-Sensor ultrasónico HC-SR04

Este componente de cuatro pin, emite un pulso de disparo y **mide en tiempo alto** el pulso de respuesta. Con esto se hace una conversión de sus bits por segundo junto con la **velocidad del sonido para transformar** su información a distancia. Es decir. La función de estos es discriminar entre distancia larga – media – corta, para así poder dar aviso al interior que un cuerpo se acerca a mayor distancia, o a distancia media con los LED. Pero también **dar aviso al exterior(con el pìezo) de que** se está a una distancia corta, llamando la atención del cuerpo y además entregarle un mensaje a través del LCD.

CARACTERISITICAS PRINCIPALES

|  |  |
| --- | --- |
| * Voltaje de funcionamiento 5V | Distancia de detección 2 cm - 450cm |
| * Corriente estática Menos de 2 mA | Alta precisión de hasta 3 mm |
| * Ángulo del sensor Menos de 15 grados | * Peso 10g |

**CODIGO:**

**CÓDIGO ULTRASONIDO**



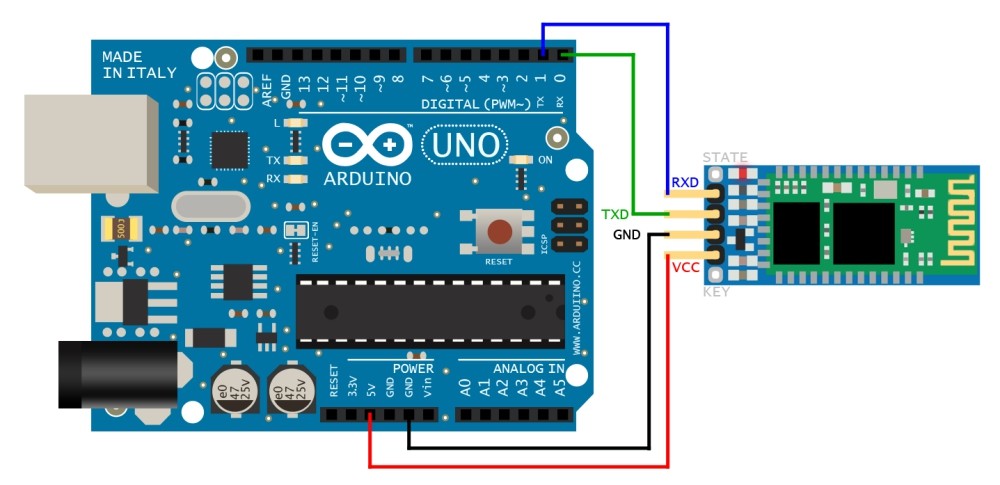
**-Placa Arduino compatible:**



-La placa Arduino se conecta a un **ordenador** a través de un USB, donde se conecta con el entorno de desarrollo Arduino (IDE). El usuario escribe el código de Arduino en el IDE, y luego lo sube al microcontrolador que ejecuta el código, interactuando con las entradas y salidas como sensores, motores y luces.

-El microcontrolador de Arduino posee lo que se llama una **interfaz de entrada**, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que conectes se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

-También cuenta con una **interfaz de salida**, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.



- Un led rojo:

Un LED es un diodo emisor de luz, es decir, **un tipo particular de diodo que emite luz** al ser atravesado por una corriente eléctrica. Los diodos (emisor de luz, o no) son unos de los dispositivos electrónicos fundamentales. Un diodo es una **unión de dos materiales semiconductores con dopados distintos. Sin entrar en detalles, esta diferencia de dopado hace que genere una barrera de potencial, que como primera consecuencia hace que el paso de corriente en uno de los sentidos no sea posible. Aquí tenemos la primera característica de los diodos, tienen polaridad, es decir, solo dejan pasar la corriente en un sentido. Por tanto, tenemos que conectar correctamente la tensión al dispositivo. La patilla larga debe ser conectada al voltaje positivo (ánodo), y la corta al voltaje negativo (cátodo).**

**CODIGO;**

|  |  |
| --- | --- |
| -Bajo consumo de energía. | - Larga vida útil |
| - Lámpara grande de 8 mm de diámetro. | - Fiabilidad de estado sólido. |
| - Fiable y resistente. | - Cumple con RoHS. |

****

- Dos resistencias; una de 1 Kohm y otra de 190 ohm

Los componentes electrónicos llamados resistencias se utilizan en los circuitos para variar los valores de intensidad y voltaje. A veces tenemos que alimentar un dispositivo y sólo disponemos de una fuente de voltaje que puede dañarlo si se conecta directamente, como ocurre por ejemplo con los ledes. Al conectarlos directamente a un pin digital de Arduino (+5V), la corriente que circula es demasiado alta para el led y una exposición prolongada puede provocar que se queme. Para evitar esto conectamos en serie con el led una resistencia eléctrica (220 ohmios) que hace que el valor de la intensidad sea menor. El led lucirá algo menos que si lo conectamos directamente pero alargará su vida útil.

CODIGO:

const int sensorPin = A0;

const int Rc = 1500; // valor de la resistencia de calibración

int V; // almacena el valor medido

long Rsensor; // almacena la resistencia calculada

void setup() {

}

void loop() {

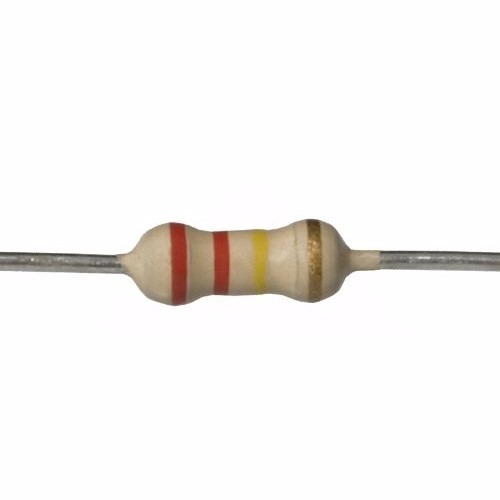
V = analogRead(sensorPin); //realizar la lectura

Rsensor = 1024L \* Rc / V - Rc; //calcular el valor de la resistencia

//...haríamos lo que quisieramos por Rsensor

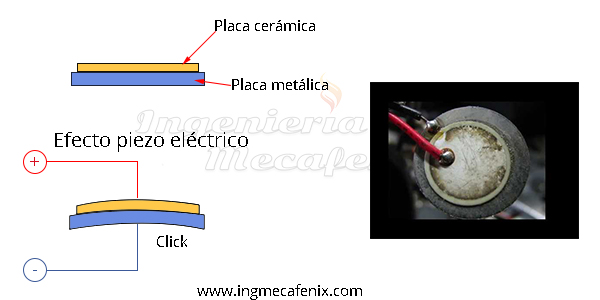
delay(1000);

}



- Buzzer pasivo;

l funcionamiento se basa en el efecto piezoeléctrico de los materiales, Este efecto funciona de tal manera que cuando aplicamos un voltaje el volumen del material cambia ligeramente. Los zumbadores están construidos con dos pequeñas placas una metálica y una cerámica, las cuales aprovechan este efecto pero solo generan un click ya que los materiales cambiaron de forma pero no regresan a su estado natural hasta que se les quita el voltaje.

[](https://www.ingmecafenix.com/wp-content/uploads/2018/10/Buzer-función.jpg)

Para que se pueda emitir un sonido continuo las placas necesitan vibrar constantemente, para eso se instala un oscilador que hace que los materiales cambien de estado una y otra vez, y así    puedan cambiar miles de veces para poder alcanzar un audio perceptible.

CODIGO:

void sonidoTerminado(){

  Serial.println("Sonido terminado");

}

void setup() {

  Serial.begin(9600);

  // Configuración del pin

  EasyBuzzer.setPin(5);

  Serial.println("Comienza el sonido");

  // Configuración del beep

  EasyBuzzer.beep(

    2000,          // Frecuencia en herzios

    100,           // Duración beep en ms

    100,           // Duración silencio en ms

    2,             // Números de beeps por ciclos

    300,           // Duración de la pausa

    1,             // Número de ciclos

    sonidoTerminado// Función callback que es llamada cuando termina

  );

**-Modulo I2C, Pantalla LCD1602A:**

El controlador de LCD I2C es un dispositivo que nos permite controlar una pantalla a través del bus I2C, usando únicamente dos cables.

El adaptador LCD a I2C tiene los pines ordenados para conectar directamente al LCD, esto lo podemos hacer a través de un protoboard o soldando directamente al LCD.

Para conectar con el modulo con el Arduino solo utilizamos los pines I2C del Arduino (SDA y SCL) y alimentación (GND y 5V), los pines I2C varían de acuerdo al modelo de Arduino con el que trabajemos, en la siguiente tabla podemos ver cuales son los pines I2C para cada modelo de Arduino.

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

//Crear el objeto lcd  dirección  0x3F y 16 columnas x 2 filas

LiquidCrystal\_I2C lcd(0x3F,16,2); //

void **setup**() {

  // Inicializar el LCD

  lcd.init();

  //Encender la luz de fondo.

  lcd.backlight();

  // Escribimos el Mensaje en el LCD.

  lcd.print("Hola Mundo");

}

void **loop**() {

   // Ubicamos el cursor en la primera posición(columna:0) de la segunda línea(fila:1)

  lcd.setCursor(0, 1);

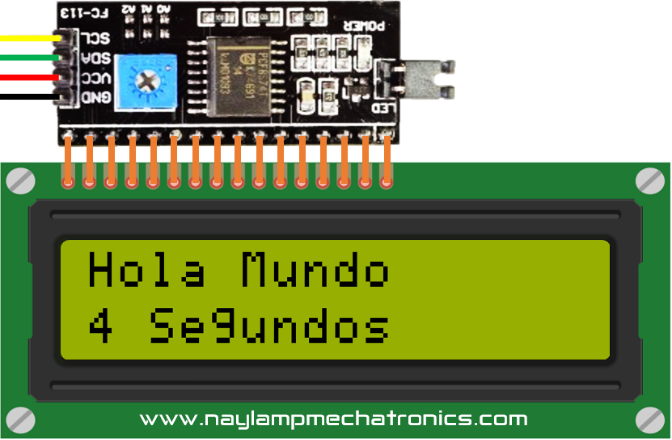
   // Escribimos el número de segundos trascurridos

  lcd.print(millis()/1000);

  lcd.print(" Segundos");

  delay(100);

}



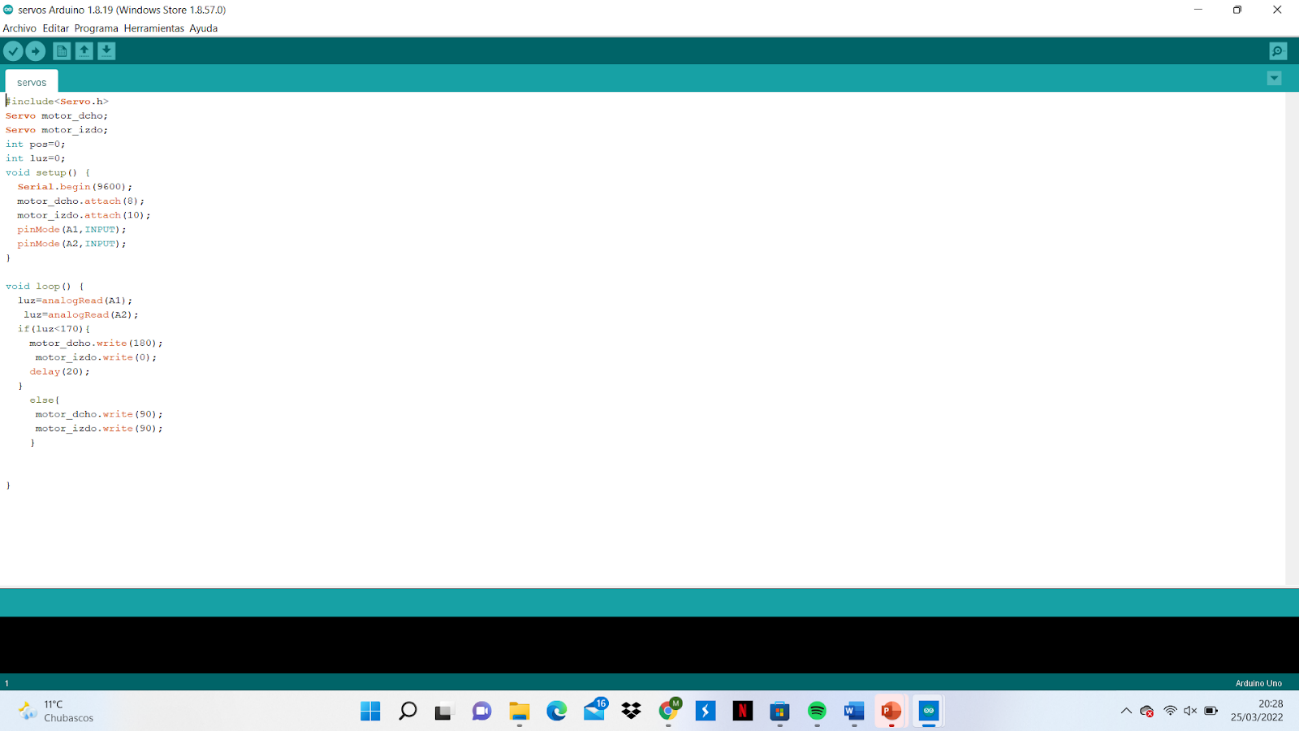
-Servomotor:

Los servomotores se controlan enviando un pulso eléctrico de ancho variable, o modulación de ancho de pulso (PWM), a través del cable de control. Hay un pulso mínimo, un pulso máximo y una frecuencia de repetición.

Por lo general, un servomotor sólo puede girar 90° en cualquier dirección para un movimiento total de 180°. La posición neutra del motor se define como la posición en la que el servo tiene la misma cantidad de rotación potencial tanto en el sentido de las agujas del reloj como en el sentido contrario. El PWM enviado al motor determina la posición del eje, y se basa en la duración del pulso enviado a través del cable de control; el rotor girará a la posición deseada.

El servomotor espera ver un pulso cada 20 milisegundos (ms) y la longitud del pulso determinará hasta dónde gira el motor. Por ejemplo, un pulso de 1.5ms hará que el motor gire a la posición de 90°. Si el tiempo es inferior a 1,5 ms, se mueve en sentido contrario a las agujas del reloj hacia la posición de 0°, y si el tiempo es superior a 1,5 ms, el servo girará en sentido de las agujas del reloj hacia la posición de 180°. Cuando se les ordena a los servos que se muevan, estos se moverán a la posición y mantendrán esa posición. Si una fuerza externa empuja contra el servo mientras el servo mantiene una posición, el servo se resistirá a salir de esa posición.

CODIGO;

****