



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores Aragón

DISEÑO Y DESARROLLO DE UN MODEM ACÚSTICO PARA
TRANSMISIÓN DE DATOS EN CANALES SUBMARINOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERA EN COMPUTACIÓN

P R E S E N T A :

KAREN AÍDA FLORES CANO

ASESOR DE TESIS: M. EN I. ARCELIA BERNAL DÍAZ



FES Aragón

Ciudad Nezahualcóyotl, Estado de México, 2017



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

En la actualidad, el ser humano desempeña actividades de diversos tipos que comprenden desde militares, comerciales hasta académicas. Por lo que para realizar las mismas, el ser humano, debido a sus propios impedimentos físicos y biológicos para permanecer debajo del agua, ha requerido del diseño y desarrollo de dispositivos que le permitan realizar estas actividades de manera remota.

El presente trabajo describe el proceso seguido para el diseño y desarrollo de un módem acústico de bajo costo con capacidad de transmisión de datos bajo el agua para la comunicación y control de módulos sensores para el monitoreo de variables físicas, químicas y biológicas del océano. Específicamente este trabajo está dirigido a ser un sistema auxiliar en la toma de imágenes estereoscópicas¹ del fondo marino en sistemas arrecifales para el estudio de especies coralinas² e ícticas³.

El diseño y el desarrollo de módems de bajo costo para la transmisión de datos debajo del agua se ha vuelto un objetivo deseable e imprescindible no sólo para aplicaciones comerciales, sino también para aplicaciones académicas que se enfocan en el estudio de los ecosistemas marinos.

¹ Imágenes que por medio de una técnica especial crean la ilusión de profundidad para la visión binocular. (Diner & Fender, 1993)

² Los corales marinos son animales coloniales, salvo excepciones, pertenecientes al filo Cnidaria, clase Anthozoa. Las colonias están formadas por hasta miles de individuos zooides y pueden alcanzar grandes dimensiones. (Squires, 1959)

³ Especies de peces.

1. Comunicaciones Submarinas

Las actividades militares, económicas, académicas y recreativas de los humanos los han hecho desarrollar diversos medios, métodos y tecnologías para transmitir información entre sí más allá de las capacidades fisiológicas de estos. Además la extensión de las actividades de los seres humanos más allá del medio terrestre conocido (por ejemplo: el espacio o medios subacuáticos, transmisiones remotas) ha promovido aún más el desarrollo y la invención de nuevas tecnologías de comunicación y transmisión de datos.

1.1. Comunicaciones submarinas por medio biológicos

Para el caso de las comunicaciones subacuáticas, sabemos que primero apareció en los organismos vivos millones de años atrás. Y, actualmente es ampliamente conocida la capacidad de los mamíferos acuáticos para comunicarse a varios kilómetros de distancia en el medio marino para orientación en ambientes extremos de oscuridad total, ecolocación⁴ de presas y predadores además de su uso en el mantenimiento de relaciones etológicas (Deecke, Ford, & Slater, 2004) (Fordyce & Barnes, 1994) (Nummela, Thwissen, Bajpai, Hussain, & Kummar, 2007) (Nummela, Thwissen, Bajpai, Hussain, & Kummar, 2007)

El agua de mar es un excelente medio para la propagación del sonido por lo que le provee un excelente medio para que los mamíferos marinos puedan desarrollar diversas

⁴ **Ecolocación**, también llamado *biosonar*, es el sonar biológico utilizado por diferentes especies animales. Los animales que utilizan la ecolocación emiten sonidos y luego escuchan los ecos que rebotan al chocar con diversos objetos. (Holland, Waters, & Rayner, 2004)

habilidades con la finalidad de permitir la ecolocación y comunicación a grandes distancias. (Reynolds, 2013) (Aguilar Soto, y otros, 2011) Las ballenas dentadas han evolucionado para usar una gran gama de diferentes sonidos para la comunicación y la ecolocación. Las señales utilizadas en la comunicación pueden consistir en silbidos, en ráfagas de llamadas, o patrones de chasquidos. (Kyhn, y otros, 2013)

La ecolocación siempre se basa en chasquidos (Kyhn, y otros, 2013) de alta frecuencia que permiten detectar los ecos producidos por el sonido al rebotar en distintos objetos. (Reynolds, 2013) Los ecos de los chasquidos permiten calcular la distancia, tamaño, forma, velocidad y el vector en el que viaja su objetivo. Además de permitirle distinguir entre dos objetos distintos. (Payne, 1983) Los chasquidos pueden variar en sus características espectrales entre especies animales y hasta entre individuos de la misma especie. (Kyhn, y otros, 2013) La gran parte de las ballenas dentadas producen señales tonales para comunicarse y chasquidos para la ecolocación. Los chasquidos de la ecolocación son producidos neumáticamente por medio del paso de aire presurizado a través de los pasajes nasales ventrales hacia los labios fónicos. Cada vez que los labios fónicos se separan, una pequeña cantidad de aire escapa por estos hacia los sacos de aire vestibulares. (Reynolds, 2013)

Un sistema funcional de biosonar requiere de un clic de un origen a niveles más elevados que el objetivo. Para obtener este nivel elevado se debe primero tener una direccionalidad elevada. Y esta direccionalidad está determinada por la razón entre el tamaño del órgano transmisor y el ancho de onda del sonido proyectado, en

consecuencia, animales pequeños requieren de frecuencias altas para lograr la ecolocación con excepción de algunas especies. (Kyhn, y otros, 2013)

Las frecuencias menores son utilizadas para obtener ecolocación a grandes distancias, esto debe al hecho que las frecuencias bajas no viajan tan lejos como las frecuencias altas debajo del agua. Las frecuencias altas son más efectivas en distancias lejanas y pueden dar una información más detallada sobre los objetivos cercanos. (Reynolds, 2013)

La ecolocación es utilizada por algunos animales como un medio de reconocimiento de su ambiente. (Reynolds, 2013) En el caso de los mamíferos marinos o cetáceos, *Odontoceti* (ballenas dentadas) y *Mysticeti* (cetáceos barbados) tienen un sistema de producción de sonido que difiere grandemente del sistema de vocalización humana. Los odontocetes⁵ producen ráfagas de chasquidos de alta frecuencia que se creen son utilizadas para la ecolocación. Los órganos especializados de los odontocetes producen clic y zumbidos con frecuencias que varían de los 0.2 a los 150 kHz que les permiten obtener información sobre el ambiente en el cual se encuentra. (Payne, 1983)

A diferencia de la producción de los chasquidos, se sabe muy poco de la producción de los sonidos tonales⁶. Se cree que los sonidos tonales se generan a partir de la vibración de los labios fónicos. Jensen et al., demostraron que los factores ambientales afectan la

⁵ Los odontocetos (*Odontoceti*) son un suborden de cetáceos que son mamíferos acuáticos. Se les conoce comúnmente como cetáceos dentados. Estos mamíferos presentan un solo espiráculo (orificio respiratorio) en la parte superior de la cabeza y una frente abultada debido a la presencia del melón, órgano utilizado en la ecolocalización. (Cushier & King, 1933)

⁶ Sonidos tonales, son aquellos sonidos que se encuentran en íntima relación entre sí en función a la consonancia sonora. (Harvard, 2003)

comunicación de algunos mamíferos marinos, estos factores son la absorción, reverberación y ruido de fondo. Esto provoca que algunos mamíferos marinos tengan que tomar en cuenta las propiedades físicas de la transmisión del sonido del ambiente en que se encuentran para poder generar sonidos. (Jensen, Perez, Johnson, Aguilar Soto, & Madsen, 2011)

Hasta el momento, todos los registros obtenidos de ballenas dentadas han tenido evidencias de grabaciones de sonidos y pulsos ultrasónicos. Estas señales se pueden agrupar en 4 distintos tipos:

- Chasquidos de corta duración (10-50 microsegundos) en ciclos de 2-3 con un espectro entre los 20-150 kHz y con un origen menos a los 220 dB re 1 microPa rms.
- Señales de 10-15 ciclos centradas alrededor de 120-140 kHz con un nivel de origen de 195 dB re 1 microPa rms.
- Chasquidos de banda ancha de 2-3 ciclos alrededor de 15-40 kHz con niveles de origen muy elevados hasta 235 dB re 1 microPa rms y direccionalmente agudos.
- Chirridos de larga duración de 200 – 300 microsegundos en frecuencias de 25 a 50 kHz (Wahlberg, Beedholm, Heerfordt, & Møhl, 2011)

1.2. Comunicaciones submarinas por medio tecnológicos

Desde la perspectiva humana, entre los primeros intentos de enviar y recibir información bajo el agua de manera remota figuran los de Leonardo Da Vinci, quien utilizó un tubo largo sumergido en el agua para la detección de embarcaciones. Sin embargo, no fue

hasta la Segunda Guerra Mundial cuando el uso del submarino como arma táctica obligó a que las comunicaciones submarinas comenzaran a desarrollarse formalmente. En 1945, se desarrolló el teléfono subacuático como dispositivo de comunicación entre submarinos. Este dispositivo utilizaba una banda lateral única (SSB) de modulación de amplitud portadora en el rango de frecuencia de 8-11 kHz, y era capaz de enviar señales acústicas a distancias de varios kilómetros. El desarrollo de la tecnología VLSI (Very Large Scale Integration- Integración a Muy Gran Escala⁷) y procesamiento digital de señales (DSP – Digital Signal Processing⁸) con requerimientos de energía moderadas permitieron, por primera vez, el procesamiento de señales complejas y algoritmos de compresión de datos en los extremos sumergidos de un enlace para la transmisión de señales debajo del agua. (Stojanovic, 1999). Eso dio como resultado la transmisión de más datos en menor tiempo.

1.2.1. Observatorios cableados

Estos observatorios permiten la transmisión de datos por medio de tendido de líneas de fibra óptica o de cables eléctricos. Hasta el momento este cableado ha probado ser muy confiable y robusto en términos de transmisión de datos y capacidad para poder ser utilizados durante tiempos prolongados. Sin embargo, el tendido de cables a lo largo del

⁷La integración en escala muy grande de sistemas de circuitos basados en transistores en circuitos integrados comenzó en los años 1980, como parte de las tecnologías de semiconductores y comunicación que se estaban desarrollando. (Barbe, 1982)

⁸El procesamiento digital de señales o DSP (sigla en inglés de digital signal processing) es la manipulación matemática de una señal de información para modificarla o mejorarla en algún sentido. Este está caracterizado por la representación en el dominio del tiempo discreto, en el dominio frecuencia discreta, u otro dominio discreto de señales por medio de una secuencia de números o símbolos y el procesado de esas señales. (Proakis & Manolakis, Digital Signal Processing: Principles, algorithms and applications, 2007)

lecho marino es costoso en tiempo, recursos e infraestructura por lo que es impráctico para algunos lugares.

1.2.1. Satélites, Radiofrecuencia y Telefonía celular

Para eliminar el tendido de cables, en algunos casos se han sustituido estos mismos por la implementación de dispositivos radiofrecuencia, satélites y/o telefonía celular para poder transmitir datos. En algunos casos estos sistemas complementan el uso de observatorios cableados o, a veces, se colocan los transmisores en boyas que sobresalen del agua. La radiofrecuencia ha demostrado su utilidad durante la transmisión de grandes cantidades de datos. Sin embargo, es necesario que el transmisor y el receptor se encuentren visibles entre sí, por lo que no son de utilidad en lugar lejanos de la costa ni en grandes profundidades.

1.2.2. Amarres Cableados

En la actualidad, también se encuentran en desarrollo estos dispositivos los cuales puedan transmitir datos desde diferentes puntos en las profundidades marinas a una boya en la superficie por medio de un cable. Estos dispositivos tienen la ventaja de poder trabajar con energía solar, además de poder transmitir datos muy confiablemente en tiempo real, a través de cables mecánicos electro-ópticos (EOM- Electro Optical Mechanical). La desventaja de estos dispositivos es que este tipo de cable es muy costoso, además de ser costoso su mantenimiento e instalación.

1.2.3. Modem Acústico

Los módems acústicos son dispositivos capaces transmitir datos a través de ondas sonoras o acústicas sin necesidad de cables. La ventaja de estos dispositivos es que son de bajo costo, fácil implementación y manejo. Además de poder complementar otros sistemas de transmisión de datos. (Benson, Chang, Graham, & Kastner, 2006)

1.2.4. Características generales

Así como en la transmisión de señales aéreas, en las comunicaciones subacuáticas existen diferentes medios de transmisión. Uno de estos medios son las ondas de radio que pueden propagarse a cualquier distancia a través del agua de mar, pero solo en frecuencias bajas (30 Hz 300 Hz), por lo que se requiere de antenas grandes y altas potencias de transmisión para una transmisión exitosa. También se pueden utilizar ondas ópticas, las cuales no sufren tanto el efecto de la atenuación, sin embargo, estas se ven afectadas por la dispersión. Por lo tanto, la transmisión de señales ópticas requiere una alta precisión para apuntar los rayos de láser y evitar este efecto. A diferencia de las comunicaciones a través de canales de radio en las vías aéreas donde los datos son transmitidos por medio de ondas electromagnéticas, en el medio submarino se utilizan ondas acústicas principalmente. Esto se debe a que las características de la propagación de las ondas de radio debajo del agua difieren de las características de propagación de las ondas de radio en el aire, además de crear un medio de propagación complicado. El ancho de banda disponible de un canal subacuático es limitado ya que depende de la pérdida de transmisión y el ruido. La pérdida de absorción acústica aumenta con el rango y la frecuencia, provocando que se limite el ancho de banda. Por lo tanto, a medida que

aumenta el rango del sistema, el ancho de banda disminuye dramáticamente. Dentro de este ancho de banda limitado las señales acústicas también están sujetas a variaciones en el tiempo de múltiples canales. (Sozer, y otros, 1999).

Capítulo 2. Formas de Transmisión de Datos Submarina

2. Formas de Transmisión de datos

Al igual que en el medio terrestre los dispositivos para transmisión de datos por debajo del agua ocupan diferentes tipos de canales de transmisión como ondas de radiofrecuencia, ondas acústicas, ondas ópticas y electromagnéticas. Estos medios de transmisión no guiados se ven afectados de distintas maneras por el medio subacuático. Por lo que, al realizar al tomar la decisión sobre el dispositivo a ocupar es importante conocer estas características para poder obtener la funcionalidad deseada.

2.1. Radiofrecuencia

La radiocomunicación es una forma común de establecer comunicaciones a largas distancias a través de lo que se denomina radiofrecuencia. La transmisión y la recepción de las ondas de radio (o radiofrecuencia) se puede realizar con una antena, un dispositivo que transforma la variación de voltaje en ondas electromagnéticas. Las ondas de radiofrecuencia ocupadas dentro de la atmósfera terrestre abarcan desde los 3Hz hasta los 300 GHz. A las frecuencias comprendidas entre los 300 MHz y los 300 GHz se les denomina microondas, las cuales son muy utilizadas en la actualidad debido a que atraviesan fácilmente la atmósfera con menos interferencia que otras longitudes de onda.

Las ondas de radiofrecuencia en el medio submarino se pueden propagar a varios kilómetros a través del agua a frecuencias bajas (30 – 300 Hz) pero requieren de grandes antenas y un alto consumo de energía. Por lo que hace costosa su implementación y mantenimiento en proyectos de transmisión de datos en el medio subacuático. Además

de las ondas de radiofrecuencia también se puede utilizar las ondas electromagnéticas para la transmisión de datos. Sin embargo, las ondas electromagnéticas, a pesar de las propiedades conductivas del agua marina, han probado ser efectivas en aplicaciones de corto alcance utilizando frecuencias altas para la transmisión rápida de datos (Goh, Shaw, & Al_Shamma'a, 2009).

2.2. Telecomunicaciones Ópticas

La telecomunicación por el medio óptico utiliza la propagación de la luz (visible o infrarroja) en la atmósfera para la transmisión de datos entre dos puntos. Esta tecnología utiliza un diodo⁹ emisor de luz o un láser como fuente de transmisión. Por su lado el receptor capta el parte del espectro transmitido para decodificar la señal. Se ha utilizado este medio para realizar transmisión de datos no sólo en el medio terrestre sino también en el espacio. En el medio subacuático, las ondas ópticas no sufren mucha atenuación, pero son afectadas por la dispersión y requieren de una gran precisión en la alineación entre receptor y transmisor, siendo solo efectivas a distancias cortas. (Stojanovic, 1999)

2.3. Ondas Acústicas

Además de los anteriores medios de comunicaciones, también se puede utilizar ondas acústicas para la transmisión de datos. Las ondas acústicas son el medio natural por el cual los seres humanos y algunas especies de animales transmiten datos. Estas ondas son ondas mecánicas que se propagan a través del medio sólido, líquido o gaseoso hasta un receptor capaz de traducir este mensaje. Por lo que, hasta el momento, la transmisión

⁹ Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. (Rashid, 2004)

datos por medio de ondas acústicas ha sido el método más empleado (Stojanivic, 1999) debido a que el sonido se puede propagar a grandes distancias debajo del agua y se puede medir o estimar su rango de manera muy precisa como lo han demostrado los sistemas de posicionamiento submarino desde 1960 (Milne, 1983).

Capítulo 3. Fundamentos de la Acústica Submarina

3. Acústica Submarina

Al estudio de la propagación del sonido debajo del agua y la interacción de las ondas mecánicas que constituyen el sonido con el agua y sus fronteras se le llama *acústica submarina*. Las frecuencias típicas asociadas con la acústica submarina se encuentran entre los 10 Hz y 1 MHz. Debido a que la propagación del sonido a frecuencias menores a 10 Hz usualmente no es posible debido a que penetran en la parte profunda del lecho marino, por otro lado, las frecuencias mayores a 1 MHz son raramente utilizadas debido a que se absorben rápidamente en el medio.

3.1. Propagación de la Onda Sonora: Velocidad, Densidad e Impedancia

La propagación de una onda sonora debajo del agua consiste en una alternancia de compresiones¹⁰ y rarefacciones¹¹ del agua. Estas compresiones y rarefacciones son detectadas por un receptor, tal como el oído humano o un hidrófono, como cambios en la presión. La velocidad del sonido c está relacionada con la frecuencia f y la longitud de onda λ de una onda de acuerdo a la siguiente fórmula:

$$c = f * \lambda$$

¹⁰ Las compresiones se refieren a la propiedad de compresibilidad del agua que es la capacidad de una sustancia de reducir su volumen sin llegar a romperse como respuesta a presión atmosférica. (Rodríguez-Mellado & Marín-Galván, 1999)

¹¹ Se denomina rarefacción al proceso por el que un cuerpo o sustancia se hace menos denso. (González-Cabrera, 1971)

Esto difiere de la velocidad v del sonido que, en este caso, se refiere al movimiento de las moléculas en el medio debido al sonido y se relaciona con el plano de onda de presión p del fluido, densidad ρ y una velocidad del sonido de

$$p = cv\rho$$

El producto de c y ρ de la fórmula anterior se conoce como *impedancia acústica característica*. El poder acústico (energía por segundo), cruzando por una unidad de área es conocido como intensidad de la onda y en un plano de onda la intensidad está dada por $I = q^2 / (\rho c)$ donde, q es la raíz de la presión acústica.

El gran contraste de la impedancia entre el aire y el agua (una tasa de 3600) y la escala de la desigualdad de la superficie da como resultado que la superficie marina se comporte como un reflector casi perfecto del sonido a frecuencias menores a 1kHz. La velocidad del sonido en el agua excede a la velocidad del aire en un factor de 4.4 y en densidad con un factor de aproximadamente de 820.

3.2. Absorción del Sonido

La causa principal de la atenuación del sonido en agua dulce y en grandes frecuencias (arriba de 100 kHz) en agua salada es la viscosidad. A esto se le agrega la contribución en bajas frecuencias a la relajación del ácido bórico en frecuencias de hasta 10 kHz y de sulfato de magnesio en frecuencias de 10 kHz a 500 kHz. El sonido se absorbe en las fronteras del fluido. En el mar las pérdidas de sonido pueden ocurrir en la capa de burbujas o en las capas de hielo, mientras que en el fondo del océano el sonido puede penetrar el sedimento y ser absorbido. (Urick, 1983)

3.3. Desafíos Planteados por el Canal

Las características de trayectoria múltiple y sus variaciones en tiempo en múltiples canales dependen de la configuración del canal, que puede ser vertical u horizontal. La propagación multicamino horizontal se produce como resultado de las reflexiones del sonido en el fondo y en la superficie. A medida que la profundidad del agua disminuye, la propagación multicamino en los canales provoca que haya una grave interferencia entre símbolos (ISI – Intersymbol Interference¹²). Además de la ISI, las características del canal variable en tiempo causan propagación Doppler, que se puede escribir en términos de velocidad y frecuencia de funcionamiento como

$$D = \frac{vf}{c}$$

Donde v es la velocidad del receptor con respecto al transmisor, f es la frecuencia portadora, y c es la velocidad de propagación de la onda (3×10^8 m/s para las ondas libres de espacio de radio). Dado que las señales acústicas se propagan a 1500 m/s, los canales subacuáticos producen una mayor propagación Doppler que los canales de radio y hacen de la recuperación de la señal portadora una tarea difícil.

Hasta hace poco tiempo, debido a la severa propagación Doppler, se utilizaban técnicas de modulación no coherentes para las comunicaciones bajo el agua. Puesto que el canal varía rápidamente, es casi imposible hacer una estimación que pudiera ser válida para el siguiente símbolo. Cuando las tasas de símbolos aumentan también aumenta la

¹² Se refiere a la elongación y combinación de diferentes señales. (Langton, 2002)

interferencia entre símbolos lo que provoca que sea difícil tener una comunicación confiable. (Akyildz, Pompili, & Melodia, 2005). Los sistemas que operan a través de varias decenas de kilómetros pueden tener un ancho de banda de sólo unos pocos kHz, mientras que un sistema de corto alcance operando sobre varias decenas de metros puede tener más de un centenar kHz de ancho de banda. En ambos casos, estos factores conducen a la velocidad de bits baja, en el orden de decenas de kbit/s para los dispositivos existentes. (Sozer, y otros, 1999).

Los desafíos planteados en un canal debajo el agua son los siguientes:

- **Pérdida de la trama (Path Loss)**

Atenuación. Principalmente provocada por la absorción debido a la conversión de la energía acústica en calor. La atenuación aumenta con la distancia y la frecuencia. La atenuación también es causada por la dispersión y la reverberación (en la superficie desigual y en el fondo del océano), la refracción y dispersión (debido al desplazamiento del punto de reflexión en la superficie causado por el aire). La profundidad del agua juega un papel clave cuando se determina la atenuación.

Dispersión geométrica. Se refiere a la propagación de la energía del sonido como resultado de la expansión de los frentes de onda. Aumenta con la distancia de propagación y es independiente de la frecuencia. Hay dos tipos comunes de la dispersión geométrica:

- Esférica (omni-direccional al punto de origen), característica en comunicaciones en agua profunda.
- Cilíndrica (radiación horizontal solamente), característica en agua poco profundas.

• Ruido

Ruido producido por los humanos. Causado principalmente por ruido de maquinaria y actividad marítima, especialmente en zonas con tráfico pesado de embarcaciones grandes.

Ruido ambiental. Está relacionado con la hidrodinámica del mar y fenómenos sísmicos y biológicos.

• Multi-Camino

La propagación multicamino puede ser responsable de una grave degradación de la señal en la comunicación acústica, ya que genera interferencia intersímbolos (ISI). La geometría multicanal depende de la configuración del enlace. Los Canales verticales se caracterizan por una pequeña dispersión temporal, mientras que los canales horizontales pueden tener largas dispersiones multicanal.

La extensión de la propagación es una fuerte función de la profundidad y la distancia entre el transmisor y el receptor.

• Retraso de alta y la varianza de retardo

La velocidad de propagación en un canal es de cinco órdenes de magnitud menor que en un canal de radio. Este gran retardo en la propagación (0,67 s/km) puede reducir el rendimiento del sistema considerablemente.

La alta variación de letras es más perjudicial para la eficiencia del protocolo diseñado ya que evita la estimación precisa del tiempo de ida y vuelta (RTT – Round Trip Time), que es el parámetro clave para muchos protocolos de comunicación comunes.

• Propagación Doppler

La propagación de la frecuencia Doppler puede ser significativo en canales subacuáticos, ya que provoca una degradación en el rendimiento comunicaciones digitales: altas tasas de transmisión de datos causan interferencia de símbolos adyacentes en el receptor. Esto requiere que haya un sofisticado procesamiento de señales para hacer frente a la interferencia intersímbolos.

La propagación Doppler genera una frecuencia simple de traducción, que es relativamente fácil compensar para un receptor y una continua frecuencia de dispersión que constituyen una señal no desplazada, lo cual es más difícil de compensar.

Si un canal tiene una propagación Doppler con un ancho de banda B y una señal tiene una duración símbolo T , entonces hay aproximadamente BT muestras no correlacionadas de su envoltura compleja. Cuando BT es menor a la unidad, se dice que el canal está poco disperso y que los efectos Doppler de la atenuación

pueden ser ignorados mientras que, si es mayor que la unidad se dice que está sobre disperso.

La mayoría de los factores descritos son causados por las propiedades químico-físicas del agua (temperatura, salinidad, densidad) y por variaciones espacio-temporales. Estas variaciones, junto con la naturaleza de guía de onda del canal, hacen que el canal acústico altamente variable temporal y espacialmente. En particular, el canal horizontal es por mucho más rápidamente variable que el canal vertical, tanto en aguas profundas y no profundas. (Akyildiz, Pompili, & Melodia, 2005)

Los canales de comunicación en canales poco profundos, por lo general, son hostiles para las altas tasas de comunicaciones submarinas debido a dos obstáculos. Uno es la excesiva dispersión retardada multitrayecto que suele ser del orden de 10-30 ms y provoca la interferencia entre símbolos (ISI) sobre 20-300 símbolos a una velocidad de datos de 2-10 kilosímbolos por segundo. Otro obstáculo es el cambio Doppler variable en tiempo debido al movimiento relativo entre la fuente (transductores) y receptor (hidrófonos), el movimiento dinámico de la masa de agua y la velocidad variable de sonido, etc. Unos pocos Hz de Doppler en canales submarinos puede ser muy significativos debido a la baja velocidad de las ondas acústicas (~ 1500 m / s). La relación de Doppler para la frecuencia portadora en los canales bajo el agua es del orden de 10^{-3} a 10^{-4} , mientras que la relación en canales inalámbricos de RF está en el orden de 10^{-7} a 10^{-9} . El importante efecto Doppler provoca no sólo cambios rápidos en la respuesta del canal, sino también la compresión o dilatación de las formas de onda de la señal. Estos dos obstáculos que el receptor coherente de los sistemas de comunicación bajo el agua

mucho más complejos que los sistemas de radiofrecuencia. (Zheng, Xiao, Yang, & Yang, 2007)

Capítulo 4. Desarrollo de Modem Subacuático Inalámbrico

4. Desarrollo del Prototipo

Desarrollo de un prototipo para la transmisión de señales ultrasónicas y su interpretación por medio de un microcontrolador. Para este prototipo se utilizarán los siguientes componentes:

Hardware:

- Microcontrolador: ATMega 328
- Sensor Ultrasónico Acoplado: HC-SR04
- Receptor Ultrasónico: UCM-R40K1

Software:

- IDE Arduino
- Visual Studio 2010 con complemento para Arduino

4.1. Componentes de Hardware

4.1.1. ATMega 328

El microcontrolador ATMega 328 es parte de una plataforma libre de dispositivos electrónicos que tiene la finalidad de hacer más fácil el uso de hardware y software.

Además de lo anterior, esta plataforma proporciona un IDE¹³ de desarrollo y una serie de

¹³ IDE (Integrated Development Environment) que se refiere al entorno de desarrollo integrado por una serie de herramientas de software para realizar la programación, en este caso, del microcontrolador ATMega328.

librerías descargables desde el sitio de Arduino¹⁴. Este es un microcontrolador de 8-bits de baja potencia de tipo CMOS¹⁵ basado en la arquitectura AVR¹⁶ mejorada de tipo RISC¹⁷. (Atmel Coproration, 2010)

Una de las características más importantes en este proyecto es la capacidad y el manejo de manejar interrupciones del ATmega 328. Esta característica en particular representó una pieza fundamental para el desarrollo a nivel programación tanto de la señal como de la interpretación de la misma.

Una interrupción consiste en un mecanismo por el cual un evento interno o externo puede interrumpir la ejecución de un programa en cualquier momento. Esto produce un salto automático a una subrutina de atención a la interrupción, ésta atiende inmediatamente el evento y retoma luego la ejecución del programa exactamente donde estaba en el momento de ser interrumpido. Las interrupciones pueden ser:

- **Externas.** Generalmente están generadas por un dispositivo periférico.
- **Internas.** Excepciones dentro de un programa.

¹⁴ Sitio de Arduino www.arduino.cc

¹⁵ Complementary metal-oxide-semiconductor o CMOS es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS configurados de tal forma que, en estado de reposo, el consumo de energía es únicamente el debido a las corrientes parásitas. (Baker, 2010)

¹⁶Los AVR son una familia de microcontroladores RISC del fabricante estadounidense Atmel. La arquitectura de los AVR fue concebida por dos estudiantes en el Norwegian Institute of Technology, y posteriormente refinada y desarrollada en Atmel Norway, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip. (Atmel Coproration, 2010)

¹⁷ En arquitectura computacional, RISC (del inglés Reduced Instruction Set Computer, en español Computador con Conjunto de Instrucciones Reducidas) es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores.

Este microcontrolador también cuenta con la característica de poseer varios *Timers* o relojes (ver Tabla 4. 1. Timers del ATmega 328). Estos timers pueden programarse a diferentes frecuencias y trabajar de manera paralela para controlar los diversos eventos que ocurren dentro del microcontrolador.

NOMBRE	BITS	RANGO DE VALORES ¹⁸
TIMER 0	8	0- 255 Utilizado en funciones: delay() y millis()
TIMER 1	16	0- 65535
TIMER 2	8	0 255

Tabla 4. 1. Timers del ATmega 328

El TIMER 0 es el timer o contador principal del microcontrolador. Este contador corre a una frecuencia de 16MHz, aunque se puede modificar a través de la programación del dispositivo. Además de poseer dos unidades de comparación de parámetros de salida y capacidad para generar pulsos modulados.

4.1.2. HC-SR04 y Sensores Ultrasónicos UCM-R40K1

El HC-SR04 es un circuito integrado por los siguientes componentes:

- Receptor UCM-R40K1
- Transmisor UCM-R40K1

¹⁸ Timers disponibles en el dispositivo ATmega 328

- Driver MAX 232
- Amplificador Operacional 074C
- Un microcontrolador no especificado que al parecer es un PIC12C508

La principal funcionalidad de este circuito es medir distancias por medio del cálculo del tiempo en que tarda en regresar una onda ultrasónica emitida por el receptor.

Las características del receptor y transmisor se describen en la siguiente Tabla 4. 2. Descripción de las características operacionales del transmisor y del receptor UCM-R40K1:

CARACTERÍSTICAS OPERACIONALES DEL RECEPTOR Y TRANSMISOR UCM-R40K1	
CARACTERÍSTICAS	UCM-R40K1
Frecuencia de Resonancia	40 kHz
Nivel de Presión Sonora	115 dB<
Sensibilidad	-64 dB
Máxima Entrada de Voltaje	10 Vrms
Directividad Típica	55°

Tabla 4. 2. Descripción de las características operacionales del transmisor y del receptor UCM-R40K1

La generación de y lectura del ultrasonido se hace a través de dos unidades piezoeléctricas en donde una de ellas actúa como emisor y la otra como receptor. De ondas de presión ultrasónicas. El emisor debe excitarse con una señal adecuada en amplitud y frecuencia. En el caso de este dispositivo puede trabajar con frecuencias que

van desde 40 kHz hasta 80 kHz y de esta forma se realiza la emisión de una onda ultrasónica que puede ser captada por un receptor.

El circuito sin modificaciones trabaja al generar ráfagas de 40 kHz con duración de 5 ms cada 65 ms, lo cual produce una onda cuadrada en trenes de pulsos. Esta onda es detectada por el transmisor provocando una interrupción en uno de los timers del microcontrolador acoplado al circuito. Se puede saber cuánto tiempo toma en regresar la onda del emisor al receptor debido a que el reloj o timer del microcontrolador inicia a contar desde el momento en que inicia la transmisión de la onda de 40 kHz, hasta el momento en que la onda es detectada a través de receptor que es cuando se detiene el reloj por medio de una interrupción. El periodo de tiempo entre la transmisión y la recepción se puede utilizar para calcular la distancia de un objeto al sensor ultrasónico.

En el caso de este circuito, debido a que cuando la onda captada por el receptor provoca una salida mayor a 4 volts desde el circuito se toma como recepción de sonido cuando hay un voltaje mayor a 4 volts en la salida del circuito, como se muestra en Ilustración 4.

1. Funcionamiento del Circuito para el Sensor UCM-40K1).

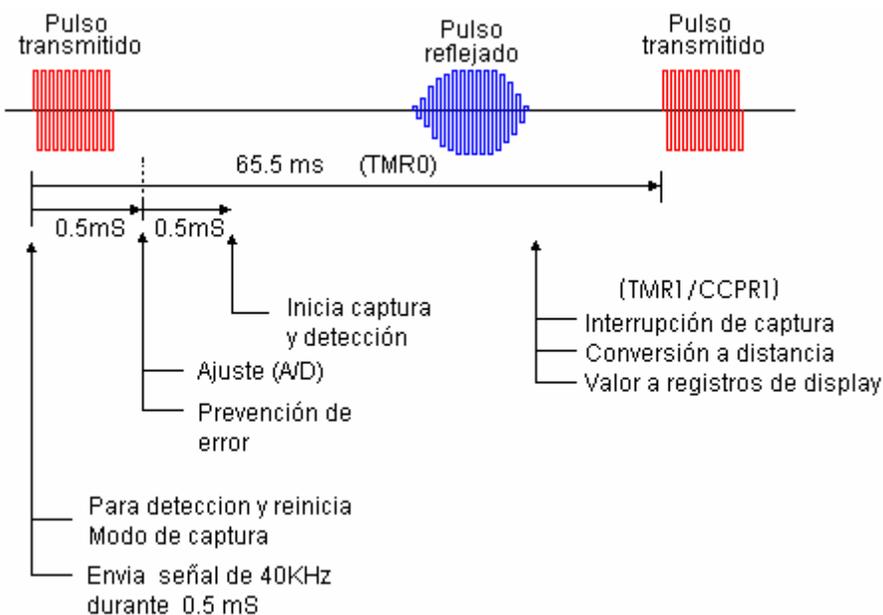


Ilustración 4. 1. Funcionamiento del Circuito para el Sensor UCM-40K1

4.2. Componentes de Software

4.2.1. IDE Arduino

Arduino constituye una plataforma de hardware libre, basada en una placa con un microcontrolador (incluye al ATM328) y un entorno de desarrollo diseñado para facilitar el uso de dispositivos electrónicos. (Arduino, 2013) El entorno de desarrollo contiene un editor de texto para escribir código, un área para mensajes, una consola de texto, una barra de herramientas con botones para las funciones más comunes y una serie de menús. Esta plataforma de desarrollo conecta al hardware del Arduino por medio de un cable USB de forma que se pueda programar el microcontrolador contenido en el hardware.

Esta plataforma de desarrollo está escrita en Java y se deriva de la plataforma de desarrollo de *Processing programming Language*¹⁹ y del proyecto *Wiring*²⁰. Los programas para esta plataforma se encuentran escritos en los lenguajes de programación C o C++. Los programas para esta plataforma se llaman **sketches**. Estos sketches se escriben dentro del editor de texto y son guardados por la plataforma con la extensión **ino**. (Arduino, 2013). La plataforma es capaz de trabajar con el paradigma de programación orientado a objetos, lo que hace más fácil la programación y la reutilización de código.

4.2.2. Microsoft Visual Studio 2010

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Este entorno de desarrollo soporta varios lenguajes de programación tales como *Visual Basic*²¹, *C++*²², *C#*²³, *J#*²⁴, así como, entornos de desarrollo web como *ASP*

¹⁹ Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab Dirigido por John Maeda. (Bohnacker, Gross, Laub, & Lazzeroni, 2012)

²⁰ Wiring es una plataforma libre de desarrollo para realizar prototipos electrónicos de un lenguaje de programación en un entorno de desarrollo integrado acoplado a una placa con un microcontrolador. Fue desarrollada por Hernando Barragán en 2003.

²¹ Visual Basic es un lenguaje de programación dirigido a eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico. (Randal & Romero Sweeney, 2006)

²² C++ es un lenguaje de programación diseñado a mediados de los años 80s por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. Por lo anterior, desde el punto de vista del paradigma de orientación a objetos este lenguaje es un lenguaje híbrido. (Alexandrescu, 2001)

²³ C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. (ECMA International, 2016)

²⁴ J# es un lenguaje de programación transicional para programadores del lenguaje Java y J++ de Microsoft. Este lenguaje fue creado con la intención que estos programadores puedan utilizar sus conocimientos anteriores. (Microsoft, 2016)

.NET²⁵ y otras plataformas de Microsoft. Una gran ventaja de utilizar este entorno de desarrollo es su capacidad de manejar el lenguaje C++ que es el lenguaje de programación para el microcontrolador ATM328, además de poseer una interfaz gráfica para poder visualizar el puerto serial y realizar interfaces gráficas. A lo anterior podemos agregar que existe un complemento desarrollado por *Codeplex*²⁶ el cual permite trabajar directamente con la plataforma de hardware de Arduino llamada *Arduino IDE for Visual Studio* (Microsoft, 2013).

4.3. Desarrollo de Circuito

4.3.1. Primera Etapa

En esta primera etapa, se desarrolló el anterior circuito de manera exitosa. Se pudo comprender de primera instancia como el HC-SR04 puede realizar la transmisión y recepción de señales ultrasónica desde un mismo circuito integrado. Además de poder hacer la conversión de señales analógicas a digitales para que el circuito y el programa embebido en el ATmega 328 sea capaz de interpretar los datos de manera correcta.

Para el caso del HC-SR04, se trató de realizar un ajuste en el programa en el ATmega 328 para que pudiera hacer una separación entre las funciones entre el transmisor y receptor y poder hacer cambios en el protocolo ya implementado en el circuito del HC-SR04. En este caso no fue exitosa la separación entre las funciones entre transmisor y receptor debido a que el circuito del HC-SR04 inicia un timer cada vez que inicia la

²⁵ ASP .NET es un framework de aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. (ECMA International, 2016)

²⁶ Codeplex es una comunidad y sitio web donde se alojan proyectos de software de código fuente abierto creado por Microsoft. <http://www.codeplex.com/> (Microsoft, 2013)

transmisión de la señal y no se puede hacer modificaciones al código contenido en el microprocesador del fabricante. El fabricante tampoco especifica el tipo de microprocesador utilizado ni se puede identificar de manera física. Al no poder hacerse una transmisión de datos de manera asincrónica, se decidió cambiar de dispositivo.

4.3.2. Segunda Etapa

Desarrollo de un prototipo para la transmisión y recepción de señales ultrasónicas y su interpretación a través de un microcontrolador con las siguientes características:

Hardware:

- Microcontrolador ATMega 328
- Transmisor: UCM-R40K1
- Receptor: UCM-R40K1

En esta etapa, se desarrolló el circuito que permita transmitir y recibir datos por medio de la modulación de ondas ultrasónicas, utilizando un microcontrolador para traducir cada uno de los signos a una frecuencia específica a un voltaje y otro microcontrolador para traducir los voltajes a símbolos entendible por un programa de software. Este circuito servirá como base para el acoplamiento de transmisores y receptores de más alto costo.

4.4. Desarrollo del Software

Se desarrolló un programa de software capaz de separar las funciones del transmisor y el receptor. Se tomó como base un programa diseñado para un microcontrolador 16F877, el cual permite la transmisión sincrónica de información, sin embargo, en nuestro caso se realizarán adaptaciones de este programa dividiéndolo en dos módulos: para transmisión

y recepción. Se tomará el diseño de una UART como modelo para transmisión y recepción de datos asíncronos. Para la modulación de datos, de manera inicial, se utilizará ASK (Amplitude Shifting Modulation – Modulación con Cambio de Amplitud) de la misma manera que lo utilizan los dispositivos HT12E y HT12D (ver Ilustración 4. 2. Medición del Tiempo de Transmisión para HT12E tomado de (Holtek Semiconductor Inc., 2000)).

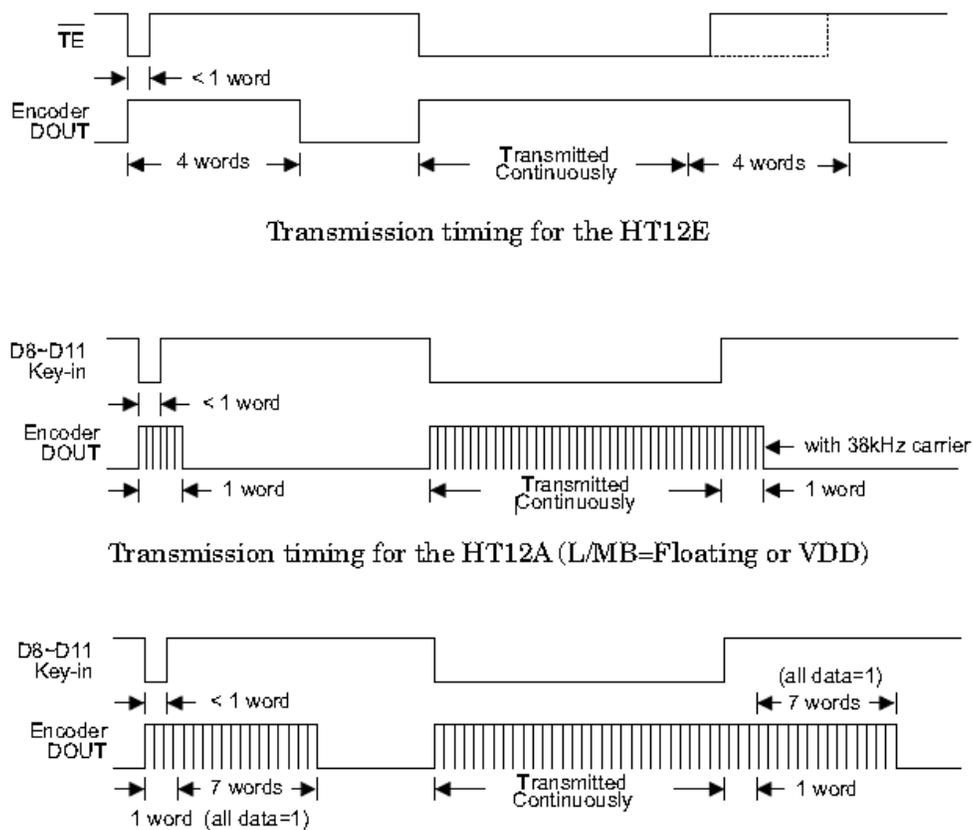


Ilustración 4. 2. Medición del Tiempo de Transmisión para HT12E tomado de (Holtek Semiconductor Inc., 2000)

4.4.1. Módulo de Transmisión

El programa para realizar la transmisión de símbolos por medio del microcontrolador ATmega 328 está basado en las interrupciones del microcontrolador. Estas interrupciones permiten que el microcontrolador pueda desempeñar diferentes funciones dependiendo de su configuración. Como ya se había mencionado anteriormente el ATmega 328 (ver. Sección 4.1.1) posee varios timers con diferentes características, en el caso de los Timers 0, 1 y 2, estos tienen la capacidad de funcionar como módulos PWM²⁷. La parte principal de un timer es su contador bidireccional. Este contador generará pulsos cada cierto tiempo de acuerdo a un reloj de pulsos interno o externo. Y, dependiendo del modo de operación utilizado, el contador es borrado, incrementado o disminuido cada cierto ciclo de reloj (clk_2) (Atmel Corporation, 2012). Sin embargo, para el este caso se utilizará el reloj interno del ATmega328 el cual transmite a una velocidad de 16 MHz, inicialmente.

Además del reloj interno, también es necesario utilizar otros registros de este microcontrolador como el Timer/Counter (TCNT2) y el Output Compare Register (OCR2A y OCR2B), los cuales son registros de 8-bits. El modo de operación de estos registros está definido por el modo Waveform Generation (WGM22:0) y el modo Compare Output (COM2x1:0). La forma más simple de operación para el WGM22:0 es la operación

²⁷ La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga. (Proakis & Manolakis, Digital Signal Processing: Principles, algorithms and applications, 2007)

normal. En este modo la dirección de conteo siempre se incrementa y no se borra al contador. El contador simplemente se desborda al llegar a un valor de 0xFF y luego reinicia en un valor 0x00. Lo anterior provoca que la bandera (TOV2) se prenda en el mismo ciclo de reloj en el que el TCNT2 se vuelve cero, el cual se fue incrementado en cada ciclo de reloj.

Sin embargo, para poder generar un pulso a una frecuencia de 40 kHz que es la frecuencia a la cual trabaja el transmisor ultrasónico, se tuvo que cambiar la funcionalidad normal del microcontrolador a un modo tipo Fast PWM. El fast Pulse Width Modulation o fast PWM se activa al escribir un 3 ó 7 y en el bit WGM22:0, lo que permite la generación de ondas de alta frecuencia. Este modo a diferencia del modo phase correct PWM, en el cual se generan ondas de dos pendientes, se generan ondas de una sola pendiente. El contador cuenta desde el valor más bajo hasta el valor más alto y vuelve a reiniciar en el valor más bajo. El valor alto está definido en un inicio como 0xFF cuando WGM2:0 = 3 y en el bit OCR2A cuando WGM2:0=7. En un modo de no inversión del bit Output Compare (OC2x) es borrado cuando son iguales los valores entre el TCNT2 y el OCR2x, además de encontrarse configurado para igualar el valor bajo. Debido a que es una sola pendiente, este modo de operación puede operar a una frecuencia dos veces más rápida que el modo Phase Correct PWM. Esta alta frecuencia lo hace ideal para aplicaciones donde se requiera regulación de potencia, rectificación y conversiones analógico-digitales.

En el modo fast PWM, el contador se incrementa hasta que el valor del contador coincide con el valor alto. Entonces, el contador es borrado en el siguiente ciclo de reloj. En este

caso el TCNT2 tiene el valor del contador y se compara con el valor del OCR2x, como se

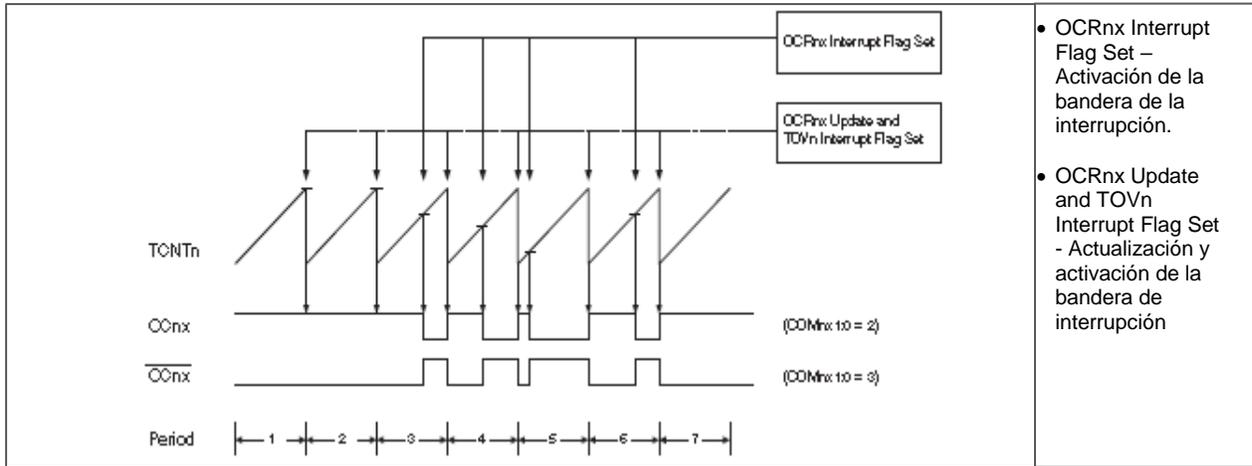


Ilustración 4. 3. Diagrama de Operación del modo Fast PWM

ve en la Ilustración 4. 3. Diagrama de Operación del modo Fast PWM. El bit Timer/Counter Overflow Flag (TOV2) se activa cada vez que el contador alcanza el valor alto. Si se activa la interrupción, se puede utilizar una rutina de interrupción (ISR) para actualizar o comparar el valor.

El modo fast PWM al final nos ayudará a generar ondas de 40 kHz las cuales pueden ser transmitidas por el transmisor ultrasónico. La finalidad de crear ondas, de acuerdo a una convención de símbolos transmitan 16 ciclos de ondas para la representación del símbolo 1 y 32 ciclos de ondas para la representación del símbolo 0. Esto se puede realizar por medio de la escritura de bits de 0 a 1 de acuerdo a la configuración necesaria. En este caso, se configuraron los siguientes parámetros: **TCCR2A – Timer/Counter Control**

Register A

Bit (0xB0)	7	6	5	4	3	2	1	0	
	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
VALOR INICIAL	0	0	0	0	0	0	0	0	

Ilustración 4. 4. Timer/Counter Control Register A (Atmel Corporation, 2012)

- **Bits 7:6 COM2A1:0: Compare Match Output Mode** Estos bits controlan el comportamiento del bit (OC2A). En la Tabla 4. 3. Modo Compare Output, del Modo Fast PWM, se muestra la forma en que se pueden configurar los diferentes comportamientos para el modo fast PWM cuando el WGM21:0 se encuentra activado.

COM2A1	COM2A0	DESCRIPCIÓN
0	0	Operación Normal del puerto
0	1	WGM22 = 0: Operación Normal del puerto, OC0A Desconectado WGM22 = 1: Invertir el valor de OC2A cuando coincidan los valores
1	0	Limpia OC2A cuando coinciden los valores, enciende el bit OC2A en el valor menor
1	1	Enciende el bit OC2A cuando hay coincidencias en la comparación, borra OC2A en valor bajo

Tabla 4. 3. Modo Compare Output, del Modo Fast PWM (Atmel Corporation, 2012)

- **Bits 5:4 – COM2B1:0: Modo Compare Match Output**

La funcionalidad de este modo depende de los parámetros del WGM22:0, lo que se explica en la Tabla 4. 4. Modo Output Compare, para el Modo Fast PWM.

COM2B1	COM2B0	DESCRIPCIÓN
0	0	Operación Normal del puerto, desconectado
0	1	Reservado
1	0	Limpia OC2B cuando coinciden los valores, enciende el bit OC2A en el valor menor

1	1	Enciende el bit OC2B cuando hay coincidencias en la comparación, borra OC2B en valor bajo
---	---	---

Tabla 4. 4. Modo Output Compare, para el Modo Fast PWM (Atmel Corporation, 2012)

- **Bits 3:2 – Reservado**
- **Bits 1:0 – WGM21:0 Modo Waveform Generation**

Combinado con el bit WGM22 encontrado en el registro TCCR2B, estos bits controlan las secuencias de conteo del contador, la fuente para el valor máximo (valor alto) del contador y el tipo de onda a utilizarse, se describen en la Tabla 4.

5.Descripción de los modos de Waveform Generation de acuerdo a su configuración de bits.

Modo	WGM22	WGM21	WGM20	Modo Operación Timer/Counter	TOP	Actualiza OCRx en	Encender bandera TOV
0	0	0	0	Normal	0xFF	Inmediato	MAX
1	0	0	1	PWM. Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Inmediato	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reservado	-	-	-
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM

6	1	1	0	Reservado	-	-	-
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Tabla 4. 5.Descripción de los modos de Waveform Generation de acuerdo a su configuración de bits (Atmel Corporation, 2012)

Nota: MAX = 0xFF; BOTTOM = 0X00

TCCR2B – Timer/Counter Control Register B

Bit (0xB1)	7	6	5	4	3	2	1	0	
	FOC2A-	FOC2B	-	-	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 6. TCCR2B- Timer/Counter Control Register B (Atmel Corporation, 2012)

- **Bit 7 – FOC2A: Force Output Compare A**

Este bit solo se implementa en ocasiones especiales y no en el modo PWM.

- **Bit 6 – FOC2B: Force Output Compare B**

Sólo activo cuando el modo Waveform Generation está en un modo no PWM.

- **Bit 5:4 – Reservados**

- **Bit 3 WGM22: Modo Waveform Generation**

Ver **¡Error! No se encuentra el origen de la referencia.** en la página **¡Error! Marcador no definido..**

- **Bit 2:0 – CS22:0: Selección de Tipo de Reloj**

Estos bits especifican el tipo de reloj a utilizarse de acuerdo a la configuración mostrada en Tabla 4. 7. Descripción de los modos de reloj de acuerdo a la configuración de bits.

CS22	CS21	CS20	DESCRIPCIÓN
0	0	0	Sin fuente de reloj (Timer/Counter detenido)
0	0	1	Clk _{T2S} /(Sin pre escala)
0	1	0	Clk _{T2S} /8 (pre escala)
0	1	1	Clk _{T2S} /32 (pre escala)
1	0	0	Clk _{T2S} /64 (pre escala)
1	0	1	Clk _{T2S} /128 (pre escala)
1	1	0	Clk _{T2S} /256 (pre escala)
1	1	1	Clk _{T2S} /1024 (pre escala)

Tabla 4. 7. Descripción de los modos de reloj de acuerdo a la configuración de bits (Atmel Corporation, 2012)

TIMSK2 – Timer/Counter 2 Interrupt Mask Register

Bit (0x70)	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R/W	R/W	R/W	TIMSK2
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 8. TIMSK2-Timer/Counter 2 Interrupt Mask Register (Atmel Corporation, 2012)

- **Bit 2 – OCIE2B: habilitar interrupción Timer/Counter 2 Output Compare Match B**

Cuando se escribe un uno el bit OCIE2B y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match B.

- **Bit 1 – OCIE2A: habilitar interrupción Timer/Counter2 Output Compare Match A**

Cuando se escribe un uno el bit OCIE2A y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match A.

- **Bit 0 – TOIE2: Habilitar interrupción Timer/Counter2 Overflow**

Cuando se escribe un uno el bit TOIE2 y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Timer/Counter 2 Overflow.

Debido a que se consideró que el modo Fast PWM era el más adecuado para la producción de ondas ultrasónicas se realizó la siguiente configuración en el registro TCCR2A. Se activó el bit COM2A0 para que en el evento de coincidencia se borrara el valor (ver, **¡Error! No se encuentra el origen de la referencia.**) y se pusieron los bits WGM21 y WGM20 en uno para activar el modo Fast PWM (ver, **¡Error! No se encuentra el origen de la referencia.**)

```
TCCR2A = _BV(COM2A0) | _BV(WGM21) | _BV(WGM20);
```

En el registro TCCR2B se realizó la configuración para el modo Fast PWM (ver **¡Error! No se encuentra el origen de la referencia.**) sin pre escala (ver **¡Error! No se encuentra el origen de la referencia.**).

```
TCCR2B = _BV(WGM22) | _BV(CS20);
```

En el registro TIMSK2 se pone el bit 2 en uno para habilitar la interrupción y de esa manera poder ejecutar una rutina de interrupción.

```
TIMSK2 = _BV(OCIE2A);
```

El valor de comparación de OCR2A se establece a 199, debido a que es necesario tener una frecuencia de 40 kHz para realizar la transmisión de los símbolos. Este cálculo se hace de acuerdo a la siguiente fórmula:

$$(a) \quad f_{OCRnx} = \frac{\frac{f_{clk_{11}}}{O}}{2 \cdot N \cdot (1 + OCRnx)}$$

donde, $\frac{f_{clk_{11}}}{O}$ es la frecuencia del reloj del microcontrolador, f_{OCRnx} es la frecuencia deseada,

N es el valor de la preescala y $OCRnx$ es el valor del registro de comparación $OCR2A$

Sustituyendo (a), con los valores que conocemos tenemos la siguiente fórmula:

$$(b) \quad 40kHz = \frac{16 \text{ MHz}}{2 \cdot 1 \cdot (1 + OCRnx)}$$

La fórmula sustituida en (b), nos da un valor de 199, por lo que se disparará la bandera de comparación (TOV2) cada 199 ciclos de reloj a 16 MHz. (Atmel Coproration, 2010)

`OCR2A = B11000111;`

El método para configuración del transductor es el siguiente:

```
void startTransducer()
{
    cli();
    TCCR2A = 0x00;
    TCCR2B = 0x00;
    TCCR2A = _BV(COM2A0) | _BV(WGM21) | _BV(WGM20);
    TCCR2B = _BV(WGM22) | _BV(CS20);
    TIMSK2 = _BV(OCIE2A);
    TIFR2 = 0;
    OCR2A = B11000111; // 199, so timer2 counts from 0 to 199 (200 cycles at 16 MHz)
    sei();
    ticks = 0;
}
```

4.4.2. Módulo de Traducción de símbolos

La configuración del 4.4.2.Módulo de Traducción de símbolos (ver, sección 0, página 31) ondas cuadradas cada 40 kHz, las cuales son enviadas como voltaje al sensor ultrasónico y este genera ondas ultrasónicas que pueden ser captadas por un receptor adecuado.

Además de este módulo de configuración también es necesario diseñar un esquema de transducción de signos recibidos por el puerto serial.

Debido a que la onda cuadrada tiene una duración se decidió utilizar la siguiente configuración mostrada en la Tabla 4. 9. Parámetros de Ciclos y Tiempo de Delay para Símbolos:

SÍMBOLO	CICLOS DE RELOJ	TIEMPO SIMBOLO	TIEMPO DE DELAY	TIEMPO TOTAL
0	16	400 μ s	800 μ s	1200 μ s
1	32	800 μ s	400 μ s	1200 μ s

Tabla 4. 9. Parámetros de Ciclos y Tiempo de Delay para Símbolos

Se utilizó un tiempo de Delay o espera entre símbolos para que se pudiera generar una onda distintiva y se pudiera diferenciar adecuadamente los símbolos enviados desde el transductor. La función del transductor se inicia al recibir símbolos desde el puerto serial. Después cada uno de estos símbolos es traducido a un número entero que corresponde a su valor en la tabla ASCII²⁸. Luego se traduce este número entero a su contraparte binaria la cual puede ser interpretada por el programa y de esta forma determinar los ciclos de reloj que se van a transmitir.

²⁸ ASCII (acrónimo inglés de American Standard Code for Information Interchange — Código Estándar Estadounidense para el Intercambio de Información), pronunciado generalmente [áski] o [ásci] , es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales. Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía. Más tarde, en 1967, se incluyeron las minúsculas, y se redefinieron algunos códigos de control para formar el código conocido como US-ASCII.

Una parte muy importante en este módulo es el inicio y la terminación de los ciclos de transmisión de las ondas para los símbolos y el tiempo que existe entre cada uno de los símbolos. Para poder a cabo esta funcionalidad se utilizaron dos timers del ATmega 328 (ver sección 4.1.1 página 21): Timer 0 y Timer 2. El Timer 0 es utilizado por este microcontrolador como contador o reloj principal para llevar a cabo la función de *delay*²⁹ y *millis*³⁰. Por lo que este contador fue utilizado para contar el tiempo de espera entre cada uno de los símbolos. El Timer 2 es un contador con la capacidad de trabajar en modo Fast PWM y de disparar interrupciones (ver. Sección 0.4.4.1. Módulo de Transmisión página 31), estas dos características pueden ser utilizadas para generar la señal adecuada para la transmisión de símbolos.

Por ejemplo, para el caso del símbolo 0:

- El puerto serial recibe un carácter alfanumérico
- El carácter es traducido a su representación en código ASCII
- Luego, la representación ASCII es traducida a su representación en un byte de 8 bits
- Se inicia el transductor y se evalúa bit por bit su representación simbólica
- En el caso del símbolo 0, el programa corre durante 15 ciclos del reloj
- Cada ciclo de reloj incrementa un contador llamado tics dentro de la rutina de interrupción del programa

²⁹ *delay()* función predeterminada del ATmega328 en el IDE Arduino la cual pausa el programa en un periodo de tiempo determinado en milisegundos. (Arduino, 2013)

³⁰ *millis()* función predeterminada del ATmega328 en el IDE Arduino la cual regresa el número de milisegundos que transcurren en el programa. (Arduino, 2013)

- Al identificar el ciclo 16 o al tener tics el valor de 16, se detiene el transductor.

Lo anterior se encuentra en el siguiente programa:

```
#include <avr/interrupt.h>

#include <avr/io.h>

#define LEDPIN 13
#define TRIGGERPIN 11
#define LONGSIMBOLO 8

enum Error{
    exito = 0,
    errlecturaSigno = 1,
    errtransmSigno = 2
};

volatile unsigned long int ticks = 0;
unsigned int intChar;
unsigned int simbolo;
String inputString = ""; // a string to hold incoming data
boolean stringComplete = false;
int tipoError = exito;

ISR(TIMER2_COMPA_vect){
    tipoError = exito;
    switch(simbolo)
    {
        case 0:
            if(ticks >16)
            {
                stopTransducer();
            }
            break;
        case 1:
            if(ticks >32)
            {
                stopTransducer();
            }
            break;
        default:
            tipoError = errtransmSigno;
            stopTransducer();
            break;
    }
    ticks += 1;
}

void startTransducer()
{
```

```

cli();
TCCR2A = 0x00;
TCCR2B = 0x00;
TCCR2A = _BV(COM2A0) | _BV(WGM21) | _BV(WGM20);
TCCR2B = _BV(WGM22) | _BV(CS20);
TIMSK2 = _BV(OCIE2A);
TIFR2 = 0;
OCR2A = B11000111; // 199, so timer2 counts from 0 to 199 (200 cycles at 16 MHz)
sei();
ticks = 0;
}

void stopTransducer()
{
  TCCR2A = 0;
  TCCR2B = 0; timer.
}

void traduceSimbolo(){
  startTransducer();

}

void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    intChar = (int)inChar;
    intChar= 1;
    conversionSimbolos();
    inputString += inChar;
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}

void conversionSimbolos(){
  int arrSimbolo[LONGSIMBOLO];
  tipoError = exito;
  startTransducer();
  for(int i = 0; i < LONGSIMBOLO; i++)
  {
    arrSimbolo[i] = bitRead(intChar, i);
    simbolo = arrSimbolo[i];
    Serial.println(simbolo);
    startTransducer();
    switch(simbolo)
    {
      case 0: delayMicroseconds(800);
              break;
      case 1: delayMicroseconds(400);
              break;
      default: tipoError = errlecturaSigno;
    }
  }
}

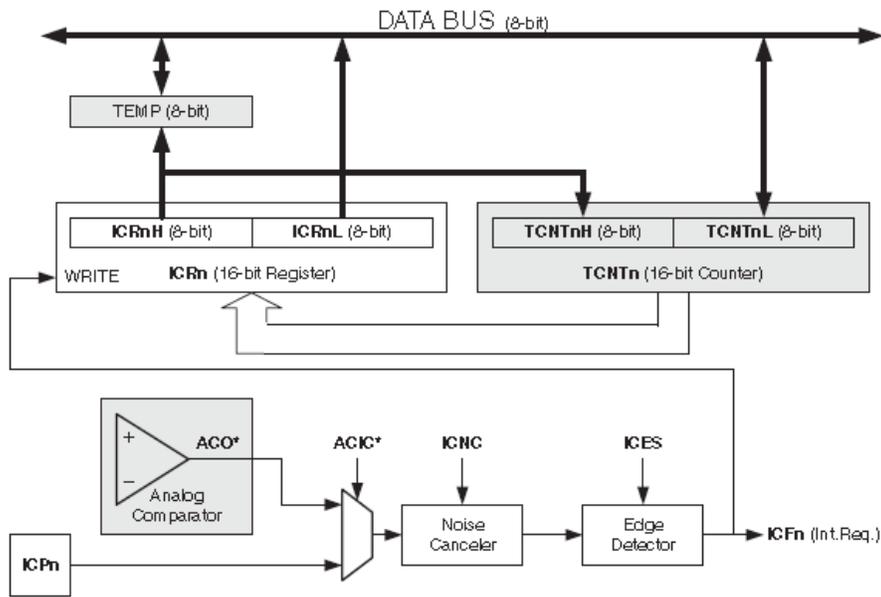
```

```
        break;
    }
}
void setup()
{
    Serial.begin(115200);
    inputString.reserve(2);
    pinMode(LEDPIN, OUTPUT);
    pinMode(TRIGGERPIN, OUTPUT);
    //startTransducer();
}

void loop()
{
    digitalWrite(LEDPIN, !digitalRead(LEDPIN));
}
```

4.4.3. Módulo de Recepción

El módulo de recepción se encarga de recibir la señal desde el receptor ultrasónico y medir el tiempo de duración de la señal en alto, en otras palabras, cuando el receptor se encuentra mandando al microcontrolador voltajes aproximadamente de 5 volts. La ausencia de un voltaje de 5 volts o un voltaje menor a 5 volts será considerada como ausencia de señal. Al detectarse la señal se contará el tiempo que la misma permanece en alto. El microcontrolador dentro de su contador o timer incorpora una unidad de captura de entradas (*Input Capture Unit*) la cual puede capturar eventos externos y asignarles una marca de tiempo para calcular el periodo total en el cual se desarrolló el evento. La señal externa del evento puede ser aplicada al pin ICP1 (*Input Capture Unit*) a través de la unidad comparadora analógica. El periodo de tiempo total obtenido a través de esta unidad puede ser utilizado para calcular la frecuencia, el ciclo de trabajo y otras características de la señal. En la Ilustración 4. 5. Diagrama de Bloques Input Capture Unit se esquematiza el procedimiento de captura de los eventos externos por el microcontrolador.



Al haber un cambio de voltaje, hay un cambio a nivel lógico en el ICP1 y alternativamente en el ACO. Este cambio se confirma al detectar el extremo que puede ser alto o bajo. Cuando se dispara la captura el valor del contador

Ilustración 4. 5. Diagrama de Bloques Input Capture Unit (Atmel Corporation, 2012)

La captura de los eventos se puede llevar a cabo con un microcontrolador a través de la funcionalidad de captura de fuentes de disparo, en inglés *Input Capture Trigger Source*.

TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 10. TCCR1A-Timer/Counter1 Control Register A (Atmel Corporation, 2012)

- Bits 7:6 COM1A1:0: Compare Match para el Canal A
- Bits 5:4 – COM1B1:0: Modo Compare para el Canal B

La funcionalidad de este modo depende de los parámetros COM1A1:0 y COM1B1:0, lo que se explica en la Tabla 4. 11. Modo Comparación de Salida no PWM.

COM1B1	COM1B0	DESCRIPCIÓN
0	0	Operación Normal del puerto, desconectado
0	1	Invertir valores OC1A/OC1B al coincidir lo valores
1	0	Limpia OC1A/OC1B cuando coinciden los valores en el valor menor
1	1	Limpia OC1A/OC1B cuando coinciden los valores en el valor alto

Tabla 4. 11. Modo Comparación de Salida no PWM (Atmel Corporation, 2012)

- **Bits 3:2 – Reservado**
- **Bits 1:0 – WGM21:0 Modo Waveform Generation**

Combinado con los bits WGM13:2 encontrado en el registro TCCR1B, estos bits controlan las secuencias de conteo del contador, la fuente para el valor máximo (valor alto) del contador y el tipo de onda a utilizarse, se describen en la Tabla 4. 11. Modo Comparación de Salida no PWM.

Modo	WGM12	WGM11	WGM10	Modo Operación Timer/Counter	TOP	Actualiza OCRx en	Encender bandera TOV
0	0	0	0	Normal	0xFF	Inmediato	MAX
1	0	0	1	PWM. Phase Correct, 8 bit	0xFF	TOP	BOTTOM
2	0	0	0	PWM. Phase Correct, 9 bit	0xFF	TOP	BOTTOM
3	0	0	1	PWM. Phase Correct, 10 bit	0xFF	TOP	BOTTOM
4	0	1	0	CTC	OCR1A	Inmediato	MAX
5	0	1	1	Fast PWM, 8 bit	0xFF	BOTTOM	TOP
6	0	1	0	Fast PWM, 9 bit	0xFF	BOTTOM	TOP
7	0	1	1	Fast PWM, 10 bit	0xFF	BOTTOM	TOP
8	1	0	0	PWM, Phase Correct, Frequency Correct	ICR1	BOTTOM	BOTTOM

9	1	0	0	1	PWM, Phase Correct, Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Inmediato	MAX
13	1	1	0	1	Reservado	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Tabla 4. 12. Descripción de los modos de Waveform Generation de acuerdo a su configuración de bits para el registro TCCR1A (Atmel Corporation, 2012)

Nota: MAX = 0xFF; BOTTOM = 0X00

TCCR1B – Timer/Counter Control Register B

Bit (0xB1)	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 13. TCCR1B - Timer/Counter Control Register B (Atmel Corporation, 2012)

- **Bit 7 – ICNC1: Input Capture Noise Canceller**

Al escribir uno en este bit la entrada del pin (ICP1) es filtrado. La señal requiere de cuatro muestras sucesivas para poder cambiar su valor.

- **Bit 6 – ICES1: Input Capture Edge Select**

Este bit selecciona cual extremo capturar en el pin ICP1 que es usado para disparar el evento de captura. Cuando se escribe cero en ICES1, el extremo negativo o en caída de la onda desencadena la captura. Por otro lado, cuando se escribe uno en ICES1, el extremo positivo o en subida desencadenará el evento de captura. Cuando se inicia la captura, el valor es copiado en el registro Input Capture (ICR1). Además de activar la bandera Input Capture (ICF1) y esto puede permitir que se habilite la interrupción.

- **Bit 5 – Reservado**
- **Bit 4:3 WGM13: Modo Waveform Generation**

Ver **¡Error! No se encuentra el origen de la referencia.** en la página **¡Error! Marcador no definido..**

- **Bit 2:0 – CS12:0: Selección de Tipo de Reloj**

Estos bits especifican el tipo de reloj a utilizarse de acuerdo a la configuración mostrada en Tabla 4. 14. Descripción de los modos de reloj de acuerdo a la configuración de bits para el registro TCCR1A .

CS12	CS11	CS10	DESCRIPCIÓN
0	0	0	Sin fuente de reloj (Timer/Counter detenido)
0	0	1	Clk _{T2S} /(Sin pre escala)
0	1	0	Clk _{T2S} /8 (pre escala)
0	1	1	Clk _{T2S} /32 (pre escala)
1	0	0	Clk _{T2S} /64 (pre escala)
1	0	1	Clk _{T2S} /128 (pre escala)
1	1	0	Clk _{T2S} /256 (pre escala)
1	1	1	Clk _{T2S} /1024 (pre escala)

Tabla 4. 14. Descripción de los modos de reloj de acuerdo a la configuración de bits para el registro TCCR1A (Atmel Corporation, 2012)

TIMSK1 – Timer/Counter 1 Interrupt Mask Register

Bit (0x70)	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R/W	R/W	R/W	TIMSK1
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 15. TIMSK1- Timer/Counter 1 Interrupt Mask Register (Atmel Corporation, 2012)

- **Bit 7,6 – Reservado**
- **Bit 5 – ICIE1: Timer/Counter1, Habilitar la interrupción Input Capture**

Cuando se escribe uno en este bit, la bandera del registro de estado se habilita (se habilitan las interrupciones globales), la interrupción Timer/Counter1 Capture está habilitada.

- **Bit 4,3 - Reservada**
- **Bit 2 – OCIE1B: habilitar interrupción Timer/Counter 2 Output Compare Match B**

Cuando se escribe un uno el bit OCIE1B y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match B.

- **Bit 1 – OCIE1A: habilitar interrupción Timer/Counter2 Output Compare Match A**

Cuando se escribe un uno el bit OCIE1A y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match A.

- **Bit 0 – TOIE1: Habilitar interrupción Timer/Counter2 Overflow**

Cuando se escribe un uno el bit TOIE1 y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Timer/Counter 2 Overflow.

TIMSK1 – Timer/Counter 1 Interrupt Mask Register

Bit (0x70)	7	6	5	4	3	2	1	0	
	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
VALOR INICIAL	0	0	0	0	0	0	0	0	

Tabla 4. 16. TIMSK1 - Timer/Counter 1 Interrupt Mask Register (Atmel Corporation, 2012)

- **Bit 7, 6 – Reservados**
- **Bit 5 – ICIE1: Timer/Counter 1, habilitar la interrupción Input Capture**

Cuando se escribe este bit en uno y la bandera del registro de estatus está habilitada se habilita la interrupción Timer/Counter 1 Input Capture. Por lo que se ejecuta la interrupción correspondiente.

- **Bit 4,3 - Reservados**
- **Bit 2 – OCIE1B: habilitar interrupción Timer/Counter 2 Output Compare Match B**

Cuando se escribe un uno el bit OCIE1B y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match B.

- **Bit 1 – OCIE1A: habilitar interrupción Timer/Counter2 output Compare Match A**

Cuando se escribe un uno el bit OCIE1A y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Compare Match A.

- **Bit 0 – TOIE1: Habilitar interrupción Timer/Counter2 Overflow**

Cuando se escribe un uno el bit TOIE1 y el estatus del I-bit en el registro de Estatus está puesto en uno, se habilita la interrupción Timer/Counter 1 Overflow.

(Atmel Coproration, 2010)

Para poder llevar a cabo la funcionalidad de captura se hicieron las siguientes configuraciones, de acuerdo a lo visto en la **¡Error! No se encuentra el origen de la referencia.** (ver página **¡Error! Marcador no definido.**). Para el registro TCCR1A,

se configuró el registro en modo normal y se desconectaron las salidas como se muestra en el siguiente código.

```
TCCR1A &= ~( _BV(WGM11) | _BV(WGM10) )  
  
TCCR1A &= ~( _BV(COM1A0) | _BV(COM1A1) | _BV(COM1B0) | _BV(COM1B1))
```

En el registro TCCR1B, también se configura en modo normal y se deshabilita el filtrado de ruido de acuerdo con la **¡Error! No se encuentra el origen de la referencia.** y la descripción de la funcionalidad de ICNC1 (ver página **¡Error! Marcador no definido.**). Al escribir el bit ICES1, se configura la captura del tiempo de entrada en el extremo positivo. También se habilita la pre escala de 1/8 por medio de la escritura de CS11 en uno y CS12:10 en cero.

```
TCCR1B &= ~( _BV(WGM12) | _BV(WGM13) | _BV(ICNC1))  
TCCR1B |= _BV(ICES1); //capture raising edge  
TCCR1B |= _BV(CS11)  
TCCR1B &= ~( _BV(CS12) | _BV(CS10) );
```

Finalmente, se habilitan las interrupciones globales por medio de la escritura del bit ICIE1 a uno.

```
TIMSK1 |= _BV(ICIE1);
```

De acuerdo a lo que ya habíamos establecido en 0 (ver página 39) se había determinado el número de ondas y de tiempo de delay que cada uno de los símbolos iba a utilizar. Por lo tanto podemos transformar la Tabla 4. 17. Tiempo en alto y su significado simbólico de la siguiente manera:

TIEMPO CAPTADO	SÍMBOLO
----------------	---------

400 $\mu\text{s} \pm 10\%$	0
800 $\mu\text{s} \pm 10\%$	1

Tabla 4. 17. Tiempo en alto y su significado simbólico

Como ya se había mencionado, en cada uno de los casos hay un delay que permite tener un total de 1200 μs para la transmisión total de un símbolo. Por lo que para determinar qué símbolo es el que se recibe se compararán las lecturas de tiempo con esta tabla de símbolos. Si el valor del periodo en alto (valor guardado por la unidad ICR1) es de 400 $\mu\text{s} \pm 10\%$, el valor será cero. Por otro lado, si el valor del periodo en alto (valor guardado por la unidad ICR1) es de 800 $\mu\text{s} \pm 10\%$, el valor será uno. Cualquier valor por abajo o por encima de los valores establecidos será considerados como erróneos por el programa y se desechará.

Por medio de operaciones de bits se irá agregando cada uno de los símbolos a una variable.

Lo anterior se muestra en el siguiente código:

```
// ISR(TIMER1_CAPT_vect)
//Función que lee el resultado captado por el timer, sí hay un cambio en el ICES
//entonces se lee el tiempo en que el valor se mantiene en alto
ISR(TIMER1_CAPT_vect){
    TCNT1 = 0;
    if(bitRead(TCCR1B, ICES1) == true)
        resultado = ICR1;
    TCCR1B ^= _BV(ICES1);
}

//setup()
//Función para iniciar el pin como entrada del Puerto serial
void setup(){
    pinMode(echoPin, INPUT);
    setup_timer1();
    Serial.begin(115200);
}
```

```

//convBinLetra
//Función que realiza los arreglos de 8 bits que componen un símbolo de un byte
void convBinLetra(){
    for(int i = 0; i < longCaracter; i++)
    {
        int num = byteSimbolo[i];
        caracter |= num << i;
    }
}

//evaluaFrec()
//Función que evalúa el valor de la frecuencia captada y determina que símbolo
//es de acuerdo a un rango de valores. Se descartan todos aquellos valores que
//están fuera de rango como ruido
void evaluaFrec(){
    long int tempResultado;
    int simbolo
    if( tempResultado >= 13333 && tempResultado <= 11764)
        simbolo = 0;
    if( tempResultado >=22222 && tempResultado <= 28571)
        simbolo = 1;
    if( simbolo != 0 && simbolo != 1)
    {
        simbolo = 3;
    }
    else{
        if(indice < longCaracter)
        {
            arrSimbolo[indice] = simbolo;
            indice += 1;
        }
        else
        {
            for(int i = 0; i < longCaracter; i++)
            {
                byteSimbolo[i] = arrSimbolo[i];
                byteCompleto = true;
            }
        }
    }
}

//loop()
//Función que se ejecuta para mostrar en una salida el resultado de las lecturas
void loop(){
    Serial.println(resultado);
    evaluaFrec();
    if(byteCompleto){
        convBinLetra();
        if ((char)caracter == '\n')
        {
            Serial.println(cadenaSalida);
        }
    }
    else
    {

```

```
    cadenaSalida += (char)caracter;  
}
```

5.1. Arquitectura del Proyecto

Al terminar el proyecto se logró construir un circuito pequeño que utilizaba una fuente de 9 volts. En la Ilustración 5. 1. Arquitectura Módem Acústico, se muestran todos los componentes operacionales del módem acústico. El proyecto se encuentra integrado por dos módulos independientes: un módulo de transmisión y un módulo de recepción.

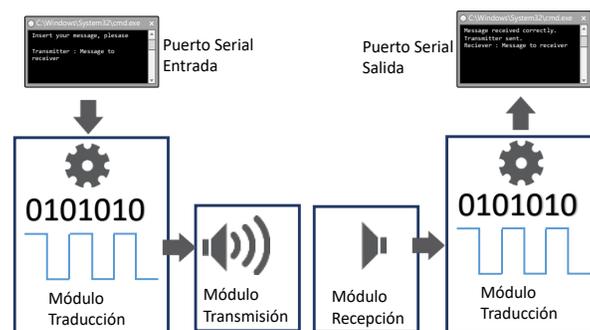


Ilustración 5. 1. Arquitectura Módem Acústico

Como se observa en esta figura las entradas del puerto serial son recibidas y procesadas por el software para ser transformadas en señales que puedan ser interpretadas por el hardware, tanto a la transmisión como a la recepción de la señal.

Al final el microcontrolador el ATmega 328 se integró junto con otros componentes para contener los módulos de transmisión y recepción de la señal. Una de las principales características del ATmega 328 es que posee 4 relojes [8] internos los cuales permiten la realización de tareas en paralelo, lo que lo hace muy adecuado para la funcionalidad del proyecto, debido a que hay tareas que se tienen que hacer

casi de manera simultánea. La arquitectura se traduce al siguiente circuito, encontrado en la Ilustración 5. 2. Configuración Final del Transmisor y Receptor:

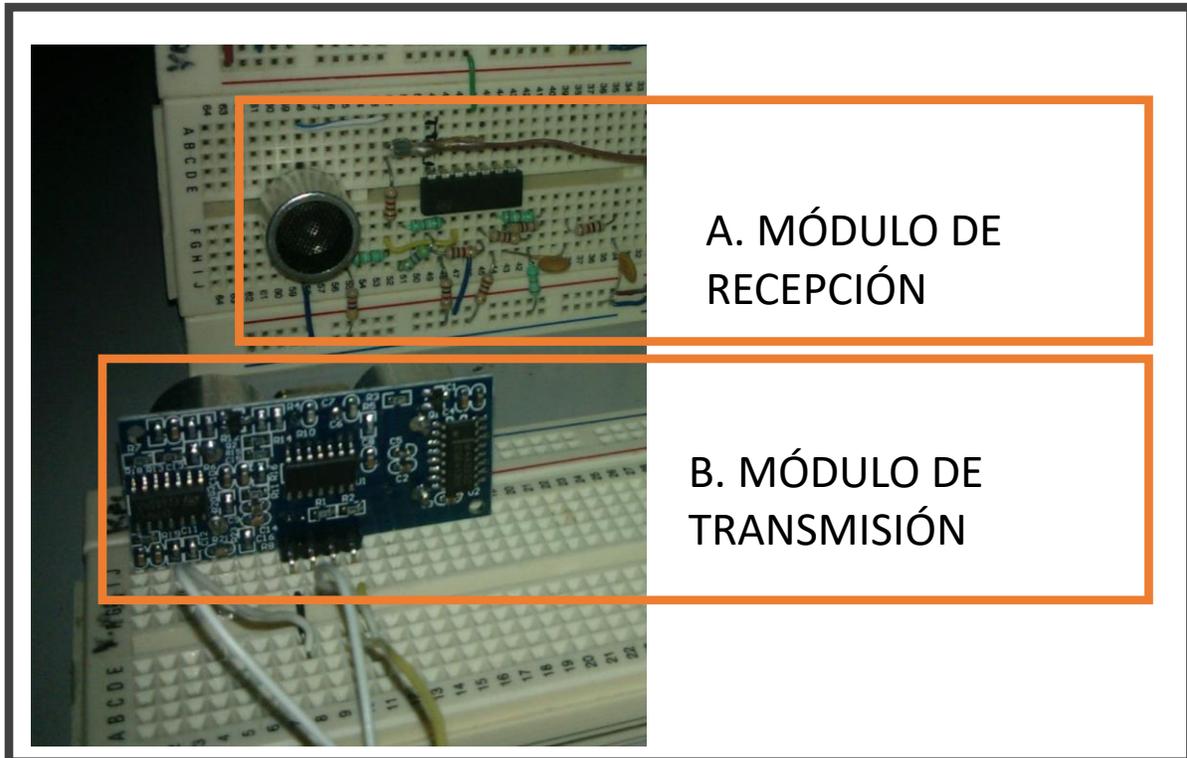


Ilustración 5. 2. Configuración Final del Transmisor y Receptor

Estos módulos se conectaron cada uno a una placa Arduino independiente para poder realizar las funciones de transmisión y recepción del mensaje de manera independiente y paralela, como se muestra en la Ilustración 5. 3. Ambos módulos trabajando y conectados a placa Arduino. Las placas de Arduino eran conectadas a una terminal en computadora que servía como puerto de salida y puerto de entrada.

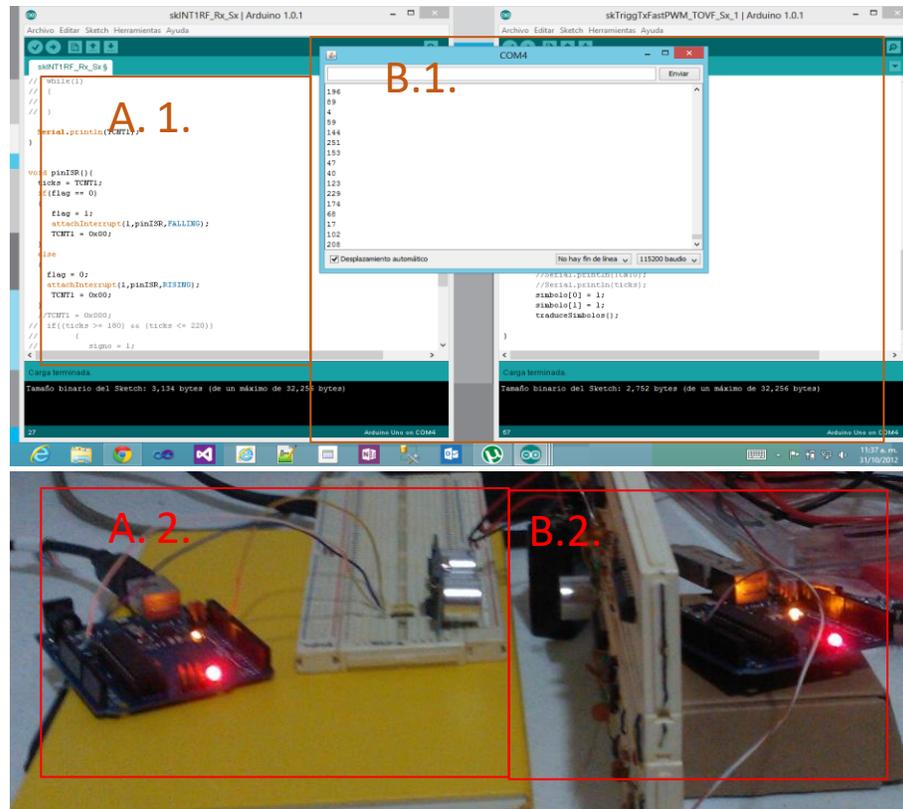


Ilustración 5. 3. Ambos módulos trabajando y conectados a placa Arduino. A.1. Interfaz gráfica para el puerto de transmisión. A.2. Circuito del Transmisor. B.1. Interfaz gráfica para el puerto del módulo de recepción. B.2. Circuito del Receptor.

5.2. Señal

Se tomó como modelo de transmisión y recepción de datos la modulación por desplazamiento de amplitud (Amplitude shift key, ASK)³¹. La transmisión de datos fue posible por medio de la generación de trenes de pulsos de diferente número de pulsos. Se estableció como convención para el caso inicial que el símbolo cero sería representado por una serie de 16 pulsos seguidos por un periodo sin pulsos. Por su parte, el símbolo uno sería representado 32 pulsos seguidos por un periodo sin pulsos. La transmisión de ambos símbolos tomaría en total el mismo tiempo

³¹ **ASK (Amplitude shift key)**, modulación por desplazamiento de amplitud es una forma de modulación en la cual se representan los datos digitales como variaciones de amplitud de la onda portadora en función de los datos a enviar

para poder identificar el inicio y el final del símbolo (Ver, ¡Error! No se encuentra el origen de la referencia.).

5.3. Hardware

5.3.1. Transmisor

El transmisor de los datos consiste en el circuito integrado por el HC-SR04 y una placa Arduino. El microcontrolador ATmega 328 que se encuentra en la placa del Arduino se encarga de crear una serie de pulsos cuadrados como resultado de la traducción de los símbolos recibidos por el software, como se muestra en la Ilustración 5. 4. Circuito de Módulo de Transmisión (HC-SR04).

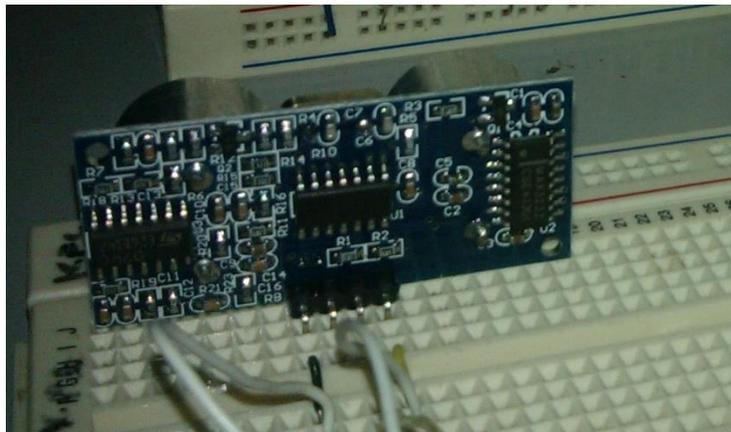


Ilustración 5. 4. Circuito de Módulo de Transmisión (HC-SR04)

Esta configuración se logró por medio de la utilización del módulo PWM³² del timer 1 del ATmega 328. Las interrupciones asociadas a este módulo y a este timer nos permiten generar una serie de pulsos con una frecuencia de 40 kHz (Ver Ilustración

³² Una señal de tipo **PWM (Pulse Width Modulated)** es una onda cuadrada que tiene un periodo fijo y un ciclo de trabajo variable. La señal PWM usa un contador cíclico que incrementa y termina al llegar al final del periodo de tiempo del PWM. En otras palabras, cuando el contador alcanza el mismo valor que el seleccionado para el ciclo de trabajo, la salida cambia de voltaje alto a voltaje bajo, por lo tanto, se crea una onda cuadrada. [10]

5. 5. Captura de señales del Osciloscopio). En esta figura se muestran los pulsos cuadrados que eran generados por el transmisor y su captura por el receptor.

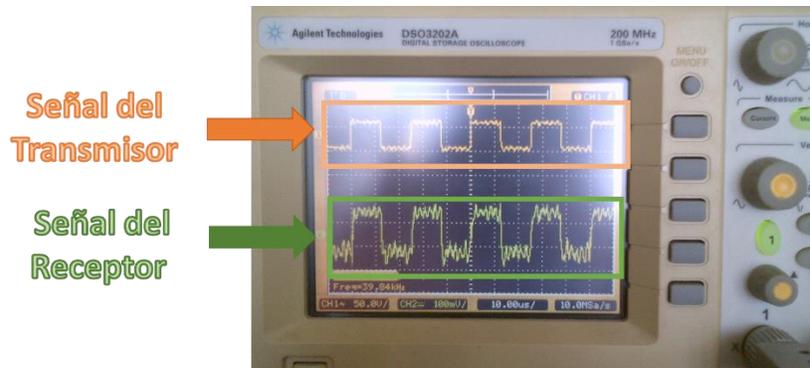


Ilustración 5. 5. Captura de señales del Osciloscopio

La finalidad de este módulo era crear trenes de pulsos a partir de un signo o carácter dentro de una palabra. En otras palabras, las palabras para poder transmitirse eran descompuestas en su unidad atómica o carácter y de acuerdo a su valor en la tabla ASCII se transmitía el valor correspondiente por medio de un número binario de 8 posiciones. Se tomó como referencia la siguiente tabla.

The ASCII code

American Standard Code for Information Interchange

ASCII control characters				ASCII printable characters				Extended ASCII characters																
DEC	HEX	Simbolo	ASCII	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo									
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ú	161	A1h	í	193	C1h	ł	225	E1h	ô
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ó	194	C2h	ŀ	226	E2h	è
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	ł	227	E3h	ó
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ü	196	C4h	ł	228	E4h	ô
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	ÿ	197	C5h	ł	229	E5h	õ
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	*	198	C6h	ł	230	E6h	ö
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	°	199	C7h	ł	231	E7h	ù
08	08h	BS	(retroceso)	40	28h	(72	48h	H	104	68h	h	136	88h	è	168	A8h	ˆ	200	C8h	ł	232	E8h	ú
09	09h	HT	(tab horizontal)	41	29h)	73	49h	I	105	69h	i	137	89h	é	169	A9h	©	201	C9h	ł	233	E9h	û
10	0Ah	LF	(salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	ê	170	AAh	¬	202	CAh	ł	234	EAh	ü
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ë	171	ABh	½	203	CBh	ł	235	EBh	ý
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACh	¾	204	CAh	ł	236	ECh	ÿ
13	0Dh	CR	(retorno de carro)	45	2Dh	.	77	4Dh	M	109	6Dh	m	141	8Dh	í	173	ADh	¸	205	CDh	ł	237	EDh	ÿ
14	0Eh	SO	(shift Out)	46	2Eh	:	78	4Eh	N	110	6Eh	n	142	8Eh	â	174	ACh	«	206	CEh	ł	238	EEh	ÿ
15	0Fh	SI	(shift In)	47	2Fh	;	79	4Fh	O	111	6Fh	o	143	8Fh	ä	175	ACh	»	207	CFh	ł	239	EFh	ÿ
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	É	176	B0h	¸	208	D0h	ó	240	F0h	ÿ
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	¸	209	D1h	ð	241	F1h	±
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	¸	210	D2h	É	242	F2h	¸
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ö	179	B3h	¸	211	D3h	Ê	243	F3h	¸
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ø	180	B4h	¸	212	D4h	Ë	244	F4h	¸
21	15h	NAK	(negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	õ	181	B5h	¸	213	D5h	Ï	245	F5h	¸
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ù	182	B6h	¸	214	D6h	Ï	246	F6h	¸
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	ú	183	B7h	¸	215	D7h	Ï	247	F7h	¸
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	¸	216	D8h	Ï	248	F8h	¸
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	ÿ	185	B9h	¸	217	D9h	Ï	249	F9h	¸
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	ÿ	186	BAh	¸	218	DAh	Ï	250	FAh	¸
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ÿ	187	BBh	¸	219	DBh	Ï	251	FBh	¸
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	ÿ	188	BCh	¸	220	DCh	Ï	252	FCh	¸
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	ÿ	189	BDh	¸	221	DDh	Ï	253	FDh	¸
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	ÿ	190	BEh	¸	222	DEh	Ï	254	FEh	¸
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	_				159	9Fh	f	191	BFh	¸	223	DFh	Ï	255	FFh	¸

Ilustración 5. 6. Código ASCII

Por lo que cada símbolo, de acuerdo a su valor hacia vibra la membrana del transductor un número de veces determinado.

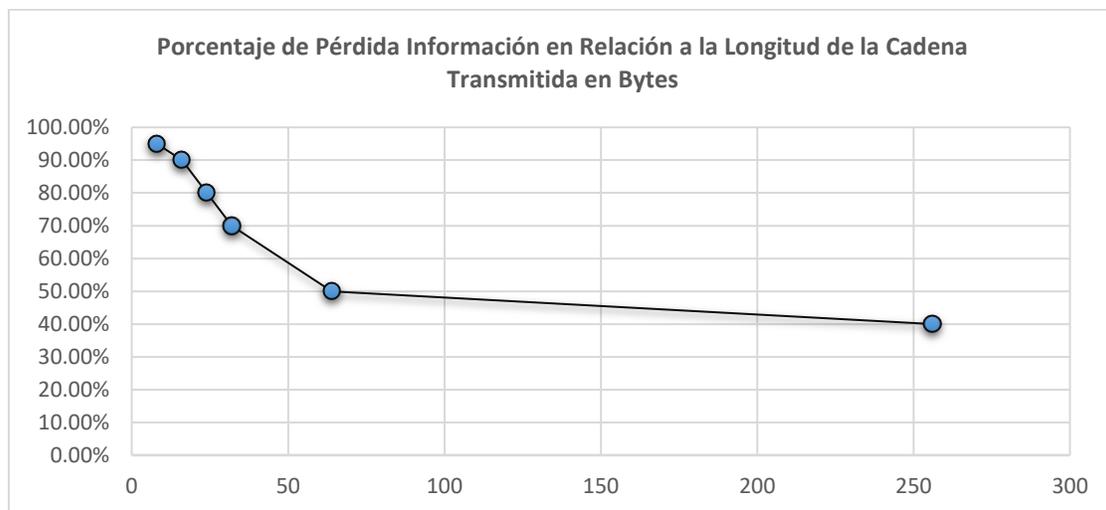
5.3.2. Receptor

La señal generada anteriormente es captada por el módulo de Recepción y por medio de otra interrupción propia del ATMEGA 328 se puede contar el tiempo en que el receptor se encuentra con voltaje en alto y definir la frecuencia. Se realizaba una transmisión y recepción efectiva de la señal. Sin embargo, el módulo de Traducción tenía problemas en determinar el inicio y el fin de un símbolo.

Debido a lo anterior, se realizó un cambio en la longitud de la señal y se determinó que los trenes de pulsos cortos son los más adecuados ya que el módulo de traducción puede determinar el inicio y el fin de un símbolo de manera más precisa.

Ahora el tren de pulsos para un cero estaba determinado se estableció en 8 pulsos y el de un uno en 16 pulsos. Además de extender el tiempo que el transmisor deja de transmitir pulsos entre símbolo y símbolo para evitar que el módulo de traducción confunda el inicio y el fin de los símbolos. Al final se tiene un tiempo de 1600 microsegundos para la transmisión de un símbolo completo no importando su valor.

Se hicieron pruebas iniciales con cadenas de 8 bytes las cuales se transmitieron exitosamente sin error en un 90% de las transmisiones. Cadenas de 64 bytes presentaban pérdida de información en aproximadamente el 50% de las transmisiones y cadenas de 256 bytes presentaban mucha pérdida de información.



Gráfica 5. 1. Porcentaje de Pérdida de Información en Relación a la Longitud de la Cadena en Bytes. En esta gráfica se puede ver como a medida que el número de bytes en la cadena aumenta (eje X) la información recibida por el receptor (eje Y) va disminuyendo.

Además del cambio en la longitud de la señal, también se hicieron cambios en el circuito de recepción en el cual se incluyó un amplificador para captar la señal. Esta modificación permitió captar mejor la señal. Este circuito actúa de la misma forma en que actúa el amplificador operacional en el circuito integrado del HC-SR04. En la siguiente ilustración se muestra el circuito amplificador del receptor, el cual toma

la señal enviada por el transmisor y la amplifica para poder leer correctamente los valores.

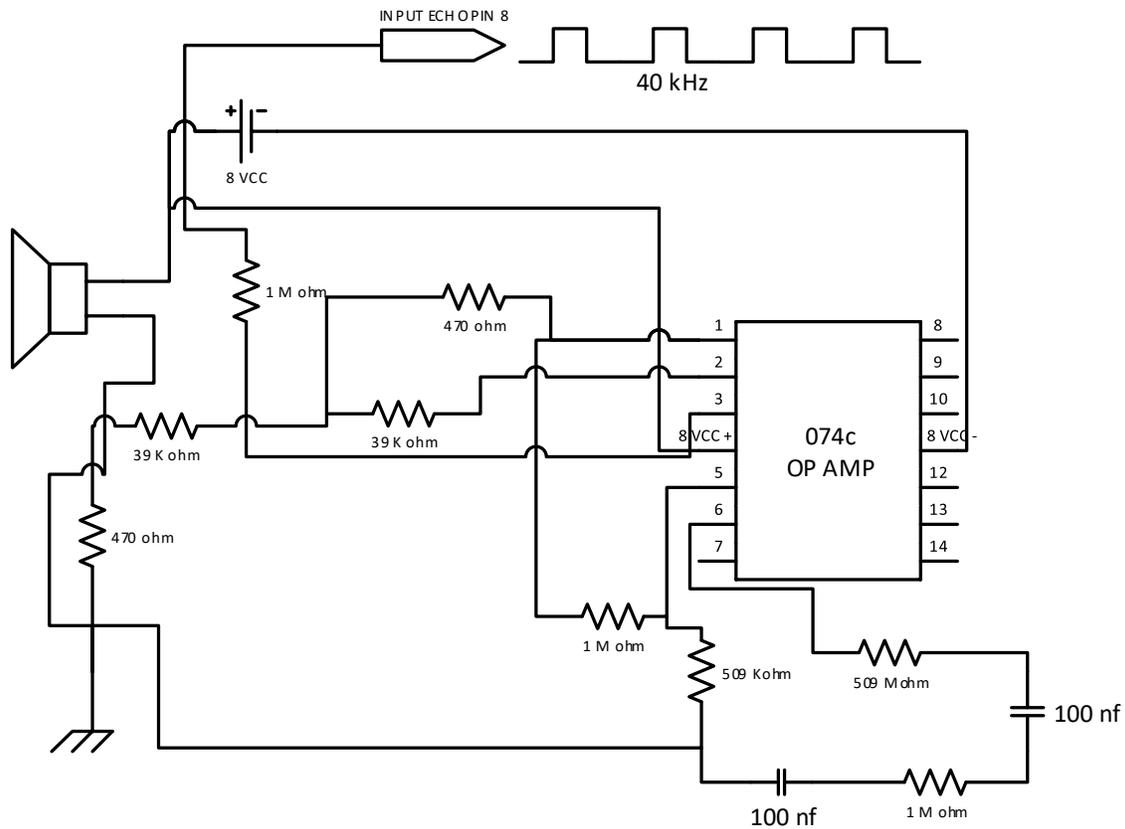


Ilustración 5. 7. Circuito amplificador del receptor

En la Ilustración 5. 7. Circuito amplificador del receptor Ilustración 5. 8. Circuito del Amplificador armado., se muestra el circuito ya armado.

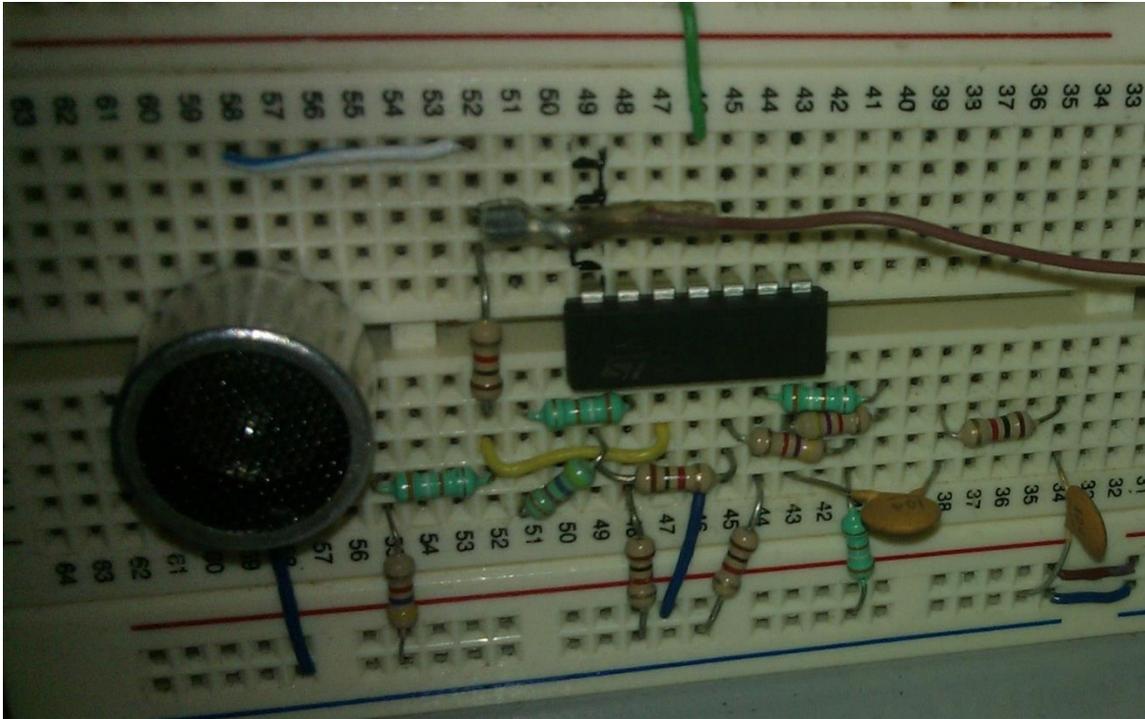


Ilustración 5. 8. Circuito del Amplificador armado.

5.4. Software

Debido a la naturaleza del proyecto el software que controla el hardware debe de contener los siguientes módulos: Módulo de Transmisión, Módulo de Recepción y Módulo de Traducción de la señal. Se utilizó Programación Orientada a Objetos para poder crear objetos de cada uno de los módulos y estos se puedan estar comunicando en un módulo global de manera instantánea.

5.4.1. Módulo de Transmisión

Desde el punto de vista del software., el módulo de transmisión se encarga de leer las frases enviadas desde un módulo de recepción a través de un puerto serial. Estas frases a su vez son descompuestas letras o símbolos y, luego, cada símbolo

es traducido a su valor en la tabla de ASCII. Estos valores en ASCII, a su vez eran traducidos a un arreglo de 8 bits o arreglos de símbolos tipo 0 (cero) y 1 (uno). Estos símbolos son lo que representarán la unidad atómica que va a ser transmitida a través de pulsos de frecuencias por el módulo de Transmisión. Por su lado, el módulo de transmisión toma la unidad atómica y la evalúa. Esto da como resultado la generación de un tren de pulsos en una frecuencia de 40 kHz dependiendo el valor de la unidad para que puedan ser captados por el módulo de Recepción. El tren de pulso es de tipo ASK (ASK-Amplitude Shift Key)³³ en onda cuadrada de 40 kHz con disparos cada cierto tiempo y un tiempo sin disparos.

5.4.2. Módulo de Recepción

El módulo de Recepción se encuentra activo todo el tiempo y está monitoreando el ambiente por pulsos de cualquier frecuencia, identifica señales de 40 kHz y cuenta cada uno de los pulsos recibidos en un tiempo establecido, lo que le permite al módulo de Traducción determinar si la señal recibida es una unidad atómica o no.

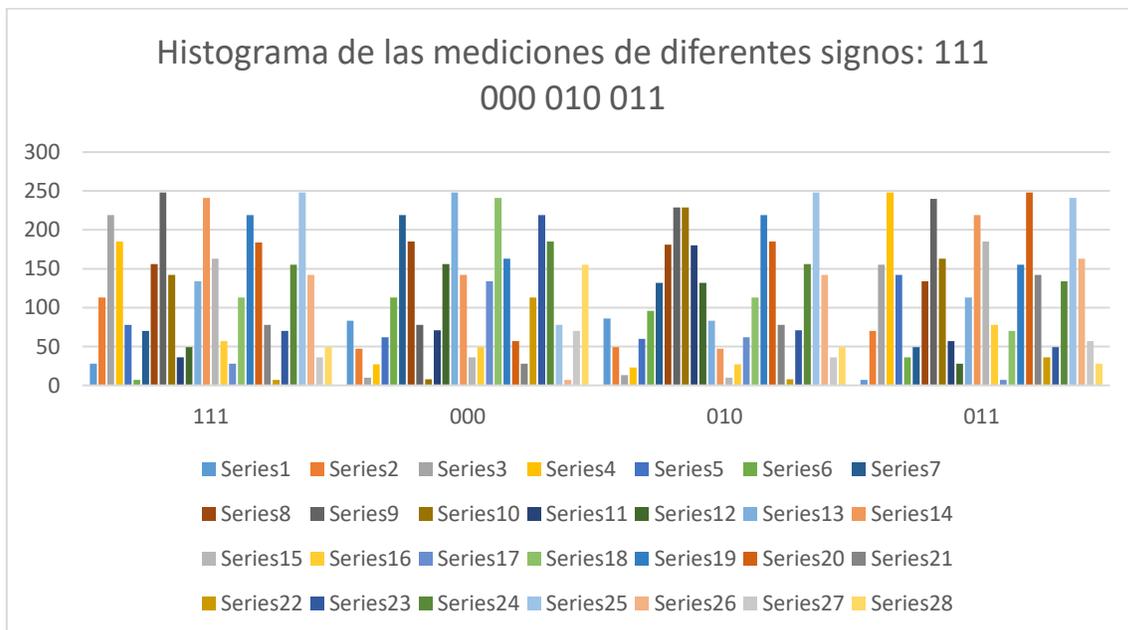
El módulo contaba las veces en que la membrana del sensor vibraba y, por lo tanto, provocaba un cambio de voltaje en el receptor. En otras palabras, el módulo de transmisión enviaba cada uno de los símbolos para que el módulo de recepción contara el número de pulsos en los cuales la membrana permanecía en alto. Si la membrana permanecía en alto 16 tics entonces se traducía como un 0. Por otro

³³ **ASK (Amplitude shift key)**, modulación por desplazamiento de amplitud es una forma de modulación en la cual se representan los datos digitales como variaciones de amplitud de la onda portadora en función de los datos a enviar. [9]

lado, si la membrana vibraba 32 veces es o significada un 1. Para evitar que el módulo de transmisión equivocara los cambios entre símbolo y símbolo se agregó una serie de tics que provocaba que la membrana lo detectara como valor bajo y baja velocidad. Esto permitió que hubiera una diferenciación entre el inicio y la terminación de la transmisión de un símbolo del número binario.

Estos números binarios se iban acumulando en un arreglo de 8 bits los cuales eran traducidos a un valor ASCII de acuerdo a la tabla de símbolos ASCII.

De acuerdo, a varias pruebas que se hicieron se podían transmitir de manera adecuada palabras o frases no mayores a 64 bytes. En la siguiente gráfica se muestran la diferencia entre las lecturas de recepción de diferentes caracteres y números binarios.



Gráfica 5. 2. Representación gráfica de la recepción de diferentes símbolos

El resultado final de cada transmisión era traducido a una palabra y esta a su vez se iba guardando en un arreglo de 8 bits, el cual al tener 8 valores era traducido a un carácter, el cual se imprimía en pantalla.

5.5. Conclusiones

Después de algunos problemas, se ha obtenido una transmisión y recepción de datos. Sin embargo, se estima que el tiempo requerido para enviar y transmitir datos es bastante prolongado y se deberían de realizar modificaciones para disminuir el tiempo y hacer la transmisión más efectiva, sin embargo, tenemos que considerar las limitaciones del canal y del medio. Por lo que, hasta el momento, se puede garantizar una buena transmisión de datos, pero con la velocidad actual la aplicación de este tipo de transmisión de datos para el envío de instrucciones de navegación no es muy conveniente. Y debido a que uno de las potenciales aplicaciones es el envío de datos para la navegación consideramos indispensable la mejora en la velocidad de envío y recepción de los datos.

En el aspecto de hardware, para el módulo de transmisión se pudo utilizar de manera transparente y sencilla el circuito integrado del HC-SR04, ya que la generación de pulsos se podía determinar fácilmente por medio del software en el programa del módulo de transmisión. Con respecto al módulo de recepción, no se pudo utilizar el circuito integrado del HC-SR04 debido a que uno de sus componentes impedía la lectura de los valores obtenidos por lo que se tuvo que utilizar al sensor ultrasónico con un circuito de amplificador armado. La transmisión y recepción de las señales en vía aérea requería de trabajo de alineación de

transmisor y receptor, sin embargo, se espera que la misma sea más potente en medio submarino.

Con respecto al software, la creación del software de transmisión y recepción requirió de realizar programación de interrupciones en el microcontrolador, lo que impidió el uso de las librerías proporcionadas por Arduino y se tuvo que crear métodos que replicaran esta misma funcionalidad, pero con la capacidad de traducir símbolos a pulsos y viceversa. Sin embargo, C++ como lenguaje de programación permitió la realización de esta tarea.

Este dispositivo se diseñó como un auxiliar en la transmisión de datos para la toma de imágenes estereoscópicas³⁴ del fondo marino. Estas imágenes tienen como propósito la reconstrucción tri-dimensional de mosaicos de imágenes, metrología de organismos y topografía circundante. Este proceso comprende la obtención de una gran cantidad de datos además de la transmisión de la misma, además de registrar al mismo tiempo diferentes parámetros físicos y químicos. Por lo que este trabajo, representa el primer paso para la transmisión de información de manera inalámbrica en entorno submarino y provee la base para que módulos más sofisticados se puedan construir sobre él o pueda trabajar junto con otros dispositivos.

³⁴ Imágenes que por medio de una técnica especial crean la ilusión de profundidad para la visión binocular. (Diner & Fender, 1993)

Referencias

- Aguilar Soto, N., Madsen, P. T., Tyack, P., Arranz, P., Marrero, J., Fais, A., . . . Johnson, M. (2011). "No shallow talk: Cryptic strategy in the vocalcommunication of Blainville's beaked whales. *Marine Mammal Science*, 1748-7692.
- Akyildiz, I., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks* 3, 257-278.
- Alexandrescu, A. (2001). *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley.
- Alim, A. (2010). Implementation and Performance Analysis of Direct Sequence Spread Spectrum Modem. *JEEE*, (3):17-22.
- Arduino. (28 de Enero de 2013). *Arduino*. Obtenido de Arduino: <http://arduino.cc/en/>
- Arduino. (28 de Enero de 2013). *Arduino Development Environment*. Obtenido de Arduino: <http://arduino.cc/en/Guide/Environment>
- Arduino. (07 de Enero de 2013). *delay*. Obtenido de Arduino: <http://arduino.cc/en/Reference/delay>
- Arduino. (08 de Enero de 2013). *Millis*. Obtenido de Arduino: <http://arduino.cc/en/Reference/Millis>

- Atmel Coproration. (2010). *8-bit AVR Microcontroller with 4/8/46/32K Bytes In-System Programmable Flash* . Atmel.
- Atmel Corporation. (2012). *Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash*. San Jose: Atmel Corporation.
- Baker, R. (2010). *CMOS: Circuit Design, Layout, and Simulation*. New York: Wiley-IEEE.
- Barbe, D. (1982). *Very large scale integration (VLSI)*. Verlag: Springer.
- Benson, B., Chang, G. M., Graham, B., & Kastner. (2006). Desing of a Low-cost Acoustic Modem for Moored Oceanographic Applications. *WUWNet*, 484-487.
- Bible, S. (6 de Mayo de 2012). *TAPR: Spread Spectrum: it's not jut for breakfast anymore!* Recuperado el 5 de Junio de 2012, de TAPR: <http://www.tapr.org>
- Bohnacker, H., Gross, B., Laub, J., & Lazzeroni, C. (2012). *Generative Design: Visualize, Program, and Create with Processing*. Princeton: Princeton Architectural Press.
- Cushier, R., & King, W. (1933). The spermaceti organ and nasal passages of the sperm whale (*Physeter catodon*) and other odontocetes. *American Museum novitates*, 1889-1944.
- Deecke, V., Ford, J., & Slater, J. (2004). The vocal behaviour of mammal-eating killer whales: communicating with costly calls. *ANIMAL BEHAVIOUR*, (69):419-455.

Diner, D., & Fender, D. (1993). Chapter 3. Methods of Viewing 3-D Images. En D. Diner, & D. Fender, *Human Engineering in Stereoscopic Viewing Devices* (págs. 35-49). New York: Springer.

ECMA International. (24 de Enero de 2016). *C# Language Specification*. Obtenido de ECMA International: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>

Fordyce, R., & Barnes, L. (1994). The evolutionary history of whales and dolphins. *ANNU REV EARTH PLANET SCI*, (22):419-455.

Goh, J., Shaw, A., & Al_Shamma'a, A. (2009). Underwater Wireless Communication System. *Journal of Physics*, 012029.

González-Cabrera, V. (1971). *Física Fundamental*. Ciudad de México: Editorial Progreso S.A. de C.V.

Harvard. (2003). Key Signature. En D. Randel, *The Harvard Dictionary of Music* (pág. 443). Harvard: Harvard University Press.

Holland, R., Waters, D., & Rayner, J. (2004). "Echolocation signal structure in the Megachiropteran bat *Rousettus aegyptiacus* Geoffroy 1810. *Journal of Experimental Biology*, 207(25): 4361-4369.

Holtek Semiconductor Inc. (11 de April de 2000). *HT12A/HT12E 2 Series Encoders*. Obtenido de Holtek: <http://www.holtek.com.tw>

Jensen, F., Perez, J., Johnson, M., Aguilar Soto, N., & Madsen, P. (2011). Calling under pressure: short-finned pilot whales make social calls during deep

foraging dives. *Proceedings of the Royal Society B*, doi: 10.1098/rspb.2010.2604.

Kyhn, L. A., Tougaard, J., Beedholm, K., Jensen, F., E, A., & al., e. (2013). Clicking in a Killer Whale *Isoturus paucus*: Harrow-Band, High Frequency Biosonar Clicks of Harbour Porpoise (*Phocoena phocoena*) and Dall's Porpoise (*Phocoenoides dalli*). *PLoS ONE*, 8(5):e63763.

Langton, C. (1 de Enero de 2002). *Inter Symbol Interference (ISI) and Root-raised Cosine (RRC) filtering*. Obtenido de Complex to Real: www.complextoreal.com

Microsoft. (29 de Enero de 2013). Obtenido de CodePlex: <http://www.codeplex.com/>

Microsoft. (1 de Enero de 2013). *Arduino for Visual Studio*. Obtenido de Codeplex: <http://visualmicro.codeplex.com/>

Microsoft. (24 de Enero de 2016). *Visual J# Product Announcement*. Obtenido de Visual Studio: <https://msdn.microsoft.com/es-mx/vstudio/bb188593>

Milne, P. (1983). *Underwater Acoustic Positioning*. Houston: Gulf Publishing.

Nummela, S., Thwissen, J., Bajpai, S., Hussain, T., & Kummar, K. (2007). Sound transmission in archaic and modern whales: anatomical adaptations for underwater hearing. *THE ANATOMICAL RECORD*, (290):716-733.

Payne, R. (1983). *Communication and behavior of whales*. California: Westview Press.

- Proakis, J., & Manolakis, D. (2007). *Digital Signal Processing: Principles, algorithms and applications*. Pearson Prentice Hall.
- Proakis, J., Rice, J., Sozer, E., & Stojanovic, M. (s.f.). *Shallow Water Acoustic Networks*. MIT, Busc año.
- Randal, R., & Romero Sweeney, M. (2006). *A tester's guid to .Net programming*. Apress.
- Rashid, M. (2004). Rectificadores con Diodos. En M. Rashid, *Electrónica de potencia* (págs. 68-120). México: Pearson.
- Reynolds, J. E. (2013). En J. E. Reynolds, *Biology of Marine Mammlas*. Smithsonian Institution.
- Rodríguez-Mellado, J., & Marín-Galvín, R. (1999). *Fisicoquímica de Aguas*. Madrid: Díaz de Santos, S.A.
- Sozer, E., Proakis, J., Stojanovic, M., Rice, J., Benson, A., & Hatch, M. (1999). Direct sequence spread spectrum based modem for under water acoustic communication and channel measurements. *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century* (págs. 228-223). Seattle, WA: IEEE.
- Squires, D. (1959). Deep sea corals collected by the Lamont Geological Observatory. 1. Atlantic corals. *American Museum Novitates*, 1-42.
- Stojanovic, M. (1999). Underwater Acoustic Communication. En *Wiley Encyclopedia of Eectricla and Electronics Engineering*. Obtenido de Wiley Encyclope.

Urick, R. (1983). *Principles of Underwater Sound*. New York: McGraw-Hill.

Wahlberg, M., Beedholm, K., Heerfordt, A., & Møhl, B. (2011). Characteristics of biosonar signals from the northern bottlenose whale, *Hyperoodon ampullatus*. *Journal of the Acoustical Society of America* , 130(5): 3077-3084.

Zheng, Y., Xiao, C., Yang, T., & Yang, W.-B. (2007). Frequency-Domain Channel Estimation and Equalization for Single Carrier Underwater Acoustic Communications. *MTS*.

Contenido

Resumen.....	1
Capítulo 1. Antecedentes	2
1. Comunicaciones Submarinas.....	2
1.1. Comunicaciones submarinas por medio biológicos	2
1.2. Comunicaciones submarinas por medio tecnológicos	5
1.2.1. Satélites, Radiofrecuencia y Telefonía celular	7
1.2.2. Amarres Cableados	7
1.2.3. Modem Acústico	8
1.2.4. Características generales	8
Capítulo 2. Formas de Transmisión de Datos Submarina	10
2. Formas de Transmisión de datos	10
2.1. Radiofrecuencia.....	10
2.2. Telecomunicaciones Ópticas	11
2.3. Ondas Acústicas	11
Capítulo 3. Fundamentos de la Acústica Submarina	13
3. Acústica Submarina.....	13
3.1. Propagación de la Onda Sonora: Velocidad, Densidad e Impedancia..	13
3.2. Absorción del Sonido	14
3.3. Desafíos Planteados por el Canal.....	15

Capítulo 4. Desarrollo de Modem Subacuático Inalámbrico.....	21
4. Desarrollo del Prototipo	21
4.1. Componentes de Hardware	21
4.2. Componentes de Software.....	26
4.3. Desarrollo de Circuito	28
4.4. Desarrollo del Software.....	29
Capítulo 5. Resultados	56
5.1. Arquitectura del Proyecto	56
5.2. Señal.....	58
5.3. Hardware.....	59
5.3.1. Transmisor.....	59
5.3.2. Receptor	61
5.4. Software	64
5.4.1. Módulo de Transmisión	64
5.4.2. Módulo de Recepción	65
5.5. Conclusiones.....	67
Referencias	69

Capítulo 1. En este capítulo se proveen los antecedentes para la creación de un módem acústico. Se discuten el tipo de comunicaciones submarinas y sus aplicaciones actuales, así como se establece la utilidad del módem acústico, tanto para la industria como para la investigación.

Capítulo 2. Aquí se discuten los diferentes canales para la transmisión de datos en el medio submarino, tanto retos como ventajas. Además de permitir exponer la ventaja de las ondas acústicas como medio de transmisión subacuática.

Capítulo 3. Este capítulo ahonda más sobre las características de las ondas acústicas como canal. Además se hace una explicación breve sobre la propagación de las ondas sonoras debajo del agua y los desafíos planteados por el canal.

Capítulo 4. En este capítulo se detalla el desarrollo del módem subacuático desde el punto de vista de software y de hardware.

Capítulo 5. En este capítulo se presentan los resultados después de haber realizado el desarrollo y la implementación del módem acústico. Se presenta la arquitectura completa del prototipo y además se presenta la forma en que se transmitirán y recibirán los mensajes. Además de que se discuten los resultados y las áreas de oportunidad para este prototipo.