**MANUAL DE PROCEDIMIENTOS PARA ENSAMBLAR PROYECTOS UTILIZANDO ARDUINO**



**GUACARA - JULIO 2017**

**MANUAL DE PROCEDIMIENTOS PARA ENSAMBLAR PROYECTOS UTILIZANDO ARDUINO** ™

**TABLA DE CONTENIDO**

*Prefacio*

*Nota para el estudiante*

Introduccion a la robotica

Introduccion a Arduino™

Componentes compatibles

Aplicaciones de Arduino™

Proyectos con Arduino™ UNO

Ejemplo de Proyecto nº1

Ejemplo de Proyecto nº2

Ejemplo de Proyecto nº3

Librería de comandos de Arduino™

**Apendice A**: Glosario de terminos

Recomendaciones de textos

**PREFACIO**

Este manual de procedimientos y/o guia ha sido realizado con la finalidad de que los estudiantes de la universidad tecnologia del centro y cualquier persona en general que esten interesados en las ramas de la robotica y sus aplicaciones puedan facilmente inicar sus primeros pasos en este mundo, por lo que se detallan ejemplos utilizando placas Arduino™ para facilitar dicha introduccion a la robotica.

**NOTA PARA EL ESTUDIANTE**

Los ejemplos encontrados en este manual de procedimientos se aplican unica y exclusivamente para emsamble de proyectos que utilizan la placa Arduino UNO, tenga en cuenta que algunos componentes pueden diferir de los que aqui se encuentran planteados. Para una mayor profundidad en las ramas de la robotica, se hacen necesarios conocimientos tanto fundamentales como avanzados en areas de electrica, mecanica e informatica.

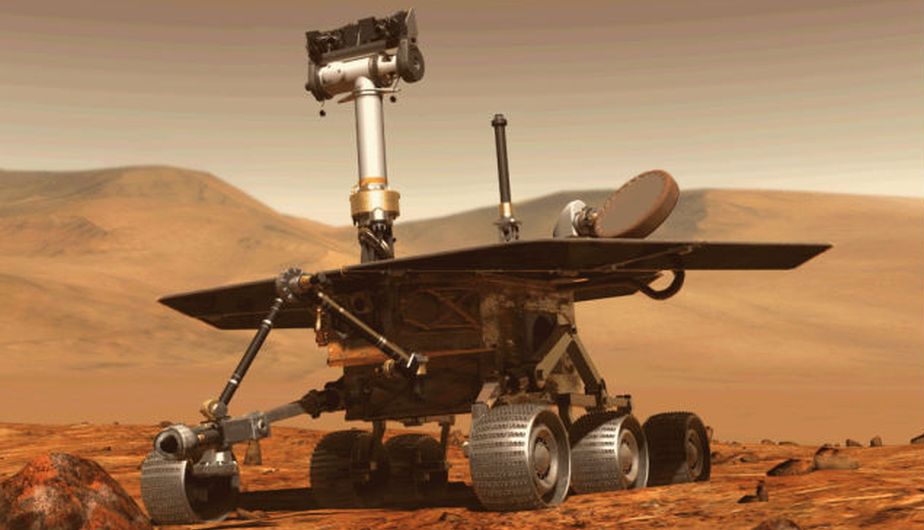
**INTRODUCCION A LA ROBOTICA**

La robótica es un termino que proviene de las palabras checas robota y robotnik (Trabajo forza y Sirviente respectivamente) y se define como la ciencia que estudia el diseño y la implementación de robots, conjugando múltiples disciplinas, como la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control, entre otras. Para definirlo en términos generales, un robot es una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas.

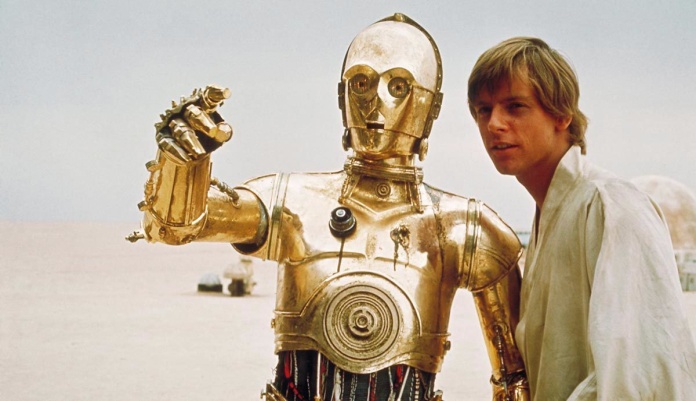
Teniendo en cuenta entonces que al hablar de robotica, se habla de robots; se hace necesario clasificar a estos ultimos. Estos se pueden clasificar en base a la estructura y componente que lo conforman: Articulados, moviles, androides, zoomorficos o hibridos.

**a) Robots Articulados:**

Son aquellos robots que pueden moverse en un determiando espacio, poseen al menos una articulacion controlada y se encuentran sobres bases fijas. Este tipo de robot se encuentra en las industrias de todo tipo y son conocidos como “manipuladores” o brazos roboticos.

**b) Robots Moviles:**

Son aquellos robots que pueden desplazarse gracias a un motor en su estructura y pueden percibir su entorno mediante sensores. Este tipo de robot emulan por lo general a sistemas de sistemas automotrices como automoviles, motocicletas, entre otros.

**c) Robots-Androides:**

Es un tipo de robot que posee aspectos humanos e incluso movimientos y funcionas propias a las que tendria un humano. Este tipo de robot suele ser un referente en numerosas peliculas y libros de ficcion.

**d) Robots-Zoomorficos:**

Es un tipo de robot similar al androide con la diferencia de que en vez de emular humanos, se emulan cualquier ser vivo del reino animal. Este tipo de robot incluye, por ejemplo: perros-robot, arañas-robot, panteras-robot, entre otros.

**e) Robots Hibridos:**

Es el tipo de robot que surge debido a la combinacion de dos o mas tipos de robots anteriormente descritos. Aquí se incluyen casi todos los tipos de robots que se desarrollan actualmente como por ejemplo: Un brazo robotico con ruedas motorizadas en su base, un androide con cintas transportadoras en vez de piernas, entre otros.

Asi como estos pueden clasificarse según su estructura, tambien pueden clasificarse según su evlocion a lo largo de los años. Actualmente la robotica se encuentra en su quinta generacion, siendo estas generaciones las siguientes:

Primera Generación: Sistemas de control basado en “paradas fijas” mecánicamente como mecanismos de relojería que mueven las cajas musicales o juguetes de cuerda. Aquí se incluyen los primeros automatas de la historia, estos funcionaban usando solo energia mecanica (Movimiento mediante engranajes).

Segunda Generación: El movimiento se controla a través de una secuencia numérica almacenada en disco o cinta magnética como en la industria automotriz.

Tercera Generación: Utilizan las computadoras para su control y tienen cierta percepción de su entorno a través del uso de sensores. Con esta generación se inicia la era de los robots inteligentes y aparecen los lenguajes de programación.

Cuarta Generación: Robots altamente inteligentes con más y mejores extensiones sensoriales, para entender sus acciones y captar el mundo que los rodea. Incorporan conceptos “modélos” de conducta.

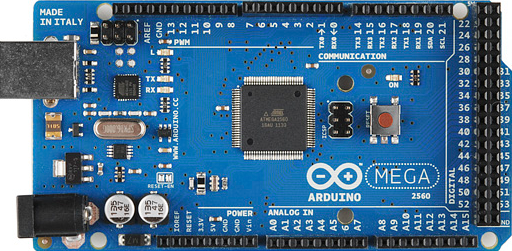
Quinta Generación: Actualmente se encuentran en desarrollo. Basarán su acción principalmente en modelos de conducta humana establecidos asi como la capacidad de reconocer expresiones.

Estas generaciones han tenido lugar en una cronologia que va desde el año 1206, donde tienen lugar los primeros automatas, hasta nuestros dias. Sin embargo, existen menciones a los robot en escritos que datan incluso antes de Cristo, estos escritos tenian como tematica a la ficcion, principal impulsor de la robotica. Para los propositos de este manual, debemos remontarnos al año 2005, donde tiene lugar, sin duda alguna, uno de los hitos mas importante en la historia del Software Libre: La invencion de una plataforma de prototipos electrónica llamada Arduino.

Lo que hizo y hace especial a Arduino es que fue desarrollado para ser facil de utilizar ya que las placas se pueden ensamblar a mano o encargarlas preensambladas; el software se puede descargar gratuitamente desde su pagina oficial y los diseños de referencia del hardware están disponibles bajo licencia open-source, por lo que el usuario es libre de adaptarlas a sus necesidades. Ademas, para iniciarse en Arduino no se es necesario una gran inversion de capital ya que esta pensado para cualquiera interesado en la electronica y la robotica. Pero a todo esto…¿Qué es Arduino?.

**INTRODUCCION A ARDUINO**™

Arduino es una marca italiana de placas de microcontroladores electronicos con una entrada USB y multiples puertos (pieza sobre la tarjeta, la cual permite hacer conexiones electricos sin necesidad de soldar) a los cuales, el usuario puede conectar componentes tales como: Motores electricos, microfonos, altavoces, sensores, luces LED, entre otros. Esta placa debe ser alimentada mediante un conector USB desde un computador o bien, una corriente directa.



Para utilizar una placa Arduino, se hace necesario el uso de un lenguaje de programacion, este se encuentra basado en C y C++ de la compañia Microsoft y puede descargarse directamente desde la pagina oficial de Arduino. Notese que en la imagen se puede observar el logo de Arduino en su esquina inferior derecha, si usted tiene una placa y noposee dicho logo, usted esta en presencia de una placa clon, bastante frecuentes en el mercado (estos clones son responsable en parte, de la popularidad que gozan las placas Arduino).

Las placas Arduino mas populares y por ende, mas solicitadas son las de “Nivel inicial” y se usan para prototipos no muy complejos, estas placas son:

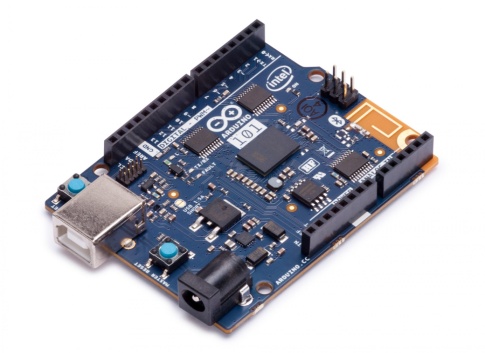
**VERSIONES NORMALES**

****

**Arduino UNO:** Es la mejor opcion para comenzar con la electrónica y la codificación si esta es su primera experiencia usando Arduino, La UNO es la placa más utilizada y documentada de toda la familia Arduino y es la que se encuentra de portada en este manual.



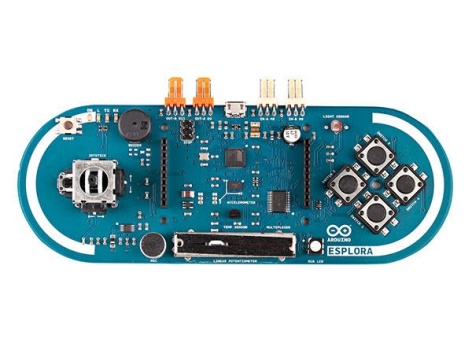
**Arduino LEONARDO:** Posee capacidades similares a la placa Arduino UNO. La LEONARDO, una vez conectada al computador, puede ser reconocida como un mouse e incluso un teclado

****

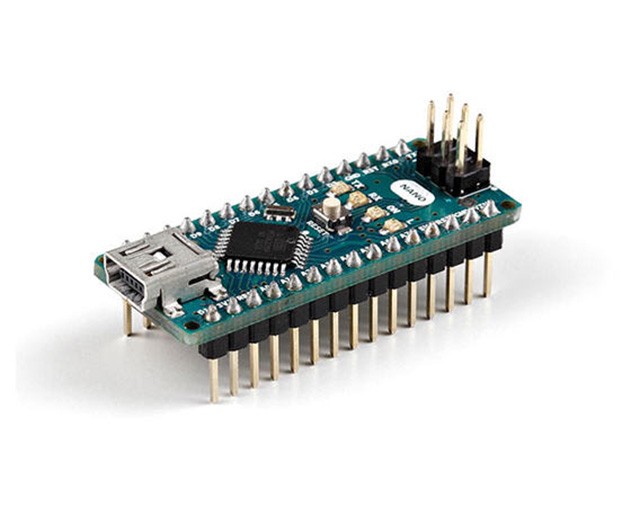
**Arduino 101:** Combinan la facilidad de uso con última tecnología. La placa reconoce gestos, y cuenta con un acelerómetrode seis ejes y giroscopio. Se utiliza para controlar proyectos a traves de conexión bluetooth



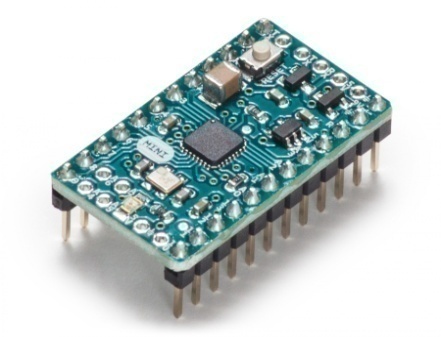
**Arduino ROBOT:** Es basicamente una placa arduino sobre ruedas, se utiliza para ciertos proyectos de tipo movil y posee carcateristicas similares a la placa Leonardo**.**

****

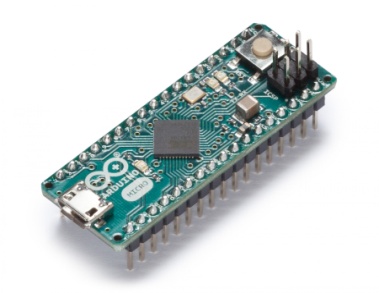
**Arduino ESPLORA:** Es una derivacion de la placa Leonardo con la diferencia de que la ESPLORA ya viene con un sensor de temperatura, un acelerómetro, un micrófono y un sensor de luz integrados.

**VERSIONES REDUCIDAS**

**Arduino NANO:** Es una version reducida de la placa Arduino UNO, carece de una toma de corriente directa, y funciona con un cable USB Mini en lugar de un cable USB estándar



**Arduino MINI:** El Arduino MINI es una versión muy compacta del Arduino NANO pero no posee una conexión USB a computador integrada.

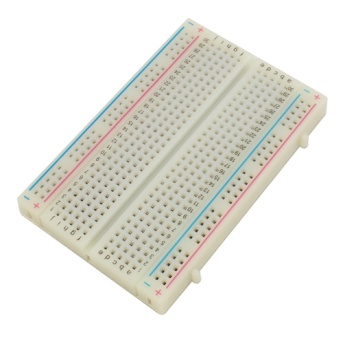
****

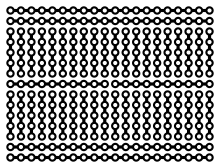
**Arduino MICRO:** Es la placa o tablero más pequeño de la familia Arduino, su tamaño hace que sea fácil de integrar en objetos cotidianos para hacerlos interactivos.

Otras placas de Arduino para proyectos avanzados o caracterisitcas complejas son: Arduino MEGA, Arduino ZERO, Arduino DUE, Arduino PRO, Arduino MOTORSHIELD, entre otros. Usted debe ser cuidadoso a la hora de obtener una de estas placas ya que estas no son las unicas y de hecho, existen modelos especiales ara modelos cuyo funcionamiento se da a traves de la conectividad inalambrica como por ejemplo: Arduino YUN, Arduino ETHERNET, Arduino TIAN, entre otros.

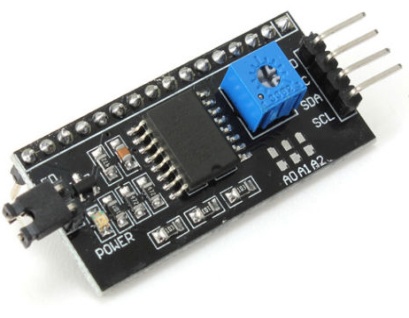
Seria correcto decir que usted no puede hacer “algo” teniendo solo la placa Arduino, se deben tener a la mano otros componentes relacionados a la electronica para realizar y ensamblar un proyecto, estos componentes son ampliamente variados y usted debe estar seguro de las especificaciones de los componentes que usted requiere y si estos son acesibles para usted. Sin embargo, existe un numero de componentes o “Shields” que suelen estar presentes en la mayoria de los proyectos de nivel inicial.

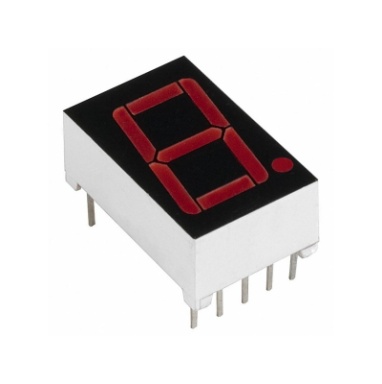
**COMPONENTES COMPATIBLES**

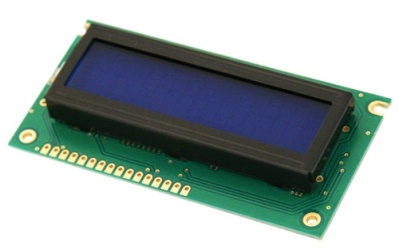
****

**Protoboard:** Es un tablero compuesto por puertos conectados electricamente entre si. Este se usa como placa de prueba para circuitos ya que en el se conectan los componentes electricos y cables. Se suele usar solo para probar prototipos.Está compuesta por bloques de plástico perforados y numerosas láminas delgadas, de una aleación de [cobre](https://es.wikipedia.org/wiki/Cobre), [estaño](https://es.wikipedia.org/wiki/Esta%C3%B1o) y [fósforo](https://es.wikipedia.org/wiki/F%C3%B3sforo), que unen dichas perforaciones. El patron de la perforaciones en las ya mencionadas laminas se puede observar en la imagen. Sobre esta placa es donde se arman los prototipos. La linea azul y roja sobre el protoboard son las lineas GND Y VCC respectivamente

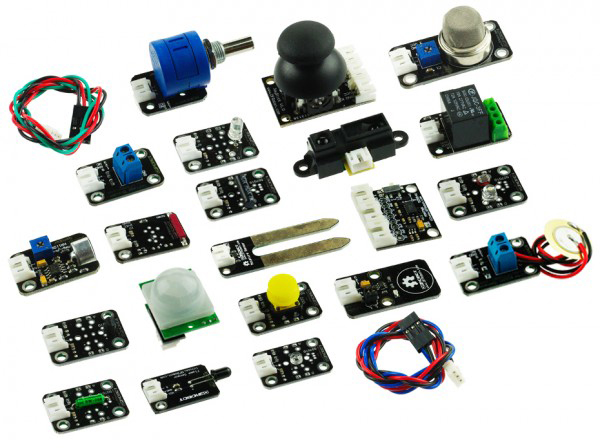
**LEDs:** Sus siglas significan Diodo Emisor de Luz. Diodo que emite luz visible cuando se energiza. Por lo general, se utiliza para reconocer cuando un sistema proporciona una entrada y reconoce una salida o por razones meramente esteticas.

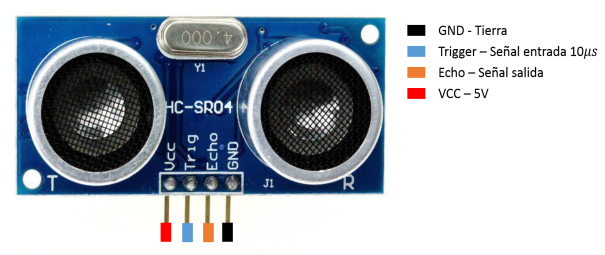
**Controlador:** Es un dispostivo con funciones preestablecidas y perifericos determinados para controlar una o mas acciones en especifico. Uno de los mas populares (En Arduino) es el controlar I2C por la cantidad de perifericos que son compatibles con el. Este dispositivo debe insertarse en el protoboard o en conjunto con el display LCD de 16x2.

**Display:** Dispositivo construido a partir de 7 u 8 LEDs, con una terminal de corriente en comun. Se utiliza para fines no muy complejos,muestran un numero y con el se pueden ensamblar contadores.

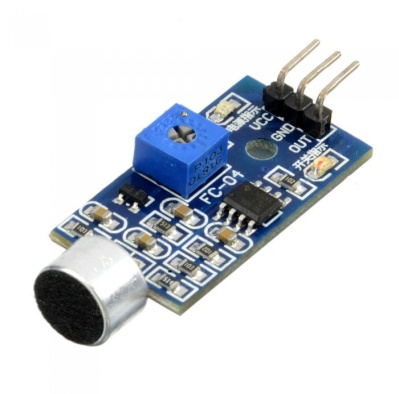


**Display LCD:** Dispositivo similar al display pero cuenta una pantalla de cristal liquido por lo que puede mostrar mucha mas infrmacion que el display convencional. Aunque existen muchos tamaños, se suele optar por un tamaño 16x2 ya que funciona bastante bien con el controlador I2C.

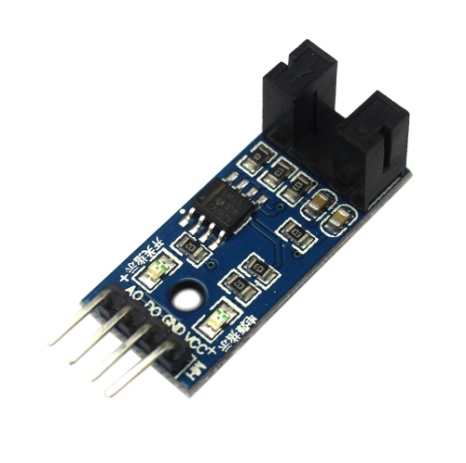
**Sensor:** Es un componente capaz de detectar una variable fisicas o quimicas. Existen varios modelos de sensores que pueden detectar mas de una magnitud pero se aconseja aduirir un sensor para una sola finalidad ya que esto representa ahorros e intercambialidad de piezas, algunos fabricantes comercializan kit con varios tipos de sensores. Estos deben ir conectados a la placa Arduino y al protoboard ademas de colocarse en una buena posicion para que estos puedan percibir facilmente la magnitud que se requiere.

****

**Sensor Ultrasonico:** Un tipo especial de sensor, un sensor ultrasonico es capaz de percibir una magnitud medida en Hertz mediante el ultrasonido, esta funcion le permite al arduino estimar la distancia de un objeto respecto al sensor. Estos tipos de sensores solo tienen un alcance cercano a los 30 grados.

**Sensor de Sonido:** Un tipo especial de sensor, un sensor de sonido o tambien llamado microfono es capaz de percibir un sonido en interpretarlo en ondas. Estos tipos de sensores usualmente se utilizan en prototipos relacionados a la musica.

**Sensor de Imagen:** Un tipo especial de sensor, un sensor de imagen consta basicamente de una camara VGA sobre una placa, la cual es capaz de percibir una imagen y es utilizado en prototipos relacionados a la vigilancia como por ejemplo, camaras de seguridad. La resolucion de la camara dependera del modelo que usted adquiera.

**Sensor de Velocidad:** Un tipo especial de sensor, un sensor de velocidad es capaz de percibir una magnitud medida en distancia por tiempo. Se utilza para comprobar en prototipos donde se necesita comprobar que tan rapido debe ir un prototipo movil ya sea automotriz o aerodinamico. Este sensor cumple su funcion gracias a una señal por medio de infrarojos.

**Puente H:** Es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avance y retroceso. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos

**Motor electrico:** Mecanismo que, mediante una energia electrica, produce un movimieto rotatorio. Este motor debe alimentarse mediante corriente directa/continua. La velocidad de giro se regula mediante una señal PWM (Tenga en cuenta la potencia maxima del motor).

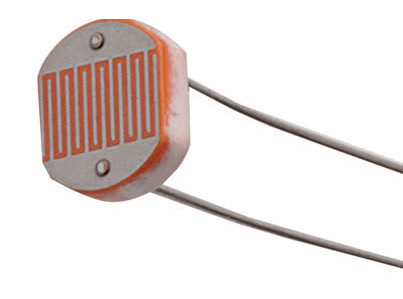
****

**Servomotor:** son también motores de corriente continua, pero en lugar de diseñarse para obtener un giro continuo que podamos aprovechar (para mover una rueda por ejemplo), se diseñan para que se muevan un angulo fijo en respuesta a una señal de control, y se mantengan fijos en esa posición. El control de un servomotor se regula mediante una seal PWM.

**Jumpers:** Se utilizan para conexiones de forma profesional, sin soldaduras, sin falsos contactos y sin desorden en el protoboard. Los cables vienen en un paquete de varias piezas, cada uno con su conector independiente. Usted puede utilizar otros cables en el protoboard pero suelen romperse y hasta atorarse en el protoboard.

****

**Resistencias:** Son componentes electrónicos que tienen la propiedad de presentar oposición al paso de la corriente eléctrica**.** Existen varios tipos de resistencias, para conocer las caracteristicas de una resistencia basta con mirar el codigo que ellas poseen representado por lineas de colores.

**Fotocelda:** Es un componente electrónico cuya resistencia varía en función de la luz. El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (puede descender hasta 50 ohms) y muy alto cuando está a oscuras (varios megaohmios).

****

**Transistor:** Es un dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada. Cumple funciones de amplificador, oscilador, conmutador o rectificado. Se le conoce tambien como resistor de transferencia.



**Condensador:** Es un dispositivo pasivo, utilizado en electricidad y electrónica, capaz de almacenar energía sustentando un campo eléctrico**.** Tambien se le conoce como capacitor.

****

**Interruptor:** Es un aparato utilizado para la interrupcion de la corriente electrica en un circuito.El interruptor de la imagen es solo un interruptor simple, existen otros tipos que se accioinan mediante señales de algun tipo.

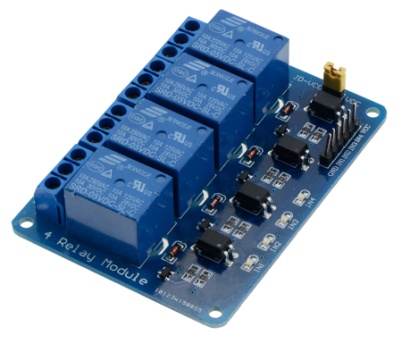


**Baterias:** Es un dispositivo electroquímico el cual almacena energía en forma química. Cuando se conecta a un circuito eléctrico, la energía química se transforma en energía eléctrica. La bateria o baterias deben colocarse en un portapilas que posteriormente debera conectarse al protoboard.

**Modulo Bluetooth:** Es un componente electonico que permite al usuario,mediante el uso de un smartphone, controlar el funcinamiento de un prototipo por via bluetooh. El modelo mas popular para Arduino es el modelo HC-06.

****

**Modulo Wi-Fi:** Es un componente electonico que permite al usuario,mediante el uso de un smartphone, controlar el funcinamiento de un prototipo por via Wi-Fi. El modelo mas popular para Arduino es el modelo ESP8266.

**Modulo de reles:** Es un componenete electronico adaptable a la placa Arduino y existen versiones de varios reles (en la imagen, 4 reles). Un rele es basicamente un electroiman que cuando se excita (recibe corriente) mueve la posición de un contacto eléctrico de conectado a desconectado o viceversa.

****

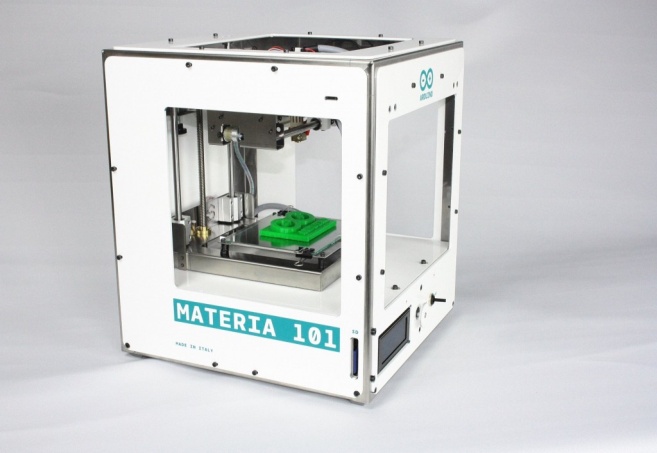
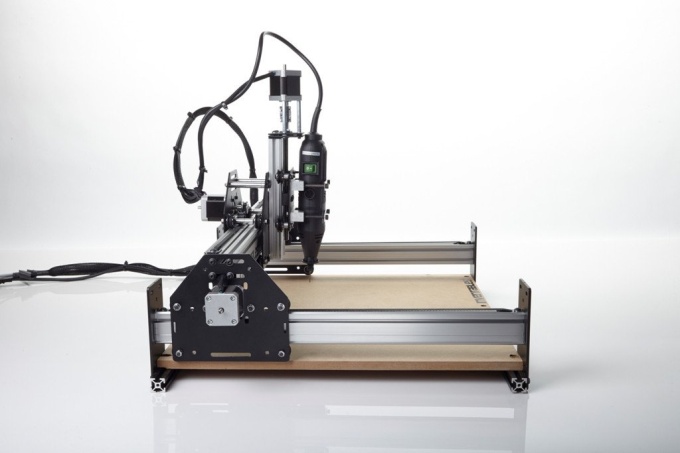
**Teclado:** Es un componente electronico que se conecta directamente a la placa Arduino, consiste en un teclado numerico 4x3 similar al de las calculadoras de bolsillo y se utiliza para proporcionar datos numericos al Arduino.

Estos son solo algunos componentes y seria imposible enlistarlos todos ya que al ser de uso libre, usted literalmente podria crear su propio componente electronico adaptable a las placas Arduino y programarlo para las funciones que usted requiera necesarias.

**APLICACIONES DE ARDUINO**™

Las aplicaciones de Arduino tienen lugar en todo tipo de areas tales como arte, musica, mecanica, electrica, informatica, entre otros. Esto se debe a que los limites de Arduino solo estan definidos por el usuario ya que las placas, al ser de uso libre, pueden modificarse a gusto de la persona que lo usa. Teniendo en cuenta esto, se hace logico la diversidad de proyectos que uno puede encontrar.

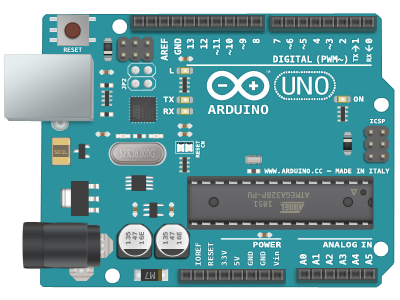
Dichos proyectos pueden facilitar en gran medida la vida de las personas y su trabajo en la industria, como lo podria ser el desarrollo de una maquina CNC (Computer Numerical Control) la cual se utiliza para el mecanizado de piezas, una impresora 3d para los muchos propositos que esta tiene o algun tipo de drone para transportar objetos livianos, entre otros. Estos con la ventaja de lo economicos que resultan en comparacion al precio que conlleva adquirir uno de estos productos en el mercado.



Este tipo de proyecto, por lo general, se encuentran comercializados en presentaciones de “Kits” donde el Kit cuenta con piezas especiales destinadas a la realizacion y ensamble de los proyectos especificados en el kit. Sin embargo, otros proyectos pueden realizarse con componentes no tan particulares. A continuacion, se señalan algunos proyectos emsamblados utilizando la placa Arduino UNO, la placa Arduino mas comun en el mercado.

**PROYECTOS CON ARDUINO**™**UNO**

Para la realizacion de cualquier proyecto, usted primeramente debera tener conocimiento sobre las caracteristicas de la placa Arduino que usted posee. En este caso, se detalla las caracteristicas de Arduino UNO:



Abreviaturas:

GND: Ground. Aquí se inserta el pin que conecta a “Tierra” en el protoboard.

A*n:* Analog *n.* Aquí se inserta el pin analogico numero *n*

*n*V: *n* Volts. Aquí se inserta un pin que trasfiere *n* Voltios

TX: Transmition. Aquí se inserta un pin para transmitir algun serial

RX: Read. Aquí se inserta un pin que lee algun serial

Especificaciones Tecnicas:

Marca: Arduino

Modelo: UNO (Modelos UNO REV X. significan Modelo Revisado por X vez. En este caso se detalla la version UNO REV3 ya que es la mas nueva al año 2017)

Microcontrolador: ATmega328P

Voltaje de entrada minimo necesario: 5 Voltios

Voltaje de entrada recomendado: 7 a 12 Voltios

Voltaje de entrada maximo permitido: 20 Voltios

Puertos para Pines Digitales: 14 pines… 6 pines son de señales PWM (pulse-width modulation)

Puertos para Pines Analogos: 6 pines

Corriente directa a suministrar por pin digital: 20 miliamperios

Corriente directa a suministrar para pin 3.3V = 50miliamperios

Entrada USB: 1 entrada

Memoria Flash: 32Kb

SRAM: 2KB

EEPROM: 1KB

Frecuencia o Velocidad de reloj (ClockSpeed): 16 Megahertz

LED\_BUILTIN: 13

Largo: 68.6mm

Ancho: 53.4mm

Peso: 25g

**EJEMPLOS CON ARDUINO**™**UNO**

En esta seccion se detallan algunos ejemplos en cuanto al ensamble de proyectos basicos utilizando la placa Arduino UNO, con la finalidad de que el lector pueda entrar a la practica de robotica y electronica. Los proyectos de ejemplo aquí a tratar, son los siguientes:

Proyecto nº 1: Reloj Digital

Proyecto nº 2: Movil esquivador de obstaculos

Proyecto nº 3: Mano robotica articulada

A continuacion, los proyectos:

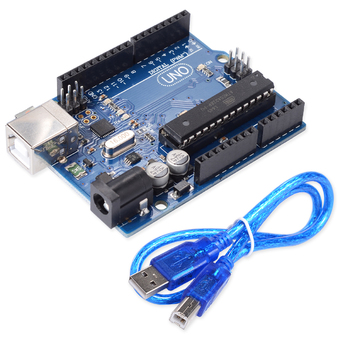
**PROYECTO Nº 1: Reloj Digital**

Para empezar con el proyecto, se hace necesario descargar el software Arduino con el cual podra desarrollar el codigo en la placa Arduino. Los componentes necesarios son:

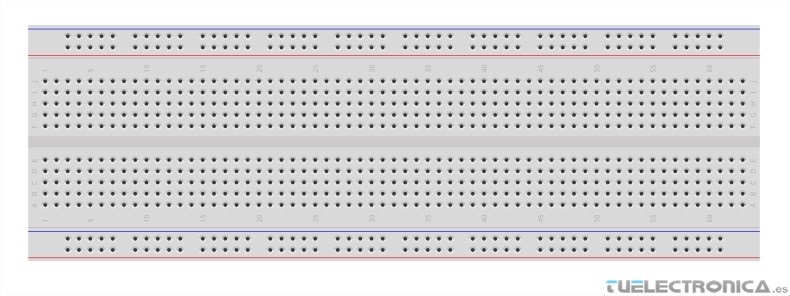
* 1 computador



* 1 placa Arduino UNO y el respectivo cable USB



* 1 Protoboard de 830 puntos



* 1 Display LCD



* 1 controlador I2C

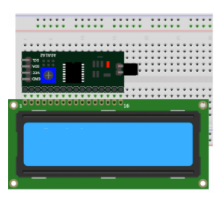


* Cables conectores de 4 pines

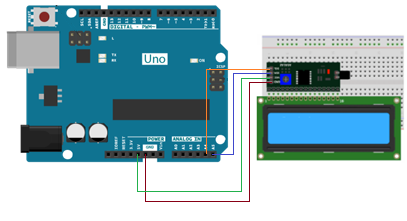


**INSTRUCCIONES:**

1- Conecte el display LCD y el controlador I2C en el protoboard de 830 puntos. El controlador posee 16 pines y el display LCD posee 16 zocalos, estos deben juntarse e insertarse en el protoboard, esto provocaria que la pantalla del display quede boca abajo. Por lo tanto, realice la conexión tal y como se muestra en la figura. (Notese que los 16 pines del controlador se encuentran paralelos a los 16 pines del display LCD)



2- Utilizando los cables conectores de 4 pines, conecte los sockets correspondientes del controlador 12C a los sockets de la placa Arduino UNO.



Las conexiones deben ser realizadas de tal manera que:

Controlador Arduino

GND GND

VCC 5V

SDA A5

SCL A4

3- Conecte un extremo del cable USB al Arduino UNO y conecte el otro extremo al computador (Esto se realizar para poder desarrolar la programacin y a su vez, propocionar energia a la placa y al display LCD, de otra forma se necesitaria una fuente de corriente directa).



4- Desarrolle el codigo en el software de Arduino previamente descargado al computador, por lo general el software suele leer el modelo de placa por lo que no tiene que especificarlo. (En el codigo se debe especificar el tamaño de pantalla del display LCD, en este caso, se supone 16x2 que es el que se muestra en la imagen). El codigo es el siguiente (se presenta de esta forma para la comprension del lector, en la practica se pretende que el codigo sea lo mas corto posible para que consuma la menor cantidad de bytes):

*#include <LiquidCrystal\_I2C.h>* //Esto es una libreria  
  
*LiquidCrystal\_I2C lcd(A0, A1, 10, 11, 12, 13);  
  
int hora;  
int minutos;  
int segundos;  
  
void setup() {  
lcd.begin(16, 2);  
hora = 0;  
minutos = 0;  
segundos = 0;  
}*// Esto se repetiria una y otra vez hasta que ocurra un reset   
*void loop() {  
lcd.clear();  
lcd.print(String(hora) + ":" + String(minutos) + ":" + String(segundos)); segundos = segundos + 1;  
if (segundos == 59) {  
minutos = minutos + 1;  
segundos = 0;  
}  
if (minutos == 59) {hora = hora + 1;}  
if (hora == 13)  
{  
hora = 1;}  
delay(10);}*

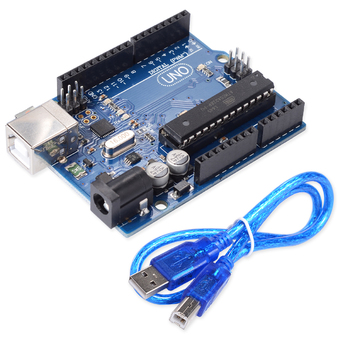
**PROYECTO Nº 2: Movil esquivador de obstaculos**

Para empezar con el proyecto, se hace necesario descargar el software Arduino con el cual podra desarrollar el codigo en la placa Arduino. Los componentes necesarios son:

* 1 computador



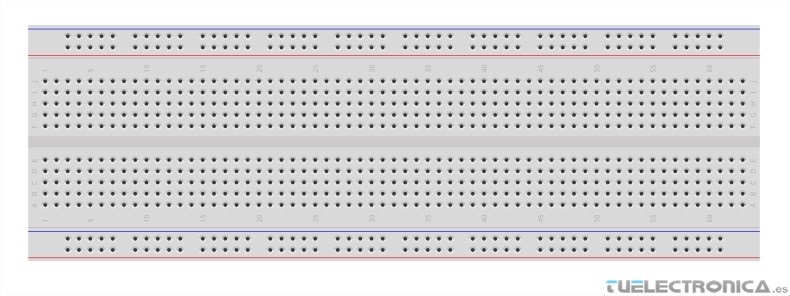
* 1 placa Arduino UNO y el respectivo cable USB

****

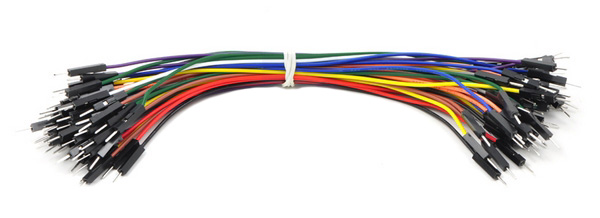
* 1 Sensor ultrasonico HC-SR04



* 1 Protoboard de 830 puntos (o menos)



* Jumpers para protoboard o Cables de conexion



* Baterias y un porta baterias (de las mismas dimensiones que las baterias)



* 2 Motores electricos



* 1 Puente H



* 2 Ruedas pequeñas (pueden ser de algun juguete)



* 1 Rueda seguidora



* Materiales de bricolaje para el chasis. Carton, Plastico y Tornillos



**INSTRUCCIONES:**

1- Tome el sensor ultrasonico y conecte sus 4 terminales a la placa Arduino UNO, el sensor contiene 4 pines: VCC, Trig, Echo, GND. Estas deben conectarse a la placa, mediante los Jumpers o un cable de 4 pines, de la siguiente manera:

Sensor Arduino

VCC 5V

Trig Pin 9 (Pin Digital)

Echo Pin 8 (Pin Digital)

GND GND

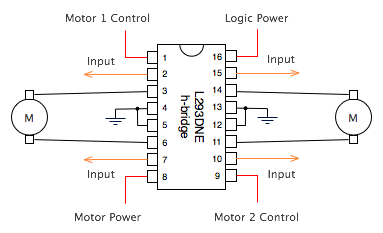
Debido a que se tienen varios componentes electronico, usted posiblemente tendra que conectar 5V y GND de la placa al protoboard, posteriormente colocar el VCC y GND del sensor al protoboard

\*VCC representa al voltaje requerido, Trig (Triggered) representa una señal de entrada, Echo repesenta una señas de salida y GND (Ground) representa a “Tierra”.

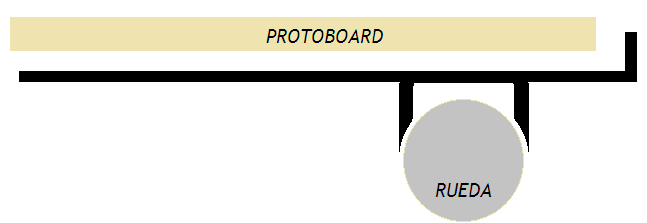
2- Construir el chasis del movil. Este se realiza de plastico o carton ya que su estructura debe ser lo mas liviana posible para no forzar los motores, ya que estos deben ser capaces de mover el peso de los compoenentes. Este chasis o armadura debe ser simple y basicamente ser un soporte para el protoboard, la placa Arduino, las baterias y el sensor, debajo de este soporte van los motores, asi como las ruedas principales y la rueda seguidora.

3- Ensamble, mediante cinta aislante, silicona o cualquier pegamento de su preferencia, los dos motores a las dos ruedas (un motor a cada rueda). Estas ruedas se venden en tiendas de electronica pero usted puede usar cualquiera siempre y cuando tenga en cuenta que estas deben ser de igual diametro, no pueden ser macisas (entre mas masa, mas fuerza de giro necesitan los motores ya que esta condicion se da por Trabajo = MasaxAceleracionxLongitud) y que estas contengan un orificio en su centro que encaje con los ejes de los motores

4- Coloque el puente H en el protoboard, se utiliza para que una vez el movil avance y detecte un obstaculo, el puente H indique que los motores puedan hacer un recorrido en reversa. Este dispositivo posee 16 “patas” las cuales controlan los motores, se insertan las patas al protoboard y se conectan los jumpers de los motores a las patas 3 – 6 y 11-14. En la imagen se detallan la funcion de cada pata del puente H.



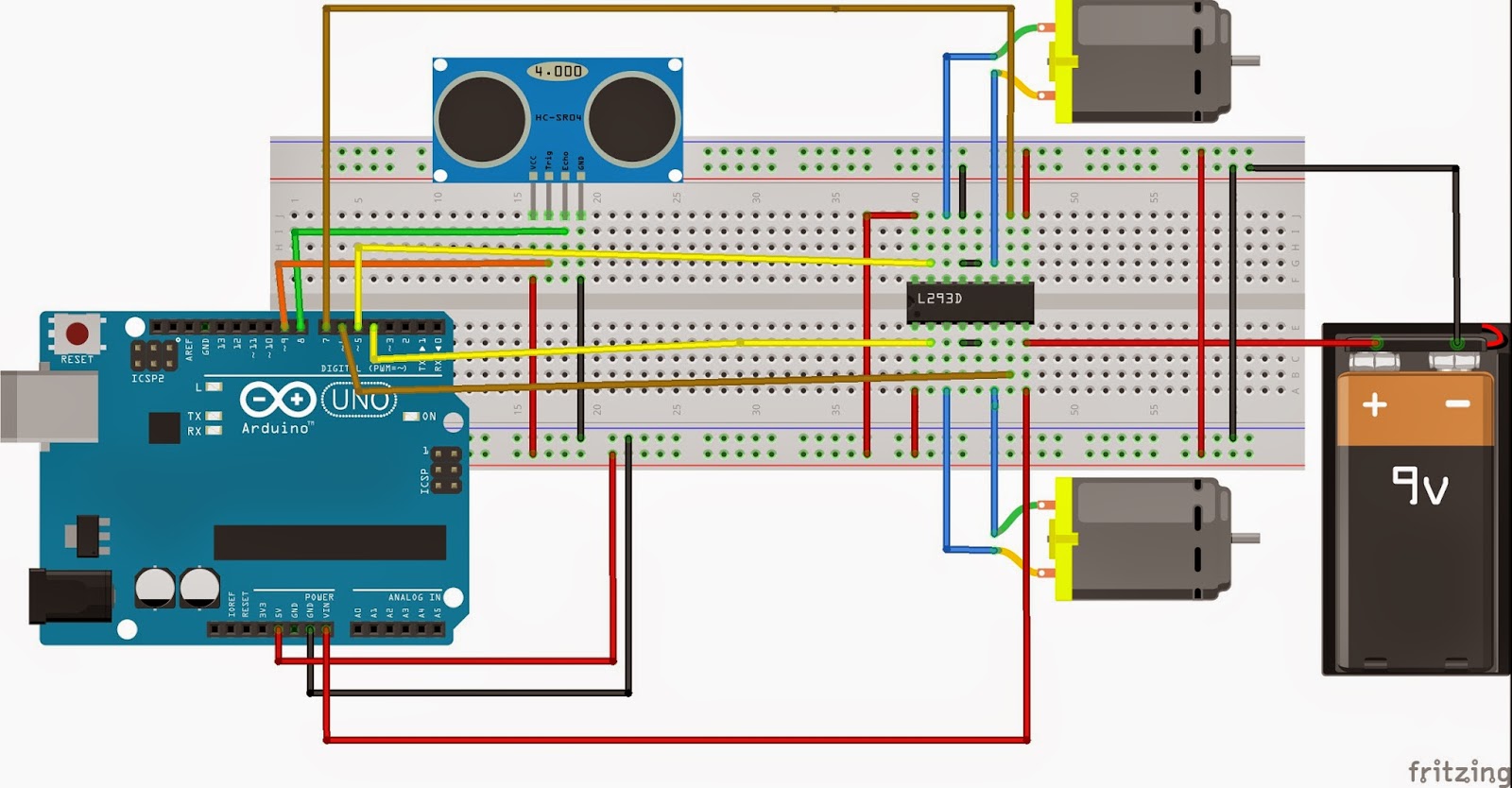
5- Para este paso, se concluye que usted tiene: Una placa Arduino y un puente H conectado al protoboard, un sensor ultrasonico conectado a la placa Arduino y al protoboard, dos motores emsamblados con dos ruedas y un chasis preparado. En este paso, junte de manera firme, con un pegamento de calidad, los dos motores y ruedas a la parte inferior del chasis y cuidadosamente, coloque el protobard en la parte superior. Usted puede saltase este paso compra el chasis en alguna tienda, el cual es un chasis que aunque algo costoso, es muy comun.



6- Coloque la rueda seguidora al final del chasis, en su parte inferior. Esta rueda se utiliza, ya que debido a las dimensiones del protoboard, para brindar estabilidad al prototipo. Otras opciones podrian seer optar a un protoboard mas pequeño con menos puntos de conexión o utilizando 4 motores y 4 ruedas hacinedo alusion a un automovil, en ambos casos, uisted tendri que hacer un gasto de mas.

7- Conecte los motores al protoboard, utilizando el diagrama suministrado en el apartado del Puente H. En los extremos de ambos motores sobresalen dos cables de colores, se tiene por norma que el cable de color mas vivo representa al polo positivo mientras que el otro representa al polo negativo.

8- Coloque la bateria o baterias en un porta baterias (las baterias deben proporcionar un voltaje entre los 6V y 10V). Dicho portabaterias se coloca encima del protoboard y en uno de los extremos del portabaterias sobresalen dos cables que son los encaragados de transmitir el voltaje, uno debe conectarse al GND del arduino o protoboard mientras que el otro se conecta al puerto VIN (Volt Input) de la placa Arduino UNO. En la siguiente imagen se muestran las conexiones finales:



9- Desarrolle el codigo en el software de Arduino previamente descargado al computador, por lo general el software suele leer el modelo de placa por lo que no tiene que especificarlo. El codigo en este caso seria el siguiente (se presenta de esta forma para la comprension del lector, en la practica se pretende que el codigo sea lo mas corto posible para que consuma la menor cantidad de bytes):

*#define echopin 8 // echo pin*

*#define trigpin 9 // Trigger pin*

*int maximumRange = 30;*

*long duration, distance;*

*void setup() {*

*Serial.begin (9600);*

*pinMode (trigpin, OUTPUT);*

*pinMode (echopin, INPUT );*

*pinMode (4, OUTPUT);*

*pinMode (5, OUTPUT);*

*pinMode (13, OUTPUT);*

*pinMode (6, OUTPUT);*

*pinMode (7, OUTPUT);*

*}*

*void loop ()*

*{*

*{*

*digitalWrite(trigpin,LOW);*

*delayMicroseconds(2);*

*digitalWrite(trigpin,HIGH);*

*delayMicroseconds(10);*

*duration=pulseIn (echopin,HIGH);*

*distance= duration/58.2;*

*delay (50);*

*Serial.println(distance);*

*}*

*if (distance >= 30 ){*

*digitalWrite(4,HIGH);*

*digitalWrite(5,HIGH);*

*digitalWrite(6,LOW);*

*digitalWrite(7,LOW);*

*delay (200);*

*}*

*else if (distance >=15 && distance <= 25) {*

*digitalWrite (4,HIGH);*

*digitalWrite (5,LOW);*

*digitalWrite (6,LOW);*

*digitalWrite (7,LOW);*

*delay (1000);*

*}*

*else if (distance < 15){*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6,HIGH);*

*digitalWrite (7,HIGH);*

*delay (1000);*

*digitalWrite (4,LOW);*

*digitalWrite (5,HIGH);*

*digitalWrite (6,LOW);*

*digitalWrite (7, LOW);*

*delay (1000);*

*} }*

\*Este prototipo aquí planteado se moveria sin ordenes a control remoto por parte de un humano, para hacer esto posible bastaria simplemente con insertar un modulo bluetooth o un modulo wi-fi en el protoboard, realizar las respectivas conexiones y controlarlo desde el smartphone del usuario, esto conllevaria tan solo unas cuantas lineas mas de codigo y la creacion de una app compatible con el sistema operativo del smartphone con el que se pretende operar el dispositivo ensamblado.

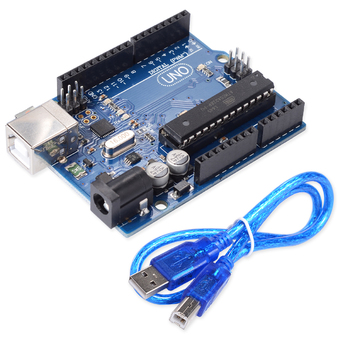
**PROYECTO Nº 3: Mano robotica articulada**

Para empezar con el proyecto, se hace necesario descargar el software Arduino con el cual podra desarrollar el codigo en la placa Arduino. Los componentes necesarios son:

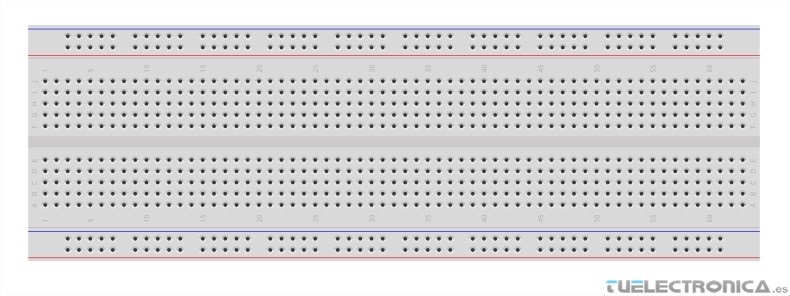
* 1 computador



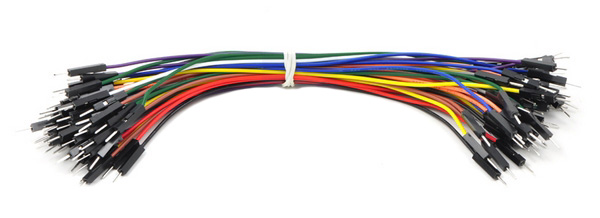
* 1 placa Arduino UNO y el respectivo cable USB

****

* 1 Protoboard de 830 puntos y 1 protoboard de 400 puntos



* Jumpers para protoboard o Cables de conexion



* 5 servomotores de al menos 9 gramos



* 5 sensores flex



* Baterias y 1 porta baterias (de las mismas dimensiones que las baterias)



* 1 Guante de mano derecha o mano izquierda



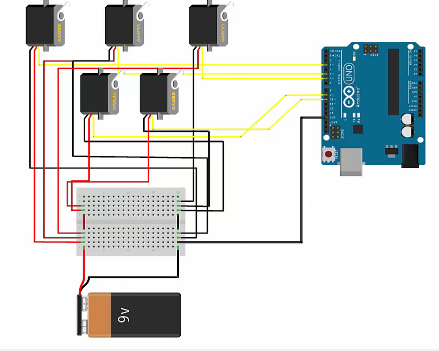
* 1 Material flexible y elementos de bricolaje. pegamento, cutters e hilos.



**INSTRUCCIONES:**

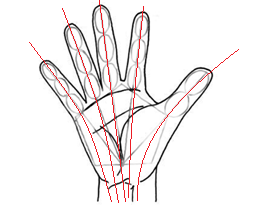
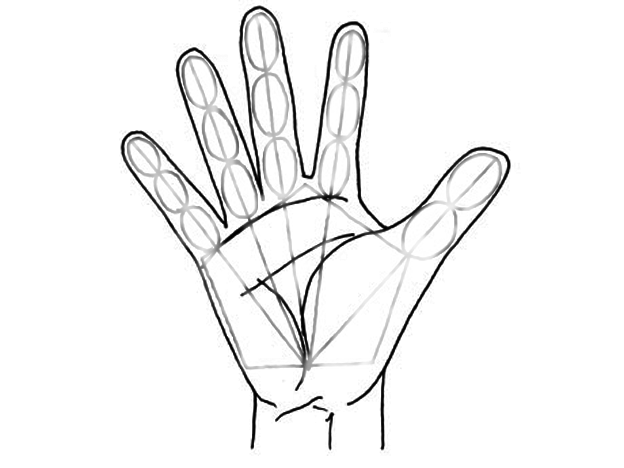
1- Tome la placa Arduino Uno y conecte el GND de la placa con el GND del protoboard de 400 puntos. Tome los servomotores, de los extremos de estos sobresalen 3 cables, por lo general suelen ser rojo (VCC), negro (GND) y un color claro (señal PWM, en este envía la señal codificada para comunicar el ángulo en el que se debe posicionar). Conecte cada cable negro de cada uno de los cinco servomotores en la “linea” GND del protoboard, conecte cada cable rojo de cada uno de los cinco servomotores en la “linea” VCC y cada cable de color claro debe ir conectado aun pin PWM de la placa Arduino, es decir un pin digital (cinco servomotores representan cincos pines digitales ocupados).

2- Se procede a conectar la fuente de alimentacion para los cinco servomotores, esta, como se ha visto hasta ahora, se conecta en los puertos del protoboard GND y VCC. La bateria debe proporcionar al menos nueve voltios, dependiendo de los servomotores se puede requerir mas voltaje. Tenga en cuenta que si usted añde resistencias al prototipo, tenga en cuenta que la resistencia es la proporcion del voltaje y la corriente en el sistema; entre mas voltaje se ubique en un sistema, mas altas deberan ser sus resistencias. Para este paso, usted deberia estar viendo lo que se muestra en la imagen

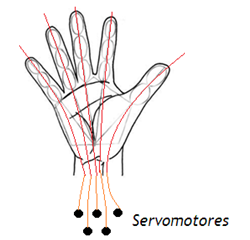


3-

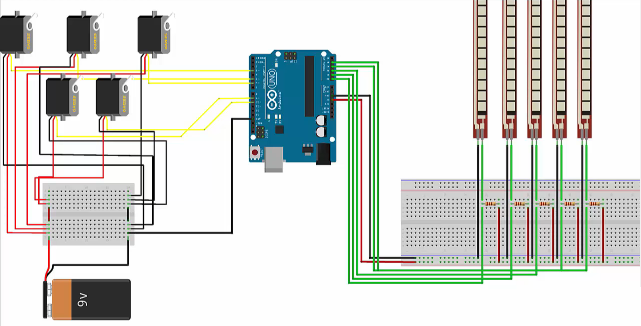
4- Utilice el material flexible como algun tipo de foami o goma espuma para crear el cuerpo y estructura de la mano robotica. No la realice como si fuera algun tipo de guante, para que este sea articulada puede guiarse de la imagen anexada, en ella se muestra la silueta de una mano derecha conteniendo a pequeñas figuras geometricas, son estas figuras las que se deben obtener del material. Suponiendo que el material es goma espuma, usted debera tener 15 piezas individuales. Inserte, con la ayuda de un alfiler u otro objeto afilado, un hilo a traves de la palma hacia cada dedo. Se puede eviatr esta trabaj teniendo la estructura de la mano impresa en 3D pero conlleva gastos que se tratan de evitar.



5- Conecte los hilos sobresalientes por el lado inferior de la palma a los ejes de los servomotores, esto provacara que haga cada revolucion o giro controlado de un servomotor represente un movimiento por parte de un dedo



6- En este paso se conectan los sensores flex, estos son sensores que existen mucho antes de la llegada de la placas Arduino y se utilizan donde la movilidad es un factor importante como lo es en este caso, por lo general su uso se dedica solo a guantes inteligentes, este sensor mide aproximadamente unos 8cm de largo. Este se conectara al protoboard de 830 puntos y no necesita mucha alimentacion en cuanto a voltaje. Estos sensores solo poseen dos cables, uno conectado al GND del protoboard y otro conectado a un puerto analogico en la placa Arduino. Esto significa que al tener cinco sensores flex, tendremos cinco puertos analogicos ocupados en el Arduino, en este paso ya se deberian tener las conexiones necesarias para la culminacion del prototipo, en la imagen se muestra el diagrama de las conexiones.



7- Coloque los sensores flex en un guante de algodón u otro material similar y fijelos firmemente con cinta aislante a los dedos, cada sensor debe ubicarse en un dedo, la posicion de cada sensor en cada dedo lo determina la posicion en la que se conectaron a los puertos analogos de la placa Arduino o a las definiciones que usted coloque en el codigo que conforma el el lenguaje de programacion Arduino basado en C/C++



En este momento, solo se requiere el codigo, se pretende que los movimientos que usted realice con su mano (mano en donde tiene colocado el guante) sean imitados por la mano que se realizo en pasos previos.

Usted no puede estar muy alejado de la mano robotica ya que recuerde que lo sensores flex siguen conectados al protoboard y que dicho sensores miden aproximadamente 8cm, existen maneras de realizar un prototipo que funcione de manera inalambrica, esto conlleva tan solo unos arreglos en el codigo asi como la implementacion de modulos para control a distancia

8- Desarrolle el codigo en el software de Arduino previamente descargado al computador, por lo general el software suele leer el modelo de placa por lo que no tiene que especificarlo. El codigo en este caso seria el siguiente:

*#include <Servo.h>*

*Servo servo\_1;*

*Servo servo\_2;*

*Servo servo\_3;*

*Servo servo\_4;*

*Servo servo\_5;*

*int flex\_1 = 0;*

*int flex\_2 = 1;*

*int flex\_3 = 2;*

*int flex\_4 = 3;*

*int flex\_5 = 4;*

*void setup()*

*{*

*servo\_1.attach(5);*

*servo\_2.attach(3);*

*servo\_3.attach(9);*

*servo\_4.attach(10);*

*servo\_5.attach(6);*

*}*

*void loop()*

*{*

*int flex\_1\_pos;*

*int servo\_1\_pos;*

*flex\_1\_pos = analogRead(flex\_1);*

*servo\_1\_pos = map(flex\_1\_pos, 1020, 1023, 180, 0);*

*servo\_1\_pos = constrain(servo\_1\_pos, 0, 180);*

*servo\_1.write(servo\_1\_pos);*

*int flex\_2\_pos;*

*int servo\_2\_pos;*

*flex\_2\_pos = analogRead(flex\_2);*

*servo\_2\_pos = map(flex\_2\_pos, 1020, 1023, 180, 0);*

*servo\_2\_pos = constrain(servo\_2\_pos, 0, 180);*

*servo\_2.write(servo\_2\_pos);*

*int flex\_3\_pos;*

*int servo\_3\_pos;*

*flex\_3\_pos = analogRead(flex\_3);*

*servo\_3\_pos = map(flex\_3\_pos, 1020, 1023, 0, 180);*

*servo\_3\_pos = constrain(servo\_3\_pos, 0, 180);*

*servo\_3.write(servo\_3\_pos);*

*int flex\_4\_pos;*

*int servo\_4\_pos;*

*flex\_4\_pos = analogRead(flex\_4);*

*servo\_4\_pos = map(flex\_4\_pos, 1020, 1023, 180, 0);*

*servo\_4\_pos = constrain(servo\_4\_pos, 0, 180);*

*servo\_4.write(servo\_4\_pos);*

*int flex\_5\_pos;*

*int servo\_5\_pos;*

*flex\_5\_pos = analogRead(flex\_5);*

*servo\_5\_pos = map(flex\_5\_pos, 1020, 1023, 180, 0);*

*servo\_5\_pos = constrain(servo\_5\_pos, 0, 180);*

*servo\_5.write(servo\_5\_pos);*

*}*

**LIBRERÍA DE COMANDOS DE ARDUINO**™

**COMANDOS DE CONTROL**

|  |  |
| --- | --- |
| **IF**  Comprueba si cierta condición se cumple y puede ser usado en conjunto con uno o más operadores de comparación (**==** igual, **!=** distinto, **<** menor, **>** mayor) | if (a == b) { **codigo**} |
| **IF…ELSE**  Permite agrupar múltiples comprobaciones | if (a < b) { **codigo1**} else { **codigo2**} |
| **FOR**  Se inicializa una variable, se comprueba una condición y ejecuta un bloque, luego se comprueba nuevamente la condición y asi sucesivamente hasta que la condición ya no sea válida | for (int a = 0; a < 10; a++) { **codigo**} |
| **SWITCH**  Compara el valor de una variable con el valor especificado en las sentencias "case". Cuando se encuentra una sentencia cuyo valor coincide con dicha variable, se ejecuta el código de esa sentencia | switch (x) { case 1: **codigo si x=1** break; case 2: **codigo si x=2**break; default: **codigo si x ≠ 1 o 2** } |
| **WHILE**  Se ejecuta un bloque hasta que la condición deje de cumplirse | while(a > b) { **codigo** } |
| **DO…WHILE**  Trabaja de la misma manera que el bucle “WHILE”*,* con la excepcion de que la condición se comprueba al final del bucle, por lo que se ejecuta al menos una vez | do { **codigo**}  while (a > b) |
| **BREAK**  Es usado para salir de los bucles DO*,*FOR*,* oWHILE, pasando por alto la condición normal del bucle. Es usado también para salir de una estructura de tipo SWITCH | while (a > b) {  **codigo**if(a == 5)  {break} } |
| **CONTINUE**  **Es usado para crear saltos en bucles DO, FOR,WHILE cuando se cumple una sentencia previamente establecida** | for (**sentencia**)  {if (**condicion**)  continue;} |
| **RETURN**  **Devuelve un valor establecido cuando se cumple una sentencia del tipo IF o IF.ELSE** | If(**sentencia**)  return **codigo1**;  else  return **codigo2** |
| **GOTO**  **No se recomienda su uso ya que muchos GOTO pueden hacer que el programa no funcione correctamente. Sin embargo en algunos casos, pueden simplificar codigos** | for (**sentencia1**) {  for(**sentencia2**) {  if(**condicion**) goto{**final**}  **n**for (**n sentencias**) }}}  **final** |

**FUNCIONES DIGITALES**

|  |  |
| --- | --- |
| **PINMODE**  Configura el pin como una entrada o una salida | pinMode(pin,  OUTPUT/INPUT) |
| **DIGITALWRITE**  Asigna un valor de 1 o 0 en un pin de salida | digitalWrite(pin, HIGH/LOW) |
| **DIGITALREAD**  Lee el valor de un pin configurado como una entrada | digitalRead(pin) |

**FUNCIONES ANALOGICAS**

|  |  |
| --- | --- |
| **ANALOGREFERENCE**  Configura el voltaje de ref usado por la entrada | analogReference(DEFAULT) |
| **ANALOGREAD**  Lee el valor de tensión en el pin analógico. Se representa con un numero entero entre 0 y 1023 | analogRead(pin) |
| **ANALOGWRITE**  Asigna un valor en señal PMW en un pin analogico | *analogWrite(pin,* ***valor****)* |

**TIPOS DE DATOS**

|  |  |
| --- | --- |
| **VOID** | Cuando se declara, indica que una funcion no devuelva ninguna informacion a la funcion de la que fue declarada. |
| **BOOLEAN** | Un booleano es aquel que solo puede ocupar dos valores: verdadero o falso (true or false) |
| **CHAR** | Un char/character es un caracter de tipo alfabetico, que puede ser tratado como un numero si se utiliza la tabla ASCII correcta |
| **BYTE** | Un byte es cualquier numero que ocupa 8-bit, puede ir desde 0 hasta 255 |
| **INT** | Int/Integer (Entero) es el tipo de dato primario para almacenar numeros |
| **WORD** | En las placas Arduino UNO y ATMEGA, es un tipo de dato que puede almacenar un numero de 16-bit. En las placas DUE y ZERO, puede almacenar un numero de 32-bit |
| **LONG** | Long es un tipo de dato para numeros de extenso tamaño y almacena hasta numeros de 32 bits (4 bytes) |
| **SHORT** | Short, en todas las placas Arduino, es un tipo de dato para numeros de hasta 16-bit (2 bytes) |
| **FLOAT** | Float es un tipo de dato para numero con punto decimal, este puede almacenar numeros de 32-bit y soportan hasta numeros con 38 decimales. Sin embargo, las operaciones matematicas de numeros flotantes solo arrojan de 6-7 digitos de precision. |
| **DOUBLE** | Se utiliza para incrementar la capacidad de almacenaje de float de 32-bit a 64-bit. Esto solo funciona en la placa Arduino Due |
| **STRING** | String puede clasificarse como un tipo de dato y/o como una clase. Este permite manipular largas cadenas de texto pero hace uso intensivo de la memoria. Se puede utilizar en dos casos:  Con (Char Arrays) = Utilizar *string*  Con (Objetos) = Utilizar *String* |
| **ARRAY** | Es una coleccion de variables a la que se puede acceder mediante un numero indice. |

*Para mas informacion sobre la estructura de los comandos aquí anexados, al final de este manual, encontrara una infografia sobre dichas estructuras.*

*Si desea profundizar en comandos y funciones avanzadas del lenguaje de programacion Arduino, visite la pagina de referencias oficial de Arduino* [*www.arduino.cc/reference/*](http://www.arduino.cc/reference/) *(Solo disponible en ingles).*

**APENDICE A: Glosario de terminos**

**Actuador**: transductor que transforma señales eléctricas en movimientos mecánicos.

**Algoritmo**: Conjunto definido de reglas o procesos que llevan a la solución de un problema en un número determinado de pasos

**Armadura**: Conjunto de componentes que se utilizan para brindar soporte o mejorar la estetica de un elemento sin entorpecer su movilidad.

**Automata**: Aparato que contiene los mecanismos necesarios para ejecutar ciertos movimientos o tareas similares a las que realiza una persona

**Chip**: Circuito electrónico de material semiconductor, especialmente silicio, en forma de cubo minúsculo, que, combinado con otros componentes, forma un sistema integrado más complejo y realiza una función electrónica específica

**Circuito**: Conjunto de conductores que son recorridos por una corriente eléctrica, y en el cual se encuentran intercalados, aparatos productores o consumidores de una corriente.

**Circuito Integrado**: Chip en el que se encuentran todos o casi todos los componentes electrónicos necesarios para realizar alguna función. Entre estos componentes se tienen: transistores, resistencias, diodos y condensadores.

**Codigo**: Sistema de símbolos y reglas que permite componer y descifrar un programa.

**Coordenadas**: Par de números reales expresados en un cierto orden que determinan la posición de un punto o objeto en una recta, en un plano o en un espacio.

**Condensador**: Dispositivo eléctrico que permite acumular cargas eléctricas

**Controlador**: Parte de cualquier software que permite contrlar un elemento o componente en particular.

**Control Remoto**: Manipulador de: Aquél en que cada grado de libertad está actuado por un dispositivo independiente, con lo que puede no estar unido cinemáticamente al actuador del operador

**Corriente Alterna**: Tipo de corriente eléctrica, en la que la dirección del flujo de electrones va y viene a intervalos regulares o ciclos

**Corriente Directa**: Tipo de corriente eléctrica que fluye de forma constante en una dirección. La corriente que fluye en una linterna o en cualquier otro aparato con baterías es corriente continua

**Diodo**: Dispositivo de dos terminales que se comporta como un interruptor común con la condición especial de que solo puede conducir la corriente eléctrica en una sola dirección.

**Dispositivo**: Mecanismo de un aparato o equipo que, una vez accionado, desarrolla de forma automática la función que tiene asignada.

**Eje**: Es cada una de las líneas por las que se puede mover el robot o algún elemento de su estructura). Cada eje define un grado de libertad del robot

**Informatica**: Conjunto de conocimientos técnicos que se ocupan del tratamiento automático de la información por medio de computadoras.

**Inteligencia Artificial**: Programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana, como el autoaprendizaje.

**Interfaz**: Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.

**Interruptor**: Dispositivo para abrir o cerrar el paso de corriente eléctrica en un circuito.

**LCD**: Sigla del inglés *Liquid Cristal Display*, 'representación visual por cristal líquido', sistema que utilizan determinadas pantallas electrónicas para mostrar información visual.

**LED**: Sigla del ingles *Light-Emitting Diode*, ‘diodo emisor de luz’, que es un tipo de diodo empleado en computadoras y paneles numéricos como relojes digitales y calculadoras

**Lenguaje de Programacion**: Programa mediante el cual se puede especificar una serie de instrucciones para que el computador efectúe una tarea específica y autónoma.

**Pin**: Pequeña clavija terminal de un circuito eléctrico.este puede ctuar como una entrada de señal digital o analogica.

**Pin Analogo**: Entrada de señal eléctrica no digital. Las señales análogas o analógicas tienen un número infinito de valores entre los límites altos y bajos de una señal portadora.

**Pin Digital**: Entrada que utiliza variaciones discretas en voltaje, frecuencia, amplitud, ubicación, etc; para cifrar, procesar o transportar señales binarias para datos informáticos, sonido, vídeo u otra información.

**Placa**: Pieza sobre la cual se encuentra un circuitos impreso y/o chips.

**Potencia**: Capacidad para producir trabajo, que se mide por la cantidad de trabajo realizado por una fuerza en una unidad de tiempo

**Protoboard**: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos

**Prototipo**: Objeto que reúne en grado máximo las características principales de cierto tipo de cosas y puede representarlas.

**Proyecto**: Idea o percepcion de una cosa que se piensa hacer y para la cual se establece un modo determinado y un conjunto de medios necesarios.

**Resistencia**: Oposición que presenta un conductor al paso de la corriente eléctrica.

**Sensor**: Dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno.

**Semiconductor**: Materiales no conductores que mediante la adición de ciertas impurezas, conducen la corriente bajo ciertas condiciones. No es directamente un conductor de corriente, pero tampoco es un aislante.

**Servo/Servomecanismo**: Dispositivo de mando y regulación que corrige automáticamente el valor de una variable para que se mantenga el valor deseado.

**Sistema**: Conjunto organizado de elementos diferenciados cuya interrelación e interacción suponen una función global.

**Solenoide**: Bobina formada por un alambre enrollado en espiral sobre una armazón cilíndrica, que se emplea en diversos aparatos eléctricos, y que crea un campo magnético cuando circula una corriente continua por su interior.

**Terminal de corriente**: Borne o hembrilla en el extremo de un hilo conductor para facilitar la conexión con un aparato.

**Transductor**: Dispositivo que tiene la misión de recibir energía de una naturaleza eléctrica, mecánica, acústica, etc., y suministrar otra energía de diferente naturaleza, pero de características dependientes de la que recibió.

**Ultrasonido**: Vibración mecánica de frecuencia superior a la de las que puede percibir el oído. Los sensores ultrasonicos pueden usar esta vibracion para percibir y medir las distancias que se encuentran frente a el.

**Voltaje**: Magnitud física que cuantifica la diferencia de potencial eléctrico entre dos puntos cualesquiera. También se le denomina tension electrica.

**RECOMENDACIONES DE TEXTOS**

La siguiente es una lista de libros y bibliografias, los cuales se enfocan en su mayoria a ramas derivadas de la electronica asi como la robotica. Estos textos son para profundizar en proyectos complejos que no requieran de alguna de las placas distribuidas y comercializadas por Arduino. A continuacion, dicha lista:

* Aguilar L. (1988). “Fundamentos de programacion: Algoritmos y Estructura de datos”. Estados Unidos. McGraw-Hill
* Deitel H., Deitel P. (1992). “Como programar en C/C++”. Estados Unidos. Pearson
* Malvino A., Bates D. (1994). “Principios de electronica”. Estados Unidos. McGraw-Hill
* Camps G., Espi J., Muñoz J. (2006). “Fundamentos de electronica analogica”. España. PUV
* Bueche F. (1991). “Fisica General”. Estados Unidos. McGraw-Hill
* Machut J. (2006). “Selección de componentes de electronica”. España. Marcombo
* Porcuna P. (2016). “Robotica y domotica con arduino”. Colombia. Ediciones de la U.