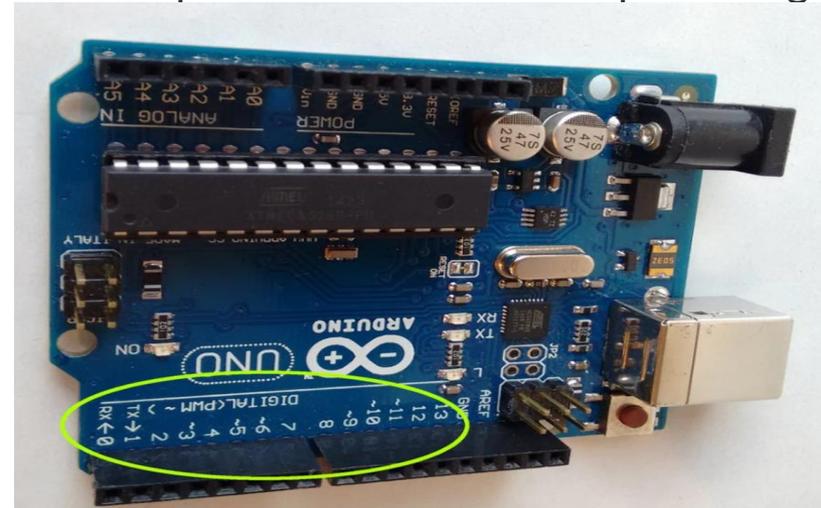


MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Entradas digitales

Las entradas digitales son las mismas que las salidas digitales, es decir, los pines que van del 1 al 13. Se diferencian de las analógicas porque éstas son capaces de “entender” sólo dos niveles de señal, LOW o valores cercanos a 0 V y HIGH o valores cercanos a 5 V. Puede parecer una desventaja pero en realidad puede ser todo lo contrario. Y no sólo porque a veces únicamente necesitemos saber dos estados (interruptor, pulsador, sensor de presencia, final de carrera....) sino porque así es capaz de leer señales de pulsos digitales. Esto significa que puede comunicarse .

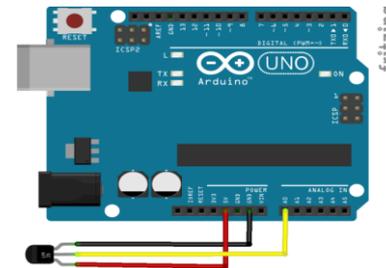
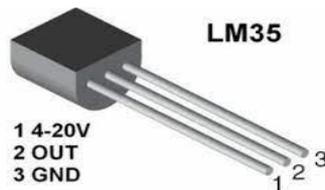


MICROCONTROLADOR

Señales Digitales Y Señales Analógicas

Entradas digitales

Por poner un ejemplo, un sensor analógico de temperatura como es el LM35 incrementaría el nivel de la tensión que llega a la placa de forma proporcional a la temperatura.



Sin embargo, uno digital como el inductivo lo que haría es cambiar la sucesión de pulsos y por tanto el mensaje que contiene el valor de la temperatura.



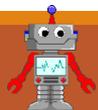
MICROCONTROLADOR

Arduino UNO

Arduino Uno está basado en el microcontrolador **ATmega328 de Atmel**. Donde los pines del Arduino UNO (pinout) consta de:

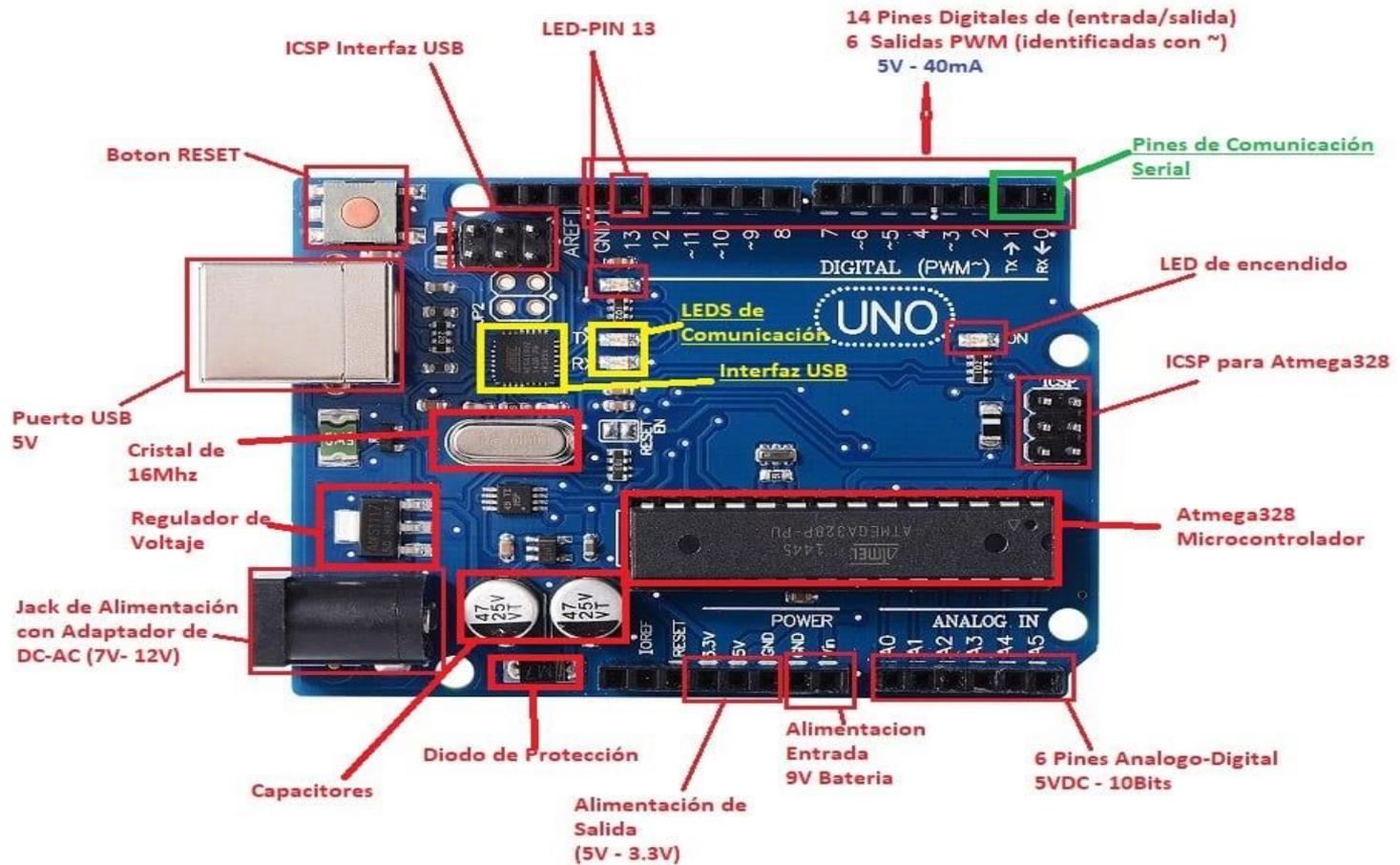
- 1.14 pines digitales,
- 2.6 entradas analógicas,
- 3.un conector de alimentación,
- 4.conexión USB
- 5.encabezado ICSP.

La versatilidad de estos pines nos irán ayudando en los diferentes proyectos que deseemos emprender con Arduino.



MICROCONTROLADOR

Arduino UNO

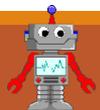


MICROCONTROLADOR

Arduino UNO - Pines

Los Pines Digitales: Pueden ser configurados como entradas o salidas y sirven para encender luces, motores, electro válvulas, relés entre otros. Si trabaja como entrada sirve para detectar cambios de estado.

Los Pines PWM: Son pines que pueden ser configurados con una señal PWM (Ancho por Modulación de Pulso) son muy usados para controlar la velocidad de motores, regular intensidades, entre otros.



MICROCONTROLADOR

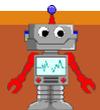
Arduino UNO - Pines

Los Pines Analogo/Digital: Sirven para capturar las señales del mundo real, como la medida de los sensores.

Pines de Alimentación: Nos ofrece dos pines, uno de 5V y otro de 3.3V, nos sirve para alimentar nuestros circuitos o protoboards.

Puerto USB: Sirve para hacer comunicación con el computador.

ICSP: (stands for In-Circuit Serial Programming) sirve para reprogramar la tarjeta con nuevos firmware.



MICROCONTROLADOR

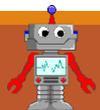
Arduino UNO - Arduino IDE

Para programar nuestros Arduinos, vamos a necesitar descargar a nuestros computadores el programa de Arduino IDE, el cual es la plataforma donde desarrollaremos todos nuestros programas y además a través de este programaremos el chip de Arduino.



```
sketch_sep25a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_sep25a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```



MICROCONTROLADOR

Arduino UNO - Estructura Básica del Programa de Arduino IDE

```
1. void setup() {  
2.   // put your setup code here, to run once:  
3.  
4. }  
5.  
6. void loop() {  
7.   // put your main code here, to run repeatedly:  
8.  
9. }
```

Donde evidenciamos **3 partes importantes:**

Zona de declaraciones

Funcion void setup

Funcion void loop



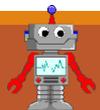
MICROCONTROLADOR

Arduino UNO - Estructura Básica del Programa de Arduino IDE

1.Zona de declaraciones: se encuentra arriba de la función “void setup”, en esta parte vamos a declarar todo lo que necesitemos en nuestro programa como variables, funciones, objetos, y estructuras.

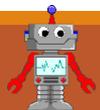
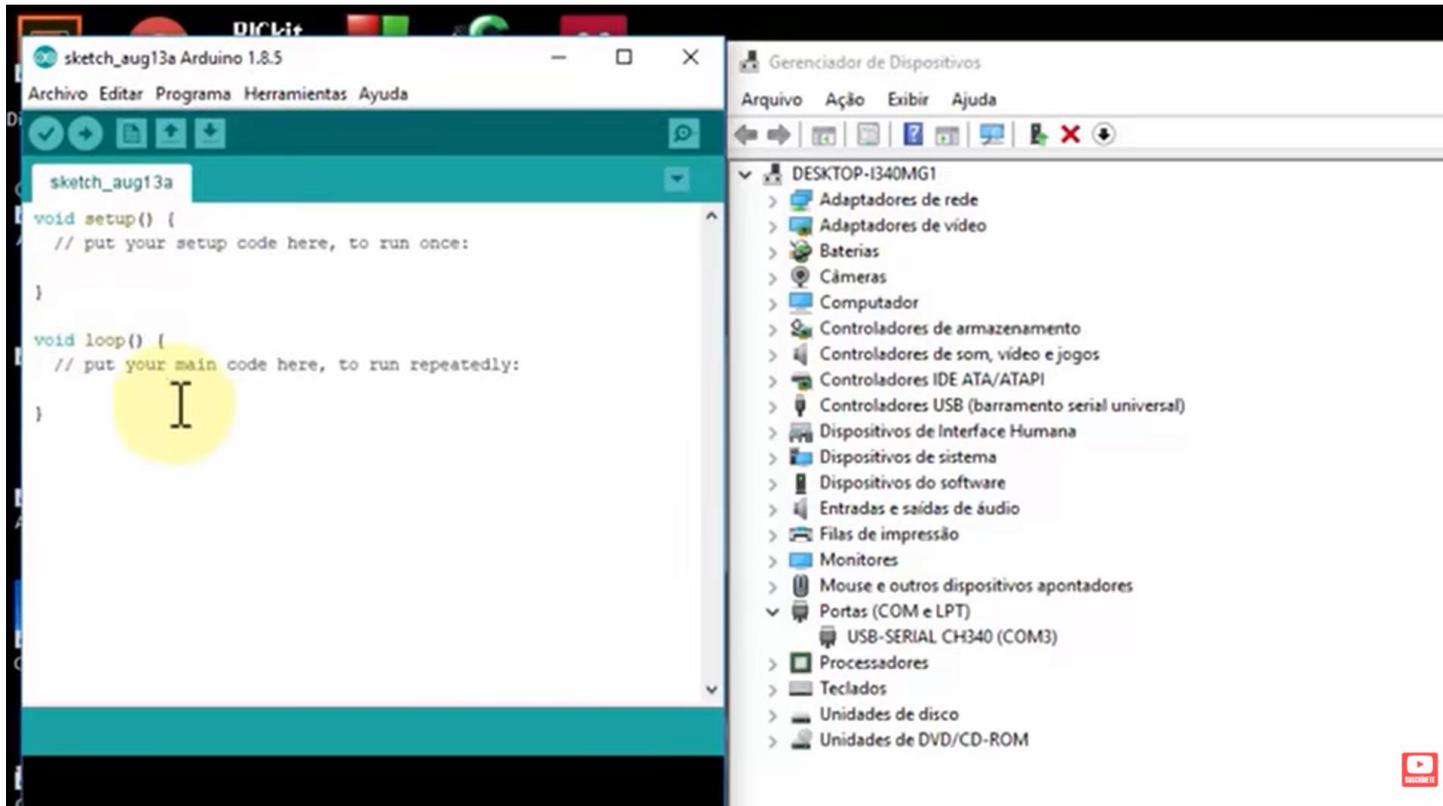
2.Funcion void setup: Esta función se ejecuta una unica vez cada que se enciende la placa de Arduino, o se pulsa la tecla Reset. Dentro de esta función se configuran todos los dispositivos perifericos que deseamos usar en nuestro proyecto de Arduino.

3.Funcion void loop: como su nombre lo indica, es una función de ciclo infinito, la cual se ejecuta infinitamente hasta que el Arduino sea desenergizado. En esa función es donde vamos a programar nuestros proyectos de Arduino.



MICROCONTROLADOR

Arduino UNO - Comprobación

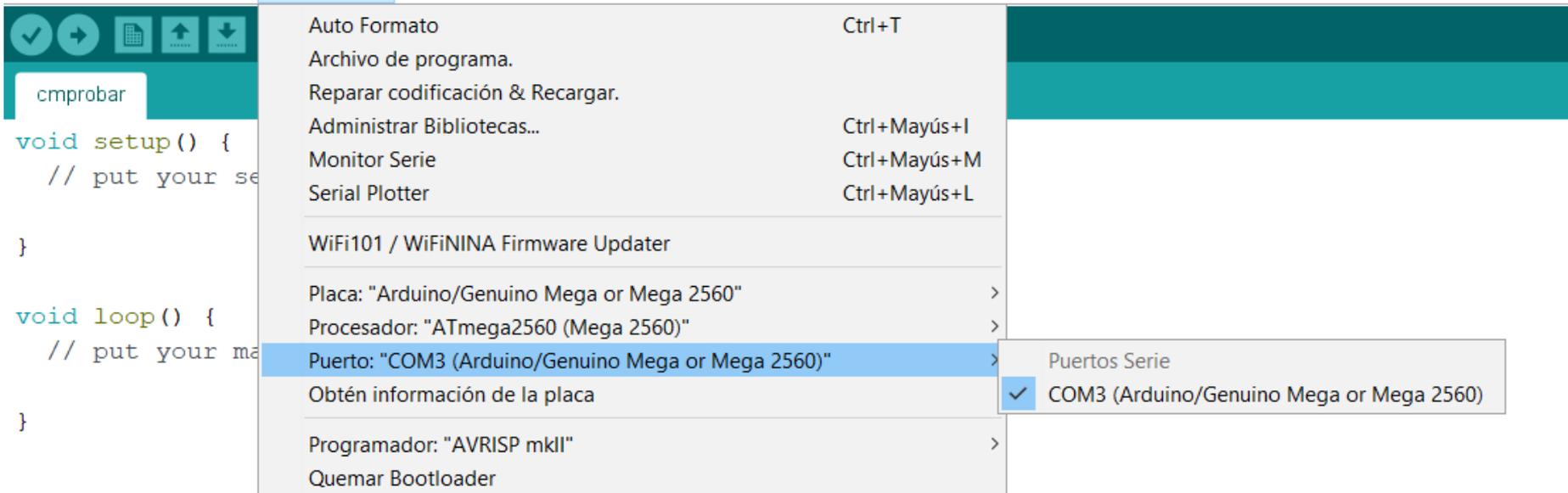


MICROCONTROLADOR

Arduino UNO - Comprobación

cmprobar Arduino 1.8.10

Archivo Editar Programa Herramientas Ayuda



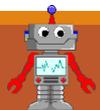
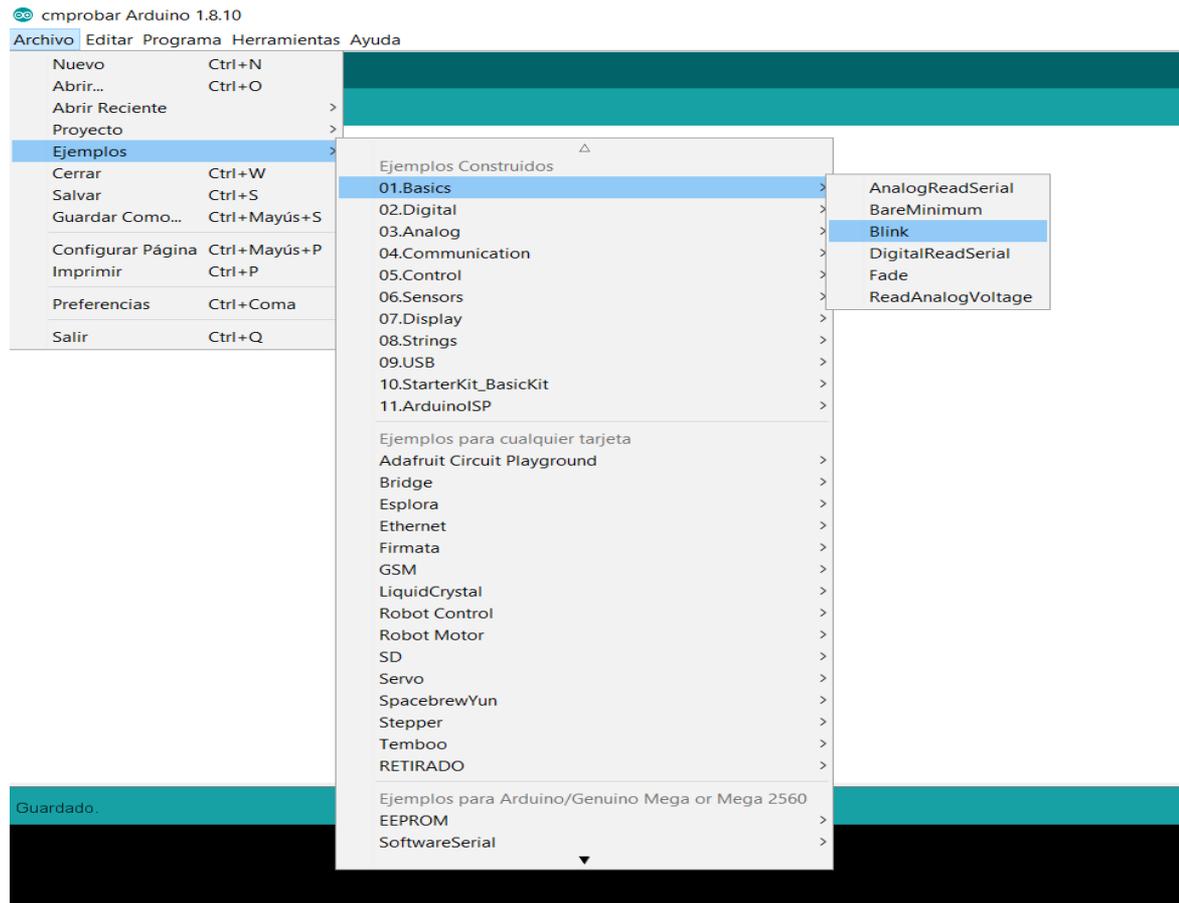
The screenshot shows the Arduino IDE interface. The 'Herramientas' (Tools) menu is open, displaying various options. The 'Puerto: "COM3 (Arduino/Genuino Mega or Mega 2560)"' option is highlighted in blue. A sub-menu is open for this option, showing 'Puertos Serie' (Serial Ports) with 'COM3 (Arduino/Genuino Mega or Mega 2560)' selected, indicated by a checkmark. The background shows the IDE's toolbar and a portion of the code editor with the following code:

```
void setup() {  
  // put your se  
  
}  
  
void loop() {  
  // put your ma  
  
}
```



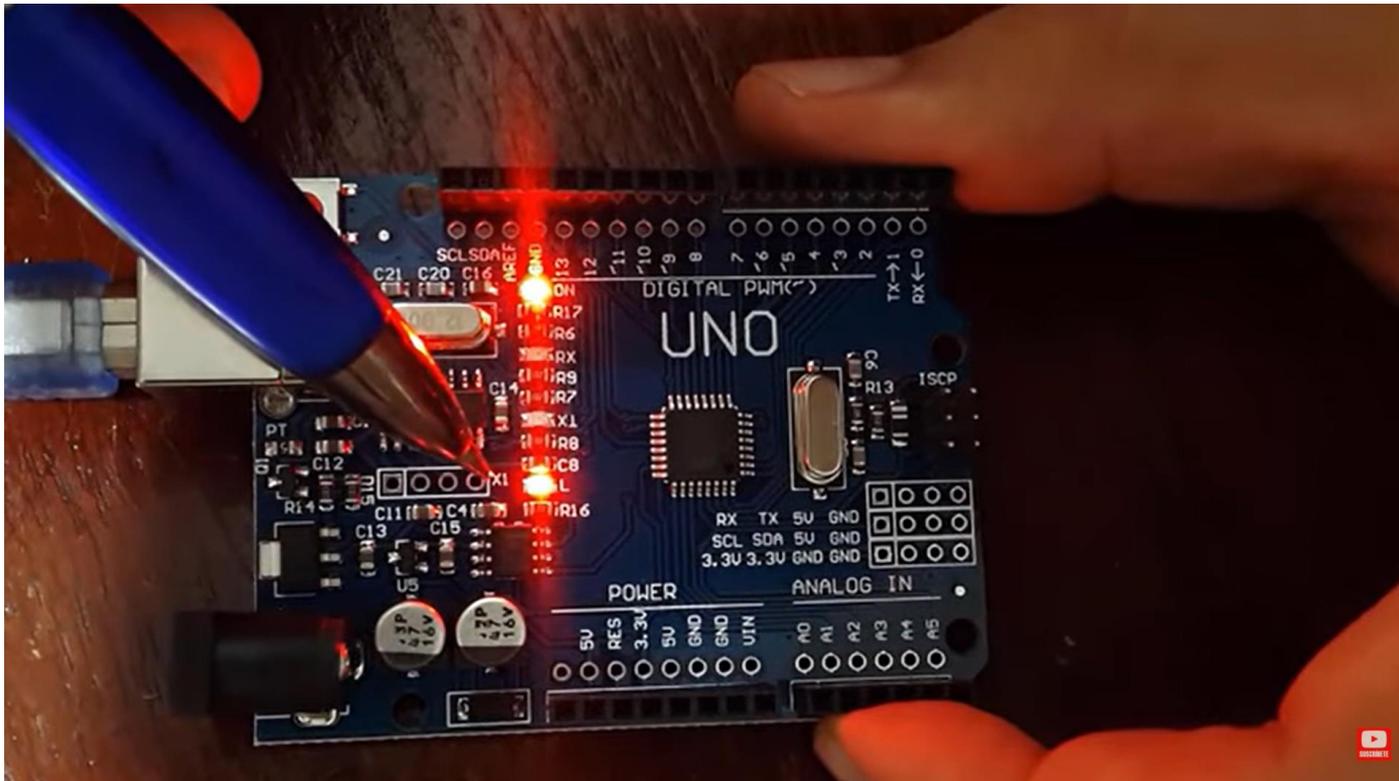
MICROCONTROLADOR

Arduino UNO - Comprobación



MICROCONTROLADOR

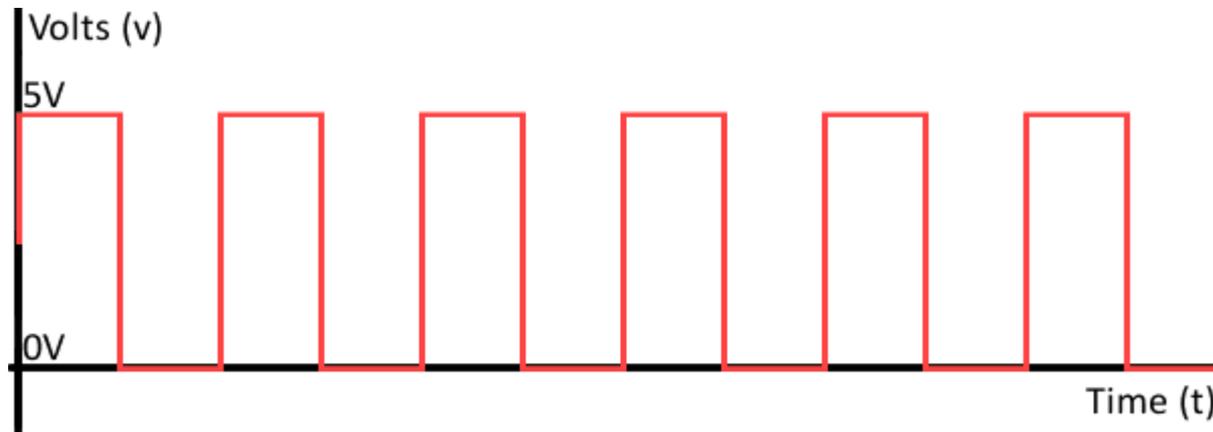
Arduino UNO - Comprobación



MICROCONTROLADOR

Arduino UNO - Que son Salidas Digitales en Arduino

Primero, vamos a recordar que una salida digital es una señal binaria, esto quiere decir que toma únicamente **2 valores**. Y esos valores en el mundo de los microcontroladores o de electrónica digital son **0V (GND) y 5V**.



Estos dos valores los conoceremos como:

- 0 – Cero Lógico – Falso – 0V
- 1 – Uno Lógico – Verdadero – 5V

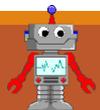


MICROCONTROLADOR

Arduino UNO - Que son Salidas Digitales en Arduino

Sin embargo, algunos modelos de ARDUINO y Microcontroladores en general operan a **3.3V** (en lugar de 5V), tales como las placas basadas en procesadores ARM (Arduino Due) o como el Arduino Nano o Mini.

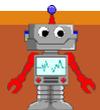
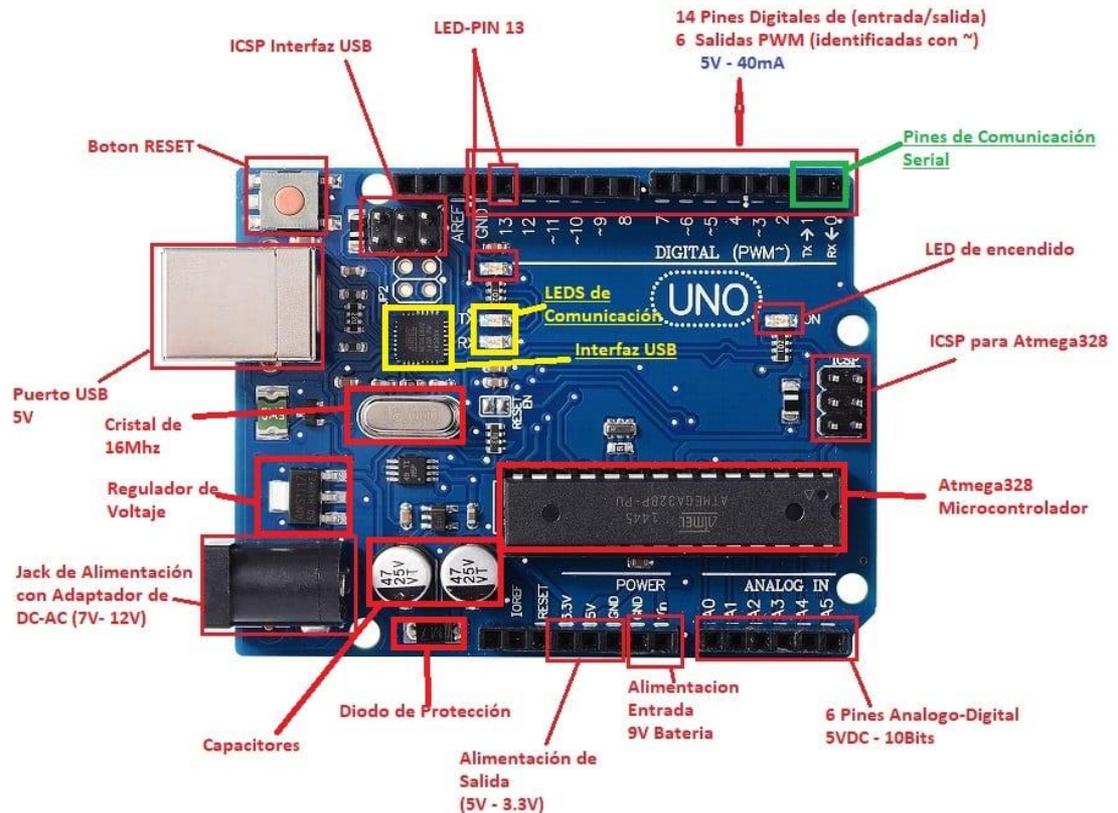
Todos los pines del puerto I/O (Input – Output) pueden configurarse para ser Salidas Digitales en Arduino



MICROCONTROLADOR

Arduino UNO – Pines Digitales

Los puertos digitales del Arduino UNO van desde **el PIN 0** hasta **el PIN 13**, como lo vemos en esta figura:

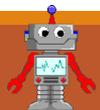
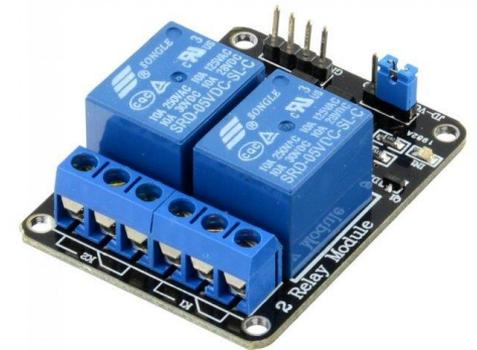
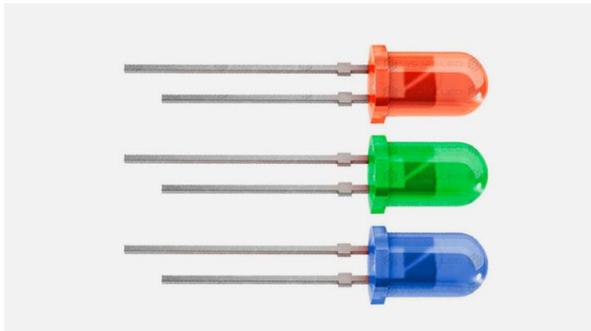


MICROCONTROLADOR

Arduino UNO – Para que Sirven las Salidas Digitales

Con las salidas digitales Arduino Mega, Uno, Leonardo o el que sea, podremos interactuar con el mundo físico de miles de formas, porque como ya lo vimos, podemos mandar una señal de dos estados (1 o 0) es decir, prender o apagar.

Entonces con esto podremos encender y apagar luces, leds, motores, relés, actuadores, etc.



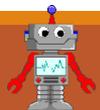
MICROCONTROLADOR

Arduino UNO – Declarar Pines de Salida en Arduino

Para programar una salida digital, debemos hacerlo dentro de la función “void setup” y vamos a utilizar la siguiente instrucción

```
1. | pinMode(PIN Digital,OUTPUT);
```

Donde dice “**Pin Digital**” debemos sustituirlo por el PIN del Arduino que nosotros queremos que sea una salida.



MICROCONTROLADOR

Arduino UNO – Declarar Pines de Salida en Arduino

Ejemplo

```
1. void setup() {  
2.     pinMode(5,OUTPUT); //Configura el PIN 5 como SALIDA DIGITAL  
3. }
```



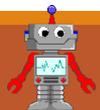
MICROCONTROLADOR

Arduino UNO – Declarar Pines de Salida en Arduino

Una vez hemos configurado los pines que queremos establecer como salidas, es hora de utilizar estas salidas, para eso vamos a utilizar una segunda función, la cual deberemos llamar dentro del “void loop”

```
1. void loop() {  
2.     digitalWrite(Pin de Salida,HIGH); //Energiza la SALIDA DIGITAL (5V)  
3.     digitalWrite(Pin de Salida,LOW); //DesEnergiza la SALIDA DIGITAL (0V)  
4. }
```

Aquí la variable “Pin de Salida” la sustituimos por el pin del Arduino que queremos energizar (5V) o desenergizar (0V). En otras palabras estamos escribiendo en la salida digital del arduino si queremos que se active o desactive, encienda o apague.



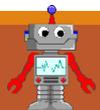
MICROCONTROLADOR

Arduino UNO – Declarar Pines de Salida en Arduino

Ejemplo

Vamos a suponer que queremos activar salida digital de Arduino en el **PIN 5**. Debemos utilizar la instrucción **digitalWrite** dentro del **void loop** usando cualquiera de las siguientes formas:

```
1. void loop() {  
2.     digitalWrite(5,HIGH); //Activa o Energiza el PIN 5  
3. }
```



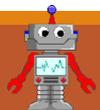
MICROCONTROLADOR

Arduino UNO – Declarar Pines de Salida en Arduino

Ejemplo

Vamos a suponer que queremos activar salida digital de Arduino en el **PIN 5**. Debemos utilizar la instrucción **digitalWrite** dentro del **void loop** usando cualquiera de las siguientes formas:

```
1. void loop() {  
2.     digitalWrite(5,HIGH); //Activa o Energiza el PIN 5  
3. }
```

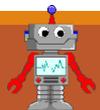
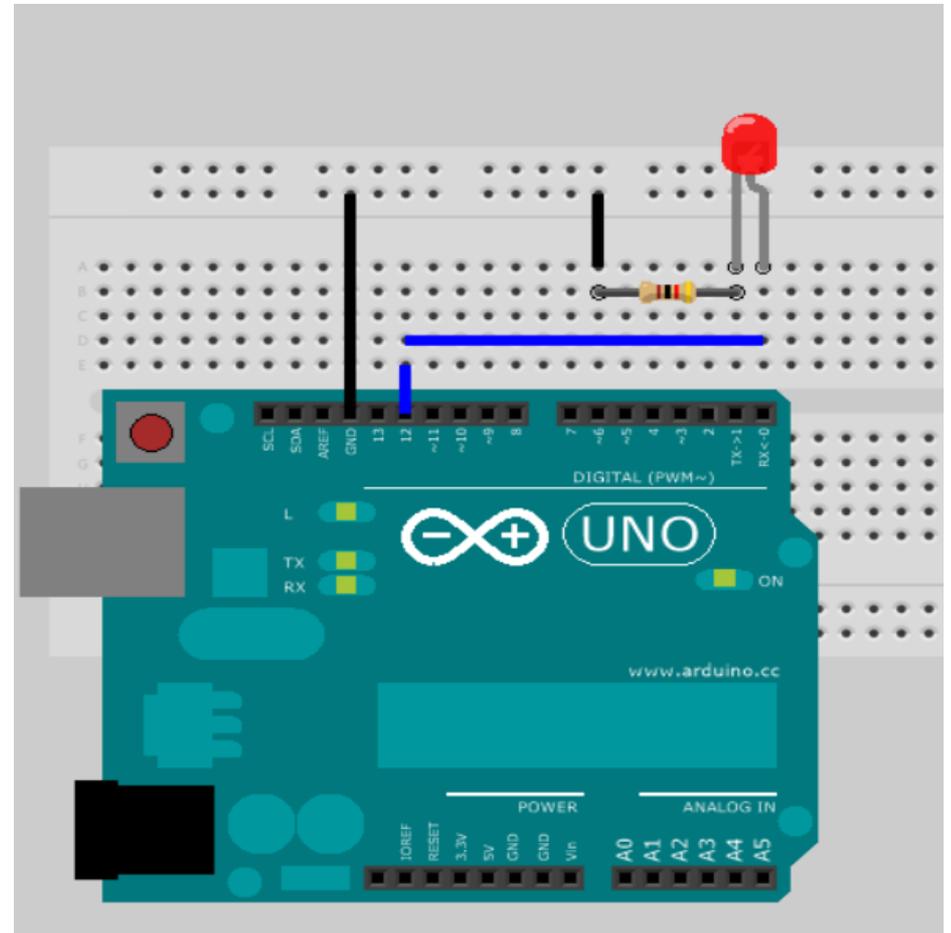
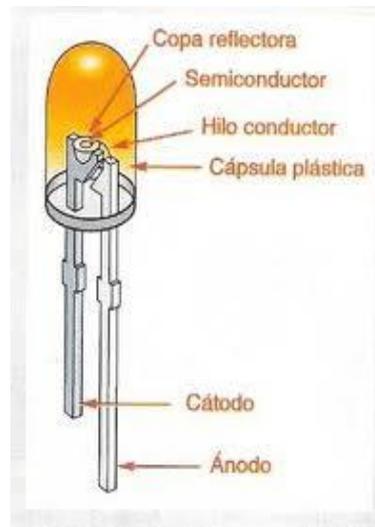


MICROCONTROLADOR

Arduino UNO – Ejemplo1

// Práctica encender y apagar un LED

```
const int LED=12;  
  
void setup()  
{  
  pinMode(LED,OUTPUT);  
}  
  
void loop()  
{  
  digitalWrite(LED,HIGH);  
  
  delay(6000);  
  
  digitalWrite(LED,LOW);  
  
  delay(6000);  
}
```



MICROCONTROLADOR

Arduino UNO – Ejemplo2

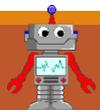
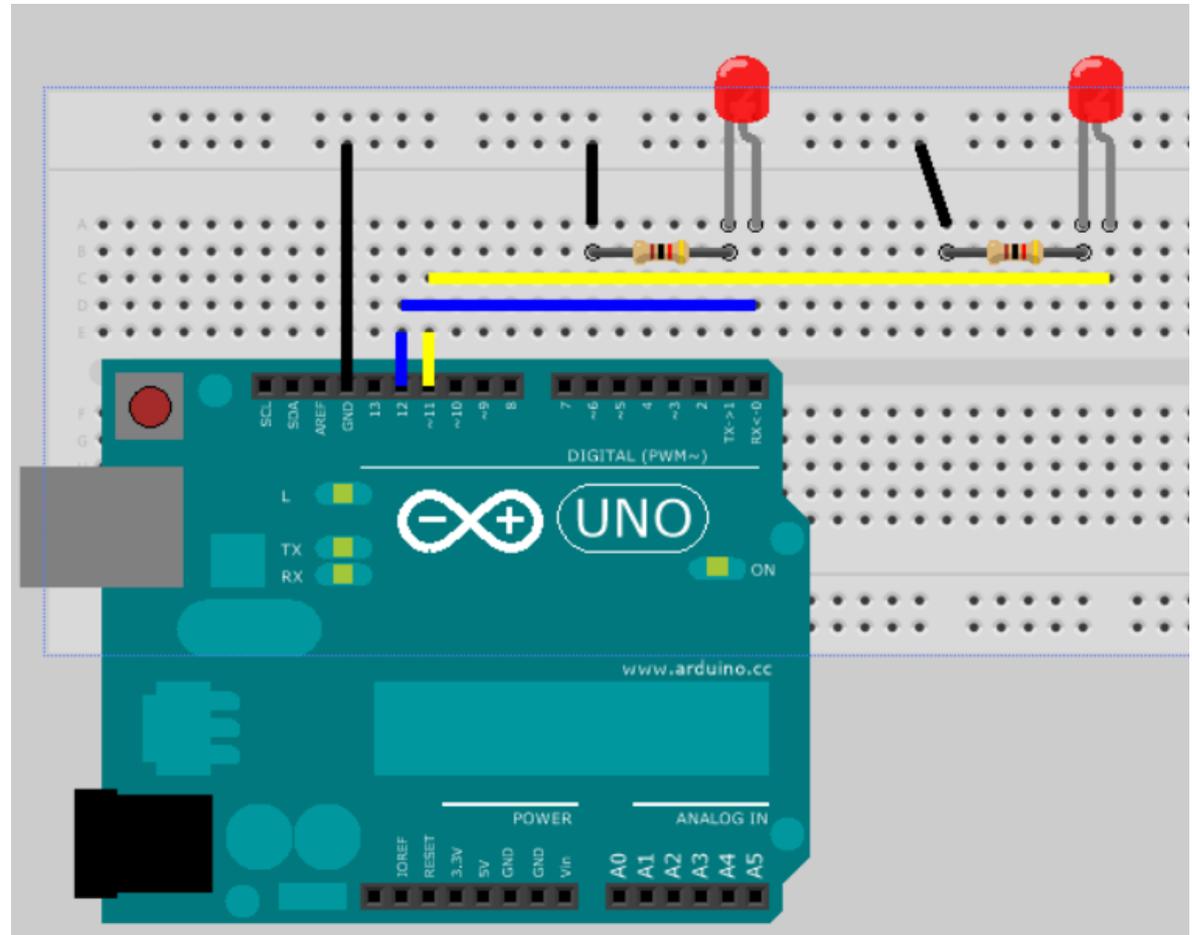
// Práctica encender y apagar dos LED

```
const int LED1=12;

const int LED2=11;

void setup()
{
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
}

void loop()
{
  digitalWrite(LED1,HIGH);
  delay(1000);
  digitalWrite(LED1,LOW);
  digitalWrite(LED2,HIGH);
  delay(1000);
  digitalWrite(LED2,LOW);
}
```



MICROCONTROLADOR

Arduino UNO – Ejemplo3

```
// Práctica encender y apagar un
LED a través de botón pulsador
const int LED=12;
const int BOTON=7;
int val;
void setup(){

  pinMode(LED,OUTPUT);
  pinMode(BOTON,INPUT);
}

void loop(){

  val=digitalRead(BOTON);

  if (val==HIGH){
  digitalWrite(LED,HIGH);
  }
  else { digitalWrite(LED,LOW);
  }

}
```

