**PONTIFICIA UNIVERSIDAD JAVERIANA**

****

**DOCUMENTO:**

Proyecto: Final Semestral

**CARRERA**

INGENIERÍA ELECTRÓNICA

**MATERIA**

LABORATORIO DE SEÑALES

**DOCENTE**

INGENIERO: JOHAN SEBASTIÁN PEÑA

**PRESENTADO POR**

DANIEL ANDRÉS PINEDA SUÁREZ

**BOGOTÁ, 2017**

**RESUMEN**

El presente proyecto se trata de la implementación de 2 artefactos. Por un lado, un sensor de proximidad que emplea un sistema configurado y programado mediante el microcontrolador Arduino Uno. Predispuesto para detectar objetos a más de 50 centímetros de distancia mediante ultrasonido (puntualmente el sensor HC-SR04) y emitir una señal dada en un pequeño parlante, a manera de pitido o timbre.

Por otra parte, se incorpora, el sistema principal, correspondiente a la base del proyecto. Un carro que detecta objetos a 15 centímetros de distancia, y los esquiva por medio de 2 motoreductores que le dan dirección y sentido que al igual que el anterior, emplea un sensor ultrasónico, el cual es programado sobre la tarjeta Arduino Mega 2560.

**ABSTRACT**

The present project deals with the implementation of 2 artifacts. On the one hand, a proximity sensor that uses a system configured and programmed by the Arduino Uno microcontroller. It is arranged to detect objects more than 50 centimeters away by ultrasound (punctually the HC-SR04 sensor) and emit a given signal in a small Speaker, as a beep or timbre.

On the other hand, it incorporates the main system, corresponding to the basis of the project. A car that detects objects at a distance of 15 centimeters, and dodges them by means of 2 geared motors that give direction and sense that like the previous one, uses an ultrasonic sensor, which is programmed on the Arduino Mega 2560 card.

**INTRODUCCIÓN**

El **ultrasonido** puede ser descrito como ondas mecánicas, esto quiere decir que requieren un medio de propagación (empleando una perturbación de dicho material). Además se caracteriza por poseer una frecuencia que se encuentra arriba del espectro audible humano (aproximadamente entre 20**Hz** -20 **kHz**).

Especies animales como los delfines y los murciélagos, quienes se valen para usarlo de forma parecida a un radar para su orientación. Las ondas emitidas por estos, son tan altas que propician un rebote en cada uno de los objetos alrededor de ellos, lo que posibilita, generar algo muy cercano a una representación de lo que se encuentra en torno a los mismos, para facilitar su orientación.

La **inspección por ultrasonido** *“se define como un procedimiento de inspección no destructivo de tipo mecánico, y su funcionamiento se basa en la impedancia acústica, la que se manifiesta como el producto de la velocidad máxima de propagación del sonido y la densidad del material”*.

Con los prototipos iniciales del instrumento, era posible medir una decadencia en la intensidad de las ondas sónicas, cuando se hacían viajar ondas supersónicas en un material, necesitando de 2 instrumentos: el receptor y el emisor.

Los instrumentos ultrasónicos empleados hoy en día permiten detectar discontinuidades superficiales e internas, dependiendo de las frecuencias seleccionadas. Aplicando el principio del ensayo de ultrasonido. Actualmente se utiliza un único aparato que funciona como emisor y receptor.

El principio básico se basa en el fenómeno productor la reflexión de las **ondas acústicas** en un objeto. Midiendo discontinuidades en la reflexión de las mismas.

**OBJETIVO GENERAL:**

Utilizar un sensor para registrar los datos de alguna variable, para obtener información sobre los valores registrados.

**DESCRIPCIÓN CONCEPTUAL DEL PROYECTO**

El proyecto versa básicamente, en 2 artefactos. El primero que tiene a su cargo dos motoreductores en L controlados, por medio del Arduino Mega 2560, el cual dirige el móvil, y lo abstiene de chocarse con elementos

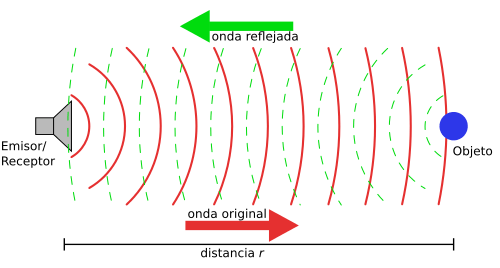
**MONTAJE**

* Dos sensores HC-SR04
* Un montaje para carro
* Arduino 1
* Arduino Mega 2560
* Dos motoreductores en L
* Batería LiPo
* Batería de 9 voltios

**PROCEDIMIENTO**

Al utilizar el módulo HCSR04, según características dadas por el fabricante, este emite un pulso modulado de ultrasonido que al propagarse en un medio ideal se mueve a 343m/s (como el vacío). Se debe tener en cuenta que el módulo posee, al igual que los actuales un receptor y un emisor incorporados.

Según siguiente que se muestra a continuación, se puede observar la onda original emitida por el sensor, que al chocar con un objeto, se refleja. Cabe resaltar, el tiempo transcurrido entre la emisión del pulso cuadrado y su posterior recepción. Para determinar la distancia es indispensable, tener una tarjeta Arduino.



*Ultrasónico.*  Recuperado de <https://liceoartabro.wordpress.com/2013/08/01/calibrado-de-un-modulo-de-ultrasonidos-hc/>

distancia = 340 \* (tiempo / 2)

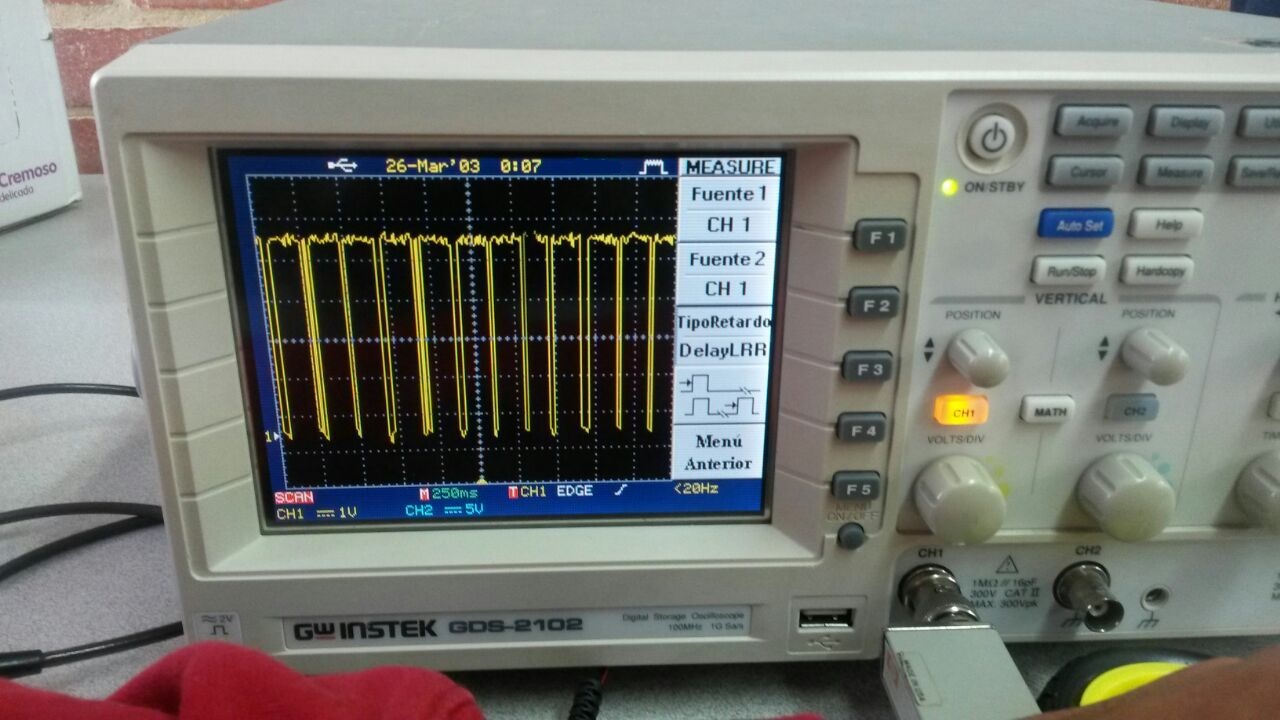
Para la configuración del puerto serial la variable utilizada es puntualmente necesario utilizar 9600 baudios, lo recomendado para iniciar la comunicación puerto serial.

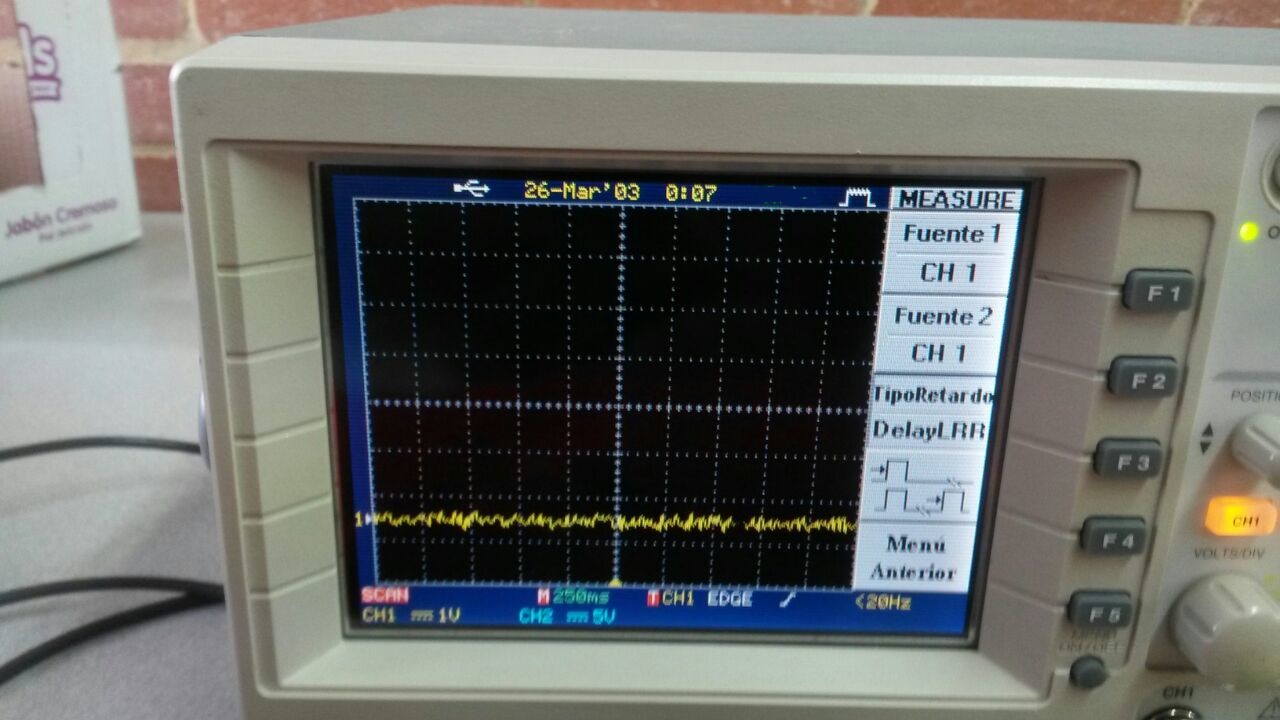
Luego, se procede a realizar la toma de medidas, se observan discrepancias en las mismas, observadas en errores entre medidas que deben ser continuas(Por ejemplo, un objeto quieto). Esto se debe a que el sensor no se encuentra calibrado. Para proceder a su calibración, se realiza una medición con una cinta métrica o regla. Encarando, de esta manera: medida real vs medida dada por el monitor serie del Arduino utilizado.

**MONTAJE GENERAL DEL ARTEFACTO**

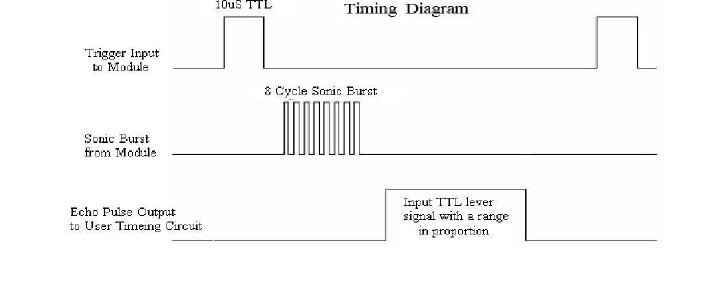
**ANALISIS Y RESULTADOS**

**PROCESAMIENTO DE IMÁGENES CON EL OSCILOSCOPIO**

1. En la siguiente imagen se puede observar la señal mostrada en el osciloscopio, luego de que el sensor no se encuentre obstruido, por ningún objeto a menos de 20cm. Esto quiere decir que a partir de un rango mayor a 20 centímetros(detectados por el sensor) el artefacto empieza a variar la visualización de la señal recibida.
2. En la siguiente imagen se observa la señal con un pulso casi nulo (o por lo menos fuera de las oscilaciones cuadradas mostradas anteriormente), debido a que se halla un objeto dentro de lo programado para realizar una nueva parte del código. Más precisamente la condicional del bucle.



Según las anteriores gráficas se puede concluir que se ve un poco de ruido en ambas señales, con respecto a la data sheet del sensor HC-SR04:

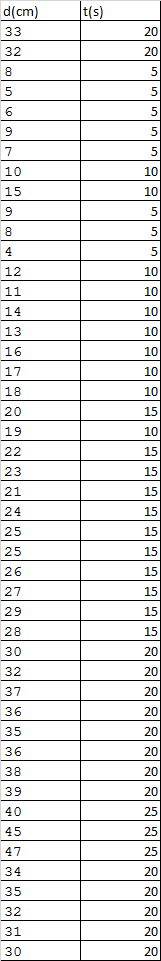


**PROCESAMIENTO DE DATOS OBTENIDOS**

Para el desarrollo del proyecto fue necesario realizar el sistema de ´pruebas que se menciona a continuación:

Se puso el carro a andar a lo largo de un recorrido con distintos obstáculos, lo que propiciaba que éste realizara el procesamiento, y posteriormente la aplicación del código.

Lo que refleja, luego de realizar su gráfica en Excel, una similar a la mostrada a continuación:



**CONCLUSIONES**

* Gracias al sensor ultrasónico es posible medir distancias con precisión por medio del rebote de las ondas sónicas.
* Por medio del sensor, es posible implementar el ultrasonido y el control de distintos aparatos a través de este.
* La realización de un proyecto con ultrasonido requiere esfuerzo, y sobre todo, una correcta calibración del mismo.
* Es indispensable realizar el código correcto, para lo que se quiere realizar, por medio de la compilación y ejecución del mismo.

**REFERENCIAS**

* Es.wikipedia.org. (2017). *Inspección por ultrasonido*. [online] Available at: https://es.wikipedia.org/wiki/Inspecci%C3%B3n\_por\_ultrasonido [Accessed 31 May 2017].
* Liceo Ártabro. (2017). Calibrado de un módulo de ultrasonidos HC. [online] Available at: https://liceoartabro.wordpress.com/2013/08/01/calibrado-de-un-modulo-de-ultrasonidos-hc/ [Accessed 31 May 2017].
* Barbus, E. (2017). *Tutorial: sensor ultrasonidos HC-SR04*. *Elcajondeardu.blogspot.com.co*. Retrieved 25 May 2017, from http://elcajondeardu.blogspot.com.co/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html
* *Medir distancia con Arduino y sensor de ultrasonidos HC-SR04*. (2017). *Luis Llamas*. Retrieved 25 May 2017, from https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/
* *Tutorial de Arduino y HC-SR04 – robologs*. (2017). *Robologs.net*. Retrieved 26 May 2017, from <https://robologs.net/2014/04/28/tutorial-de-arduino-y-hc-sr04/>

**ANEXO 1: CÓDIGO AKKA 3000**

/\*

Pontificia Universidad Javeriana

Daniel Andrés Pineda Suárez-

Proyecto Primer semestre\_senales

AKA3000

\*/

int Trig= 38;// Variable que los define pines del sensor de ultra sonido

int Echo= 36; // Variable que define los pines del sensor de ultra sonido

#include <AFMotor.h> // La siguiente librería describe el movimiento y control de los motores mediante

AF\_DCMotor right(1,MOTOR12\_64KHZ); // Frecuencia de PWM (para controlar la velocidad de giro del motor) 64kHz primer motor(recomendado por el módulo de Adafruiten su respectiva datasheet)

AF\_DCMotor left(2,MOTOR12\_64KHZ); // Hz segundo motor

void setup() {

Serial.begin(9600); //La mayoría de ordenadores emplean la variable como 9600, que es lo recomendado por Arduino

pinMode(Trig, OUTPUT); // Se define el Trig como pin de impulso o salida

pinMode(Echo, INPUT); // Se define el Echo como pin de recepción o entrada

right.setSpeed(250); // velocidad de ambos motores

left.setSpeed (250);

}

void loop() { //creo la función loop para la realización de un bucle

digitalWrite(Trig, LOW); // Trigger apagado

delayMicroseconds(2); //Se utilizó la librería de arduino de delay

digitalWrite(Trig, HIGH); // Se enciende trigger

delayMicroseconds(15); // Se envía un 1 durante 15 microsegundos

digitalWrite(Trig, LOW); // se define trigger como bajo

long tiempo, espacio; // Se crean 2 variables, para almacenar con tiempo en ms y espacio en cm

tiempo = pulseIn(Echo, HIGH); // Calcular la duracion del Pulso

espacio = (tiempo/2) / 29.1; // distancia a tiempo como sale en data sheet Test distance = (high level time×velocity of sound (340M/S) / 2,

Serial.println(espacio);

if (espacio < 20 ) { // Si la distancia es menor a 20cm girar a la izquierda right.run(FORWARD); // Avanza el motor de la derecha (donde BACKWARD es atrás, FORWARD es adelante)

{

left.run (BACKWARD);

delay (150);

right.run(BACKWARD);

delay (150);

}

left.run (BACKWARD); // Se cambia la polaridad del motor de la izquierda

delay (150); // Se pone el tiempo de giro

right. run(RELEASE);// Se detiene el motor

delay (0);

}

else {

delay (20); // Estabilizar los motores mediante un retraso o pausa de 20 milisegundos

right.run(FORWARD); //siga avanzando

left.run(FORWARD);

}

}

**ANEXO 2: CÓDIGO INDICADOR SONORO**

/\*Pontificia Universidad Javeriana

Indicador sonoro de objetos

Daniel Andrés Pineda Suárez\*/

#include "Ultrasonic.h" //Se incluye la librería del arduino para el HCSR04

Ultrasonic ultrasonic(7,6); //Se establece las variables: Impulso, recepción respectivamente

// Señal acustica

#include "pitches.h" //Se incluye las librerías que contiene cada tono

int timbre; //variable

int nota = NOTE\_E5; //Existen distintas notas para la librería de arduino, en mi caso elegí la E6 Tono o nota elegida para que suene

void setup()

{

Serial.begin(9600); //Establece la comunicación con el puerto serial

}

void loop()

{

timbre = ultrasonic.Ranging(CM);

if (timbre > 15) //Mientras que la distancia sea mayor a 15 cm la frecuencia del tono se intencifica

{

delay(timbre);

noTone(7); //Detiene la generación de ondas cuadradas por impulsos

tone(8,timbre); //Aviso

delay(250); //Duración del timbre

noTone(8); //Apagamos el sonido

}

timbre = ultrasonic.Ranging(CM); // se verifica la distancia para repetir el bucle

}