

Implementation of a distance detection system using active SONAR

Desarrollo de sistema de detección de distancia en agua mediante sonar activo.

Recibido: 12 de Julio de 2018 - Aceptado: 20 de Agosto de 2018

Belman Jahir Rodríguez ^{1*},

Felipe A. Vallejo¹,

Daniel Fernando Ramírez¹,

Miguel Ángel Olivares Tenorio¹



¹GRIAP-Grupo de Investigación en Acústica Aplicada, Ingeniería de Sonido, Universidad de San Buenaventura, Bogotá, Colombia.

brodriguez@usbog.edu.co

Abstract

We present a sonar system to measure the distance to targets for short-range objects, in real-time, in air and water by means of active SONAR and ultrasonic transducers. This device is to be used as a system of detection of aquatic species and underwater visualization of the sea bed in shallow waters. We implement ultrasonic transducers of different costs available in the market and test their performance in both air and water mediums. We present a detailed evaluation of the operation of these devices, some of their applications, the signals they can handle in terms of both frequency and amplitude, the processing of signals to measure the distance to targets. We also discuss how aspects such as salinity and temperature can affect the sound speed in the water and thus the performance of the device.

* Corresponding Author.
E-mail: brodriguez@usbog.edu.co

Keywords

Signal processing, SONAR, active sonar technologies, ultrasound.

Resumen

Se presenta un sistema tipo sonar activo para medir la distancia a objetivos a corto alcance en aire o agua usando transductores acústicos de ultrasonido. Se pretende desarrollar un sistema de detección de especies acuáticas y la visualización del suelo marino en aguas de poca profundidad. Se adaptan a este sistema transductores de ultrasonido de diferentes costos que existen en el mercado y se evalúa su desempeño al usar el sistema en agua y aire. Adicionalmente, se presenta una evaluación detallada del funcionamiento de estos dispositivos, algunas de sus aplicaciones, las señales que pueden manejar en cuanto a frecuencia y amplitud, el procesamiento de señales para la obtención de distancia cuando las ondas sonoras se propagadas en agua y como interfieren aspectos como la salinidad y la temperatura en este fenómeno de propagación sonora.

Palabras clave

Procesamiento de Señales, Sonar, Tecnologías de Sonares Activos, Ultrasonido.

1. Introduction

Many of the current technologies are inspired on behavior observed in nature. Ecolocalization is a form of navigation and identification of distances and objects most commonly used by animals such as bats, and marine species such as whales and other aquatic mammals. Since 1966 at the first Animal Sonar Systems convention, held in Frascaty, Italy, it was determined that there was a need for research on the design and development of technological systems with properties that approximate biological systems. Since then, there have been significant developments related to biological sonar systems and their applications in the engineering field [1].

SONAR technology, which stands for the words SOund NAVigation and Ranging, can be thought of as an acoustic version of RADAR. These SONAR and RADAR technologies are directly or indirectly related to governments and military agencies engaged in Anti-Submarine Warfare (ASW). Its operating principle is acoustic and consists of a transducer that emits ultrasonic waves (20 KHz - 500 KHz) that are incident to a target, and a receiver, which captures the reflected waves emitted by the target.

The use of SONAR is quite widespread in these days. Their applications include from electronic devices for vehicle parking, to specialized systems for fishing and determination of biomass in oceans [2].

SONAR systems are devices that are used in the processes of navigation, communication or detection of objects in the air or water. In general, these devices are divided according

to their construction in: active sonars, systems that emit and simultaneously detect sound waves and, passive sonars which can only detect sound disturbances [3], [4].

In this article, we present a sonar system to measure the distance to targets for short-range objects, in real-time, in air and water by means of active SONAR and ultrasonic transducers. For this end, we must consider the influence of different factors such as temperature, salinity and others that might affect the sonars performance. Salinity, temperature and ultrasonic transducers are integrated to a Raspberry Pi microprocessor [5], that will be implemented for signal processing and visualization in a graphical interface of distance versus time.

2. Processing acoustic signals in water

The processing of acoustic signals in water is usually classified into three main tasks: Detection, Estimation and Classification [3], [6]. In Detection, the main objective is to determine if the signal received by the detector has or not a desired signal, which can be reduced to a binary decision problem (the signal exists or does not exist). In the estimation problem, the usual task is to extract the numerical values of certain physical parameters that are specific to the detected signal, the source of this or some parameter of the medium. The most complex estimation problems involving both detector properties, the medium and the source are often referred to as Inverse Problems [3], [6], examples of which are the extraction of seabed properties (density, temperature, etc.) from the received signal or the characterization of a source or target, to this inverse problem is sometimes known as classification.

The proposed system performs the acoustic signals processing by means of the detection mechanism, in this case the objective is to determine if the signal is received or not, i.e. if it is reflected or not, to estimate a distance in case of existence reflections to the object to study. Initially, we show an analysis showing distinct commercial sensors for air distance detection to determine the ultrasonic signals generated, their characteristics (in terms of amplitude and frequency) and in this way to generate a proper system for detection of objects in water.

3. Detection of distance by ultrasound sensor in air

We implement to our device emitter/receiver pairs commonly used in active SONAR systems, such as HC-SR04 and JSN-B02 shown in Figure 1.



Figure 1. (Top left) HC-SR04 sensor, (top-right) ultrasonic transducer, (down) JSN-B02 sensor.

The sensor shown in the Top Left image in Figure 1, the HC-SR04, is a device that can be implemented in Arduino or Raspberry Pi. This device has independent emitter and receiver transducers that operate at an ultrasonic frequency of 40KHz and it is classified as a distance sensor. The JSN-B02 device is similar to the HC-SR04, it may be water resistant but does not work for calculating distances in acoustic environments. The JSN-B02 sensor implements a single transducer such as the one observed in the upper right image in Figure 1. This transducer fulfills the function of being simultaneously both the emitter and receiver at a sample rate 40KHz, therefore it is also considered as active SONAR. The transducer is commonly used as a proximity alarm system in some vehicles.

The output transducer is responsible for reproducing the high frequency acoustic signal (ultrasound) and the input transducer is in charge of receiving the existing reflections emitted by the output transducer in the propagation medium. This input transducer must have the physical characteristics to receive the high frequency signals.

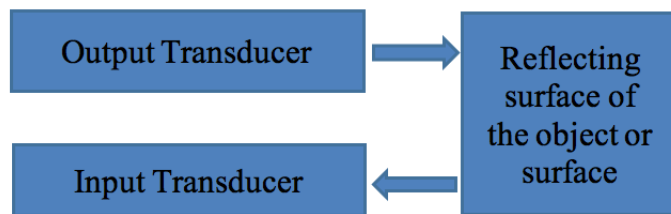


Figure 2. General detection scheme

The output transducer, or emitter, consists of a piezoelectric capsule designed to generate its greatest amplitude at a specific frequency, 40 KHz. The emitted signal propagates in the medium, in this case air and it is reflected when it reaches some surface. Part of the energy incident to this surface is absorbed and another part is reflected, this depends on the composition of the surface material, this property is related to the absorption coefficient of the surface material [4]. The reflected acoustic waves return to the input transducer, which is also a piezoelectric capsule, and transforms the received acoustic energy into a voltage signal of less amplitude than the signal sent by the output transducer.

The airborne devices HC-SR04 and JSN-B02, are similar in operation, but differ in the ultrasonic transducers they have. These devices can be integrated to an Arduino microcontroller or Raspberry Pi microprocessor, and are used for generating the activation of the ultrasonic signal and, register the sending and receiving time of the ultrasonic signals.

The HC-SR04 and JSN-B02 devices have four pins: Vcc, Gnd (ground), Echo (Input), Trig (Output). The Echo and Trigger pins are digital input and digital output respectively. The operation of these devices starts with the activation (+ 5v TTL) of a pulse, of 10µs of duration, sent from a microcontroller to the Trigger pin. This activation enables the device to generate a periodic burst of eight rectangular pulses of 40 KHz for the generation of ultrasonic acoustic signal as depicted in Figure 3. This burst of pulses is emitted periodically with a separation of approximately 60ms, a space for reception of reflections, therefore in each burst can be analysed a distance every 60ms.

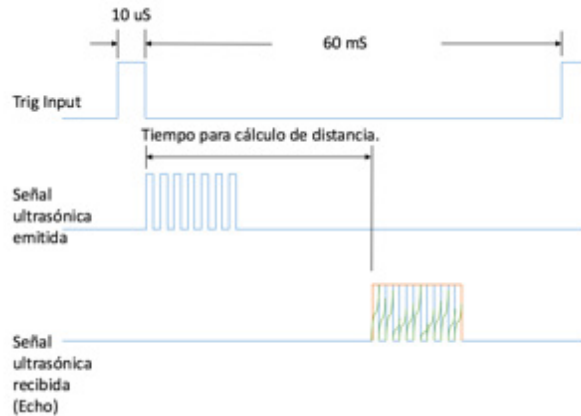


Figure 3. Time diagram showing the emitted and received ultrasonic pulses.

The acoustic signal propagates in the air and can be reflected or not. If it gets reflected it can return to the other ultrasonic transducer receiving the ultrasonic signal. The reception transducer converts the acoustic signal into a voltage signal, then the voltage signal passes through an envelope detector to demodulate the signal and finally, the signal passes through a voltage comparator whose function is to generate a logical data item "1" if the received signal is higher than the system noise (detection case) or "0" if there is no reflection better said, the signal is lower than the value defined in the comparator.

The comparator output is a TTL logic signal, is sent to an Arduino microcontroller or Raspberry Pi [5]. The microcontroller or microprocessor records the difference between the emission time (trigger on) and the transmission time (Echo on). Then the microcontroller determines the distance to the target as the time difference between the Trig output pulse and the Echo input pulse times the speed of sound divided by 2:

$$Dist = \frac{(T_{echo} - T_{trig}) * c}{2} \quad (1)$$

Here, T_{Trig} is the activation time of the generated ultrasonic pulse, T_{Echo} is the reception time of the reflected signal and c is the speed of sound. In equation (1) we divide by two because the pulse travels back and forth to the reflecting surface, therefore the distance is only measured between the device and the reflecting surface.

From equation (1) and the pulse time characteristics of the HC-SR04 and JSN-B02 systems, see Figure 3, a maximum available range can be computed as the time separation between contiguous emitted bursts, which we set to 60ms. Therefore, applying equation (1) and taking into account that the velocity of sound in air is 343.2 m/s for normal conditions of temperature (20 degrees centigrade) and humidity (50%). Then the maximum range achievable for an echo signal received 60 ms after the trigger is:

$$Dist = \frac{(0.06s - 0s) * 343.2 \text{ m/s}}{2} = 10.29 \text{ meters.} \quad (2)$$

Therefore, the distance meter depends on the separation of the burst pulses, the longer separation between pulses the greater the distance that can be recorded by the system.

We provide an example of an Arduino script that calculates the distance in air, for the HC-SR04 and JSN-B02 sensors [7]:

```

1 long tiempo;
2 int trigger_salida= 7;
3 int echo_entrada= 8;
4 float distancia;
5 void setup()
6 {
7   pinMode(trigger_salida, OUTPUT);
8   pinMode(echo_entrada, INPUT)
9   Serial.begin(9600);
10 }
11 void loop()
12 {
13   digitalWrite(trigger_salida, HIGH);
14   delayMicroseconds(10);
15   digitalWrite(trigger_salida, LOW);
16   tiempo = (pulseIn(echo_entrada, HIGH))/2;
17   distancia = float(tiempo * 0.343);
18   Serial.println(distancia);
19   delay(1000);
20 }

```

The first four lines define the variables to be used, which are input and output which are sent through pins 7 and 8 of the microcontroller. Lines 7 to 9 define that the variables “trigger” and “input” that are output and input, also defines the speed of the serial port to display the data on the screen by the serial monitor. In lines 13 to 15 a logic “1” signal (+ 5v) with a duration of 10 μ s is generated, it is used to activate the ultrasonic signal of the sensor. This signal is sent to the trig pin of the sensor, as can be visualized in the trigger input wavelet of Figure 3. Line 16 records the reflection’s return time. Line 17 calculates the distance as the given time in microseconds multiplied 0.343 corresponding to the speed of sound. Therefore, if the time it takes is 25 ms in and out of the ultrasonic signal, the resulting distance will be $((25000/2) * 0.343) = 4.28$ meters. Finally, on lines 18 and 19 the distance value is printed and a second is waited for the next data to be taken.

We have already explained a distance detection system by means of ultrasound that can operate in air, the next step is to implement ultrasonic transducers that operate in water. However, in this case other factors such as salinity and temperature also influence the propagation of acoustic waves in water, here is how to manipulate these variables for the calculation of the speed of sound in water.

4. Sound propagation in water

The velocity of sound in water is small compared to the velocity of electromagnetic waves in air by a factor of (2×10^5).

4.1 Sound velocity in water

The variations of the velocity of sound, c , in the ocean are relatively small. Generally, c is between 1450 and 1540 m/s. Nevertheless, these small changes affect the propagation of the sound in the water [8]. Different factors such as salinity, temperature, turbidity, depth and frequency of propagation of the sound wave, are properties that are influenced to each other, and influence the speed of the propagated wave.

Salinity and temperature are characteristics that determine the density of water in the oceans. The density of the water increases as the water gets colder and the more salt it contain, it is also affected by precipitation. All of this results in the stratification of the ocean which results in zones where the speed of sound vary.

Salinity is relevant in some climate change processes such as dense water formations in high latitudes or in atmospheric circulation in tropical latitudes. The salinity is commonly measured using either conductivity sensors, ultrasonic sensors, radiometric interferometry with satellites, or CTD (Conductivity, Temperature and Depth). [9]

The effect of factors such as salinity and temperature in water can cause the propagation speed of sound to change. Thus, it is necessary to keep track of changes in these variables in real time to maintain an adequate operation of the sonar system. To this end, we implemented a digital temperature sensor (DS18B20) and an analogue salinity or conductivity sensor (A1003v1) to keep track of these changes.

4.2 Conductivity sensor

Electrical conductivity is defined as the property of inorganic salts in solution (electrolytes) to conduct electric currents. Positive and negatively charged ions are the ones that conduct the current, and the amount driven will depend on the number of ions present and their mobility [11]. To measure the conductivity of a fluid there are three groups of sensors: inductive, capacitive and cells of two or more electrodes [10]. Therefore, salinity also influences the propagation of sound waves in water. If the water is denser the sound waves will travel at a higher speed.

An analogue salinity sensor, A1003v1, is implemented, consisting of three elements: the probe with the electrodes, a conversion element and a connection composed of three terminals: Gnd, Vcc and Out. This voltage output is of analog type and therefore must go to a digital analog converter to communicate with the Raspberry Pi microprocessor, which only has digital inputs in its GPIO (General Purpose Input Output) ports [10]. The salinity sensor measures the conductivity in a range of $0\text{-}5000\mu\text{S}/\text{cm}$, (3.2ppt at $17\text{ }^\circ\text{C}$). To calibrate the device we use the following conversion factor:

$$35 \text{ ppt} = 53 \text{ mSimen/cm} \quad (3)$$



Figure 4. Salinity sensor A1002v1

4.3 Temperature sensor

The DS18B20 is a device that obtains the water temperature and communicates it digitally to the microcontroller (Raspberry Pi). The sensor uses a communication protocol called OneWire [12] which uses a single cable to send and receive data; these are sent in groups of 8 bits.

This sensor has an operation range of $-50\text{ }^{\circ}\text{C}$ to $125\text{ }^{\circ}\text{C}$ and its interface has three pins: Vcc, Gnd and digital, which performs bidirectional OneWire communication from the sensor to a GPIO input pin of the Raspberry Pi. Note that it is necessary to use a pull-up resistor connected to the power supply. The connection diagram can be seen in Figure 6.



Figure 5. Temperature sensor DS18B20

The propagation of acoustic waves in water is given by equation (4):

$$c = 1449.2 + (4.6 * t) - (0.055 * t^2) + (0.00029 * t^3) + (1.34 - 0.01 * t) * (s - 35) + (0.016 * d) \quad (4)$$

Where t is temperature s is salinity in parts per thousand % and d is depth. Equation (4) is valid for temperatures between 0 and 35 degrees centigrade, salinity between 0 and 45‰ and depths between 0 and 1000 meters. The speed of sound increases with temperature, salinity and depth. [9]

Consider the case in which we deal with the sound propagation in fresh water, at a temperature of 20 degrees centigrade, without salinity and at a depth of 10 meters then c is approximately ~ 1481 m/s. This is value that we will be taken into account for tests in fresh water.

It is clear then that the temperature has a greater influence on the speed of sound than the salinity or the depth, but this factor loses importance as the depth increases. Taking into account the parameters that influence the propagation of sound in the water, to calculate the distance, the sensors that calculate these variables must be integrated into the microprocessor as shown in Figure 6.

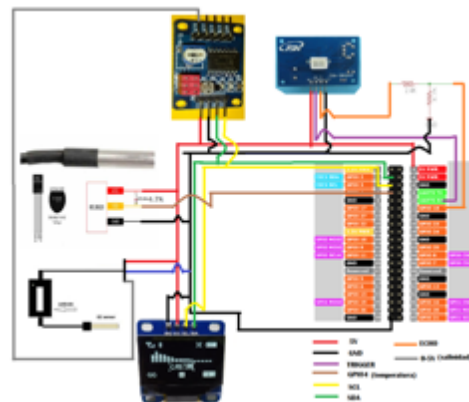


Figure 6. Sensor-microprocessor connection diagram Raspberry Pi

Finally, it is necessary to implement at this stage the aquatic ultrasound transducers for the complete system for calculating distance in aquatic environments.

4.4 Ultrasonic transducer for the water

It is necessary to implement ultrasonic transducers that work in aquatic environments, as we did with the HC-SR04 and JSN-B02 sensors implemented in air, to determine the distance to some reflecting surface in water, either an object or the aquatic surface. Two transducers are implemented as shown in Figure 7. According to the manufacturer, these transducers operate at an ultrasonic frequency of 200 KHz.

It is necessary to evaluate the performance of these transducers when operating either as an emitter or as a receptor. We do this by evaluating their frequency response by using a sweep frequency signal with the aid of a signal generator. Figure 7 shows the results of some of these tests and the resulting signals with the 200 KHz transducers. These emitter and receiver transducers are located to achieve the maximum emission / reception ratio which is achieved by placing them in the same plane as shown in Figure 7.

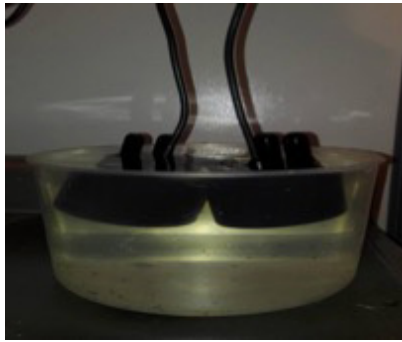


Figure 7. Signal flow water detection system

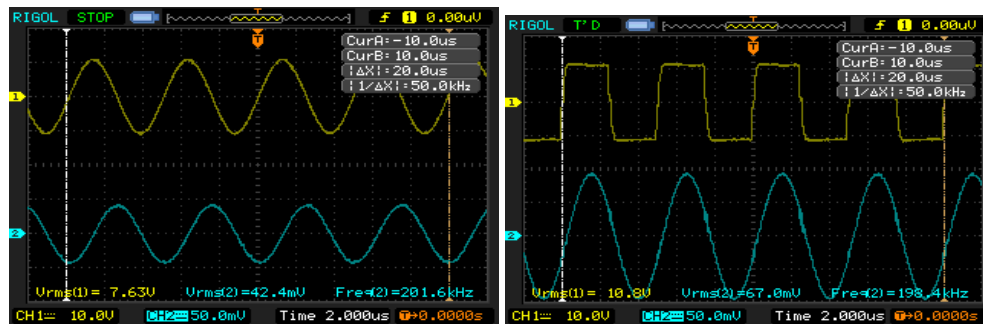


Figure 8. Yellow signal (Transmitter 8v, 200KHz), Blue signal (Receiver)

The images in Figure 8 show that a maximum reception is achieved when the input signal is tuned to 200 KHz and when the shape of the pulse is set to a square wave. For frequencies below and above 200 KHz, and for different wave shapes, the received signal shows lower amplitude than the input signal.

5. Implementation of detection system

After evaluating the operation characteristics of the ultrasonic, temperature, and humidity transducers; We continue by implementing a system that generates the ultrasonic signal and that processes the received signal. Our proposed system is shown in Figure 9.

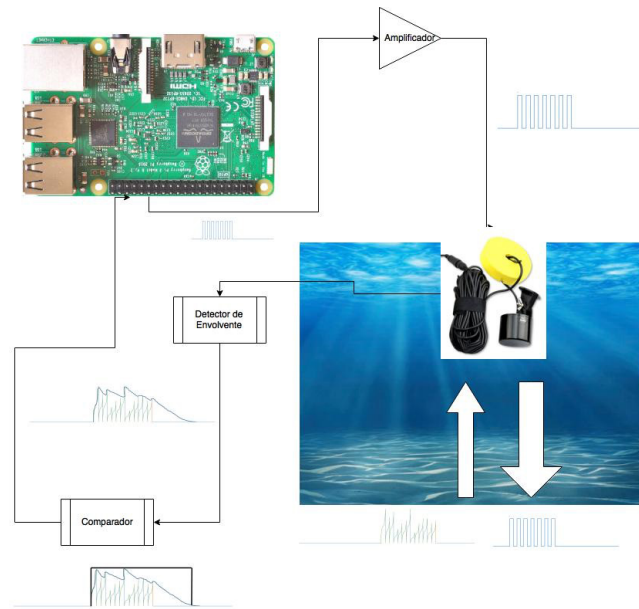


Figure 9. Assembly of emission and reception test

As observed in Figure 9, the first stage of the system consists of the generation of the ultrasonic signal, the amplification of this signal, and finally, the emission of the ultrasonic pulse by the ultrasonic emitter. During the second stage, the signal is propagated in water and scattered by a reflecting surface of the target. This reflection is different from the emitted signal having variations in amplitude and wave shape that depend on properties of the scatter and medium of propagation. The reflected pulse returns to the transducer which converts the acoustic signal into voltage signal. During this final stage, the received signal must be amplified again and passed through an envelope detector. The resulting signal is sent to a comparator that generates a binary signal (the signal is detected or not) then, the Raspberry Pi microprocessor records the sent and received times to compute the distance to the target.

The initial step is to generate an intermittent square signal with a carrier frequency of 200 KHz with the aid of the microcontroller. This signal is then sent to the HC-SR04 and JSN-B02 transducers.

5.1 200KHz intermittent square signal pulse generator

The microcontroller, either Arduino or Raspberry Pi, is programmed generate an intermittent signal in order to continuously detect the difference between the emitted pulses and received pulses, as described in Figure 3. In Figure 8 we show the pulse-train as recorded at the output pin of the microprocessor. To generate this signal, it was necessary to activate and deactivate a digital output at the microcontroller with a 50% duty cycle and duration $2.5\mu\text{s}$ in a logic one and $2.5\mu\text{s}$ in a logic 0. This signal must last long enough to excite the transducer and obtain a reflection, then it must be kept in logical 0 to wait for the received signal. The duration of the test pulse-train corresponds to 8 periods of ultrasonic signal, that is, $40\mu\text{s}$.

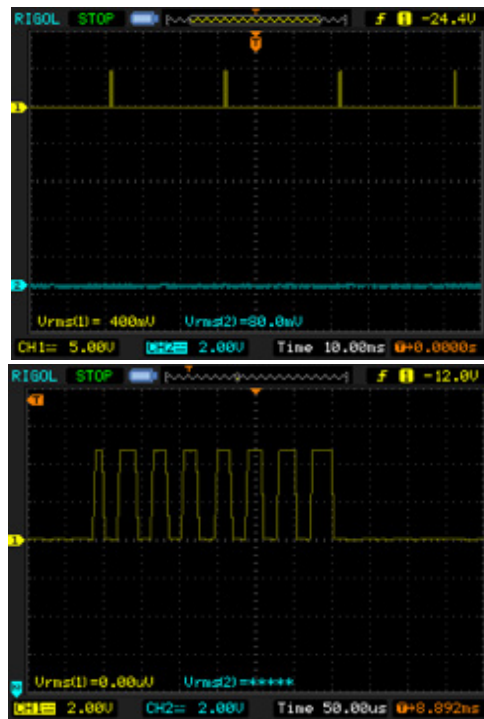


Figure 8. Pulse-train as recorded at the output pin of the microprocessor. The upper image shows the intermittent pulse bursts, the bottom part shows the ultrasonic signal burst recorded with a smaller time division.

The separation between the bursts of pulses is chosen to 30 ms and this value sets the maximum distance range that could be measured with our active sonar device in water. If the sound velocity in fresh water is approximately 1481 m/s and 1500 m/s in salt water and taking into account equations (1) and (4). With T_{Trig} set to 0 and T_{Echo} to 30ms, the maximum distance measured in fresh water for these conditions is 22.2m and the maximum distance measured in salt water for these conditions is 22.5 m.

If a greater distance is to be measured, the spacing between the emitted pulses must be greater so that the reflection may take longer time to reach the receiver. After this stage, the signal must be amplified to increase the emitted signal power so that the reflected pulse has enough energy to attain an adequate signal-to-noise ratio and to be able to properly detect it at the receiver.

5.2 Ultrasonic signal amplifier

The amplifying stage at the output signal of the microcontroller must be able to operate at the frequency range at which the transducer is driven, that is $\sim 200\text{KHz}$. For that reason, we chose the LF353 operational amplifier, since according to its specification sheet, works at ultrasonic frequencies.

5.3 Envelope detector

As depicted in Figure 3, the received pulses must pass through a demodulation stage to reduce the signal's bandwidth to be able to digitalize the signal's information with the sampling rate provide by our microprocessor. This is achieved with an envelope detector composed of a diode, a capacitor and a resistance, as shown in Figure 11.

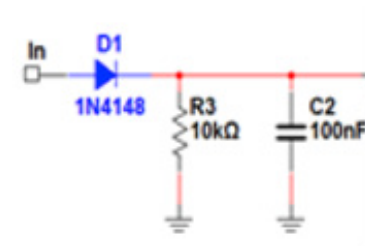
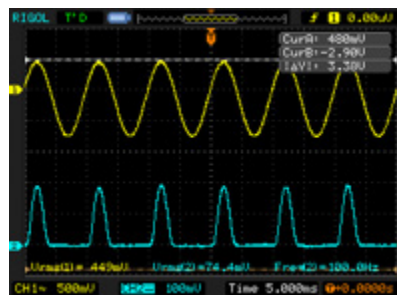


Figure 9. Envelope detector circuit

The diode allows the passage of the positive half-cycles, while the capacitor and the resistance maintain the voltage to detect the envelope of the received signal. This behavior is similar to the operation of an AM radio receiver, which implements an envelope detector as demodulator to extract the information transmitted in the AM signal. The calculation of the characteristic time of this circuit depends on the value of resistance and the capacitor, on the frequency of the received signal and its bandwidth according to the following expression.

$$\frac{1}{f} \ll RC \ll \frac{1}{W} \quad (5)$$

Here f is the frequency of the 200KHz transducer and W is the signal bandwidth, which in this case was approximately 10KH since the transducer's frequency response is constant between 195 KHz and 205 KHz. Using equation (5) we determined that resistance and capacitor values should be $R = 10K\Omega$, and $C = 100nF$. The operation of the envelope detector stage is then evaluated by inputting different signals at different frequencies to verify their behavior.



a.

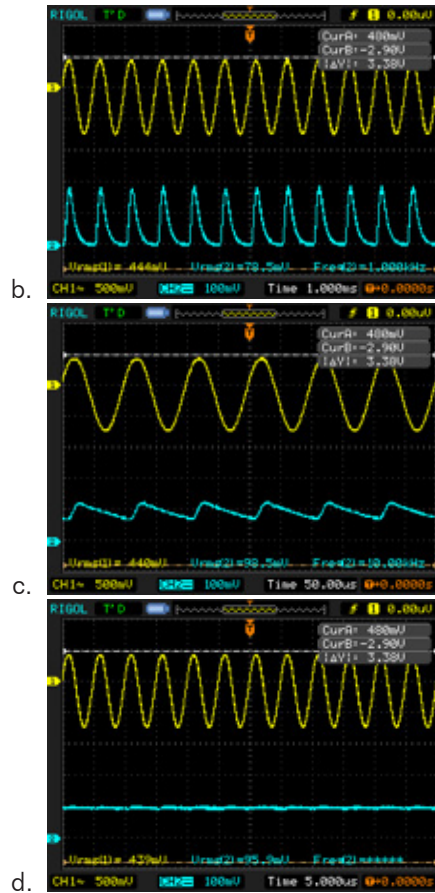


Figure 10. Envelope detector response, Yellow (Input signals), Blue (output signals). Response for a sinusoidal input signals centered at: (a) 100Hz, (b) 1KHz, (c) 10KHz, and (d) 200KHz.

Note that the received signal must have an amplitude greater than 0.7 V since the diode allows current passage for voltages higher than this value. If the received signal is lower, the diode will not rectify the incoming signal properly.

5.4 Comparator

After the envelope detector stage, a comparator whose function is to generate TTL logic values, signal detected or not, from the analogue voltages picked up by the ultrasonic receiver. The output of this digital signal is subsequently entered into the digital input ports of the microprocessor, and be it either an Arduino or Raspberry Pi.

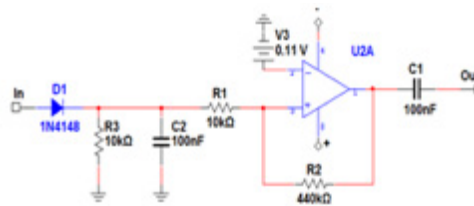


Figure 11. Comparator circuit.

The comparator is implemented with the circuit shown in Figure 13 where the aforementioned operational amplifier, the LF353, is used. In Figure 14 we show the performance tests of the comparator when excited a 200 KHz input signal. Note that at this stage, the input signal has a shorter frequency bandwidth due to demodulation occurring at the envelope detector stage.

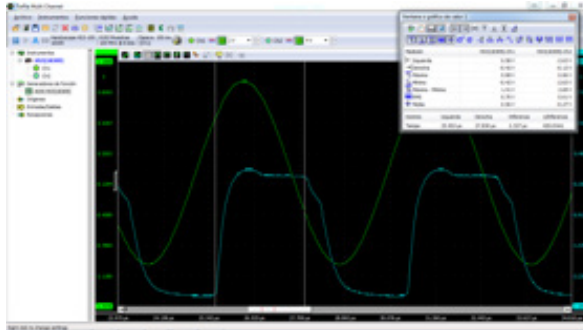


Figure 14. Comparator operation with 200KHz signal. Green (Signal Input), Cyan (Signal Output)

Since the output signal of the comparator has logic TTL values of 0 and 1, the signal at this stage is properly read through the digital ports of the microcontroller. Then the microcontroller software is programmed to record the time between two successive detections, and with this information determine the distance from the transducer to the reflecting surface of the target. It is important to point out that since for higher ultrasonic frequencies the transducer become more directional the target must be located, within a valid range, in front of the transducers.

5.5 Distance measurement with the proposed system

Having implemented successfully each of the stages of our system, we proceed to present the complete operation of our system. We created a system that displays the distance to the target vs time in quasi real-time. It is important to clarify that since the emitter and receiver transducers are identical and, since the transduction mechanism for these transducers is reciprocal, we used a single transducer as both emitter and receiver simultaneously.

The signal generated by the microprocessor is converted into an acoustic signal at the transducer, which at this point works as the emitter. The pulse propagates through the medium (water) until it encounters an obstacle or reflecting surface. At this point, part of the signal gets reflected directly back to the transducer, which now works as a receiver. The received signal is passed through the circuit in Figures 9 and 13, which adapts and converts the analog signal to a digital signal that can be processed by the microcontroller. The microcontroller then computes the distance to the target and transmits this information to the display by means of serial communication to display the distance data almost in real-time. In Figure 15 we show the signal captured at the output of transducer (green), the signal at the output envelope detector stage (blue, top) and the output signal of the comparator (blue, down). The latter corresponds to the signal that the microcontroller will use to that display the distance to the target vs time in quasi real-time.

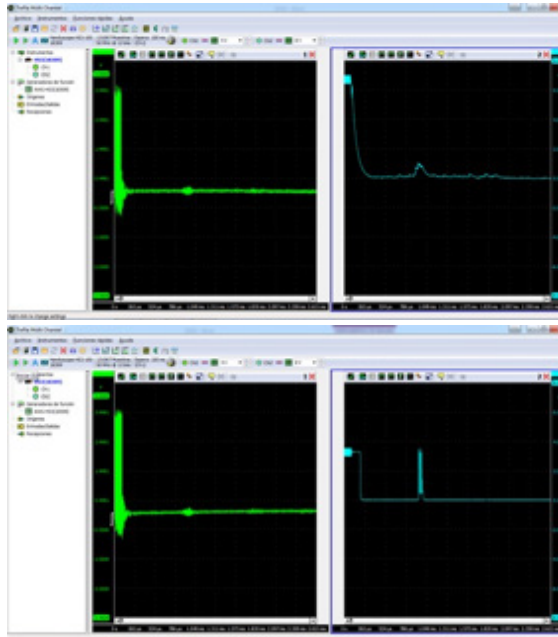


Figure 15. Signal captured at the output of transducer (green, top and down), the signal at the output envelope detector stage (blue, top) and the output signal of the comparator (blue, down).

In the Figure 15, the results displayed by the oscilloscope correspond to an approximate distance range of 60 cm. In Figure 16 we present a screenshot of the display of our system while operating in real-time. The figures show the *distance to the target vs time* graph generated by a Python script implemented in a Raspberry Pi 3 and using the JSN-B02 transducer, this script is similar to the one explained above for an Arduino microprocessor.

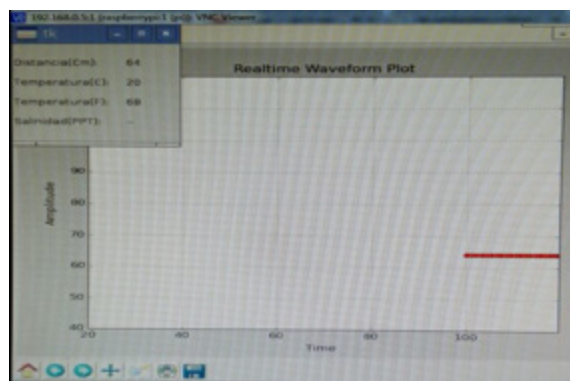


Figure 16. Screenshot of the display of our system while operating in real-time.

The software shows an approximate value to the estimated by the oscilloscope analysis, a temperature of 20 degrees centigrade and without salinity, this test is developed with fresh water.

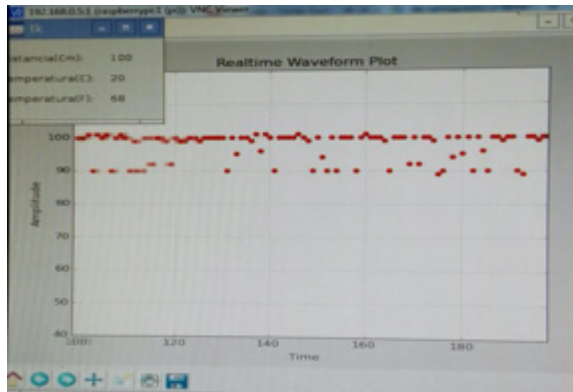


Figure 17. Screenshot of the display of our system while operating in real-time

In Figure 17 we show the graphical user interface of our device operating in real-time and when the target is placed in front of the transducer one meter apart. Note that varying the position of the target changes the distance shown at the display.

For our system, we used a Raspberry Pi microprocessor since this device offers a reduced size and, TTL GPIO digital input and output ports that allow digital communication (UART, I2C, SPI) to different sensors and Linux operating system. The scripts were written in Python which allows for an open source license and the Arduino microcontroller. An initial version of the software computes the distance directly into the Raspberry Pi, and another version captures the distance from Arduino and sends it by digital communication to the Raspberry Pi, which shows in a two-dimensional graph the time vs. distance, in real time.

The system is composed of two parts: sensors and processors. The implemented sensors are the digital temperature sensor ds18b20 integrated through the OneWire protocol, the analogue salinity sensor A1003v1 that needs an analog-digital converter. We use the AD/DA (PCF8591T), which has a resolution of 8 bits and communicates to the Raspberry Pi through the I2C protocol. Finally, we used an OLED screen to verify variables of salinity and temperature. For the emission and reception of ultrasound in water a transducer of 200 KHz was used.

6. Conclusions

We implemented a sonar system to measure the distance to targets for short-range objects, in real-time, in air and water by means of active SONAR and ultrasonic transducers. To this end, we implemented to the system several ultrasonic emitter and receiver transducers, a series of sensors to determine more accurately the speed of sound for varying conditions of the medium and, analogue and digital signal processing for optimal extraction of the distance information out of the emitted and received acoustic signals produced by our system.

The digital signal processing was done using two distinct platforms. We wrote Python scripts running on a Raspberry Pi to compute the distance to the target out of the acoustic

information sent to the processor's input GPIO ports. The distance information was also extracted using an Arduino script that calculates the distance to the target using the emission and reception times of the ultrasonic pulses. In either case, the information is sent to the Raspberry Pi which uses a OLED graphical user interface to display, in real time, the distance in water to a given target or reflecting surface.

The sonar system for measuring the distance to targets is also able to operate in air. It is only necessary to implement ultrasonic transducers that operate in air like mediums and, use the correct the speed of sound since in air it is approximately 344 m/s. Corrections in the speed of sound in air should also account for changes in temperature and humidity of the medium when implementing the visualization software.

Finally, we point out that in order to achieve higher distance-ranges of operation, it is necessary to implement high power transducers that produce high energy pulses to increase the signal-to-noise ratio of the acoustic signals. These transducers require high power supplies that must be adapted to the system. This would be necessary to implement this sonar systems in ocean like conditions.

7. References

- [1] Griffin, D. "Animal Sonar Systems" *Plenum Press*, New York, pp.11-14 (1979).
- [2] J. Burczynski. "Introducción al uso de sistemas sonar para la detección de la biomasa de peces." *Food & Agriculture Org*, pp.1-6 (1982).
- [3] Havelock, D. & Kuwano S, *Handbook of Signal Processing in Acoustics.*, Springer, New York, USA (2008)
- [4] L. Kinsler, *Fundamentals of acoustics*, John Wiley & Sons, Inc. New York, USA (2000)
- [5] Kurniawan, A. *Getting Started with Raspberry Pi 3*, PE Press. Berlin, Alemania (2016)
- [6] F. Vallejo, B. Rodríguez, M. Herrera, *Detección, estimación y problemas inversos con sistemas sonar*, Ingenium, vol.18, No 36, pp. 67-80, (2017)
- [7] Caruthers, J. *Fundamentals of marine acoustics*. Texas A&M University. Madison, USA (1973)
- [8] Brekhovskikh, L & Lysanov, Yu. *Fundamentals of Ocean Acoustics*. Springer Series, Berlin, Alemania (1991)
- [9] Posada, N. & Zuluaga, C. "Instrumentación para la medición de salinidad, temperatura y corrientes en océanos". *Revista Educación en Ingeniería*, pp 10, 17 (2011)
- [10] Rosón, V. "Humidificación isotérmica v/s humidificación adiabática". *Frío y Calor, Volúmen 87* (2006)
- [11] GPIO: Raspberry pi models A and B. Raspberry pi GPIO Online [Jun 2017]
- [12] Texas Instrument, 1-Wire Enumeration. Online [2017]