



Escuela Técnica Superior de Ingenieros de
Telecomunicación
Universidad de Valladolid

Sistema de comunicación agua-aire mediante radio y ultrasonidos con geolocalización

Water-air communication system by radio and ultrasound with geolocation

Autor: Miguel Alonso Felipe
Tutor: Ramón De La Rosa Steinz

Valladolid a 10/09/2019

*A mi familia, pareja y amigos
que han estado presentes en la elaboración
de este proyecto y me han apoyado y soportado.
También a mi tutor y profesor Ramón
por mostrarme los entresijos de la electrónica analógica
y guiarme por sus intrincados senderos.*

Resumen

En este trabajo se van a estudiar y desarrollar los métodos que se pueden emplear para establecer una comunicación que atraviese dos medios tan dispares como el aire y el agua. Se analizarán los distintos modelos teóricos que caracterizan la propagación ondulatoria en cada uno de los medios de interés y los posibles módulos funcionales implementables. Se explicará y se realizará un prototipo práctico que mediante un sistema que ejerza de nexo entre ambos medios ofrezca una implementación con las tecnologías económicamente más apropiadas para así lograr, un enlace por el que se puedan transmitir y recibir datos de una forma sencilla y eficaz.

Palabras clave:

Comunicación subacuática, hidrófono, transmisor acústico, PLL, geolocalización, código Morse, boya, FM, radiocomunicación, comunicación aire-agua

Abstract

In this work we will study and develop the methods that can be used to establish a communication that propagates through two different means such as air and water. We will analyze the different theoretical models that characterize the wave propagation in each of the means of interest and the possible implementable functional modules. A practical prototype will be explained and developed through a system that acts as a link between the two means, proposing an implementation with the most economically appropriate technologies. This all works towards achieving a link where data can be transmitted and received through in a simple and efficient way.

Keywords:

Underwater communication, hydrophone, acoustic transmitter, PLL, geolocation, Morse code, buoy, FM, radiocommunication, air-water communication

Índice general

1. Introducción	11
1.1. Motivación	16
1.2. Objetivos	18
1.3. Estructura del documento	20
2. Elección y explicación de las soluciones empleadas	21
2.1. Visión general	21
2.2. Elección de los canales de transmisión adecuados y de las tecnologías empleadas	23
2.2.1. Tecnologías subacuáticas	23
2.2.2. Tecnologías aéreas	33
2.2.3. Tecnologías electrónicas complementarias	43
2.3. Desarrollo y análisis del módulo de geolocalización	52
2.4. Desarrollo hardware y software para la generación de tonos	55
2.4.1. Modalidad de generación de los tonos mediante hardware	55
2.4.2. Modalidad de generación de los tonos mediante software	57
3. Explicación detallada del sistema de comunicación basado en boyas	63
3.1. Módulo tranceptor aerotransportado	63
3.2. Módulo boya	69
3.3. Módulo tranceptor acústico submarino	80
4. Pruebas y resultados	86
4.1. Pruebas de subsistemas	86
4.1.1. Pruebas de generación de tonos mediante software	86
4.1.2. Transmisión-Recepción radio en la banda de 2,4 GHz a 2,5 GHz	90
4.1.3. Transmisión-Recepción radio en la banda de FM	92
4.1.4. Pruebas sobre el funcionamiento del enlace radio entre el avión y la boya, incluyendo la detección de tonos en el avión	93
4.1.5. Amplificación de señales de audio recogidas con el micrófono	95
4.1.6. Transmisión y recepción subacuática de tonos	97
4.1.7. Prueba con el generador de tonos basado en VCO	99
4.2. Pruebas de sistema	101
4.2.1. Prueba de recepción en submarino de mensaje del avión	101
4.2.2. Prueba de recepción en avión de mensaje del submarino	102
4.2.3. Prueba de obtención de posición GPS	104

5. Conclusiones	105
6. Líneas de mejora	106
Anexos	109
A. Código para la generación de tonos (transmisión del mensaje desde avión y desde el submarino)	110
A.1. Octave	110
A.2. Matlab	112
A.3. Arduino	117
B. Código de control de los módulos de Arduino	120
B.1. Transmisor en la banda de FM comercial. FM Transmitter v2.0	120
B.2. Receptor en la banda de FM comercial. Receptor basado en el módulo de radio TEA5767	120
B.3. Módulo tranceptor nRF24L01 (2,4GHz-2,5GHz)	121
B.3.1. Módulo 1	121
B.3.2. Módulo 2	122
C. Esquemas electrónicos y módulos auxiliares	124
C.1. Esquema de amplificador operacional con alimentación simple	124
C.1.1. Valores de la etapa de amplificación basada en el integrado TL082 empleada en la caja del avión	126
C.1.2. Valores de la etapa de amplificación doble basada en el integrado TL082 empleada en la caja de la boya y del submarino	127
C.2. Esquema electrónico del conexionado del CD4046B (PLL). Sistemas tipo 1 de orden 2	128
C.2.1. Valores de los esquemas electrónicos basados en PLL	131
C.3. Integrado CD4046 (VCO)	134
C.3.1. Valores del VCO empleados en la caja de la boya	135
C.4. Amplificador de potencia	136
D. Diagramas de flujo	138
D.1. Diagrama de flujo de generación de código Morse	138
D.2. Diagrama de flujo del bloque avión	139
D.3. Diagrama de flujo del bloque boya	139
D.4. Diagrama de flujo del bloque submarino	140
E. Información sobre submarinos	141
E.1. Distribución de antenas por el casco del submarino	141
E.2. Tablas de los sistemas de antenas usados por los submarinos	142
E.3. Tabla de frecuencias usadas por los submarinos y su utilidad	144
F. Código de las cajas del avión, la boya y el submarino	145
F.1. Código de la caja del avión (recepción)	145
F.2. Código de la caja de la boya(transmisión y recepción)	150
F.3. Código de la caja del submarino (recepción)	159

Índice de figuras

1.1.	Situaciones de comunicación en un submarino clase 688 ([14],[15]). . . .	11
1.2.	Ejemplos de alargamiento eléctrico de antenas cortas utilizando el efecto inductivo[15, cap. 4].	12
1.3.	Esquema que muestra las tecnologías de comunicación típicas instaladas en los vehículos.	13
1.4.	(a) Muestra que un transmisor de radio no puede comunicarse porque las señales de radio sufren demasiado amortiguamiento en el agua. (b) Muestra que las señales acústicas se reflejan en la superficie del agua. (c) Muestra que un receptor TARF (<i>Translational Acoustic-RF</i>) emplea un radar para detectar en la superficie vibraciones causadas por ondas de presión acústica y las emplea para decodificar el mensaje.	14
1.5.	Las vibraciones superficiales se traducen en una modulación de fase. La fase de la reflexión inalámbrica cambia con vibraciones superficiales diminutas[2, pág. 4].	15
1.6.	Distribución de antenas en un submarino clase 688.	17
1.7.	Comunicación entre avión y submarino con el sistema de boyas.	18
1.8.	Estructura del documento.	20
2.1.	Esquema del funcionamiento global del sistema de comunicación basado en boyas.	22
2.2.	Se muestran dos tipos de enlace utilizando tecnología óptica. En el enlace A se muestran tres situaciones de incidencia; incidencia normal con transferencia total de potencia, incidencia con un $\theta < \theta_c$ e incidencia con $\theta > \theta_c$ con reflexión total. En el enlace B se muestra como sería la tecnología óptica implementada con nuestro sistema boya.	25
2.3.	Penetración de las ondas electromagnéticas en el agua salada[13].	26
2.4.	Perfil típico de la velocidad de la propagación del sonido para latitudes medias[9].	27
2.5.	Variación de la velocidad del sonido obtenida por las ecuaciones de Del Grosso y Chen y Millero con la temperatura[9].	28
2.6.	Variación del coeficiente de absorción con la frecuencia[9].	29
2.7.	Micrófono piezoeléctrico con etapa de amplificación.	30
2.8.	Se muestran un altavoz electromecánico y un micrófono de carbón de los empleados en la transmisión acústica subacuática.	31
2.9.	Modulación 2-FSK empleada en el enlace de subida (dirección avión) y en el enlace de bajada (dirección boya).	33
2.10.	Transceptor NRF24L01 con antena impresa.	34
2.11.	Transceptor NRF24L01 con antena externa.	34

2.12. Montaje del sistema transceptor en la banda de 2.4GHz a 2.5GHz. . . .	36
2.13. Captura de pantalla que muestra el funcionamiento del programa para los módulos nRF24L01.	36
2.14. Esquema de transmisión FM.	38
2.15. Método para duplicar el tamaño efectivo de una antena para la banda de FM.	39
2.16. Transmisor FM de la casa <i>ELECHOUSE</i>	39
2.17. Diagrama de bloques del integrado principal del Transmisor FM. Se observa que es un sistema de modulación FM basado en sintetizador de frecuencias, lo que estabiliza la frecuencia portadora.	40
2.18. Módulo receptor en la banda de FM <i>TEA5767</i>	41
2.19. Diagrama de bloques del integrado <i>TEA5767</i>	42
2.20. Módulo amplificador de potencia.	44
2.21. Módulo amplificador de potencia empleado en los montajes.	44
2.22. Esquema de un montaje de dos etapas de amplificación usando el TL082.	45
2.23. Montaje de un amplificador operacional de dos etapas.	46
2.24. Esquema electrónico del conexionado del integrado CD4046B.	47
2.25. Diagrama de bloques del PLL.	47
2.26. Función de transferencia del PLL.	48
2.27. Respuesta al escalón de un sistema tipo I de orden II.	49
2.28. Montaje del integrado CD4046B para ser usado como PLL.	50
2.29. Función del filtro a la salida de la patilla 10 del PLL.	50
2.30. Ilustración del rizado de la señal a la salida de la patilla 10. En amarillo una señal de entrada de 5 KHz (patilla 14), en azul la señal de tensión con rizado a la salida de la patilla 10.	50
2.31. Montaje del módulo GPS con la placa Arduino UNO	53
2.32. Datos recogidos por el módulo GPS sin usar la librería TinyGPS	53
2.33. Datos recogidos por el módulo GPS con la librería TinyGPS	54
2.34. Módulo generador de onda cuadrada, triangular y sinusoidal.	55
2.35. Esquema de conexionado del integrado CD4046B empleando su VCO.	56
2.36. Recreación simplificada de alto nivel de la generación de tonos software.	57
2.37. Diagrama de flujo explicativo de la traducción de código ASCII a Morse mediante tonos de distinta frecuencia.	58
2.38. Programa de Octave generador de Morse con tonos de distinta frecuencia.	59
2.39. Programa de Matlab generador de Morse con tonos de distinta frecuencia.	60
2.40. Esquema eléctrico del conexionado del <i>Buzzer</i>	60
2.41. Programa de Arduino generador de Morse con tonos de distinta frecuencia.	61
3.1. Diagrama de flujo explicativo de funcionamiento del bloque avión.	64
3.2. Funcionamiento del sistema de traducción vía software del avión.	65
3.3. Diagrama de conexionado del módulo avión.	65
3.4. Esquema electrónico del conexionado del integrado CD4046B.	68
3.5. Montaje con una etapa de amplificación usando el TL082.	68
3.6. Fotos del montaje de la caja del avión.	69
3.7. Vista frontal y dorsal de la caja del avión.	69
3.8. Diagrama de flujo explicativo de funcionamiento del módulo boya.	70

3.9. Se ilustra el método de codificación empleado. En rojo : se manda el número de dígitos antes de la coma decimal. En azul : se manda el signo de la coordenada, donde p equivale a positivo y n a negativo.	71
3.10. Diagrama de conexionado del módulo boya.	72
3.11. Esquema electrónico del conexionado del integrado CD4046B.	75
3.12. Esquema de conexionado del integrado CD4046B empleando su VCO.	75
3.13. Montaje total de dos etapas de amplificación usando el TL082.	76
3.14. Se muestra la fuente de alimentación de 12V empleada.	77
3.15. Se muestra el convertidor dc-dc que reduce la tensión de 12V a 8V para alimentar el Arduino.	78
3.16. Fotos del montaje de la caja de la boya.	78
3.17. Fotos del exterior de la caja de la boya.	79
3.18. Diagrama de flujo explicativo de funcionamiento del bloque submarino.	80
3.19. Diagrama de conexionado del módulo submarino.	81
3.20. Esquema electrónico del conexionado del integrado CD4046B.	83
3.21. Montaje total de dos etapas de amplificación usando el TL082.	83
3.22. Fotos del montaje de la caja del submarino.	84
3.23. Fotos del exterior de la caja del submarino.	85
4.1. Fotografía de la prueba de ejecución del código Octave.	87
4.2. Fotografía de la prueba de la petición GPS vía software.	87
4.3. Fotografía de la prueba de ejecución del código MatLab.	88
4.4. Fotografía de la prueba de ejecución del código en C++ implementado en el microcontrolador de Arduino.	89
4.5. Fotografía del conexionado del <i>buzzer</i> con el Arduino.	89
4.6. Fotografía de la prueba realizada con los módulos de transmisión radio en la banda de 2,4 GHz a 2,5 GHz. Se señala en verde la recepción de datos en el transceptor 1, en azul cuando el transceptor 1 toma el rol de emisor y en rojo el momento en el que el transceptor 2 abandona la comunicación.	90
4.7. Fotografía del montaje de los módulos transceptores radio en la banda de 2,4 GHz a 2,5 GHz.	91
4.8. Fotografía de la prueba realizada con los módulos de transmisión radio en la banda de FM y la antena provisional del transmisor de FM.	92
4.9. Se muestra uno de los tonos transmitidos desde el ordenador (500 Hz).	93
4.10. Se muestra el enganche del PLL a uno de los tonos (1000 Hz).	94
4.11. Aplicación móvil de Android generando un tono puro de 800 Hz.	95
4.12. Se muestra el montaje empleado para la recepción, amplificación y enganche de un tono de 800 Hz empleando un receptor acústico.	96
4.13. Se muestra la pantalla del osciloscopio del laboratorio de docencia, donde se grafica en amarillo el tono de 800 Hz captado por el micrófono y en azul la salida de la patilla 14 del PLL. Se recuadra en rojo como la frecuencia de enganche del PLL se corresponde con la transmitida desde el teléfono móvil.	96
4.14. Se muestra el generador de funciones fijado a una frecuencia de 500 Hz para una onda sinusoidal. El tono se genera sin offset (el cursor está desviado) y a máxima amplitud.	97

4.15. Se muestran el transmisor y el receptor acústico impermeabilizados estableciendo el enlace de transmisión del tono de 500 Hz bajo el agua.	98
4.16. Se muestran la doble etapa de amplificación basada en el integrado TL082.	98
4.17. Se muestran la frecuencia de 500 Hz extraída del receptor acústico y amplificada con la doble etapa de amplificación basada en el integrado TL082.	98
4.18. Se muestra el VCO insertado en el zócalo con su red resistiva a la entrada de la patilla 9.	99
4.19. Se muestra en el osciloscopio la salida de la patilla 4 del VCO al aplicar una tensión de 5V a la entrada de la red resistiva.	100
4.20. Se muestra en el osciloscopio la salida de la patilla 4 del VCO al aplicar una tensión de 0V a la entrada de la red resistiva.	100
4.21. Se muestra una visión general del montaje total del sistema.	101
6.1. Ejemplo de una de las situaciones del protocolo de comunicación donde hay un avión aliado y uno enemigo. C.REQ: Communication Request, Petición de establecer conexión; ACK SUB.: ACKnowledge, Asentimiento de conexión por parte del submarino; ENEMY SIGNAL: Señal enemiga	108
C.1. Montaje de una etapa de amplificación usando el TL082.	124
C.2. Montaje total de dos etapas de amplificación usando el TL082.	125
C.3. Esquema electrónico del conexionado del integrado CD4046B.	128
C.4. Diagrama de bloques del PLL.	129
C.5. Función de transferencia del PLL.	129
C.6. Respuesta al escalón de un sistema tipo I de orden II.	131
C.7. Esquema de conexionado del integrado CD4046B empleando su VCO.	134
C.8. Módulo amplificador de potencia.	136
C.9. Módulo amplificador de potencia empleado en los montajes.	137
D.1. Diagrama de flujo explicativo de la traducción de código ASCII a Morse mediante tonos de distinta frecuencia.	138
D.2. Diagrama de flujo explicativo del funcionamiento del bloque avión.	139
D.3. Diagrama de flujo explicativo del funcionamiento del bloque boya.	139
D.4. Diagrama de flujo explicativo del funcionamiento del bloque submarino.	140
E.1. Distribución de antenas en un submarino clase 688.	141
E.2. Tipo de sistemas de antenas usado en un submarino (1 de 2).	142
E.3. Tipo de sistemas de antenas usado en un submarino (2 de 2).	143
E.4. Bandas de frecuencia utilizadas para la comunicación en un submarino.	144

Índice de tablas

2.1. Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz.	51
2.2. Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.	57
3.1. Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.	67
3.2. Los valores RC empleados para el diseño del circuito con el integrado TL082.	67
3.3. Los valores de los componentes electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.	74
3.4. Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz.	74
3.5. Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.	76
3.6. Los valores RC empleados para el diseño del circuito con el integrado TL082.	77
3.7. Los valores de los componentes electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.	82
3.8. Los valores RC empleados para el diseño del circuito con el integrado TL082.	84
C.1. Los valores RC empleados para el diseño del circuito con el integrado TL082.	126
C.2. Los valores RC empleados para el diseño del circuito con el integrado TL082.	127
C.3. Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja del avión.	132
C.4. Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja de la boya.	133
C.5. Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz en la caja de la boya.	133
C.6. Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja del submarino.	134
C.7. Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.	135

Capítulo 1

Introducción

Las telecomunicaciones engloban un amplio rango de aplicaciones que pretenden lograr comunicaciones en situaciones cada vez más complejas, veloces y eficientes. Uno de los grandes promotores de estudios sobre las telecomunicaciones es el ejército, donde cubrir las comunicaciones en cualquier tipo de lugar y circunstancia es un tema crucial. De entre todo el marco de recursos móviles con los que cuenta el ejército, un vehículo de especial importancia en la estrategia marítima son los submarinos. Estos últimos, si bien es cierto que cuentan con tecnología puntera y un gran campo de investigación y desarrollo, se ven ciertamente limitados en cuanto a temas de comunicación concierne. Los submarinos, a pesar de contar con numerosos sistemas de radiofrecuencia y multitud de antenas repartidas por todo su casco, no cuentan con un sistema que les permita establecer fluidamente una comunicación con objetos externos al medio subacuático (véase el Anexo E). En la Figura 1.1 se muestra como el submarino tiene la **necesidad de emerger** progresivamente para poder transmitir en bandas de frecuencia más elevadas.

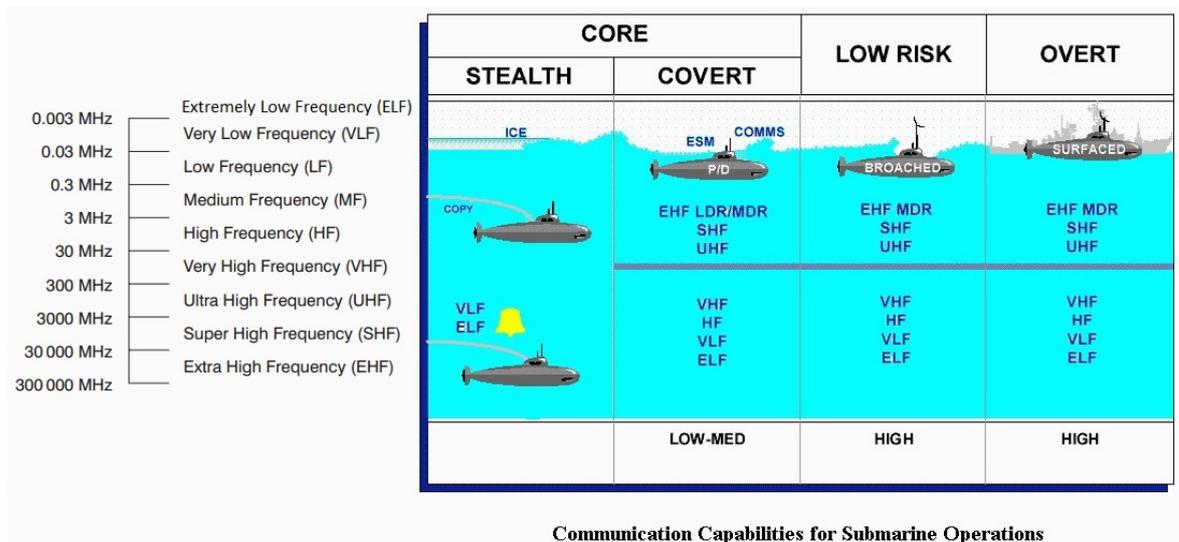


Figura 1.1: Situaciones de comunicación en un submarino clase 688 ([14],[15]).

El submarino únicamente puede comunicarse por radio bajo el agua desplegando una larga antena tras él. El tamaño necesario para una antena en una instalación terres-

tre con la que se pueda transmitir información a tan baja frecuencia es desmesurado; incluso empleando alargamiento eléctrico, que son técnicas por las cuales se consigue simular una antena de una determinada longitud a partir de una antena más corta (por ejemplo haciendo más inductiva la antena más corta añadiendo bobinas)[15, cap. 4, pág. 109-110], Figura 1.2). Realizando unos cálculos rápidos y según la ecuación (1.1), para la banda de ELF (de 3 Hz a 300 Hz) se obtiene $\lambda = 29.979 \text{ Km}$ para $f = 10\text{Hz}$, que es la frecuencia más baja a la que opera la antena OE-315 integrada en el submarino clase 688¹ (se adjunta en las tablas del Anexo E). Sabiendo que la longitud de la antena tiene que ser como mínimo la mitad de la longitud de onda[15, cap. 4], supone una antena de $\lambda/2 = 14.989,5 \text{ Km}$, que tras emplear técnicas de alargamiento eléctrico se logra reducir a tamaños de decenas de kilómetros. A pesar de que esta tecnología permite una comunicación directa con el submarino, únicamente se utiliza para transmitirle datos y voz, con lo que el submarino es un mero receptor y no puede transmitir con la antena que incorpora (Anexo E).

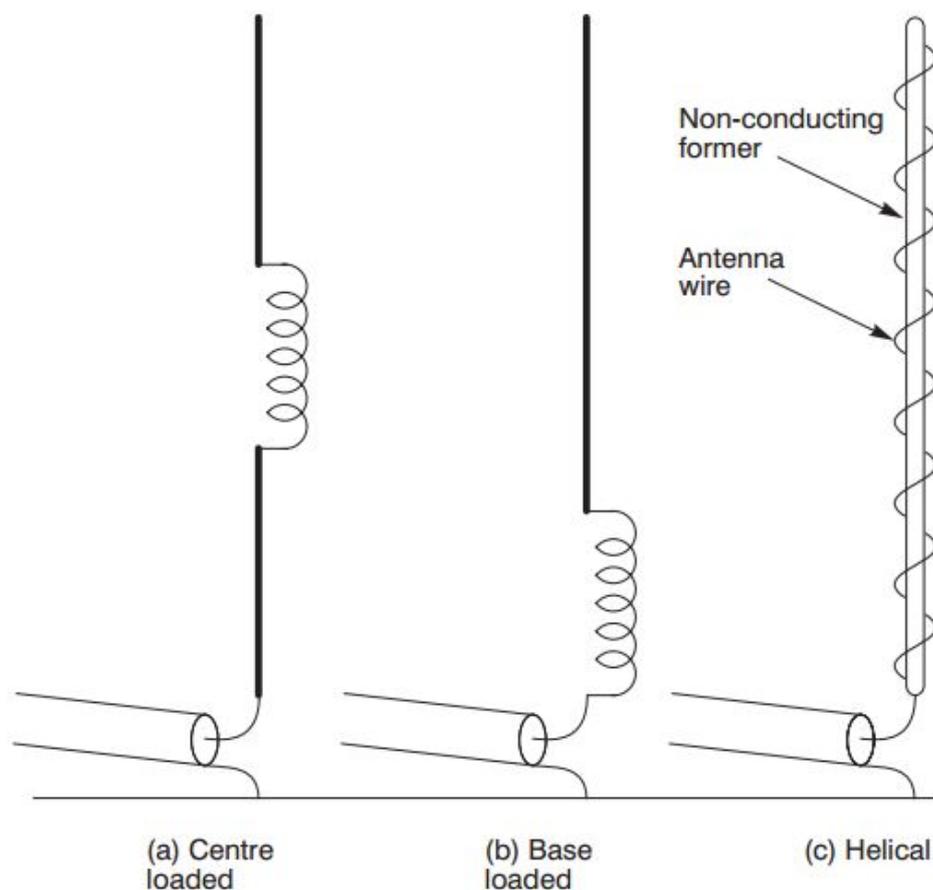


Figura 1.2: Ejemplos de alargamiento eléctrico de antenas cortas utilizando el efecto inductivo[15, cap. 4].

¹Serie de submarinos nucleares de ataque de la Armada de los Estados Unidos construidos entre 1972 y 1996.

$$\lambda [m] = \frac{c [m/s]}{f [s^{-1}]}, \text{ donde } c = 299.792.458 \text{ m/s (velocidad de la luz en el vacío)} \quad (1.1)$$

Estos resultados convierten al submarino en un objeto aislado de las tecnologías de telecomunicación convencionales que incorporan el resto de equipos aéreos y terrestres, dado que los vehículos marítimos de superficie como los barcos pueden emplear medios de transmisión-recepción acústicos para comunicarse con el submarino bajo el agua, como se muestra en la Figura 1.3.



Figura 1.3: Esquema que muestra las tecnologías de comunicación típicas instaladas en los vehículos.

En la Figura 1.3 se ilustra como las tecnologías típicas basadas en radio no consiguen una comunicación directa con el submarino, ya que las longitudes de onda que emplean no consiguen atravesar la interfaz acuática con la suficiente intensidad como para penetrar hasta una profundidad relevante.

A la vista de que es inviable transportar antenas de decenas de kilómetros para poder realizar una comunicación directa con un submarino, un estudio del MIT (*Massachusetts Institute of Technology*) realizado por *Francesco Tonolini* y *Fadel Adib* [2], propone una alternativa. En esta publicación se busca una solución que no suponga emplear una antena desmesurada y que **no comprometa el secreto de la ubicación del**

submarino. Para lograr una comunicación de este tipo se confía en la elevada capacidad energética y recursos tecnológicos de los submarinos, dónde el submarino consigue rizar la superficie del agua transmitiendo ondas acústicas con un increíble despliegue de potencia. Esta idea se ilustra en la Figura 1.4.

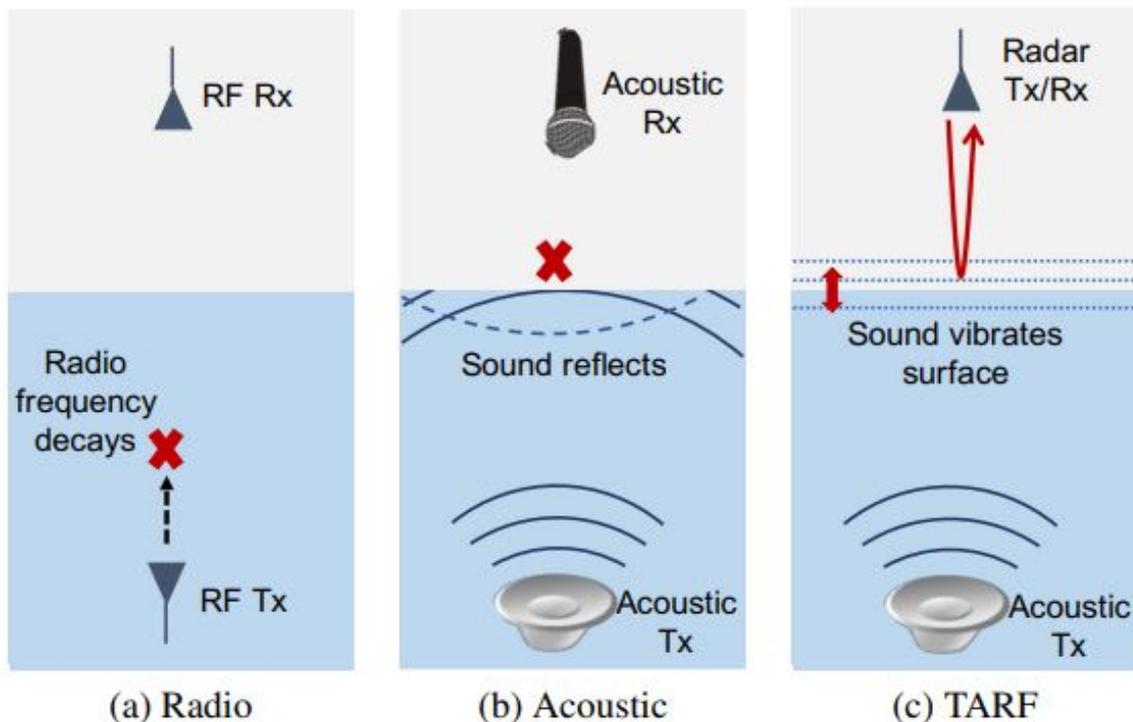


Figura 1.4: (a) Muestra que un transmisor de radio no puede comunicarse porque las señales de radio sufren demasiado amortiguamiento en el agua. (b) Muestra que las señales acústicas se reflejan en la superficie del agua. (c) Muestra que un receptor TARF (*Translational Acoustic-RF*) emplea un radar para detectar en la superficie vibraciones causadas por ondas de presión acústica y las emplea para decodificar el mensaje.

Este sistema de comunicación propuesto por el MIT emplea un complejo sistema de comunicación inalámbrica entre un objeto aéreo y uno subacuático. Para ello implementa un transmisor sumergido en el agua (TARF) que envía paquetes de información como señales acústicas comprendidas entre los 100 y los 200 Hz, que es el rango típicamente utilizado por los submarinos y AUVs (*Autonomous Underwater Vehicle*) para las comunicaciones subacuáticas debido a su baja atenuación y largo recorrido de distancias bajo el agua[16, pág. 435–440]. La señal acústica emitida viaja como una onda de presión por el agua que golpea con la superficie e induce oscilaciones proporcionales a la onda de presión emitida por el TARF. En el otro extremo de la comunicación se localiza un receptor TARF, que consiste en un radar de ondas milimétricas FMCW (*Frequency Modulated Carrier Wave*). Este radar emite una señal de banda ancha (centrada en torno a los 60 GHz) de la que luego mide su reflexión con la superficie del agua. Al reflejarse la señal se ve afectada en la fase debido a las vibraciones de la superficie del agua ocasionadas por la señal acústica transmitida por el submarino. El receptor recoge estos cambios de fase y los demodula para recuperar la información transmitida

por el submarino. En la Figura 1.5 se ilustra el funcionamiento de este sistema. Es interesante resaltar que **este sistema solo implementa un enlace de subida** (del objeto sumergido al objeto aéreo).

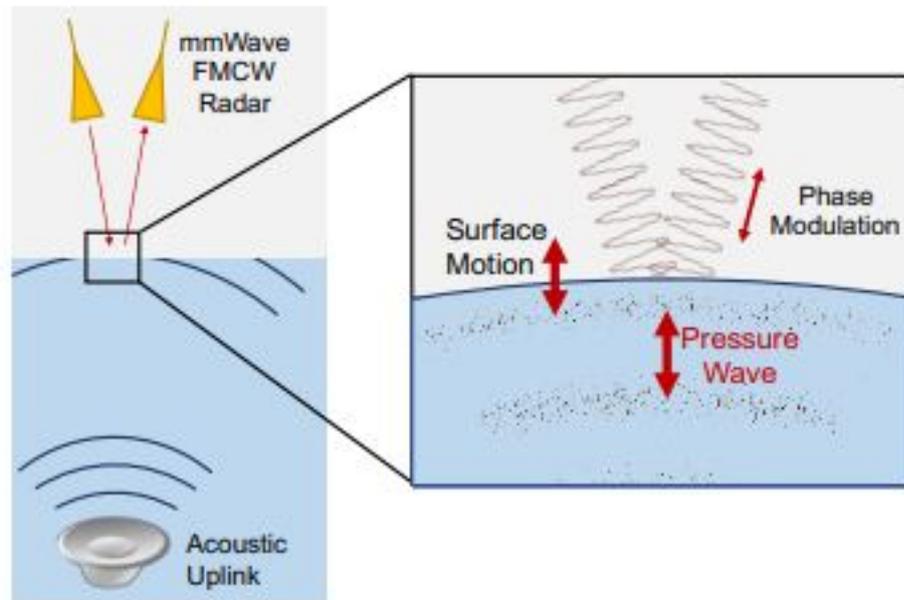


Figura 1.5: Las vibraciones superficiales se traducen en una modulación de fase. La fase de la reflexión inalámbrica cambia con vibraciones superficiales diminutas[2, pág. 4].

Inconvenientes del sistema actual:

- Solo ofrece canal de recepción. El submarino sigue sin poder recibir datos de objetos aéreos o terrestres mientras se encuentra sumergido.
- Es un sistema complejo y caro.
- Difícil seguimiento desde el avión. Tanto localizar la comunicación como mantener el láser en la zona de emisión.
- Clutter con el rizado de las olas. La comunicación depende del clima y del estado del oleaje.
- Graves efectos sobre la vida marina debido a las altas potencias de transmisión a las que se ve obligado a operar el submarino.

1.1. Motivación

La idea que impulsa este trabajo nace tras consultar una publicación en una página web de divulgación para ingenieros electrónicos llamada *Electronic Design* [1], en ella se encuentra un resumen sobre unos experimentos llevados a cabo por dos miembros del MIT *Francesco Tonolini y Fadel Adib* [2] (desarrollado en la introducción). La publicación trata sobre una serie de pruebas llevadas a cabo para resolver el problema que surge al intentar establecer una comunicación que atraviesa dos medios de propagación distintos, en este caso, entre el aire y el agua. La investigación que se desarrolla a lo largo de la publicación se centra en no comprometer la posición de los submarinos cuando estos tienen necesidad de comunicarse con un objeto exterior (p.ej: un avión, un dron, etc).

Un escenario de comunicación entre dos medios se da también en los enlaces satelitales, en los que la señal pasa de la atmósfera terrestre al vacío y donde los principales problemas que degradan la transmisión se localizan en el mismo medio de propagación (p.ej: por distancia, interferencias solares o con otras emisiones, reflexiones y cambio de polarización por la ionización de la atmósfera, etc...); sin embargo, la situación que vamos a afrontar presenta graves problemas tanto en la propagación a través del medio (en este caso será el agua), como problemas para atravesar la interfaz que separa ambos medios. Un escenario aire-agua supone un reto de gran envergadura, ya que un cambio de medio con índices de refracción tan dispares, supone un desafío para la transmisión de datos, debido a que la onda que los transmite se verá severamente afectada en potencia y calidad. Este tipo de comunicación despierta el interés por conocer que tipo de tecnología puede resolver esta situación y si los conocimientos adquiridos durante la carrera son suficientes como para entenderla y mejorarla en la medida de lo posible.

Una vez enmarcado el problema de este sistema de comunicación, se empieza a investigar sobre los submarinos y sobre que tipo de tecnología disponen. Se estudia la publicación *SUBMARINE FORCE STRATEGIC PLAN, "Submarine Communications Master Plan, 1995"* [14] y de ella se extrae información de utilidad para conocer las limitaciones y los puntos fuertes que presentan los submarinos como: tipo de modulaciones que utilizan, frecuencias que usan para comunicarse, estructura y distribución de las antenas de que disponen, etc, tal y como se ilustra en la Figura 1.6 y según las tablas que se recogen en el Anexo E.

Tras consultar publicaciones sobre las tecnologías estudiadas para resolver este tipo de comunicaciones, se hace evidente el gran potencial de los submarinos y se comienza a investigar como emplearlo para resolver este complicado enlace de comunicación aire-agua. A raíz de la complejidad tecnológica que supone el método actual propuesto por el MIT (en el cual se hace una lectura de un rizado provocado por el submarino en la superficie del agua mediante un láser instalado en el avión) y a la vista de que ni siquiera se consigue un enlace bidireccional, surge la inspiración de este TFG, que busca dar una solución con un sistema de comunicación en el que ambos extremos puedan transmitir información e implementarlo de una forma más sencilla, eficiente y robusta.

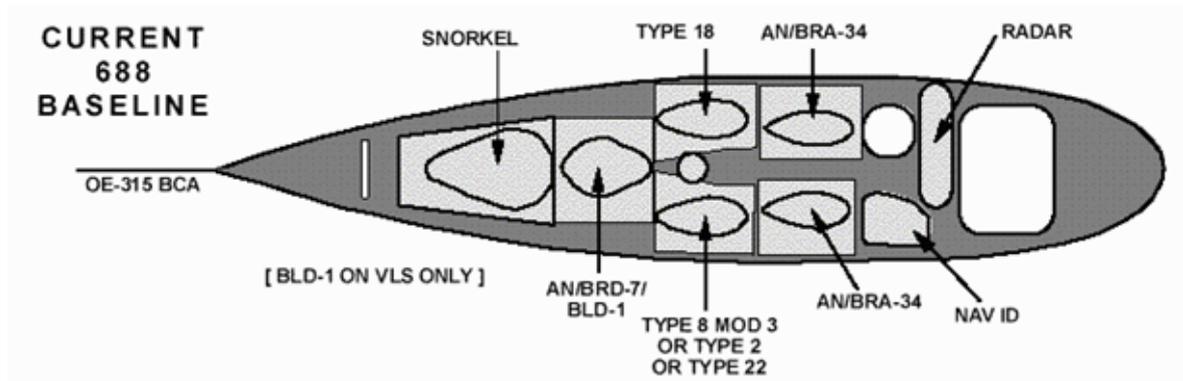


Figura 1.6: Distribución de antenas en un submarino clase 688.

Este proyecto supone la oportunidad de cubrir una necesidad personal de afianzar conocimientos sobre todos los campos de las telecomunicaciones estudiados durante la carrera, supliendo así la inquietud de terminar mis estudios sin poner a prueba las habilidades adquiridas en lenguajes de programación, capacidad de desarrollo de montajes electrónicos y conocimientos sobre sistemas radio y propagación de ondas electromagnéticas.

1.2. Objetivos

En este trabajo se busca establecer un enlace aire-agua entre dos nodos de comunicación situados en medios de propagación distintos. Para ello, se pretende diseñar un sistema que facilite el paso de la señal de un medio a otro, en este caso, se opta por implementar un sistema de boyas, como se muestra en la Figura 1.7. Estas boyas deben llevar un sistema de comunicación mediante ondas acústicas para el medio acuático y un sistema radio para el medio aéreo, además de establecer un vínculo entre ambos. Para lograr un montaje seguro, el sistema debe estar completamente impermeabilizado para evitar la intrusión de agua que pueda dañar los componentes electrónicos. No obstante, el material que lo componga debe permitir el paso de señales acústicas y electromagnéticas (en este trabajo solo se impermeabilizan los componentes necesarios por falta de recursos). En el desarrollo del trabajo se pretende buscar un compromiso económico entre la electrónica y la eficiencia en el montaje; por consiguiente se buscarán soluciones ingeniosas que sustituyan sistemas más complejos y económicamente inaccesibles.

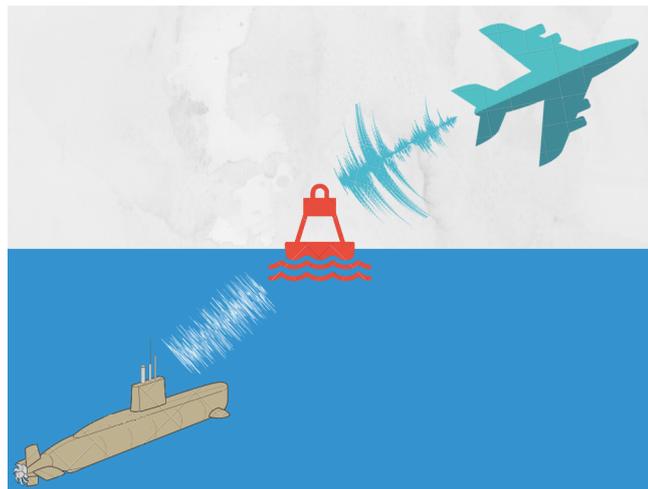


Figura 1.7: Comunicación entre avión y submarino con el sistema de boyas.

Este sistema incluirá además un módulo GPS. Este módulo será útil para mantener localizada la boya. Si en el sistema se dispone de varias boyas, se podrá geolocalizar al transmisor subacuático mediante la triangulación de las transmisiones.

Así pues, una vez explicado de modo general el problema que se pretende resolver y la solución que se va a emplear, pasamos a enumerar de una forma ordenada los objetivos concretos que este trabajo fin de grado pretende conseguir:

1. **Sobre competencias adquiridas o reforzadas por el alumno:** realizar un proyecto que incluya competencias de gran parte de los campos estudiados en la carrera, con la intención de afianzar los conocimientos previos sobre estos.
 - (I) Adquirir habilidades de diseño y construcción de circuitos electrónicos funcionales.

- (II) Ampliación de los conocimientos sobre instrumentación de laboratorio.
 - (III) Desarrollo y dominio software en diversos entornos de programación.
 - (IV) Dominio de herramientas de edición de textos y gráfica para la elaboración de documentos técnicos, diagramas y figuras en proyectos.
2. **Diseño de un sistema de comunicación aire-agua para enlazar en modo *full-duplex* un avión (u otro elemento situado fuera del agua) y un submarino (u otro elemento situado bajo el agua).**
- (I) Estudiar otros tipos de comunicaciones y establecer una opción de mejora con un sustituto económico.
 - (II) Diseñar una estrategia que permita establecer una comunicación *full-duplex* empleando el sistema.
 - (III) Elegir las tecnologías de transmisión adecuadas y las tecnologías electrónicas más convenientes para el desarrollo y construcción de un prototipo del sistema.
3. **Construcción, comprobación y ensamblado de todos los subsistemas (unidades funcionales) necesarios para completar el prototipo de sistema final.**
- (I) Diseño y construcción de los subsistemas para el prototipo (diseñar y construir un transceptor radio funcional, diseñar y construir un sistema transceptor subacuático económico y diseñar y construir el resto de los subsistemas de apoyo necesarios).
 - (II) Desarrollo del software necesario para el funcionamiento del sistema.
 - (III) Efectuar diversas pruebas de funcionamiento para comprobar la viabilidad del sistema y de las soluciones tecnológicas escogidas.

1.3. Estructura del documento

A lo largo del documento se explicarán gradualmente las tecnologías empleadas. En primer lugar se dará un enfoque global de la solución que se ha ideado para una comunicación aire-agua. Después, tras un punto de vista teórico sobre las comunicaciones aéreas y subacuáticas, se estudiarán las tecnologías complementarias que formarán parte de los módulos principales y las distintas opciones contempladas, donde se explica cuáles han sido escogidas y descartadas por temas de viabilidad y eficiencia. Una vez definidos los bloques adicionales del proyecto se explica el hardware y el software que se emplea en la codificación de la información a transmitir. Explicadas las tecnologías que integra el proyecto, se exponen los módulos principales que componen el montaje final junto con sus diagramas de funcionamiento. Por último, se exponen las pruebas realizadas, las conclusiones extraídas de este trabajo y las líneas de mejora. Además, en este trabajo los **Anexos** se han definido como una guía rápida que facilite la consulta del lector en cualquier momento sobre los componentes electrónicos empleados.



Figura 1.8: Estructura del documento.

Capítulo 2

Elección y explicación de las soluciones empleadas

2.1. Visión general

En este apartado se explica el principio general de funcionamiento del sistema completo de comunicación entre avión y submarino (o bien aéreo - subacuático). La justificación de la elección de los medios de transmisión utilizados y de las tecnologías concretas empleadas se realizará en los siguientes apartados. El sistema en su conjunto cuenta con tres bloques fundamentales para la comunicación. El primer bloque emula al sistema de comunicación aerotransportado, donde se dispondrá de un transceptor radio y el cual se explica más en detalle en la sección 3.1. El bloque final es el submarino y cuenta con un sistema transceptor acústico. Este bloque se analiza en detalle en la sección 3.3. El bloque intermedio y encargado de ser el nexo del primer y último bloque es la boya. La boya se encarga de transmitir la comunicación vía radio con el equipo aéreo y vía acústica con el submarino como se explica en la sección 3.2.

Tras comentar brevemente las características de los enlaces y los nodos del sistema de comunicación, se procede a explicar el funcionamiento básico de la transmisión de datos. Para iniciar la comunicación el avión envía un identificador al submarino (este sistema de identificación no se ha implementado en el código de este proyecto para no complicar más la transmisión de datos). Si el identificador es correcto se comienza con la transmisión de datos entre ambos extremos a través de la boya. El submarino además podrá pedirle la ubicación GPS a la boya (ubicación que la boya obtendrá con el módulo GPS que lleva instalado). La información proporcionada por el GPS podrá ser utilizada por el submarino para saber su ubicación (para mayor precisión se podría realizar triangulación utilizando los datos de ubicación de múltiples boyas). Las peticiones de GPS¹ y los mensajes se escriben en código ASCII y se codifican en Morse en todas las comunicaciones. La codificación se realiza de manera automática por un programa para cumplir con los requisitos de la comunicación. El código Morse resultante se traduce a dos tonos de frecuencias distintas para definir el punto y la raya; de este modo, el receptor submarino distinguirá el código según la frecuencia que

¹De la codificación y transmisión de la ubicación GPS se encargará íntegramente la boya con el equipo electrónico que esta incluye.

reciba en cada instante de tiempo. En la Figura 2.1 se ilustra el funcionamiento general.

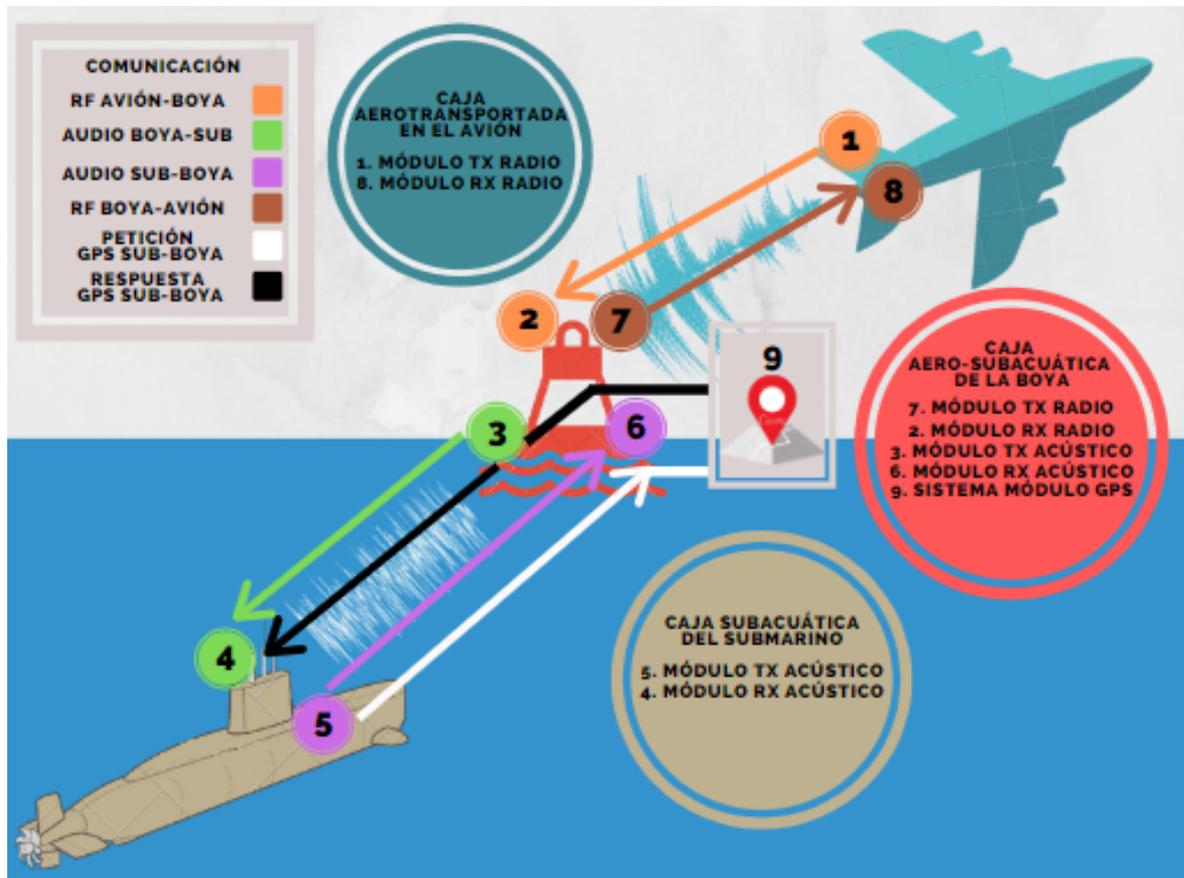


Figura 2.1: Esquema del funcionamiento global del sistema de comunicación basado en boyas.

La Figura 2.1 explica el funcionamiento básico del sistema. Según los trayectos:

- **Trayecto 1-2 (naranja):** comunicación radio entre el módulo transmisor radio instalado en la caja del avión y el módulo receptor radio instalado en la boya.
- **Trayecto 3-4 (verde):** comunicación mediante ondas acústicas entre el módulo transmisor acústico instalado en la caja de la boya y el módulo receptor acústico instalado en la caja del submarino.
- **Trayecto 5-6 (morado):** comunicación mediante ondas acústicas entre el módulo transmisor acústico instalado en la caja del submarino y el módulo receptor acústico instalado en la caja de la boya.
- **Trayecto 5-6 (blanco):** comunicación mediante ondas acústicas entre el submarino y la boya. Transmite la petición de la ubicación GPS.
- **Trayecto 3-4 (negro):** comunicación mediante ondas acústicas entre la boya y el submarino. Transmite la respuesta con la ubicación GPS pedida por el submarino a la boya.

- **Trayecto 7-8 (marrón):** comunicación radio entre el módulo transmisor radio instalado en la caja de la boya y el módulo receptor radio instalado en la caja del avión.

2.2. Elección de los canales de transmisión adecuados y de las tecnologías empleadas

En este apartado se explicarán los posibles mecanismos físicos de comunicación que podrían emplearse para llevar a cabo el sistema propuesto (vía luz, radio o acústico) tanto en medio aéreo como en el subacuático y se justificará la elección realizada anteriormente. Además, se discutirán diferentes opciones tecnológicas para la materialización de dicha comunicación (tipos de transmisores de radio y soluciones para el enlace acústico), eligiendo justificadamente algunas de ellas. La concreción del uso del hardware utilizado y del software necesario para hacerlo funcionar, cumpliendo las especificaciones del sistema marcadas en el apartado de objetivos, se expondrá después de este apartado.

Para lograr este sistema de comunicación entre dos medios se han estudiado y analizado diversas implementaciones posibles. Buscando siempre la solución óptima se exponen los sistemas desarrollados, tanto los que han sido seleccionados finalmente, como los que han sido descartados por temas referentes a su viabilidad y eficiencia, para dar lugar a la tecnología más adecuada para cumplir con una labor específica en este proyecto.

2.2.1. Tecnologías subacuáticas

Se comienza analizando sistemas de comunicaciones subacuáticas, ya que supone el mayor reto tecnológico. Entre los múltiples sistemas que se han desarrollado se centra el estudio en dos implementaciones principales. La primera se basa en transmisión de información mediante ondas acústicas, las cuales presentan unas características idóneas para propagarse largas distancias por el agua. La segunda implementación se basa en transmisión mediante ondas electromagnéticas, tanto dentro como fuera del espectro visible, de las cuales se analizaron en detalle los sistemas actuales disponibles. Rápidamente **se descartaron los sistemas radio**, dado que las implementaciones actuales precisan antenas terrestres demasiado grandes como para ser aerotransportadas (debido a las largas longitudes de onda que se emplean para que la onda pueda propagarse suficiente distancia en el agua).

Descartados los sistemas radio de longitudes de onda elevadas se procede a estudiar qué es lo que sucede cuando una onda electromagnética en otras longitudes de onda alcanza la interfaz entre los medios. Cuando la onda incide con un ángulo superior al ángulo crítico θ_c , esta se refleja en su totalidad (véase la particularización de la *Ley de Snell* en (2.1)) y no es capaz de pasar al segundo medio. Para ángulos inferiores a θ_c , se transfiere parte de la potencia de la onda al segundo medio, mientras que el resto se refleja y solo para una incidencia ortogonal al medio se consigue una transferencia casi

total de potencia.

$$\theta_c = \arcsin(n_2/n_1) \quad (2.1)$$

Donde n_1 es el índice de refracción del agua, n_2 el índice de refracción del aire (con $n_1 > n_2$) y donde θ_c es el ángulo crítico. Para el caso del agua salada el índice de refracción varía según la presión, la salinidad, la cloridad, la temperatura y la longitud de onda. Los valores típicos varían según las tablas recogidas en el estudio de ROSWELL W. AUSTIN Y GEORGE HALIKAS[17]. Para 10°C, salinidad de 34,998 ‰, cloridad de 19,373 ‰ y a presión atmosférica, se obtiene un $n_1 = 1,35040$, para el caso del aire tenemos un $n_2 = 1,00029$ (sabiendo que el índice de refracción del vacío es $n_0 = 1$). Lo que nos da un $\theta_c = 47,79416^\circ$, ángulo de incidencia a partir del cual se producirá reflexión total de cualquier tipo de onda electromagnética. Y a pesar de que parte de la potencia de la onda se transfiera para ángulos inferiores al θ_c , se tendrá que lidiar con la fuerte atenuación del agua salina. Otro problema debido al cambio de medio es la refracción de la onda, que representa la fracción de potencia de señal que es capaz de transmitirse al segundo medio y la cual sufre una desviación respecto del ángulo de incidencia. Es cierto que esto es un problema únicamente para un haz muy directivo (como podría ser el de un láser), dado que para la incidencia de una onda plana la información llegaría a pesar de la refracción (suponiendo que la atenuación no acabase con su propagación). En este punto se puede decir que utilizar una comunicación radio íntegra que atravesase de un medio al otro en la banda de frecuencias en cuestión no es viable y se procede a estudiar la propagación si el foco de emisión se encuentra ya inmerso en el agua.

Tras analizar las posibilidades se centra el estudio en la banda de frecuencia más prometedora para una comunicación subacuática, las ondas electromagnéticas en el espectro visible generadas directamente dentro del agua, evitando así los problemas de reflexión y refracción anteriormente comentados. Se muestra en la Figura 2.2 la idea de un enlace óptico directo y del enlace que se emplearía con nuestro sistema basado en boyas.

Para implementar el sistema de comunicación en estas longitudes de onda se pensó en utilizar LEDs RGB y fotodiodos, es decir, uso de tecnología VLC (*Visible Light Communication*). Para analizar la viabilidad de estos sistemas se consultan las publicaciones de CHAO WANG, HONG-YI YU, YI-JUN ZHU, "A long distance underwater visible light communication system with single photon avalanche diode"[13], GROBE ET AL, "High-Speed Visible Light Communication Systems"[10] y MODELOS DE PROPAGACIÓN DE SEÑALES ACÚSTICAS EN ENTORNOS SUBACUÁTICOS[9] del departamento de electrónica de la Universidad de Alcalá. Estas publicaciones arrojan resultados poco esperanzadores en cuanto al empleo de ondas electromagnéticas para cumplir con las necesidades de nuestro sistema de comunicación, tales como que las ondas electromagnéticas sufren atenuaciones del orden del **dB/m para una frecuencia de 30 KHz**, mientras que una onda acústica sufre una atenuación del orden de **dB/Km para esa misma frecuencia**. Por otro lado, las frecuencias ópticas se ven altamente

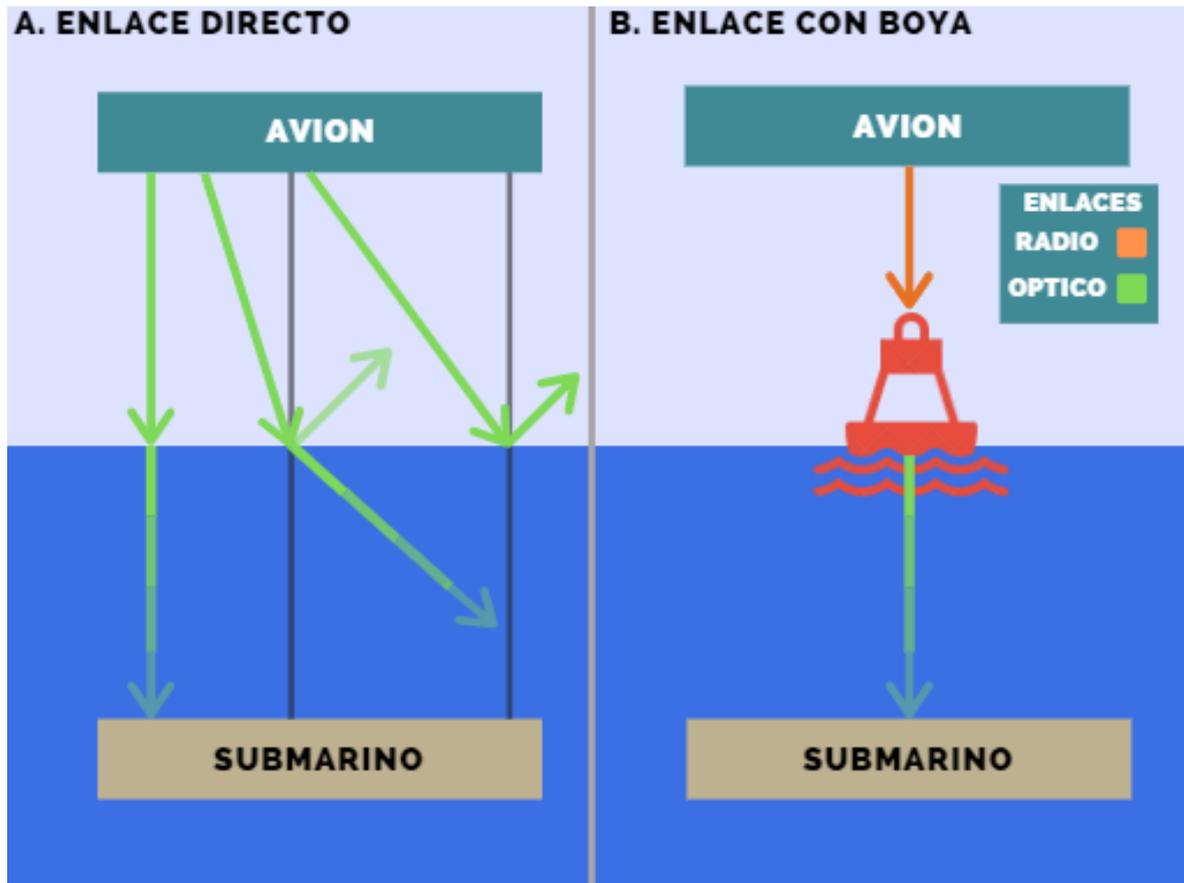


Figura 2.2: Se muestran dos tipos de enlace utilizando tecnología óptica. En el enlace A se muestran tres situaciones de incidencia; incidencia normal con transferencia total de potencia, incidencia con un $\theta < \theta_c$ e incidencia con $\theta > \theta_c$ con reflexión total. En el enlace B se muestra como sería la tecnología óptica implementada con nuestro sistema boya.

afectadas por el *scattering* y la dispersión, hasta el punto en el que el agua se vuelve opaca para distancias superiores a los 200 m [9]. Este resultado es poco favorable para las características que se precisan en este proyecto, ya que la distancia a la que consigue propagarse la información es un factor relevante en nuestro sistema de comunicación. Se pretende proteger la ubicación del submarino y si este necesita estar próximo a la boya para comunicarse puede correr el riesgo de delatar su posición.

En las publicaciones ya mencionadas ([9],[10] y [13]) se trata un tema que es de utilidad en este proyecto, la viabilidad de establecer una comunicación óptica inalámbrica en un medio subacuático. Los experimentos de estas publicaciones se realizan para las longitudes de onda en el espectro visible menos afectadas por las pérdidas introducidas por el agua (longitudes de onda comprendidas entre el azul y el verde); sin embargo, a pesar de ser las longitudes de onda más favorables para una propagación subacuática, no se obtiene un rendimiento equiparable al de las comunicaciones con ondas acústicas[10, pág. 61, *Future Visions and Main Applications*]. Según [13, pág. 11, *Conclusion*] las simulaciones estiman una distancia máxima de penetración de las ondas electromagnéticas de **500 metros**(Figura 2.3), la cual no es una distancia com-

parable al orden de kilómetros que pueden recorrer las ondas acústicas y además solo se obtiene para simulaciones en las que el emisor óptico transmite a **100 W** de potencia, lo cual si bien es viable en un submarino no se corresponde con un diseño de bajo consumo como el que debe integrar la boya, ni a un diseño económico, dado que para esas profundidades se tendrían que emplear láseres muy directivos. Por este motivo se descartan ideas como la transmisión de información utilizando transmisores y receptores ópticos en este proyecto; dado que a pesar de ser útil en un prototipo a pequeña escala, no cumpliría con los requisitos de una implementación real aún utilizando la tecnología más puntera actual.

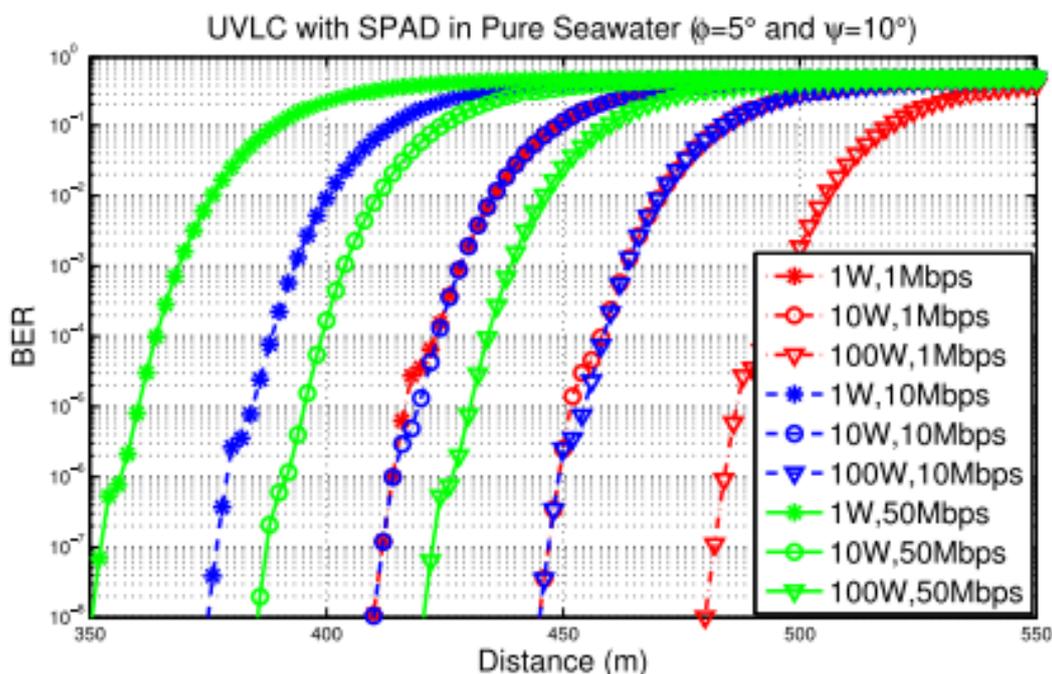


Figura 2.3: Penetración de las ondas electromagnéticas en el agua salada[13].

En la Figura 2.3, ϕ es el ángulo formado con el eje perpendicular al LED para el cual la intensidad luminosa del haz emitido se reduce al 50%, ψ el ángulo de visión del fotodiodo *SPAD* (*Single Photon Avalanche Diode*). Las pruebas se realizan para potencias de transmisión de $P = 1W$, $10W$ y $100W$, para distintas velocidades de transmisión y se enfrentan con la *BER* (*Bit Error Rate*) obtenida. Se puede observar que para profundidades superiores a 500 m la *BER* es demasiado elevada.

Llegados a este punto se descarta la opción de utilizar las ondas electromagnéticas para una comunicación subacuática tras los estudios y problemas expuestos, la complejidad que supone establecer el enlace de comunicación entre los extremos y la baja eficiencia que presenta en el medio acuático. En su lugar se emplearán las ondas acústicas, dado que presentan una respuesta mucho más prometedora en este medio de propagación.

Elegido un sistema de transmisión acústico, se consulta la publicación de *MODELOS DE PROPAGACIÓN DE SEÑALES ACÚSTICAS EN ENTORNOS SUBACUÁTICOS*[9]

para la comprensión de la propagación del sonido a través del agua. De esta publicación se pueden extraer datos interesantes como la variación que sufre la velocidad de propagación del sonido según la presión(Figura 2.4) y la temperatura(Figura 2.5) del agua. Este es un dato que puede utilizarse en un sistema más complejo para diseñar estrategias en la transmisión, por ejemplo para calcular los retardos de la señal en un sistema de triangulación de posición empleando múltiples boyas. Se adjuntan dos imágenes extraídas del documento para ilustrar el fenómeno.

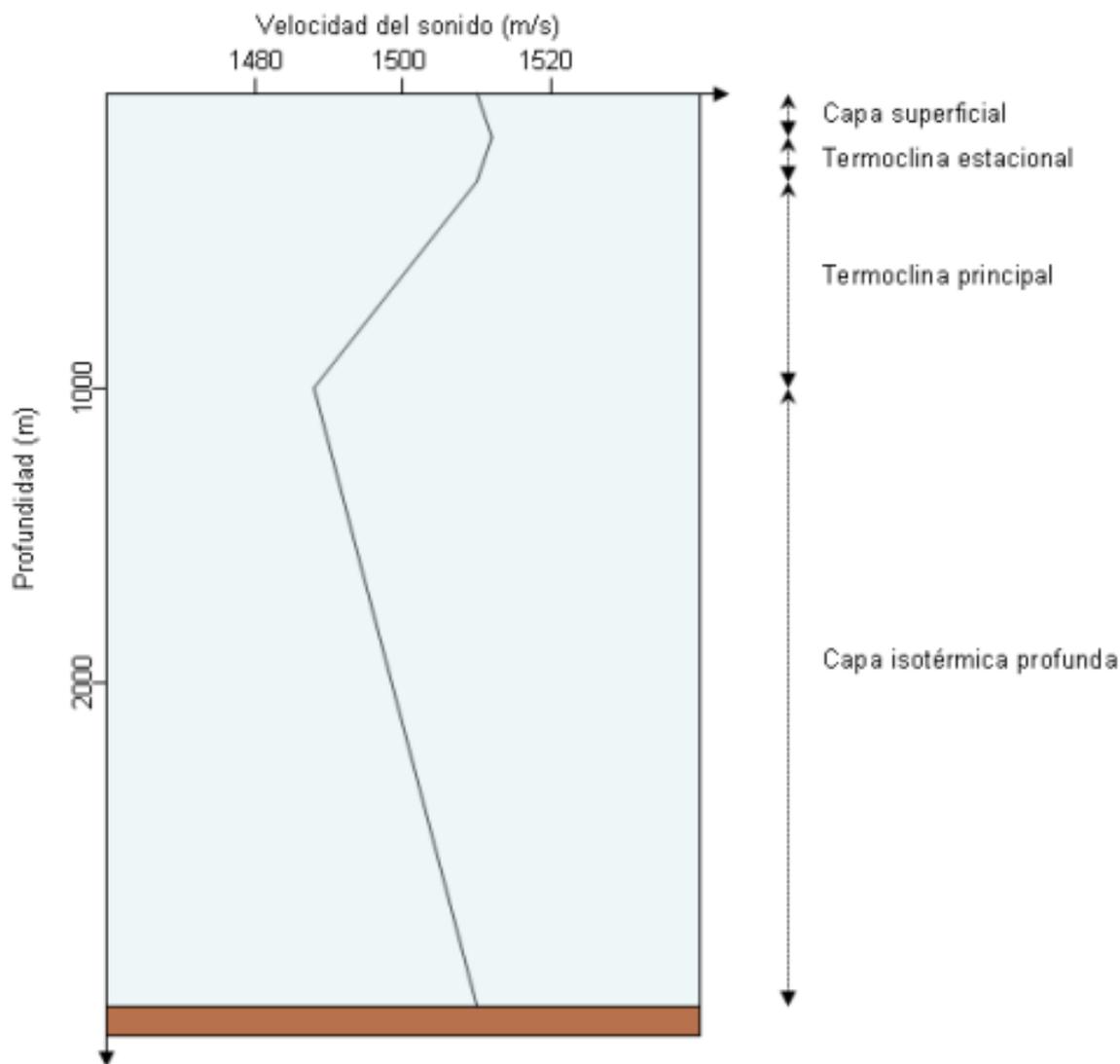


Figura 2.4: Perfil típico de la velocidad de la propagación del sonido para latitudes medias[9].

A la vista de las imágenes se puede comprender mejor la importancia que tiene la temperatura y la presión en la velocidad de propagación del sonido. Se puede observar que estos parámetros son factores importantes a tener en cuenta, sin embargo, el dato más valioso que se recoge en esta publicación para este proyecto se encuentra en la ecuación del coeficiente de absorción del agua de *Francois y Garrison* (2.2). Esta ecuación fue desarrollada a principios de los años 80 y es válida para frecuencias de

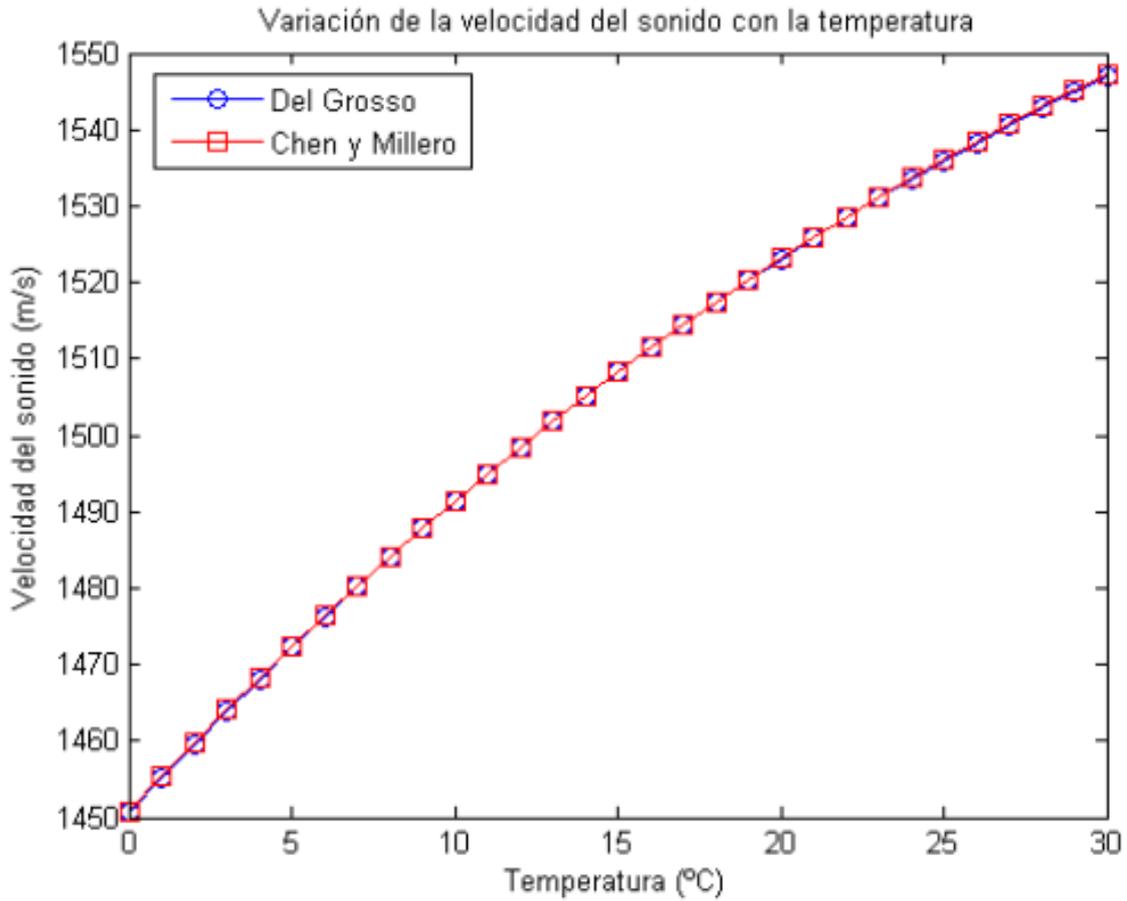


Figura 2.5: Variación de la velocidad del sonido obtenida por las ecuaciones de Del Grosso y Chen y Millero con la temperatura[9].

100 Hz a 1 MHz.

$$\alpha = \underbrace{\frac{A_1 P_1 f_1 f^2}{f^2 + f_1^2}}_1 + \underbrace{\frac{A_2 P_2 f_2 f^2}{f^2 + f_2^2}}_2 + \underbrace{A_3 P_3 f^2}_3 \quad (2.2)$$

Donde α tiene unidades de dB/Km y f es la frecuencia de la señal en KHz. El primer término proviene de la contribución del ácido bórico, el segundo de la contribución del sulfato de magnesio y el tercero de la viscosidad del agua, f_1 y f_2 son las frecuencias de relajación del ácido bórico y el sulfato de magnesio respectivamente, en KHz, y los coeficientes A_i y P_i pueden depender del pH, la salinidad, la profundidad, la temperatura y la velocidad del sonido [9, pág. 7].

Mediante la ecuación de *François y Garrison* 2.2 se genera la Figura 2.6. Para ello se fija un pH de 8, salinidad de 35 ‰, temperatura de 5°C y una profundidad de 0 m.

Tras observar la Figura 2.6 se puede sacar una conclusión a la hora de elegir las frecuencias de transmisión en el sistema. Para transmitir a grandes distancias del orden de varios kilómetros se tendrían que emplear decenas de KHz (de lo contrario la absor-

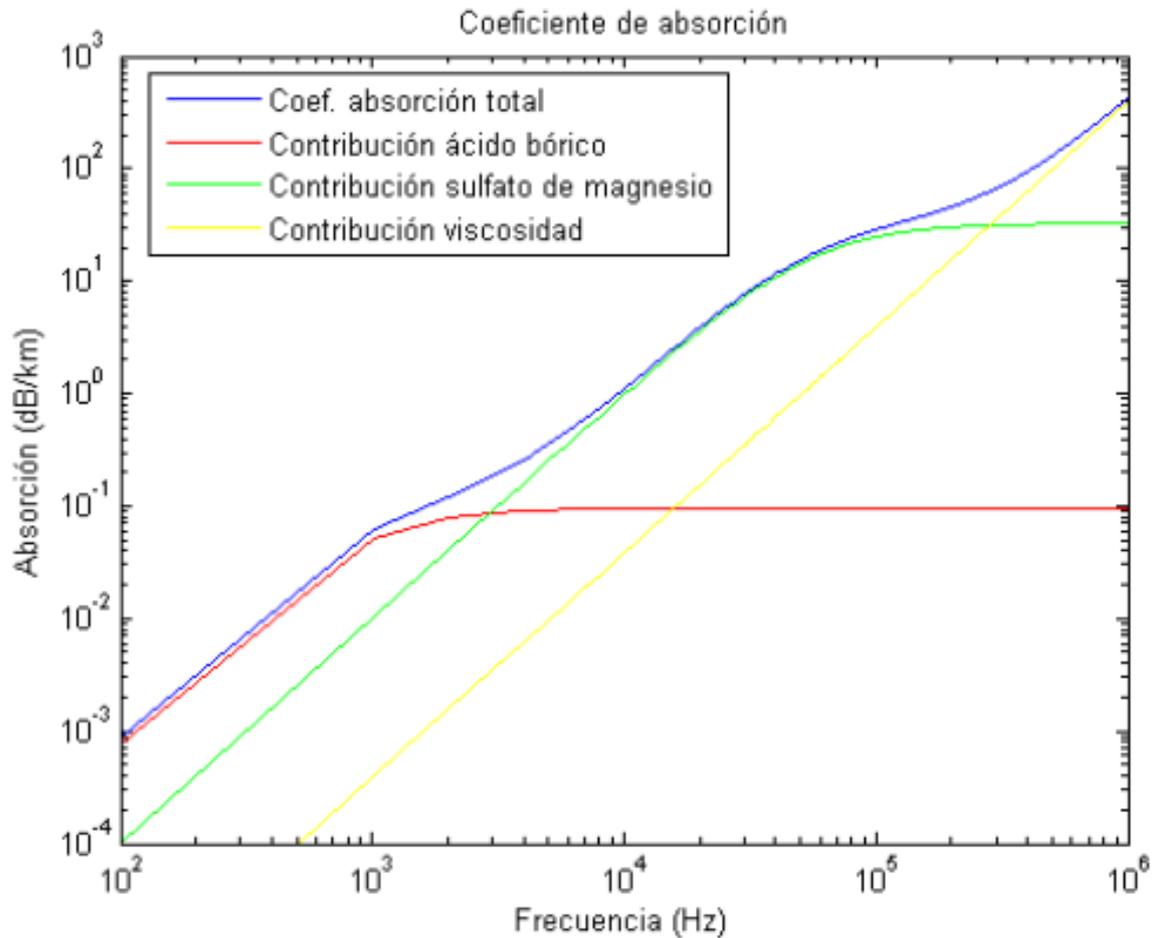


Figura 2.6: Variación del coeficiente de absorción con la frecuencia[9].

ción es demasiado elevada). Para transmisiones cercanas (menos de 1 Km) se pueden emplear centenas de KHz.

El empleo de altas frecuencias tiene inconvenientes en lo que a pequeños obstáculos respecta, como pueden ser las burbujas generadas por las olas o cualquier cuerpo con un tamaño parejo a la longitud de onda. Esto significa que los rebotes a estas frecuencias en la superficie suponen grandes pérdidas en la señal. Sin embargo, para bajas frecuencias nos encontramos con que obstáculos más grandes e incluso el ruido generado por el oleaje interfieren en la calidad de la transmisión; no obstante, las ondas de baja frecuencia sufren menos interferencia con los rebotes en la superficie del agua. Dependiendo de la aplicación se pueden emplear unas u otras. Para aplicaciones militares como puede ser el sonar se emplean tanto bajas como altas frecuencias, desde un rango audible (20 Hz a 20 KHz) a un rango de alta frecuencia que comprende (80 KHz a 350 KHz). La transmisión en alta frecuencia para estas aplicaciones puede ser útil para ganar resolución en los objetos en los que rebota la onda (esto se debe a que si el tamaño del objeto es muy pequeño en comparación a la longitud de onda este pasará desapercibido para el sonar).

Analizada la teoría sobre propagación acústica en el agua se procede al análisis de

los distintos dispositivos disponibles para establecer una comunicación en este medio. Tanto los transmisores como los receptores han de ser impermeables. Se exploraron soluciones comerciales que por su diseño pudieran ser sumergidas sin dañarse. Los hidrófonos son componentes especializados muy caros y se intentó encontrar alguna opción de micrófono económico como corresponde a los equipos que deben ser usados en una boya desechable. Para la transmisión-recepción subacuática se emplearán entonces micrófonos y altavoces impermeabilizados. Para preparar el equipo se utiliza cualquier tipo de recubrimiento impermeable que sea lo suficientemente elástico como para poder vibrar con las frecuencias a las que se opera en el sistema y así, no reducir demasiado la sensibilidad de los receptores de audio ni la potencia de emisión de los transmisores acústicos. Los sistemas de transmisión y recepción acústica se pueden extraer de cualquier equipo electrónico que cuente con ellos (como en este caso que se han extraído de teléfonos antiguos, radios, etc...) o comprando algún módulo. Se asume entonces que en algunos dispositivos sería necesario realizar algún tipo de impermeabilización mecánica, con la consecuente pérdida de sensibilidad. Hay que remarcar el hecho de que en el submarino no existen estos problemas prácticos-económicos puesto que estas naves disponen de excelentes equipos de recepción y transmisión acústica (aunque para nuestro prototipo se empleará el mismo hardware que para la boya). Una solución comercial que se ha probado consiste en unos micrófonos capacitivos muy económicos, que cuentan con una etapa de amplificación (Figura 2.7) y cuyo uso principal es extraer el sonido de la guitarra mediante la vibración que esta experimenta en su caja al rasgar las cuerdas. Si bien es cierto que estos micrófonos desempeñaban a la perfección su papel cuando se situaban sobre la superficie que está vibrando a una determinada frecuencia, no funcionaban tan bien a la hora de recibir una onda acústica transmitida por el aire o el agua. Se pensó que en el agua, al ser un medio más denso y que propaga mejor el sonido que el aire, la onda acústica llegaría con la suficiente fuerza como para hacer vibrar significativamente el sensor piezoeléctrico del micrófono; sin embargo, a pesar de transmitir en un medio más favorable para la propagación del sonido, las placas no contaban con la sensibilidad necesaria como para recoger la señal acústica con una buena SNR (Signal to Noise Ratio o Relación Señal a Ruido).

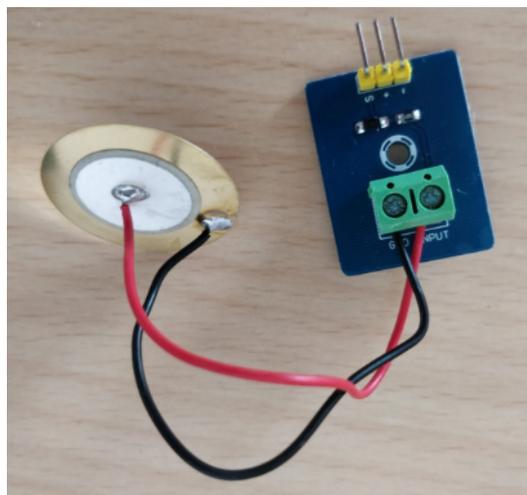


Figura 2.7: Micrófono piezoeléctrico con etapa de amplificación.

Tecnología subacuática empleada

Se utilizarán entonces para desempeñar la labor de comunicación en un entorno subacuático micrófonos y altavoces impermeabilizados extraídos de teléfonos antiguos (Figura 2.8). El micrófono es de carbón, por lo que cuenta con una cápsula con gránulos de carbón cuya resistencia es variable según como se presionen estos gránulos. Como las ondas acústicas comprimen la capsula con una determinada frecuencia la resistencia de la cápsula varía a esa frecuencia y por consiguiente varía la corriente que la atraviesa. Precisamente por estar diseñados de esta forma estos micrófonos no tiene polaridad a tener en cuenta al conectarlos a los equipos del proyecto. El altavoz es electromecánico, es decir, se basa en un electroimán que al ser alimentado con una señal analógica con una determinada frecuencia atrae una lámina metálica que hace oscilar el aire a esa misma frecuencia.

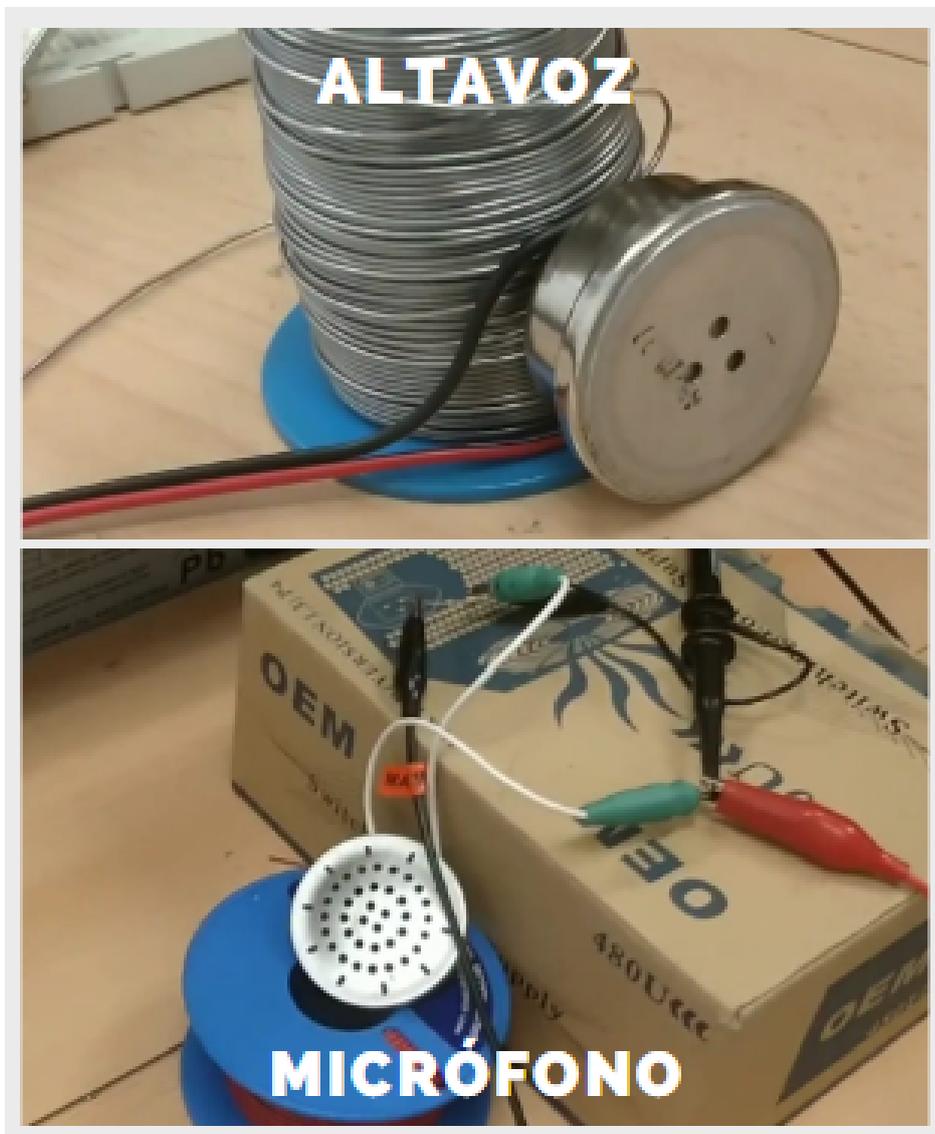


Figura 2.8: Se muestran un altavoz electromecánico y un micrófono de carbón de los empleados en la transmisión acústica subacuática.

Se escogen los altavoces y micrófonos extraídos de los teléfonos dado que ambos dispositivos son muy resistentes y toleran condiciones extremas, lo cual los convierte en excelentes candidatos para las pruebas a las que van a ser sometidos. Estos elementos acústicos no están diseñados para una transmisión subacuática y al impermeabilizarlos pierden potencia en la emisión y sensibilidad en la recepción. Por consiguiente, ambos dispositivos necesitarán de una amplificación.

En cuanto al método elegido para transmitir la información se usará una doble traducción mediante software, en la cual se pasará un mensaje escrito en ASCII por el usuario a Morse y el código Morse se traducirá por dos tonos de distintas frecuencias asociados al punto y a la raya (tal y como se ha explicado en la sección 2.1). Este método se ha escogido debido a que en un medio acuático supone un sistema de comunicación más robusto que transmitir la misma frecuencia y distinguir por la duración del tono, tal y como se emplea el código Morse donde la raya tiene una duración temporal superior a la del punto. Además, optimizando este sistema se puede transmitir a mayor velocidad dado que los tonos duran el mismo tiempo. En cuanto a las frecuencias elegidas para la transmisión se han escogido acordes a la respuesta frecuencial que presentan los dispositivos acústicos empleados. Dado que fueron diseñados para el rango audible (ya que se han extraído de teléfonos antiguos) teóricamente se pueden escoger frecuencias arbitrarias comprendidas entre los 20 Hz y los 20 KHz. En la práctica los micrófonos y altavoces no tienen una respuesta frecuencial plana y se tendrán que ajustar las frecuencias elegidas al rango óptimo de los equipos. Los dispositivos elegidos no cubren todo este rango ya que están optimizados de 300 Hz a 3400 Hz, por consiguiente **se realizará un estudio previo de la respuesta frecuencial de los componentes a emplear antes de escoger las frecuencias**². Se muestra en la Figura 2.9 un diagrama de frecuencias empleadas en cada canal para cada símbolo. Se ha procurado mantenerse en el rango óptimo de los equipos, teniendo en cuenta las posibilidades técnicas para desarrollar los montajes a cada frecuencia y la necesidad de separar los canales para evitar que se interfieran.

En la Figura 2.9 se representa el rango de frecuencias audible teórico, en el cual se distribuyen las frecuencias escogidas para nuestros dos símbolos, el punto y la raya (en el sistema Morse implementado). En ambas direcciones la frecuencia más baja se emplea para representar el punto y la frecuencia más alta para la raya. Si bien recorrer la circunferencia es barrer en frecuencias, esta se utiliza para representar un mismo nivel de tensión que es compartido por todos los símbolos, es decir, el sistema no realiza ninguna discriminación por nivel de tensión de la señal recibida. De esta forma se determina teniendo en cuenta las limitaciones de los transductores escogidos y la separabilidad de las frecuencias en recepción:

- **Frecuencia de 1800 Hz:** define el símbolo *punto* en el sistema de codificación Morse en el canal de subida (dirección avión).
- **Frecuencia de 2800 Hz:** define el símbolo *raya* en el sistema de codificación Morse en el canal de subida (dirección avión).

²Una mala elección de frecuencias para los equipos que se vayan a emplear puede provocar que el sistema no funcione correctamente

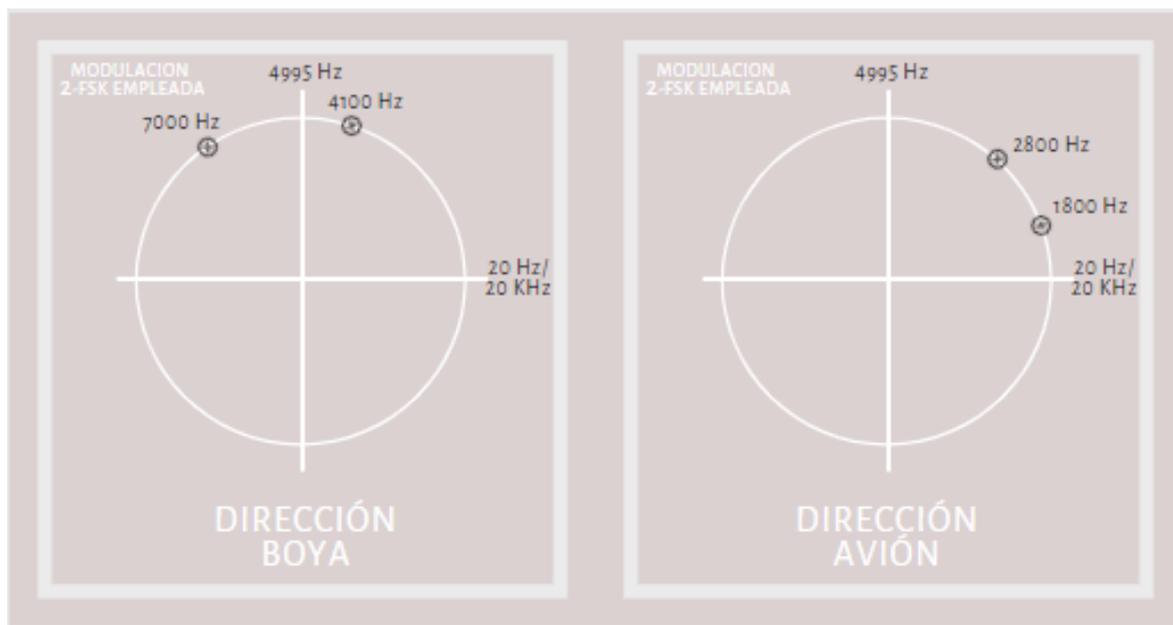


Figura 2.9: Modulación 2-FSK empleada en el enlace de subida (dirección avión) y en el enlace de bajada (dirección boya).

- **Frecuencia de 4100 Hz:** define el símbolo *punto* en el sistema de codificación Morse en el canal de bajada (dirección boya).
- **Frecuencia de 7000 Hz:** define el símbolo *raya* en el sistema de codificación Morse en el canal de bajada (dirección boya).

2.2.2. Tecnologías aéreas

En este punto por suerte se cuenta con numerosas opciones, dado que la transmisión radio a través del aire es un sistema ampliamente extendido. Para estudiar la transmisión aérea y las bandas de radiofrecuencia se consulta el libro NEWNES GUIDE TO RADIO AND COMMUNICATIONS TECHNOLOGY [15, cap. 2, 5 y 6].

Para el estudio de los sistemas radio se comenzó analizando unos módulos de transmisión inalámbrica en frecuencias entre 2,4GHz y 2,5GHz (que es el rango de frecuencias de corta distancia en el que se encuentra una de las bandas ICM (industrial, científica y médica) de uso libre con una alta disponibilidad tecnología de bajo coste [15, cap. 11, pág. 282]³). En concreto, se analizó el módulo *NRF24L01* [7]. Existen dos versiones de módulos que incorporan el *NRF24L01*, uno con antena integrada y un alcance máximo de unos 20-30 metros (Figura 2.10), y la versión de alta potencia que incorpora amplificador y antena externa, con un alcance máximo de 700-1000 metros (Figura 2.11). Ambos son módulos ampliamente empleados y de bajo coste; por tanto, se decide evaluar la aplicación del módulo que integra amplificador y cuenta con una antena externa. Un factor de gran relevancia es la alimentación del módulo, donde será necesario que

³Las otras bandas más empleadas son en 915 MHz y 5 GHz

se alimente a 3,3V⁴.

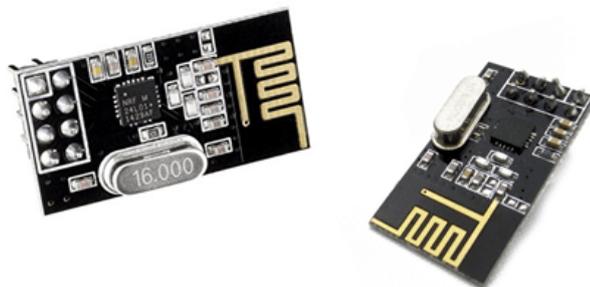


Figura 2.10: Transceptor NRF24L01 con antena impresa.



Figura 2.11: Transceptor NRF24L01 con antena externa.

El módulo de transmisión en la banda de 2,4GHz a 2,5GHz tiene un inconveniente: A pesar de ser un módulo que integra la capacidad de trabajar como transmisor y receptor, este módulo no es completamente *full-duplex*, dado que intercala el papel de transmisor y receptor con su pareja. Esta última característica impulsará el proyecto a estudios de otros módulos para desarrollar el papel de transceptor *full-duplex*. A pesar de que este módulo no sea muy útil para la aplicación que se busca en el proyecto, cumple perfectamente como transceptor con una tecnología de reparto de la capacidad del enlace basada en *TDM* (*Time Division Multiplexing*).

A continuación, se procede a analizar los puntos más relevantes del código empleado para conectar ambos transceptores. El código no es muy complejo y se puede encontrar toda la información pertinente a las funciones utilizadas en *GitHub*[11]. Se puede localizar el código al completo en el Anexo B.3.

Examinando el código cabe resaltar unas líneas de especial importancia como las que se adjuntan.

⁴Se puede alimentar de 1,9V a 3,6V.**NO ALIMENTAR A 5V** como otros módulos del mercado; esto podría dañarlo de manera irreversible.

```
1 | const byte addresses[][6] = {"00001", "00002"};
```

En esta línea se declara un array que puede contener hasta 6 direcciones, donde todas ellas deben compartir los primeros 32 bits y solo el byte menos significativo debe ser distinto (los ceros a la izquierda también se codifican). Estas direcciones son todas las que el transceptor puede estar escuchando simultáneamente, y por tanto, establecer comunicación con ellas. En el caso de nuestro sistema podría usarse para atender múltiples peticiones de distintos aviones. No obstante, para probar el funcionamiento del sistema de comunicación radio en la banda de 2,4GHz a 2,5GHz emplearemos únicamente dos módulos.

Para la coordinación entre los módulos se emplean las direcciones que se han declarado para determinar quién se encuentra en estado de transmisor y quién en estado de receptor. Para asignar estas direcciones hay que tener en cuenta que la dirección que uno de los módulos use para escribir tiene que estar siendo utilizada por el otro módulo para escuchar en ese mismo intervalo de tiempo. Se puede apreciar en el código cómo para el módulo 1 se emplea una dirección de escritura 00002 (que se corresponde con la dirección de lectura del módulo 2) y una de lectura 00001 (que se corresponde con la dirección de escritura del módulo 2). Además del número de tubería (se puede elegir 0-5). Se elige el número 1, ya que el número 0 tiene otros usos por parte del programa⁵.

```
1 | //PARA EL MODULO 1
2 | radio.openWritingPipe(addresses[1]); // 00002
3 | radio.openReadingPipe(1, addresses[0]); // 00001
4 | //PARA EL MODULO 2
5 | radio.openWritingPipe(addresses[0]); // 00001
6 | radio.openReadingPipe(1, addresses[1]); // 00002
```

Se adjunta una imagen del montaje y se analiza el conexionado como se muestra en la Figura 2.12. También se adjunta una captura de pantalla del montaje funcionando en la Figura 2.13.

- El cable rojo (pin VCC del módulo nRF24L01) se conectará a 3.3V.
- El cable negro (pin GND del módulo nRF24L01) a la GND común del montaje electrónico total.
- El cable morado (pin CE del módulo nRF24L01) al pin DIGITAL 7(debe estar declarado en el código del programa).
- El cable azul (pin CSN del módulo nRF24L01) al pin DIGITAL 8(debe estar declarado en el código del programa).
- El cable marrón (pin MISO del módulo nRF24L01) al pin DIGITAL 50.

⁵**Importante:** La tubería numero 0 es usada también por la tubería de escritura, con lo cual, antes de escuchar en este número se debe cerrar la escritura.

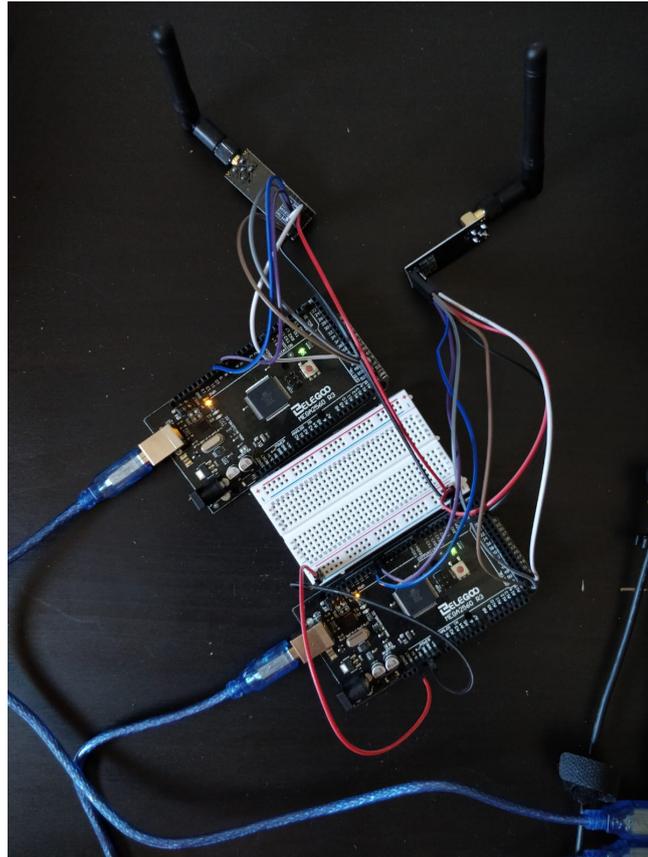


Figura 2.12: Montaje del sistema transceptor en la banda de 2.4GHz a 2.5GHz.

- El cable blanco (pin MOSI del módulo nRF24L01) al pin DIGITAL 51.
- El cable gris (pin SCK del módulo nRF24L01) al pin DIGITAL 52.

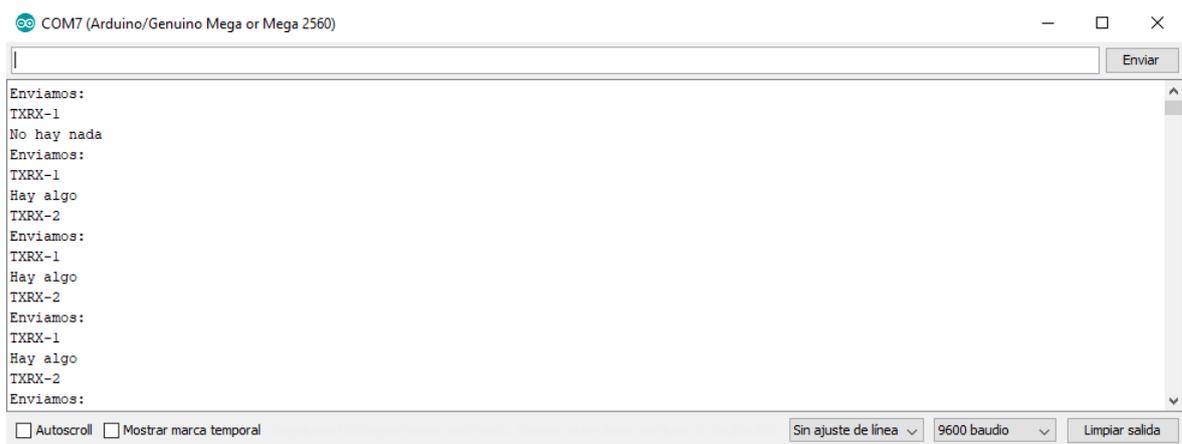


Figura 2.13: Captura de pantalla que muestra el funcionamiento del programa para los módulos nRF24L01.

En la captura de pantalla (Figura 2.13), se puede observar cómo los módulos se dividen los tiempos para transmitir y recibir información. Se muestra la salida *Serial*

de uno de los módulos, en este caso el que se ha denotado como módulo 1 en el código (Anexo B.3). El módulo 1 toma el rol de transmisor y comienza transmitiendo por el puerto 00002 el mensaje "TXRX-1"; seguidamente, se pone a escuchar por el puerto 00001. Tras no recibir información por el puerto 00001 en la ranura de tiempo prefijada en el código, se muestra por pantalla el mensaje "No hay nada". Pasado este tiempo de espera para recibir información, se pasa de nuevo al estado de transmisor. Una vez transmitido el mensaje en la ranura de tiempo asignada en el código por el usuario, pasará de nuevo a la escucha por el puerto 00001. Esta vez sí que recibe información, y tras notificarlo con un mensaje de "Hay algo", se muestra el mensaje recibido (el mensaje transmitido por el módulo 2): "TXRX-2". La secuencia de intercambio de papeles entre los módulos continúa indefinidamente.

Explicado el funcionamiento del módulo se procede a analizar su viabilidad en el proyecto. El módulo *NR24L01*, a pesar de tener un alcance tan elevado como se ha comentado anteriormente, presenta numerosos factores que pueden llegar a reducir esta distancia hasta unos 2-3 metros, sin embargo no es un factor de gran relevancia en un prototipo que emplea una banda de frecuencias de corta distancia que no se utilizaría en una implementación real (en este caso la banda de VHF-FM se asemeja más a las bandas de frecuencia militares que se usarían). Este módulo además necesita visión directa, pero como se usará en espacio abierto este aspecto no daría ningún problema en una futura implementación. En cuanto al impacto de diseño sobre la boya, se planificó en una primera aproximación al sistema final, que este contase con la tecnología necesaria para traducir a Morse un mensaje transmitido vía radio en ASCII por el módulo *NR24L01*. Se esogió esta estrategia de transmitir caracteres ASCII debido a que el módulo *NR24L01* no está pensado para proporcionar a su salida señales de audio (en este caso los tonos que diferencian el punto y la raya). Entonces, esto suponía que tanto la tarea de traducir el mensaje en ASCII, como la de generar los tonos⁶ hacia el receptor submarino recaía en la boya y dado que la boya debe de ser un sistema lo más económico y eficiente posible, es razonable pensar que la carga computacional debe recaer en el avión y en el submarino; con intención de reducir al máximo el consumo y asegurar así un mayor tiempo de autonomía a la boya. Es entonces cuando se decide usar el módulo transmisor de radio *KT0803K* en conjunto con el módulo receptor de radio *TEA5767*, los cuales cuentan con un puerto jack para introducir y extraer respectivamente la señal de audio y cuyas prestaciones en cuanto a alcance y consumo son similares.

Tecnología aérea empleada

Se procede a explicar los módulos de transmisión y recepción de radio empleados, el *KT0803K* y el *TEA5767* respectivamente. Se comienza a desarrollar el módulo transmisor que trabaja en la banda de FM comercial, lo cual implica que se usa una banda de frecuencias más bajas que el módulo *NR24L01* y por consiguiente es más robusta ante interferencias por objetos (aunque como ya se ha comentado este sistema se utiliza en espacio abierto). Este módulo transmisor radio cuenta con una gama de ver-

⁶Estos tonos representan el punto y la raya del código Morse. Se desarrolla más en detalle a lo largo del documento.

siones que mejoran la SNR y la detección de silencios para mejorar el consumo, como la *KT0803M* y la *KT0803L*. Para este proyecto se ha empleado el de menor gama, la versión *KT0803K* ya que consigue cumplir con las especificaciones para operar como sistema de transmisión radio en este proyecto.

La idea es conseguir una comunicación *full-duplex* transmitiendo a frecuencias distintas entre los dos extremos de la comunicación RF (avión-boya) y recibiendo la información en sus respectivas frecuencias. De este modo la comunicación entre ambos extremos es constante y no requiere de ningún sistema que reparta la capacidad del enlace; es decir, ambos transmiten y reciben aprovechando todo el ancho de banda del canal. Se decide así pues, usar la frecuencia $f_1 = 87.5\text{MHz}$ para la comunicación avión-boya y la frecuencia $f_2 = 108\text{MHz}$ para la comunicación boya-avión, tal y como se presenta la Figura 2.14.



Figura 2.14: Esquema de transmisión FM.

El módulo transmisor radio empleado es el modelo *FM Transmitter v2.0* (100mW de potencia; 17mA de corriente de operación) de la casa *Elechouse* que permite un recorrido desde los 70MHz hasta los 108MHz. El módulo transmisor utilizado se muestra en la Figura 2.16. Para transmitir se extrae una antena telescópica de una radio comercial reutilizada, ya que el módulo no cuenta con ninguna. Hasta la obtención de la antena telescópica se utilizó un cable cuya longitud se puede calcular mediante la ecuación (1.1) ya empleada en la Introducción. De modo que $\lambda = \frac{c}{97,75\text{MHz}} = 3,0669\text{m}$ y dado que 3m es poco práctico al ser demasiada longitud y elevar la impedancia de antena, se emplea un cuarto de onda $\lambda/4 = 76,673\text{cm}$, que usando un plano de tierra (PEC, conductor perfecto) para duplicar la longitud efectiva se puede reducir a una antena de $38,337\text{cm}$ [15, cáp. 4]. Se ilustra este efecto en la Figura 2.15.

Se adjunta también el esquema del integrado más relevante (Figura 2.17), el *KT0803K* [5]; encargado de la generación de la portadora, la modulación y la amplificación de

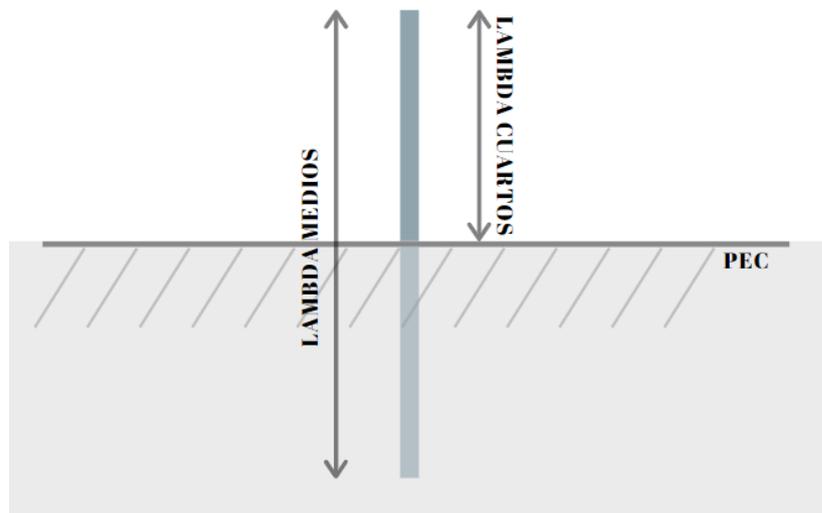


Figura 2.15: Método para duplicar el tamaño efectivo de una antena para la banda de FM.

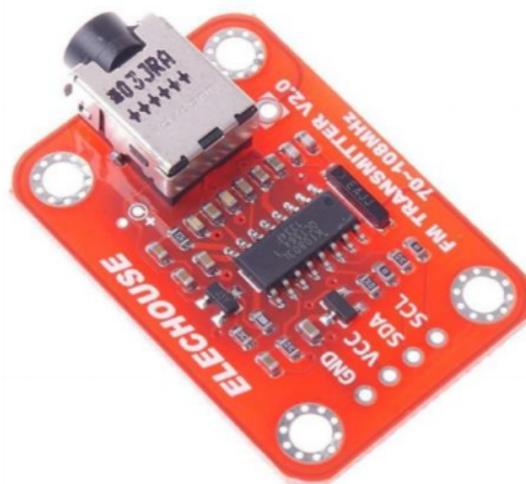


Figura 2.16: Transmisor FM de la casa *ELECHOUSE*.

potencia en el transmisor.

En cuanto al software, se recurre al uso de la librería `FMTX`⁷, que simplifica la comunicación con el módulo. La frecuencia de transmisión se fija en el código y el Arduino utiliza las salidas analógicas SDA (Serial Data Line) y SCL (Serial Clock Line) (protocolo I2C (Inter-Integrated Circuit)) para controlar la comunicación con el módulo transmisor mediante el pin SDA y el *clocking* (reloj) mediante el SCL. Es pues tarea del módulo de FM realizar el proceso de transmisión de la señal analógica recibida a través del jack de entrada que incorpora.

⁷**NOTA IMPORTANTE:** Debido a un problema de incompatibilidad software de la librería, esta **NO** funciona para versiones de IDE de Arduino superiores a la 1.8.8

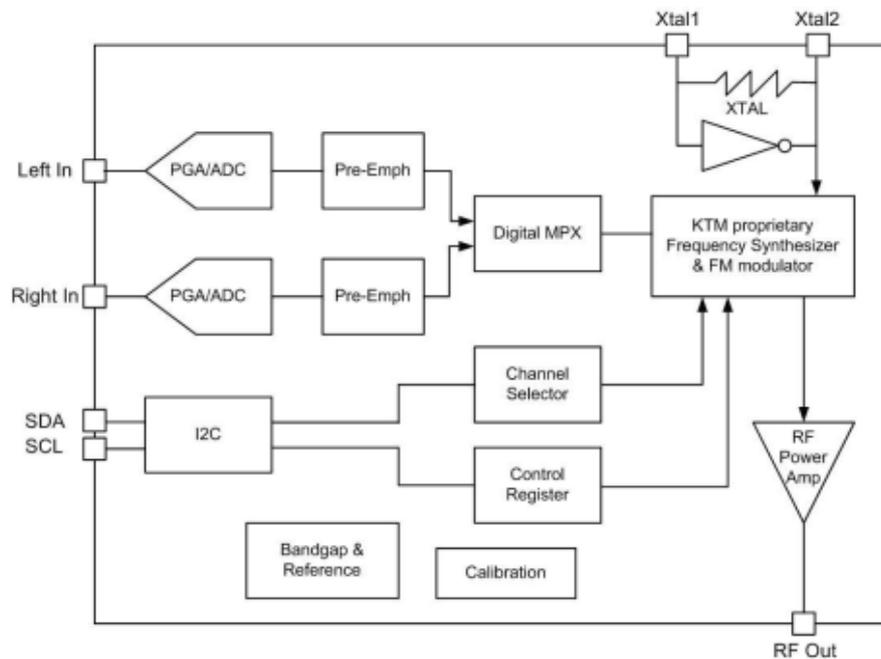


Figure 1: KT0803K System Diagram

Figura 2.17: Diagrama de bloques del integrado principal del Transmisor FM. Se observa que es un sistema de modulación FM basado en sintetizador de frecuencias, lo que estabiliza la frecuencia portadora.

El código simplificado para el uso en nuestro sistema es realmente sencillo; se adjunta con los comentarios pertinentes para su comprensión ⁸.

```

1  #include <FM_TX.h> // Libreria para controlar el transmisor FM
2  float fm_freq = 108; // Frecuencia de transmision
3  void setup(){
4  Serial.begin(9600); // Inicializa el puerto serie a 9600 baudios
   (valor por defecto)
5  Serial.print("FM-TX TFG\r\n"); // Imprime por el monitor serie
6
7  fmtx_init(fm_freq, EUROPE); // Para transmitir en el rango de
   frecuencias europeas
8
9  Serial.print("Canal: ");
10 Serial.print(fm_freq,1);
11 Serial.println("MHz");
12 }
13 void loop(){
14 // Podria usarse para controlar constantemente el puerto serie
   y variar en tiempo real la frecuencia de transmision, por
   ejemplo. Dado que para nuestro sistema esto no tiene mayor
   relevancia se deja simplificado el codigo.
15 }

```

⁸También se encuentra en el Anexo B.1.

En cuanto al receptor radio FM, se ha utilizado el módulo de radio *TEA5767*. Este módulo ronda los 11mA de corriente de operación y cuenta con un estado de bajo consumo cuando no se recibe portadora en la banda a la que está sintonizado. Además cuenta en su hardware con dos puertos jack. Uno de ellos, se utiliza para extraer el audio de la banda en la que se haya sintonizado y el otro se utiliza para acoplar una antena telescópica que viene incluida con el módulo. Se adjunta una imagen del módulo en su conjunto en la Figura 2.18.

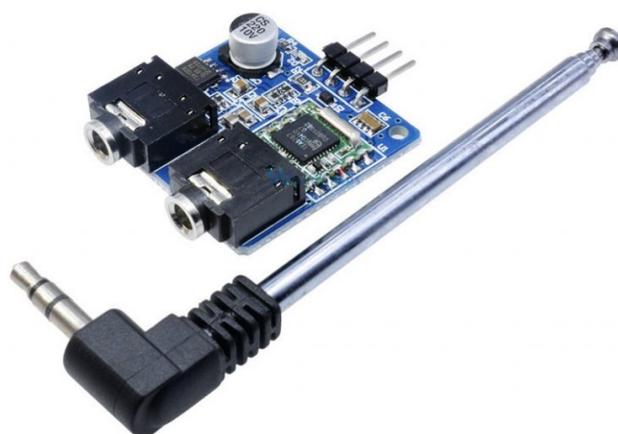


Figura 2.18: Módulo receptor en la banda de FM *TEA5767*.

El módulo receptor de FM recibe su nombre por parte de su integrado principal, el TEA5767 [8], cuyo diagrama se adjunta en la Figura 2.19.

Se puede observar que es la configuración básica de un receptor de FM basado en sintetizador de frecuencias para fijar la frecuencia de recepción. Cuenta también con un control automático de ganancia y con un bloque de desplazamiento a frecuencia intermedia típico de un receptor heterodino.

En cuanto al software, el uso de las librerías *radio.h* y *TEA5767.h* reduce el código significativamente. Se adjunta el código (Anexo B.2) necesario para fijar la frecuencia de recepción y conseguir sintonizar una emisora.

```
1 #include <radio.h>
2 #include <TEA5767.h>
3
4 #define FIX_BAND RADIO_BAND_FM // Sintonizamos FM
5 #define FIX_STATION 10800 // Definimos la estacion a 108.00 Mhz
6
7 TEA5767 radio; // Creamos un objeto radio
8
9 void setup(){
```

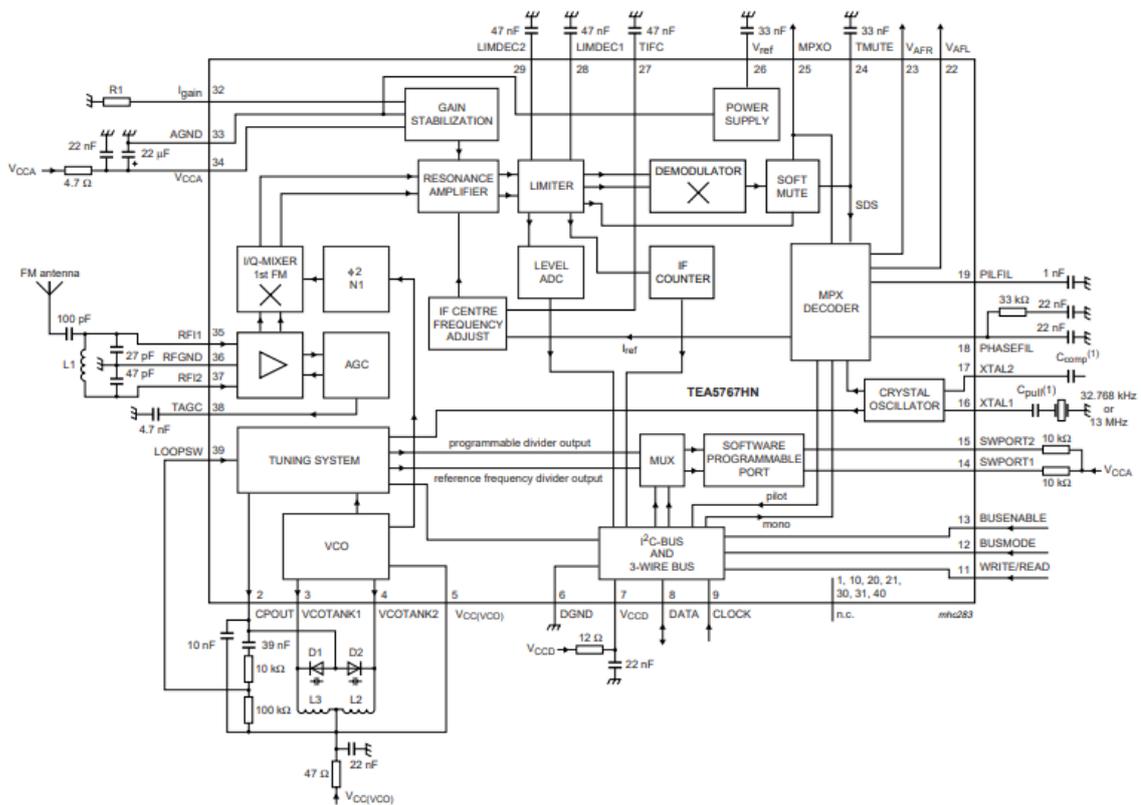


Figura 2.19: Diagrama de bloques del integrado *TEA5767*.

```

10 Serial.begin(57600); // Establecemos a 57600 baudios por ser el
    valor que necesita el modulo FM para comunicarse con el
    Arduino. De esta forma podremos imprimir por pantalla la
    informacion.
11
12 radio.init(); // Inicializar el objeto radio
13 radio.setBandFrequency(FIX_BAND, FIX_STATION); // Seleccionamos
    la emisora 108.00
14 radio.setVolume(15); // De 0 a 15
15 radio.setMono(true); // Para recibir en mono, ya que es el tipo
    que se utiliza en este sistema
16 }
17
18 void loop(){
19 // Se puede emplear el loop para imprimir informacion acerca de
    la banda en la que esta fijado el receptor, si por ejemplo
    se implementa un sistema para cambiar de cadena con un
    potenciometro.
20 }

```

2.2.3. Tecnologías electrónicas complementarias

Una vez definidos los módulos principales que integran el sistema del avión y el submarino, se procede a analizar los dispositivos electrónicos adicionales que intervienen en el correcto funcionamiento del sistema de comunicación basado en boya. Llegados a este punto se ha desarrollado la tecnología necesaria para establecer el enlace radio y acústico, pero no se ha explicado el método empleado para reconocer la codificación Morse mediante tonos de distinta frecuencia ni se ha planteado un sistema que acondicione la señal tras la atenuación introducida por la propagación. Por consiguiente, será necesario un elemento que sea capaz de interpretar los tonos que se van a transmitir y una solución para corregir el nivel de señal, tanto en la recepción para obtener una amplitud suficiente, como en la transmisión para lograr que la señal alcance al receptor con la potencia necesaria.

Módulos de amplificación

Se comienzan analizando los bloques de amplificación para señales acústicas encargados de acondicionar la señal a un nivel de tensión adecuado. Dado que la señal tendrá dificultades en la transmisión tales como la atenuación introducida por el medio o en el caso acústico el agravante de la sensibilidad reducida a causa de la impermeabilización (problema menos grave en un sistema en el que se empleen componentes acústicos de transmisión y recepción comerciales y no la solución económica a la que se ha tenido que recurrir). Para amplificar la señal se emplearán dos sistemas distintos, uno para la transmisión que aporte suficiente potencia a la señal antes de ser transmitida por el agua mediante ondas acústicas y otro para la recepción (tanto en sistemas acústicos como en radio) que consiga rescatar una señal atenuada hasta niveles de tensión con los que se pueda operar. Para la transmisión acústica se empleará un amplificador de potencia tipo D (Anexo C.4) y para la recepción una amplificación en dos etapas utilizando los dos amplificadores operacionales que contiene el integrado TL082 (Anexo C.1).

En primer lugar se explica el amplificador de potencia empleado en la transmisión acústica. Para amplificar la señal de audio antes de transmitirla por el altavoz a través del agua, se utiliza un amplificador de potencia de clase D. Que sea de clase D implica que utiliza el modo conmutado de los transistores (corte y saturación) para regular la entrega de potencia, esto hace que se pierda menos energía y consigue que el sistema de amplificación sea más eficiente. De esta forma, dado que irán integrados en la boya (además de en el submarino), se conseguirá alargar la vida útil de la batería de esta. Este módulo cuenta con un potenciómetro que regula la amplificación de la salida, una entrada de alimentación, una entrada de señal de audio mediante un puerto jack y salidas mediante un conector de 4 puertos para obtener una amplificación mono (empleando solo una salida) o estéreo (empleando las dos salidas). Se ilustra el módulo estudiado en la Figura 2.20.

Otras características del amplificador de potencia son: una potencia nominal de salida de 10W, unas frecuencias de operación de 20 Hz a 50 KHz (más que suficiente para el rango que se emplea en nuestro sistema), con un voltaje de funcionamiento de

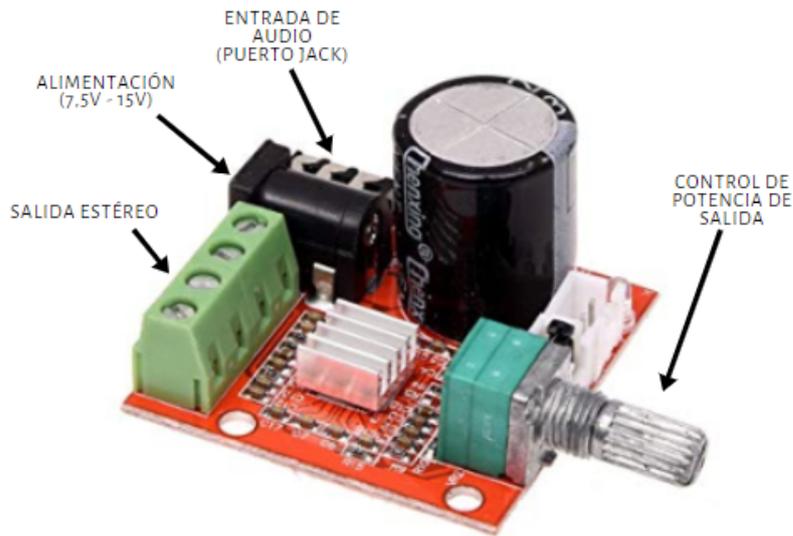


Figura 2.20: Módulo amplificador de potencia.

entre 7,5V y 15V (el fabricante recomienda alimentarlo con fuentes de alimentación de 9V ó 12V) y una corriente de reposo de 20 mA.

En el proyecto se acaba empleando un módulo de características de conexionado similares, con la mitad de potencia (5W) y con una alimentación de 1,8V a 12V. Se escoge definitivamente este módulo ya que facilita la integrabilidad en las cajas y cumple perfectamente con los niveles de amplificación necesarios para el altavoz. Se adjunta en la Figura 2.21.

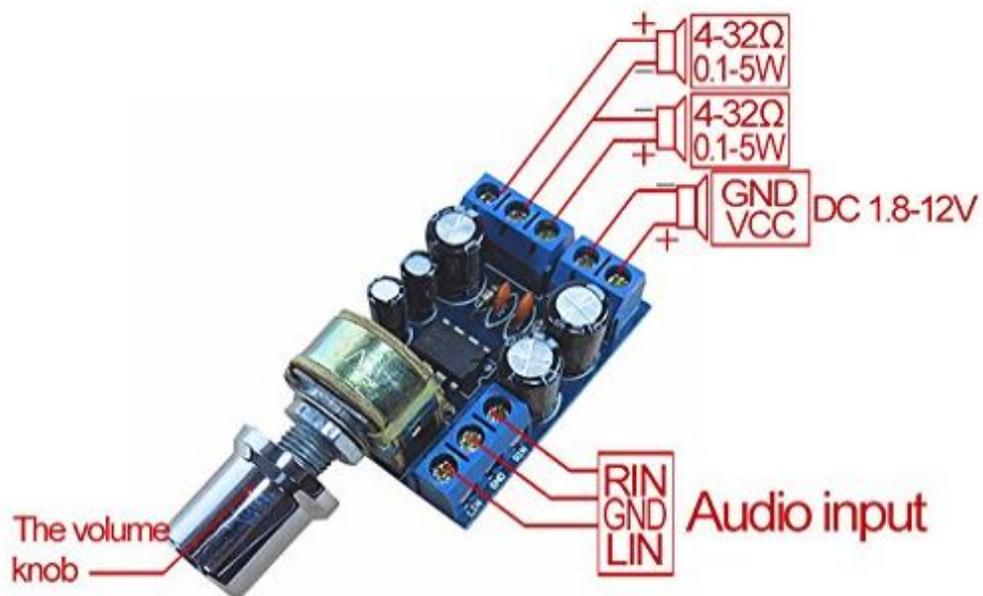


Figura 2.21: Módulo amplificador de potencia empleado en los montajes.

A continuación se explica detalladamente la etapa de amplificación que se aplica tras el micrófono para las señales acústicas recibidas. Para mejorar la calidad de la señal en la recepción, todos los receptores cuentan con una etapa de amplificación de audio, que ha sido implementada con un integrado TL082 usando una arquitectura de dos etapas para controlar mejor el offset de voltaje. Es importante controlar el nivel de offset de la señal, dado que si existe un nivel de offset DC excesivo puede acabar saturando la salida del amplificador debido a su elevada ganancia. Esta es también la razón de amplificar en 2 etapas, puesto que el offset amplificado de la primera se elimina mediante el condensador C2 a la salida del operacional y así no se acumula en el siguiente amplificador inversor. Un esquema electrónico de uno de estos montajes se observa en la Figura 2.22 (para más información consultar el Anexo C).

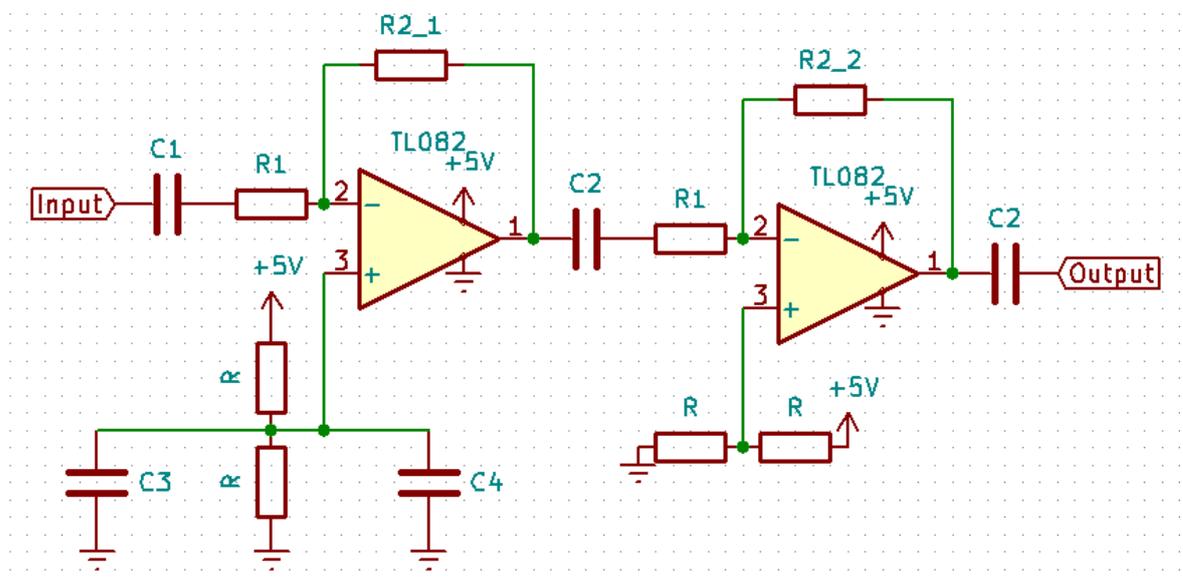


Figura 2.22: Esquema de un montaje de dos etapas de amplificación usando el TL082.

Se emplea una configuración de alimentación simple que es de gran utilidad en los montajes ya que reduce el número de fuentes en el sistema, lo que permite una mayor integrabilidad dado que de la placa de Arduino se puede extraer un nivel de 5V de tensión que proporciona uno de sus pines. Además se añade un filtro paso bajo en la referencia compuesto por C3 y C4 que elimine cualquier contenido parásito de alta frecuencia en la referencia, ya que es una "falsa masa" flotante y no una masa verdadera a tierra (donde se supone que no puede haber ruido y está a 0V). La siguiente etapa podría llevarla también pero no es tan crítico porque el ruido que pudiera producirse en la primera referencia se amplificaría $N_1G_1G_2$ (donde G_i se calcula según 2.3) y el ruido producido en la referencia de la segunda solamente sería N_2G_2 . Se adjunta una imagen de uno de los montajes en la Figura 2.23. En la imagen se puede observar como R2.1 y R2.2 se han implementado de forma que se puedan cambiar para modificar la ganancia del circuito (2.3).

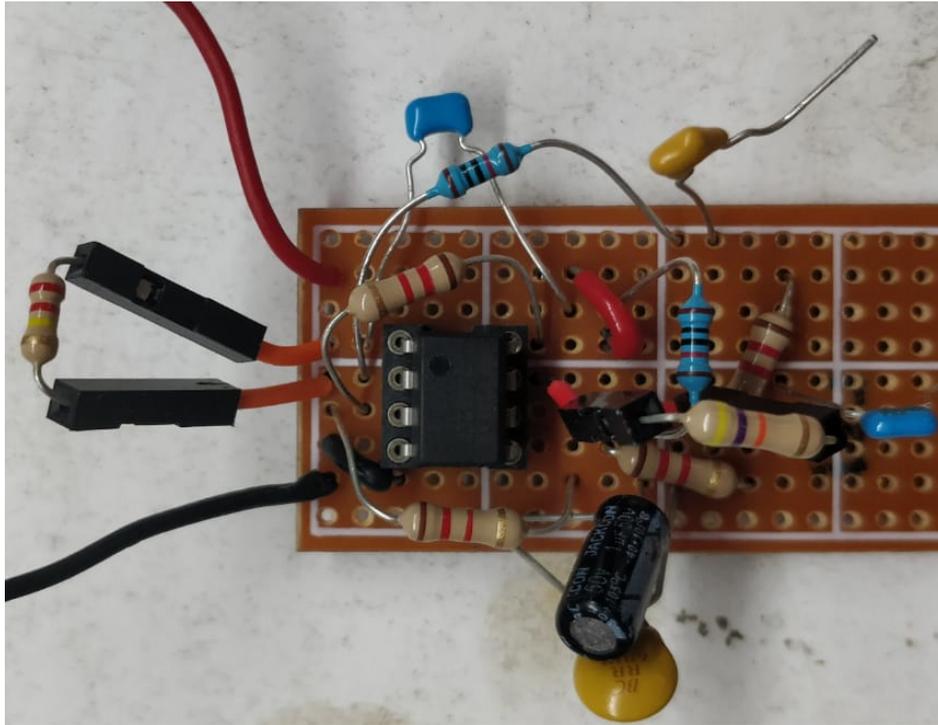


Figura 2.23: Montaje de un amplificador operacional de dos etapas.

Se adjunta también la ecuación que define la ganancia de cada etapa:

$$G = -\frac{R_2}{R_1} \quad (2.3)$$

Módulo de detección de tonos

Por último se analiza el bloque encargado de interpretar el código empleado en la comunicación entre el avión y el submarino. La señal que se va a transmitir será código Morse, mediante tonos de distintas frecuencias para definir el punto y la raya. El bloque que se encargará de distinguir esta codificación frecuencial es un montaje electrónico basado en el integrado CD4046B[12]. El integrado CD4046B (Anexo C.2) tiene una gran importancia en el funcionamiento del sistema, dado que es una pieza fundamental para poder interpretar los tonos en la recepción y también para generar tonos (señales cuadradas) de la frecuencia que nosotros queramos transmitir sin la necesidad de usar otro tipo de módulos o un ordenador. A continuación explicaremos resumidamente la teoría de operación de un PLL y la función que desempeñan los componentes electrónicos pasivos que se utilizan y se pueden observar en la Figura 2.24. Después de esto se explicará cómo se realiza la detección de tonos usando el PLL.

El CD4046B incluye un bloque interno para construir el oscilador controlado por voltaje (VCO) requerido en un PLL. El rango de frecuencias de funcionamiento del VCO se ajusta con tres componentes externos, R1, R2 y C1. Fijando el valor de C1 es posible establecer la frecuencia mínima de oscilación del VCO calculando R1 y la máxima calculando R2 según las especificaciones del fabricante⁹.

⁹Si dejamos la patilla 12 sin conectar a tierra a través de una resistencia R2, es decir, en alta

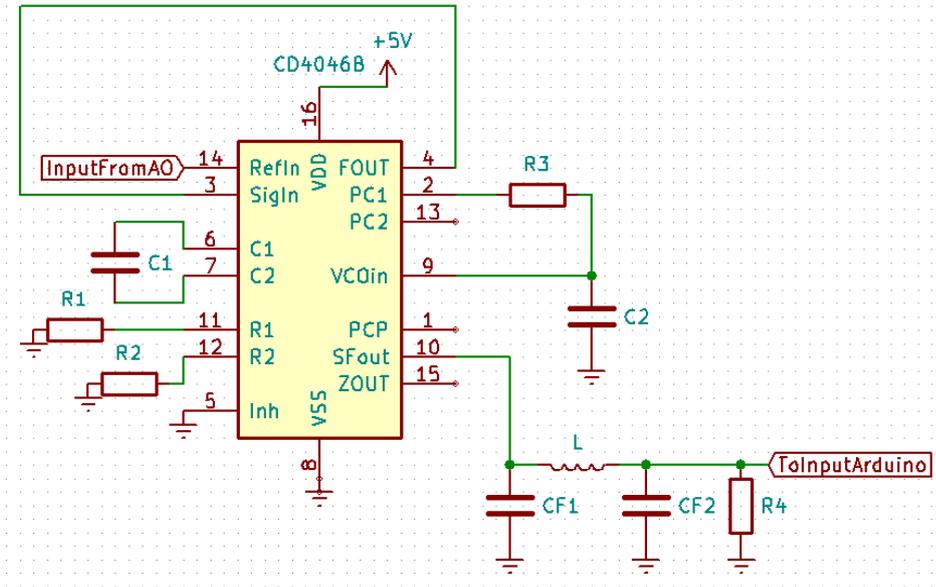


Figura 2.24: Esquema electrónico del conexionado del integrado CD4046B.

Queda entonces por comentar la función del resto de elementos pasivos del circuito. En primer lugar se analiza el conjunto R3 y C2 (formando un filtro paso bajo) conectado entre las patillas 2 (salida del comparador de fase) y 9 (entrada al VCO). Este conjunto RC (cuya respuesta frecuencial queda definida por la función de transferencia $F(s)$ (2.4)) fija dos parámetros fundamentales del PLL, que son: la w_n (pulsación natural) y la ξ (coeficiente de amortiguamiento). Además este bloque RC integra la salida pulsada del comparador digital de fase (puerta XOR)¹⁰ de forma que se otorga un nivel de tensión idealmente continuo¹¹ a la entrada del VCO. Se observa la localización de este bloque en el diagrama de la Figura 2.25.

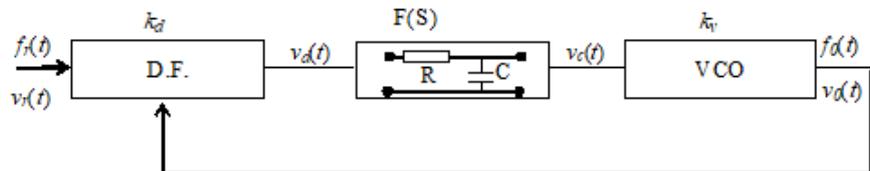


Figura 2.25: Diagrama de bloques del PLL.

$$F(s) = \frac{1}{1 + sRC} \quad (2.4)$$

impedancia, el recorrido comienza en 0 Hz.

¹⁰Una señal de pulsos cuadrados con anchura variante que indica cómo de diferente es la fase de la señal de entrada con respecto la fase de la señal de referencia.

¹¹En la práctica se puede observar que presenta un rizado.

Estos parámetros afectan al tiempo que tarda el PLL en engancharse a una frecuencia de entrada (que esté en su recorrido de frecuencias), además de definir la función de transferencia del PLL $H(s)$ (2.5). Esta función de transferencia es tipo paso bajo (como se ilustra en la Figura 2.26) y presenta un codo para la pulsación natural w_n . Para frecuencias de entrada superiores a w_n (y tratando casos de frecuencias dentro del margen de operación) la amplitud de la excursión de fase del PLL se verá atenuada por un factor 40dB/década, es decir, el PLL conseguirá seguir a la entrada, pero a medida que esta frecuencia se aleje de la pulsación natural, la amplitud de la excursión de fase de la señal de salida sufrirá cada vez una mayor atenuación.

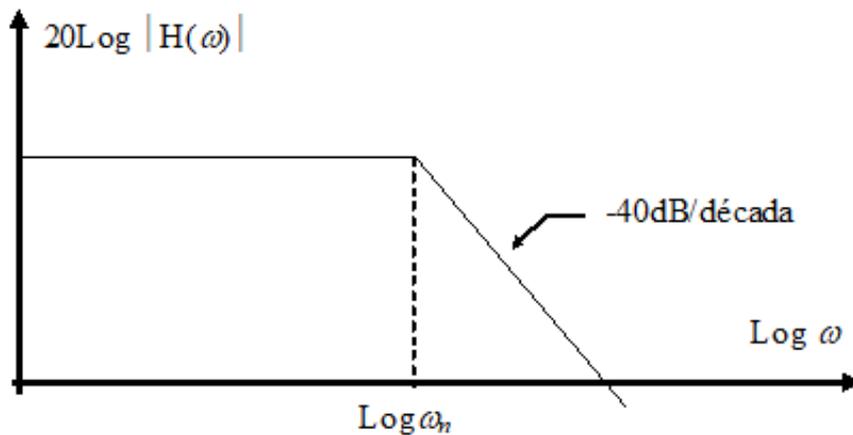


Figura 2.26: Función de transferencia del PLL.

$$H(s) = \frac{\phi_0(s)}{\phi_r(s)} = \frac{KF(s)}{s + KF(s)}, \text{ donde } K = 2\pi k_d k_v^{12} \quad (2.5)$$

Siguiendo las fórmulas (2.6) se puede determinar que a menor pulsación natural w_n , es decir, cuanto más paso bajo sea la función de transferencia del PLL (Figura 2.26) implicará una respuesta más lenta por parte del PLL a la hora de engancharse a una frecuencia que esté dentro de su recorrido. También se puede extraer la conclusión de que a mayor coeficiente de amortiguamiento ξ , el PLL oscila menos antes de engancharse a la frecuencia de entrada. Este fenómeno se puede observar en la Figura 2.27, donde se suele considerar que el PLL está enganchado para valores entorno a $\pm 0,1$ del valor unidad $\theta_0(t) = 1$.

$$\tau = RC; \quad \omega_n = \sqrt{\frac{K}{\tau}}; \quad \xi = \frac{1}{2\tau\omega_n} \quad (2.6)$$

¹²Donde k_d es la pendiente de la función de transferencia del comparador de fase en (V/rad) y k_v la pendiente de la función de transferencia del VCO en (Hz/V).

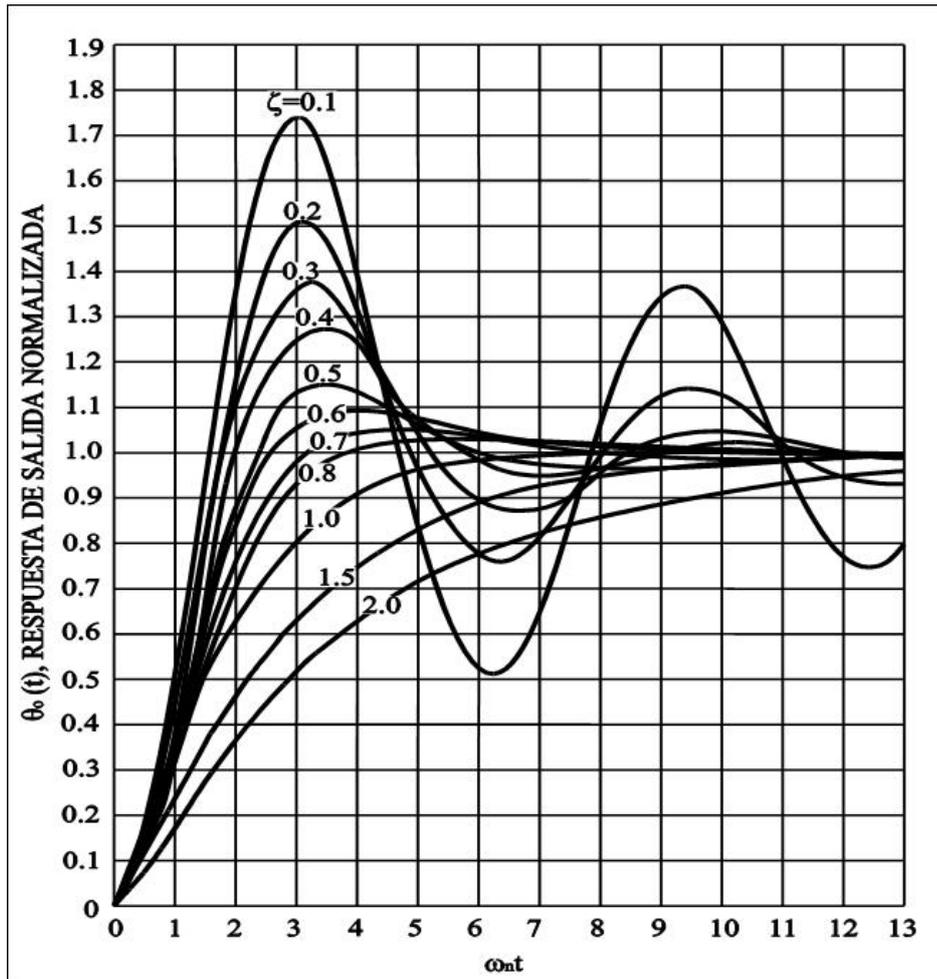


Figura 2.27: Respuesta al escalón de un sistema tipo I de orden II.

Si ahora se sustituye (2.5) en (2.6):

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.7)$$

Por último, se comenta el bloque con dos condensadores CF1, CF2 y L. Su función principal es filtrar por completo el rizado (Figura 2.29, Figura 2.30) del nivel de continua que aparece por la patilla 10 (que será 0,7V menor que la de la patilla 9 por ser la misma salida tras un seguidor de emisor). La señal que se obtiene en la patilla 9 está relacionada con la salida del filtro del bucle $F(s)$ y dada la naturaleza digital del detector de fase, se produce un rizado digital sobre la componente continua de la señal obtenida cuya frecuencia está relacionada con la de la señal de entrada. Este rizado ocasiona errores en la medición por parte del Arduino dado que la excursión del nivel de tensión llega a ser muy significativa y alcanza valores de tensión que se alejan mucho del esperado para la frecuencia a la que se encuentra enganchado el PLL.

Se adjunta una imagen de ejemplo de uno de los montajes en la Figura 2.28.

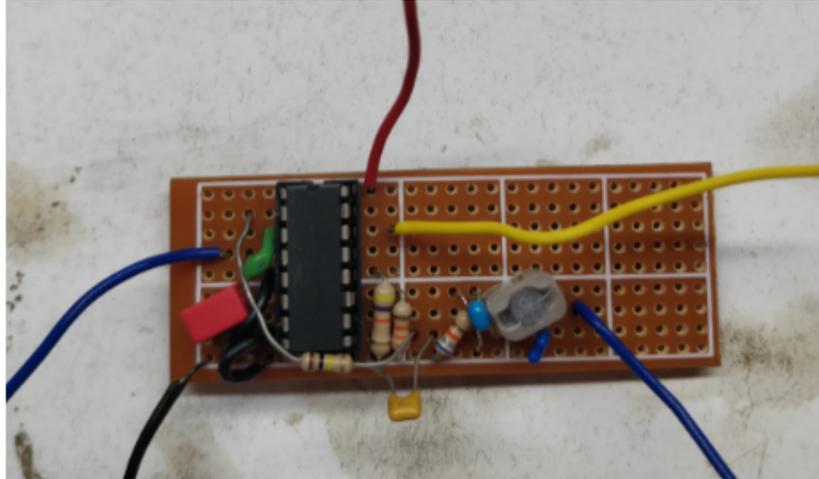


Figura 2.28: Montaje del integrado CD4046B para ser usado como PLL.

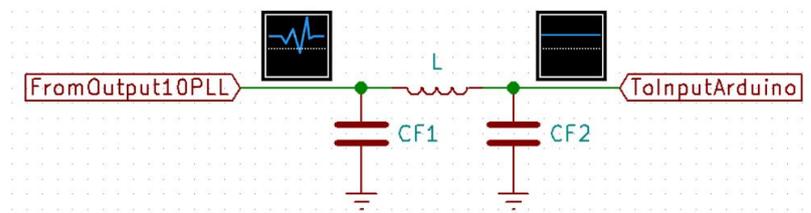


Figura 2.29: Función del filtro a la salida de la patilla 10 del PLL.

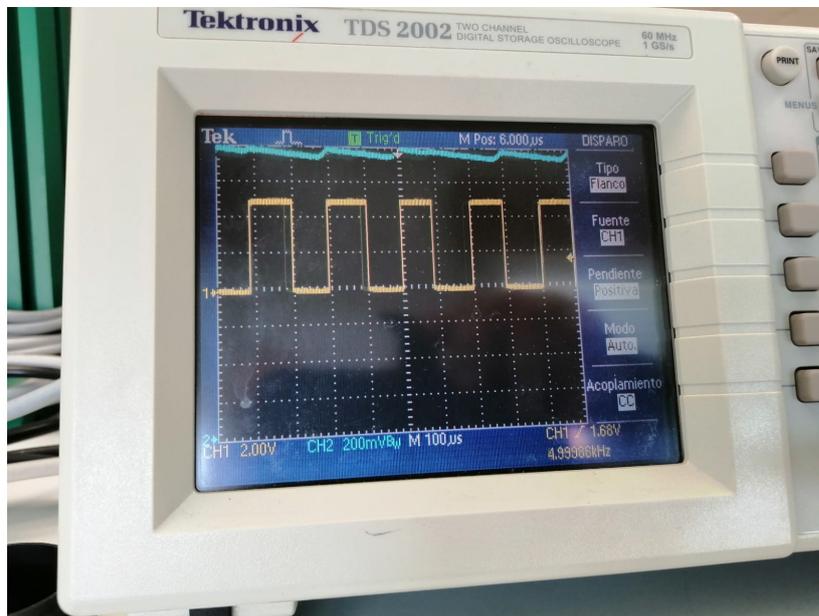


Figura 2.30: Ilustración del rizado de la señal a la salida de la patilla 10. En amarillo una señal de entrada de 5 KHz (patilla 14), en azul la señal de tensión con rizado a la salida de la patilla 10.

Tras el análisis teórico se procede a explicar el proceso de detección de tonos. Siguiendo el esquema electrónico de la Figura 2.24 la señal amplificada se introduce en la patilla 14. Esta señal será un tono cuya frecuencia varía dependiendo de si se está

recibiendo un punto o una raya y debe tener suficiente amplitud como para que el PLL pueda funcionar. El PLL se engancha a la frecuencia de esta señal y genera un nivel de tensión proporcional y unívoco para cada frecuencia de entrada a la salida de la patilla 10 (o como ya se ha explicado, de la patilla 9). Dado que estos niveles de tensión caracterizan por completo la frecuencia que se está recibiendo, se emplea para realizar una lectura con el Arduino y así poder diferenciar el código Morse mediante tonos de distintas frecuencias.

Receptor del tono de petición GPS

Para la solicitud GPS del submarino se empleará un esquema PLL como el analizado en el punto anterior 2.2.3. El diseño del PLL se adapta para detectar la petición del submarino mediante un tono de 500 Hz. El conjunto componentes electrónicos pasivos empleado en el esquema electrónico del PLL para poder engancharse a dicha frecuencia se recoge en la Tabla 2.1.

CD4046B (para GPS)	
L	1 mH
C1	10 nF
C2	6,8 nF
CF1	10 nF
CF2	100 nF
R1	270 K Ω
R2	∞
R3	82 K Ω

Tabla 2.1: Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz.

2.3. Desarrollo y análisis del módulo de geolocalización

En este apartado se estudia el modo de integrar un sistema GPS en la boya. Este módulo entra en funcionamiento cuando el submarino solicita la ubicación GPS a la boya. Implementar un módulo GPS es de especial relevancia en el proyecto, dado que la posibilidad de geolocalización por parte del submarino es una característica estratégica adicional muy importante que añadir al sistema de comunicación.

Para implementar esta funcionalidad se emplea un módulo comercial GPS, en concreto, el modelo GPS6MV2. Este módulo se conecta con el microcontrolador Arduino y de una forma muy sencilla y eficaz se obtienen con precisión datos como: la ubicación (coordenadas), la fecha y hora actual, el número de satélites visibles, la altitud con respecto a nivel del mar (en este proyecto dado que es un sistema de boyas no será de utilidad¹³) e incluso la velocidad a la que se desplaza el dispositivo que lo incorpora. Un factor con gran relevancia es la zona donde se use el dispositivo, dado que en zonas de interior el funcionamiento se puede demorar hasta **15 minutos** en lo que el módulo GPS localiza satélites; por tanto, se recomienda su uso en espacio abierto para un funcionamiento más dinámico que se ajuste mejor a los cambios que pueda sufrir el módulo. Debido a que el módulo se utilizará en espacio abierto esto no resulta un inconveniente, sin embargo, es un dato a tener en cuenta a la hora de realizar pruebas antes de desplegarlo.

En cuanto a la comunicación entre el Arduino y el módulo GPS se utilizará un puerto serie o se habilitarán los pines correctos mediante software con la librería `SoftwareSerial` [3]¹⁴. Se ha realizado el montaje con las dos placas más conocidas y usadas de Arduino (la UNO y la MEGA2560). En la Figura 2.31 se muestra uno de los montajes.

El montaje a nivel hardware es realmente sencillo, dado que solo requiere un conexionado básico con la alimentación ofrecida por el Arduino (o una fuente independiente¹⁵) y con un puerto `Serial`¹⁶ para la comunicación serie bit a bit con la placa Arduino. Para mayor detalle y siguiendo el código de colores:

- El cable rojo (pin VCC del módulo GPS) se conectará a 3,3V o 5V.
- El cable negro (pin GND del módulo GPS) a la GND común del montaje electrónico total.
- El cable azul (pin TX del módulo GPS) al pin DIGITAL 4.

¹³Para cualquier punto de la superficie terrestre se requiere una calibración manual por parte del usuario.

¹⁴No todos los pines se pueden habilitar como puerto serie, dado que no todos permiten interrupciones de cambio. Esto depende de la placa de Arduino en uso, vease el apartado de **Limitaciones** en [3].

¹⁵Siempre teniendo en cuenta que para el resto de elementos electrónicos dependientes entre si se debe mantener la misma tierra.

¹⁶En este caso y según [3], es necesario utilizar la librería `SoftwareSerial` y habilitar dos pines para la transmisión serie de datos.

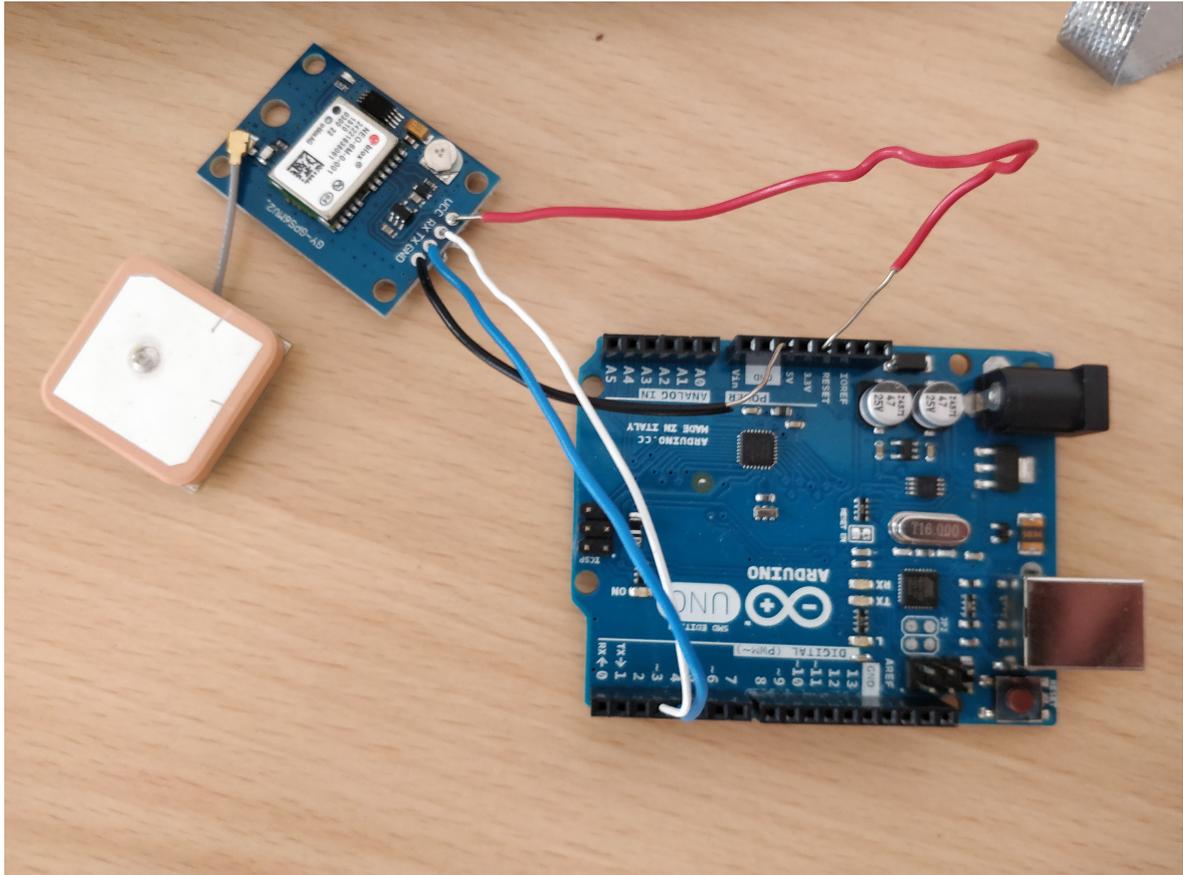


Figura 2.31: Montaje del módulo GPS con la placa Arduino UNO

- El cable blanco (pin RX del módulo GPS) al pin DIGITAL 3.

Sobre el software que permite extraer toda esta información, cabe resaltar la utilidad de usar la librería TinyGPS[4]. La necesidad de utilizar esta librería, surge como consecuencia de la dificultad que conlleva interpretar los datos que el módulo GPS envía al Arduino. Se puede apreciar la diferencia en las figuras Figura 2.32 y Figura 2.33.

```

COM6
Enviar
4
$GPGSV,3,1,10,01,44,277,36,03,17,209,25,08,85,089,34,10,28,046764$GPRMC,16430,W8G36,.$GPRMC,16447211S,035$GPRMC,16441
023,105$GPRMC,16441
023,103$GPRMC,16446P.3,,,7,$GPRMC,16441
023,701$GPRMC,1645,,610,1,4$GPRMC,1640A652,4,2,$GPRMC,164A60,1,B,34$GPRMC,164,
8M,2017$GPRMC,164190517563,$GPRMC,1646P,.,,3P2,,$GPRMC,164.V4207G121$GPRMC,1649T1,3,S,46$GPRMC,1640,6,,,3,91$GPRMC,164N,188,,,20$GPRMC,164-
*$GPRMC,1648N4,.8,3P$GPRMC,1648N4,.8,7P$GPRMC,164,08,0,,G,$GPRMC,1644W540402P4$GPRMC,1644.K,1$,1V6$GPRMC,16443.73*576LGPRMC,1644,1,,0,18:
023,102$GPRMC,16441G02102,0$GPRMC,1642,A,0,6,64$GPRMC,1640A622,5,2.$GPRMC,1648,A610,1,4$GPRMC,164A12,1,9,33$GPRMC,164,0M02335$GPRMC,164.V
$GPRMC,1641
023,111$GPRMC,16446P.3,,,71$GPRMC,1644,18,0,,G*,$GPRMC,1640G1,18,24$GPRMC,164.,,18,1381.$GPRMC,164,648,477,0$GPRMC,164.,,18,1381.
Autoscroll [ ] Mostrar marca temporal Retorno de carro 115200 baudio Limpiar salida

```

Figura 2.32: Datos recogidos por el módulo GPS sin usar la librería TinyGPS



Figura 2.33: Datos recogidos por el módulo GPS con la librería TinyGPS

Por último, se facilita al lector el código utilizado para la extracción de la ubicación, el dato de interés en este proyecto. El resto de datos se pueden extraer de forma análoga consultando la información de la librería TinyGPS.

```

1  #include <SoftwareSerial.h>// Incluimos la libreria
    SoftwareSerial que nos permite declarar ciertos pines como
    puertos serie
2  #include <TinyGPS.h>// Incluimos la libreria TinyGPS que nos
    ayuda a tener una extraccion de datos mas sencilla del modulo
    GPS
3
4  TinyGPS gps;// Declaramos el objeto gps de la clase TinyGPS
5  SoftwareSerial serialgps(4,3);// Declaramos el pin 4 Tx y 3 Rx
    (nuevos puertos serie)
6
7  void setup(){
8      Serial.begin(115200);// Iniciamos el puerto serie a
        115200 baudios, que es la velocidad de simbolos por
        segundo necesaria para mostrar los datos extraidos
        del modulo GPS
9      serialgps.begin(9600);// Iniciamos el puerto serie por
        el cual se comunican el modulo GPS y el Arduino
10 }
11
12 void loop(){
13     while(serialgps.available()){// Mientras haya datos
        disponibles en el puerto serie...
14         int c = serialgps.read();// Extraemos la
            informacion del puerto serie
15
16         if(gps.encode(c)){// Devuelve "true" cuando los
            nuevos datos se han descodificado
            completamente y se pueden utilizar.
17         float latitude, longitude;
  
```

```

18     gps.f_get_position(&latitude, &longitude);//
        Obtenemos la latitud y la longitud
19
20     // Se imprime por el monitor serie del
        ordenador la informacion obtenida
21     Serial.print("Latitud/Longitud: ");
22     Serial.print(latitude,5);
23     Serial.print(", ");
24     Serial.println(longitude,5);
25     }
26 }
27 }

```

2.4. Desarrollo hardware y software para la generación de tonos

2.4.1. Modalidad de generación de los tonos mediante hardware

En este apartado se buscará una solución para la generación de tonos mediante hardware que se usará en la boya para transmitir sus coordenadas GPS en caso de que el submarino las pida.

Se estudió la posibilidad de emplear un módulo generador de ondas cuadradas, triangulares y sinusoidales, el módulo *AD9833*, el cual se adjunta en la Figura 2.34. El uso de este módulo se desestimó debido a que realizar esta función con un VCO es un método más transparente (debido al estudio en profundidad que se realiza sobre el integrado CD4046B en este proyecto) y económico.

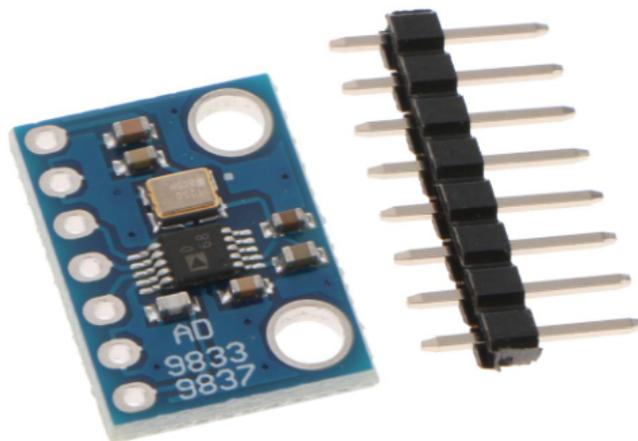


Figura 2.34: Módulo generador de onda cuadrada, triangular y sinusoidal.

Como segunda opción y definitiva se realizó una implementación con el integrado

CD4046B(subsección 2.2.3). Esta vez empleándolo como un VCO al cual se le aplicará una tensión en la patilla 9, que estará controlada por un PIN digital de Arduino. De este modo, se obtendrá una señal cuadrada a la salida de la patilla 4 del CD4046B con una frecuencia proporcional a esa tensión de entrada. A pesar de ser una señal cuadrada, al transmitir a tan alta frecuencia los tonos, los armónicos en los múltiplos impares del tono caerán fuera del rango de enganche del PLL. Además, la respuesta frecuencial del micrófono cae para frecuencias muy elevadas, con lo cual, no será preciso filtrar los armónicos de la señal cuadrada. Se ilustra el conexionado en la Figura 2.35.

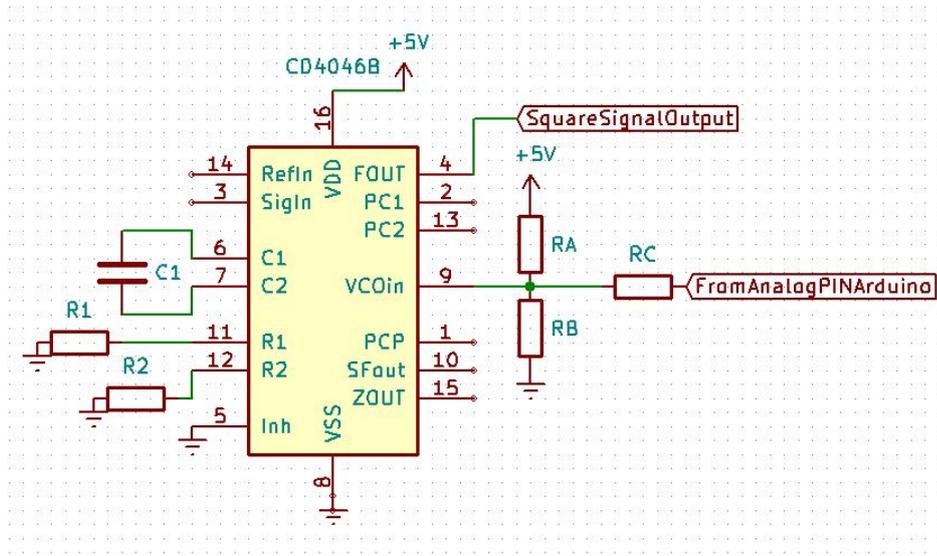


Figura 2.35: Esquema de conexionado del integrado CD4046B empleando su VCO.

Con respecto a los valores de los componentes electrónicos pasivos R1, R2 y C1, se escogen para que el VCO se mueva en el rango de frecuencias para una transmisión hacia el submarino, es decir 4100 Hz y 7000 Hz tal y como se explica en la subsección 2.2.1. En cuanto a RA, RB y RC se despejan mediante un sistema sencillo en el cual se necesita obtener 1,4V cuando se ponga en alta (5V) un pin digital de Arduino y 3,2V cuando se ponga en baja (0V) este mismo pin. Se fija $R_B = 10\text{ K}\Omega$ y se resuelven los sistemas que se adjuntan (2.8, 2.9).

$$V_A = V_{cc} \frac{R_B // R_C}{R_A + R_B // R_C}; \text{ donde } R_B = 10\text{ K}\Omega, V_A = 1,4\text{ V}; V_{cc} = 5\text{ V} \quad (2.8)$$

$$V_B = V_{cc} \frac{R_B}{R_B + R_A // R_C}, \text{ donde } R_B = 10\text{ K}\Omega, V_B = 3,2\text{ V}; V_{cc} = 5\text{ V} \quad (2.9)$$

Los valores de todo el circuito se recogen en la Tabla 2.2:

CD4046B (VCO)	
C1	6,8 nF
R1	47 K Ω
R2	82 K Ω
RA	12,86 K Ω \rightarrow 12 K Ω
RB	10 K Ω
RC	10 K Ω

Tabla 2.2: Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.

2.4.2. Modalidad de generación de los tonos mediante software

La generación de los tonos también se ha realizado en las cajas del avión y del submarino, empleando la salida de un PC y por tanto generadas vía software. El código estará contenido en el programa que realiza la función de recoger los mensajes en ASCII introducidos por el usuario y traducirlos a código Morse, tal y como se observa en la representación de alto nivel simplificada de la Figura 2.36.

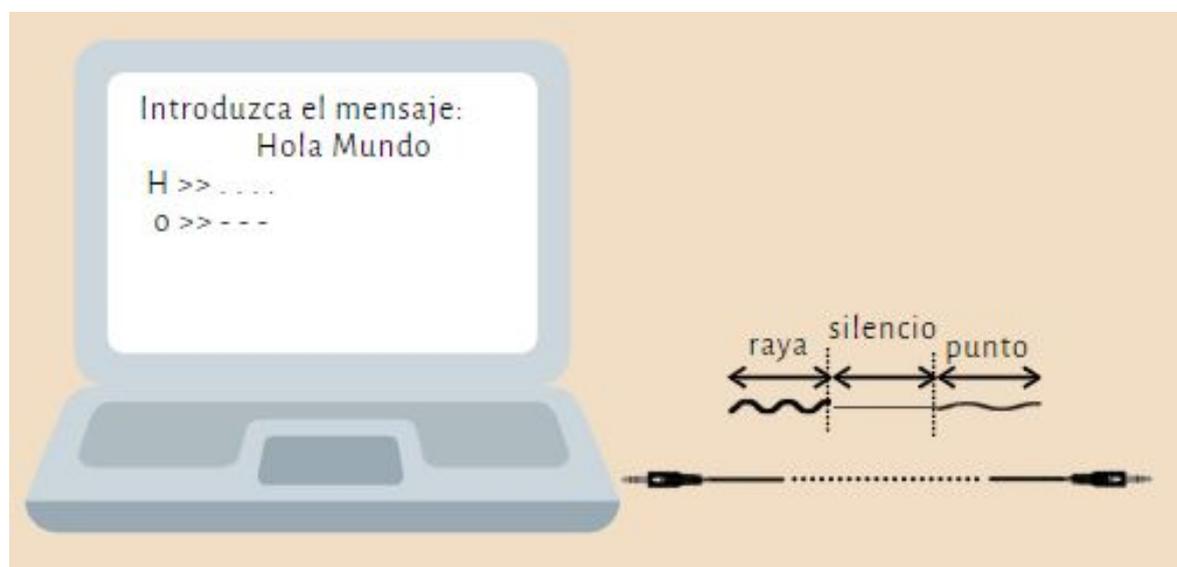


Figura 2.36: Recreación simplificada de alto nivel de la generación de tonos software.

El código se ha desarrollado de tres maneras diferentes: en C++ para Arduino, en Octave y en MatLab usando la interfaz gráfica GUI¹⁷. Se opta por la solución con

¹⁷Graphical User Interface. Herramienta de MatLab que permite programar software a través de una interfaz gráfica; de este modo, se elimina la necesidad de escribir comandos para generar nuestra

Octave por ser software gratuito y por descargar al Arduino de una tarea que puede realizar el ordenador. Para simplificar la comprensión del código se adjunta un diagrama de flujo¹⁸ que explica su funcionamiento (Figura 2.37).

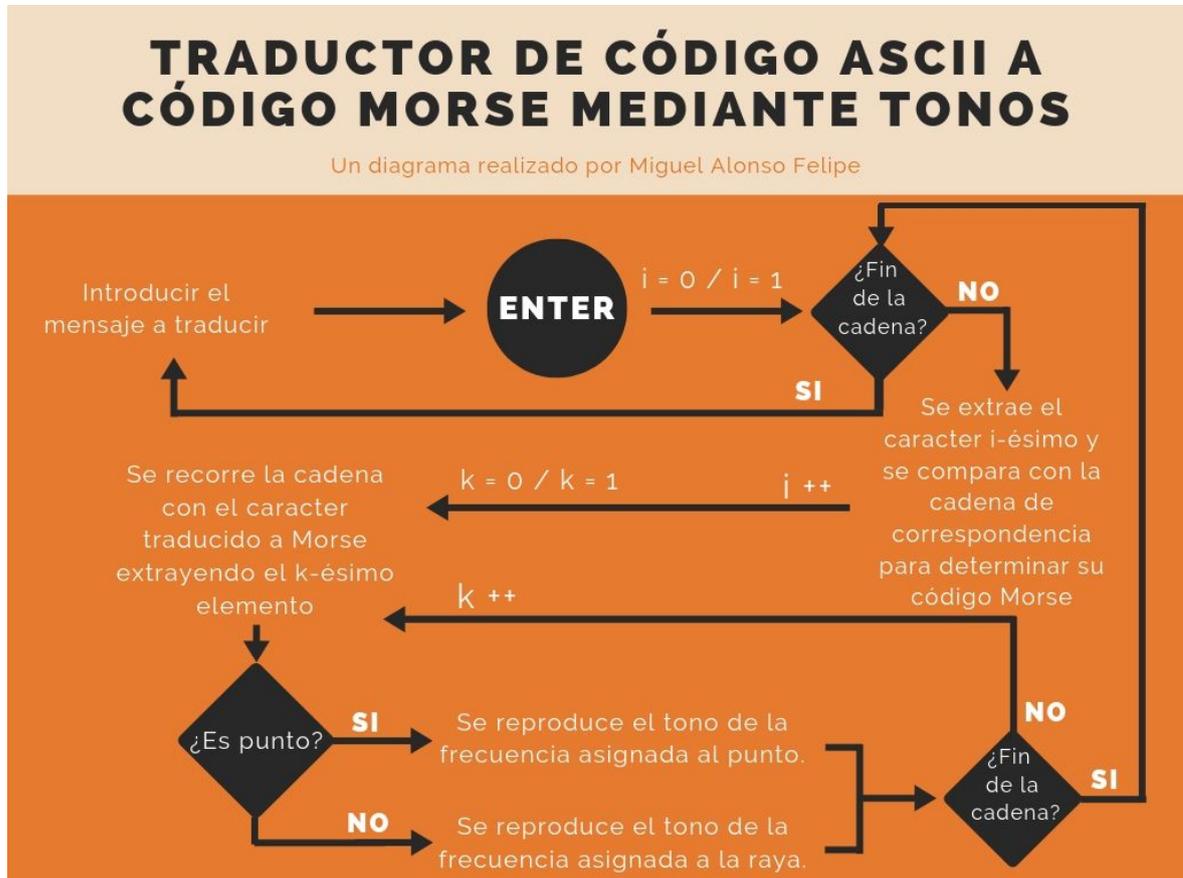


Figura 2.37: Diagrama de flujo explicativo de la traducción de código ASCII a Morse mediante tonos de distinta frecuencia.

Se puede observar en primer lugar el código realizado con Octave en el Anexo A.1, que es el código que se utiliza en el sistema final.

Tras ejecutar el código en la aplicación, se solicita al usuario introducir el mensaje. Una vez introducido el mensaje y tras pulsar *ENTER*, se puede observar el funcionamiento del programa. Se comienza analizando el mensaje carácter a carácter y se traduce cada uno a código Morse. Seguidamente, se reproducen los tonos correspondientes a los puntos y las rayas de cada uno de los caracteres de forma progresiva según son enviados (Figura 2.37). Se muestra el programa en funcionamiento en la Figura 2.38.

A continuación, se presenta la segunda opción con el código realizado con MatLab 2018a (código en el Anexo A.2). Para comprender parte de este código hay que estar

aplicación.

¹⁸También se encuentra en el Anexo D.1.

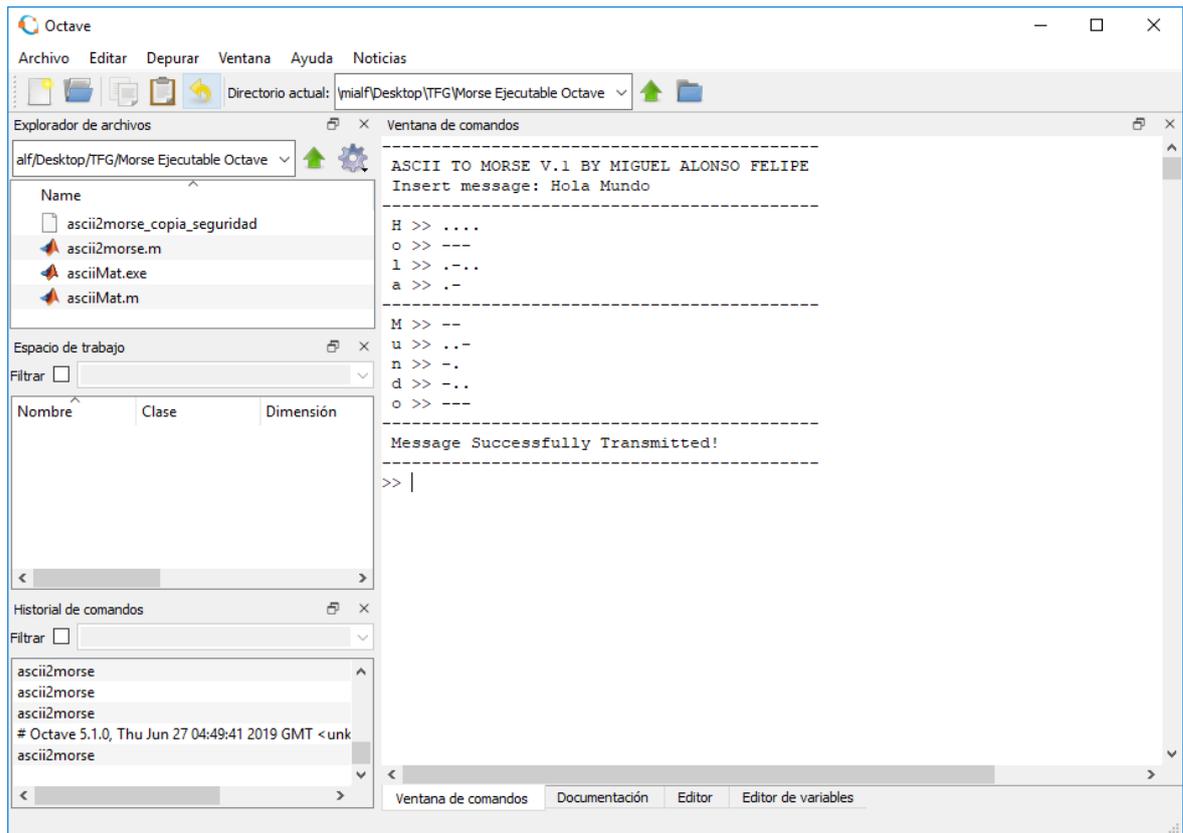


Figura 2.38: Programa de Octave generador de Morse con tonos de distinta frecuencia.

familiarizado con el uso del GUI de MatLab; no obstante, se comentan las funciones desarrolladas por el usuario y las generadas automáticamente por el asistente gráfico de MatLab y la utilidad de cada una.

Se puede observar que el código es bastante más complejo que el realizado con Octave, esto se debe a la interfaz gráfica que se lanza tras compilarlo. Sin la interfaz gráfica el código de Octave y el de MatLab serían idénticos. Se muestra la interfaz gráfica generada por el código en la Figura 2.39.

En la interfaz gráfica se irá mostrando el proceso de la transmisión de los caracteres en la línea *Ascii process*, mientras que el proceso de transmisión del código Morse se irá imprimiendo en la línea *Morse process*, donde los caracteres son separados por una barra vertical y las palabras son separadas por una doble barra (//).

Para concluir con la explicación de la generación de tonos vía software, el último lenguaje en el que se ha implementado es en C++ (Anexo A.3). Se usarán funciones avanzadas de entrada/salida (rotuladas como *Advanced I/O*), que se pueden encontrar fácilmente en la página web de Arduino [6]. Para comprobar el correcto funcionamiento podemos usar un *Buzzer* pasivo (como se muestra en la Figura 2.40). El código escrito en C++ muestra evidentes similitudes con el escrito en MatLab y Octave; no obstante, ha sido adaptado para poder correr sobre el microcontrolador de Arduino. También cabe resaltar que los tonos generados, si bien tienen la frecuencia que se estipula en la

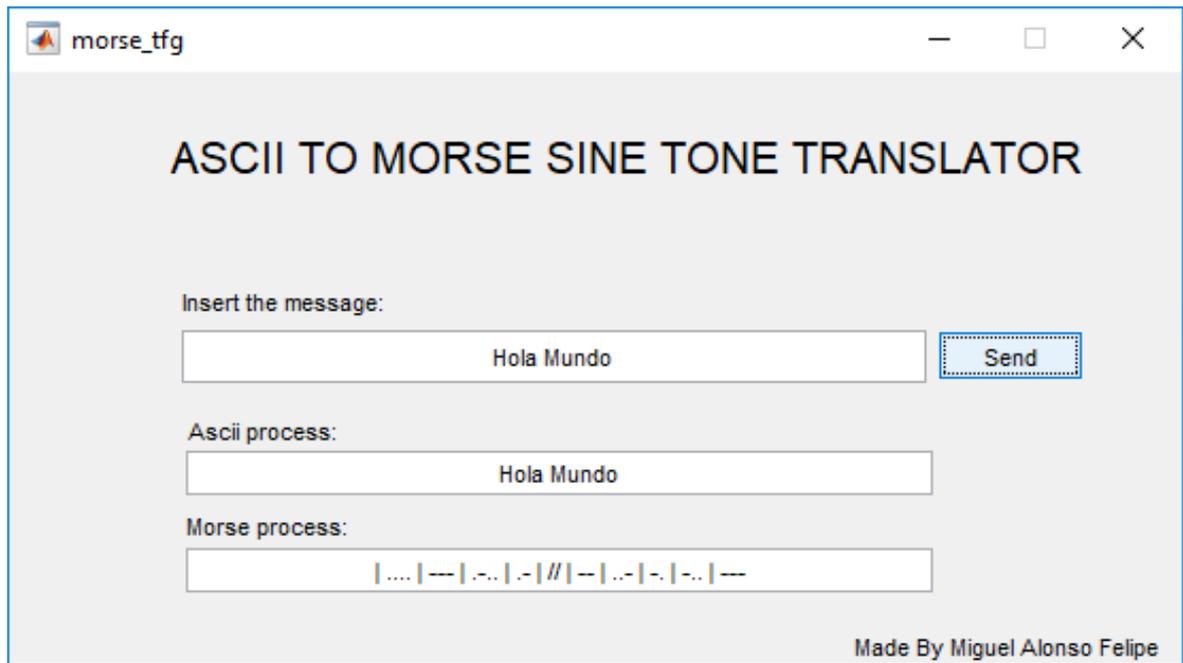


Figura 2.39: Programa de Matlab generador de Morse con tonos de distinta frecuencia.

función, estos son en realidad una señal cuadrada, es decir, para generar un tono puro necesitaríamos realizar un filtrado paso bajo para eliminar los armónicos en los múltiplos impares de la frecuencia fundamental que podrían perturbar el funcionamiento del transmisor FM y de su modulación. Un filtro a tan bajas frecuencias necesitaría de un orden lo suficientemente elevado como para adquirir la selectividad frecuencial necesaria. A consecuencia de este inconveniente y debido a que se busca minimizar la carga computacional sobre el Arduino como ya se ha comentado con anterioridad, se descarta esta idea de la implementación final.

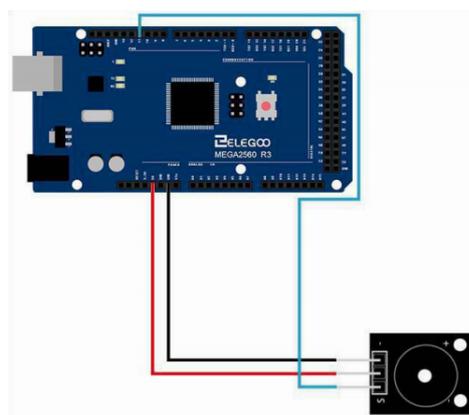


Figura 2.40: Esquema eléctrico del conexionado del *Buzzer*.

Se adjunta con objetivo de clarificar el comportamiento del código una captura del monitor serie durante la ejecución del programa. La captura se muestra en la Figura 2.41. La traducción comienza tras enviar la cadena de caracteres introducida por el

usuario.

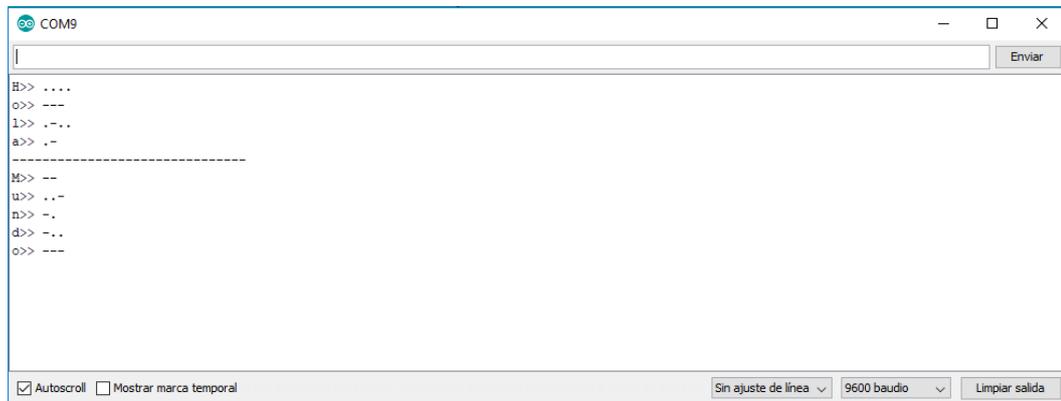


Figura 2.41: Programa de Arduino generador de Morse con tonos de distinta frecuencia.

Mejora sobre el código Octave

En vísperas de concluir el proyecto, surge una idea que mejora notablemente la fiabilidad del sistema. Esta idea se basa en utilizar una frecuencia no usada por la comunicación en ninguna de las direcciones, de forma que para realizar una petición de GPS a la boya solo haya que transmitir esa frecuencia (se ha elegido 500 Hz) y no tenga que descifrar un mensaje (en este caso el mensaje era "GPS"), ya que este método se oponía a descargar a la boya de procesado adicional. Se implementa esta utilidad **únicamente** en el código que se utilizará finalmente de **Octave**; no obstante implementarlo en C++ (Arduino) y Matlab resulta una práctica trivial que carece de mayor atención. Las únicas líneas que se añaden con respecto al código original localizado en el Anexo A.1 (a partir de la línea 21) se adjuntan a continuación (y por supuesto, hay que añadir la condición al resto del código).

```
1  % Declaramos la bandera que recoge si se ha hecho una petición
   de gps o no
2  gps = 0;
3
4  % Si llega una petición GPS se tramita
5  if(cadena == "GPS" || cadena == "gps")
6      display('GPS REQUEST SENDING...');
7      fs=8000;
8      duracion=2;
9      freq=500;
10     val=0:1/fs:duracion;
11
12     a=sin(2*pi* freq*val);
13     sound(a,fs);
14     gps = 1; % Para que no se transmita este mensaje al
               avion innecesariamente
```

15 | **end**

Se puede observar que el código es muy simple y que solo recoge si el usuario teclea "GPS" o en su defecto "gps" y genera un tono de 500 Hz. Además, añade un *flag* denominado **gps** que se utilizará para evitar que se recorra el resto del código del programa innecesariamente.

Capítulo 3

Explicación detallada del sistema de comunicación basado en boyas

3.1. Módulo transceptor aerotransportado

El sistema que irá integrado en el avión está constituido por las tecnologías de transmisión-recepción radio en la banda de FM comercial explicadas en la subsección 2.2.2. Si bien es cierto que en una implementación real se usarían bandas militares, para este prototipo se ha decidido utilizar la banda de FM comercial, ya que a efectos prácticos arroja los mismos resultados: la adquisición de módulos que trabajan en estas frecuencias es más factible y se prefiere no interferir en las bandas de frecuencias militares realizando pruebas (a pesar de que no se transmita a mucha potencia). El módulo también incluye componentes electrónicos adicionales necesarios para su funcionamiento, tales como: integrados CD4046B y TL082, ambos localizables en el Anexo C y componentes pasivos RLC.

La composición y el funcionamiento de este módulo viene descrito por el diagrama que se adjunta en la Figura 3.1.

Partiendo del canal dirección avión-boya y siguiendo el diagrama desde el **Ordenador**, se introduce el mensaje en código ASCII y se traduce a Morse con alguno de los programas desarrollados y explicados en la subsección 2.4 y cuyo diagrama de bloques se puede localizar en el Anexo D.1. Estos programas realizan la traducción del mensaje introducido por el usuario en ASCII a código Morse. El código Morse es interpretado por el programa y traducido a su vez a una señal analógica de audio, donde al punto se le asigna una frecuencia de $\approx 4,1$ KHz y a la raya de ≈ 7 KHz. Hay que tener en cuenta que para las frecuencias de transmisión se pueden elegir valores próximos a los descritos, ya que la recepción e interpretación de los tonos¹ se distingue por umbrales de tensión definidos en el código, no por el nivel de tensión exacto que da la patilla 9 (o la patilla 10 a 0,7V menos) del PLL. La señal analógica de audio generada tras el proceso de traducción se transmite por el puerto jack del **Ordenador** al del **TX Radio** instalado en la caja del módulo avión. Después, el **TX Radio** modula los tonos en FM con portadora en 108 MHz y se transmiten hacia el **RX Boya** (receptor radio

¹Véase el código de recepción del submarino en el Anexo.

DIAGRAMA DE FUNCIONAMIENTO DEL BLOQUE AVIÓN

Un diagrama realizado por Miguel Alonso Felipe



Figura 3.1: Diagrama de flujo explicativo de funcionamiento del bloque avión.

instalado en la boya).

En cuanto al canal dirección boya-avión, se comienza recibiendo la señal modulada en FM desde **TX Boya** (transmisor radio instalado en la boya) con portadora en 87,5 MHz, a una frecuencia lo más alejada posible de la de transmisión a 108 MHz para evitar cualquier tipo de interferencia². La señal modulada en FM se desmodula en el **RX Radio** (receptor de radio en el avión) y se amplifica (**AMP. DE AUDIO**) utilizando una etapa de amplificación³ con un TL082 (véase Anexo C.1). Tras amplificar la señal hasta saturarla para que la frecuencia de la señal recibida (la cual fue generada en el submarino y tendrá o bien $\approx 1,8$ KHz ó $\approx 2,8$ KHz) pueda ser interpretada por la entrada del **PLL** (dado que si el nivel de tensión de la señal es demasiado bajo el PLL no se enganchará), la tensión de salida de la patilla 10 convenientemente suavizada mediante un filtro paso bajo LC. La tensión de salida del **PLL** se introduce en el PIN analógico A0 del **Arduino**. Con la lectura del nivel de tensión se determinará si se ha recibido un punto o una raya y se mostrará la secuencia de llegada del mensaje en Morse recibido a través del monitor del **Ordenador**. Una vez recibido el mensaje y

²Al encontrarse transmisor y receptor implementados tan cerca el uno del otro hay peligro de que parte de la señal del transmisor se pueda acoplar al receptor.

³En este caso al igual que en el submarino bastará con una sola etapa de amplificación; no obstante, el resultado tras dos etapas sería idéntico, basta con que la señal sature de forma que el PLL la interprete.

tras esperar el tiempo que se defina en el código por el usuario, se interpretará que el mensaje ha concluido y se mostrará por pantalla la traducción completa del mensaje Morse a ASCII. Se ilustra el resultado en la Figura 3.2 (véase el código en el Anexo F.1).

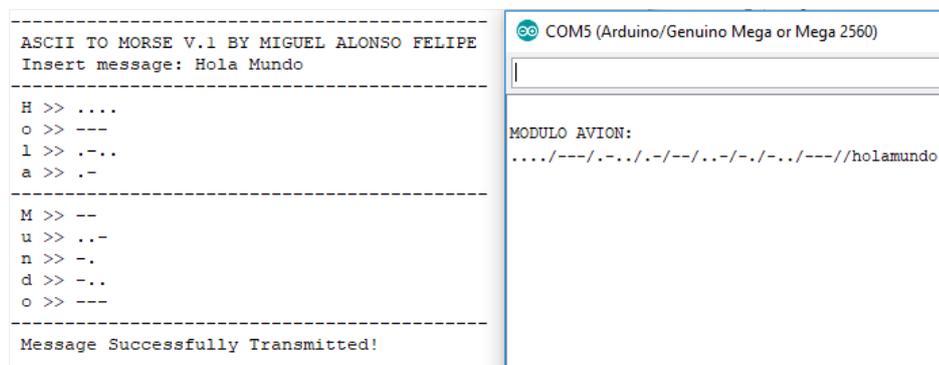


Figura 3.2: Funcionamiento del sistema de traducción vía software del avión.

Explicado el funcionamiento se procede a analizar el conexionado del montaje. Para ello se adjunta el diagrama de la Figura 3.3.

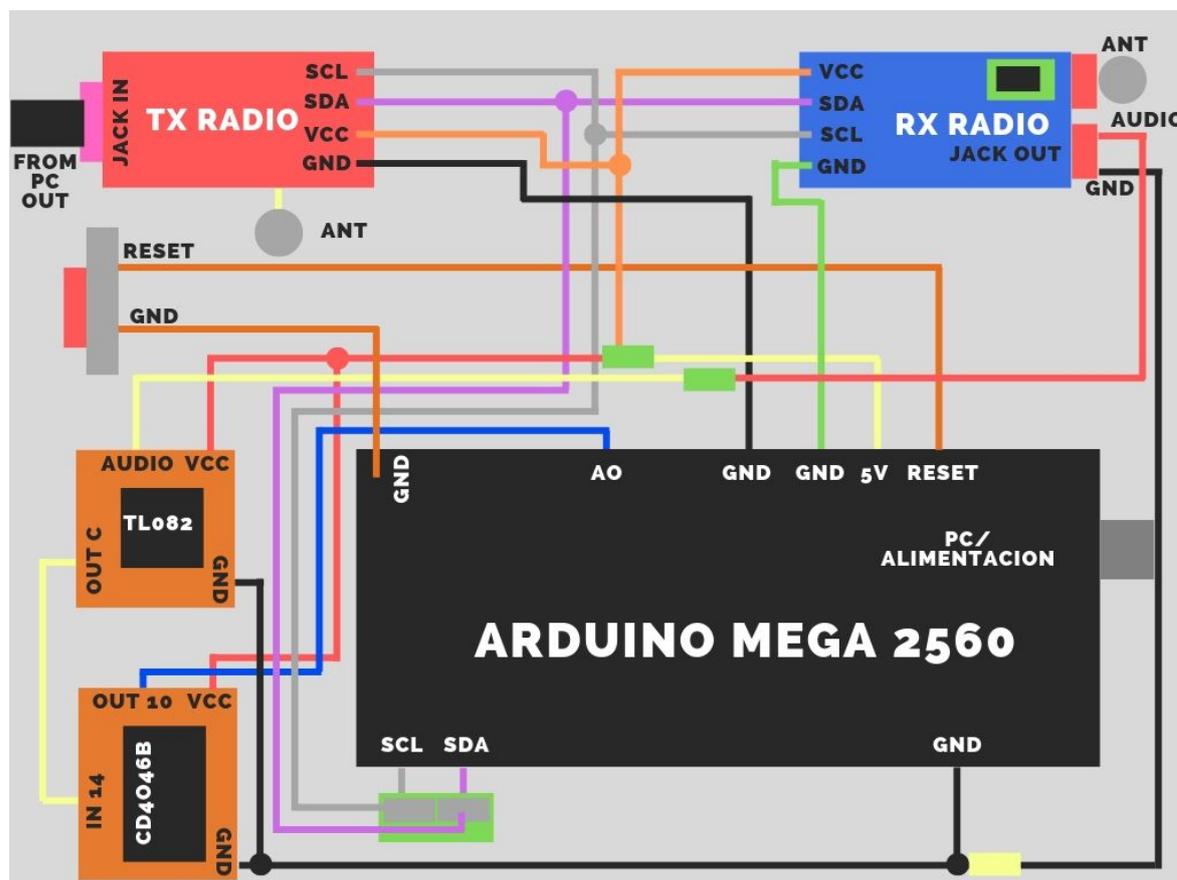


Figura 3.3: Diagrama de conexionado del módulo avión.

Siguiendo el diagrama se explica más en detalle punto a punto el conexionado:

- El cable amarillo del Arduino (PIN de 5V de salida) se usa para alimentar todos los componentes electrónicos de la caja (etiquetados con VCC en el diagrama). Todos ellos están soldados entre sí y aislados con un termorretráctil verde.
- Los pines de GND del Arduino se utilizan para dar una tierra común a todos los elementos de la caja. Todos los cables negros son tierra, **exceptuando el del receptor radio que es verde** para no confundirlo con la GND del jack out.
- Los cables marrones se usan para el botón de inicio de la transmisión (botón rojo) conectado al pin RESET de Arduino y a GND.
- Los cables morados se usan para la salida analógica SDA. Se cortocircuitan las salidas SDA del transmisor y el receptor radio y se conectan al pin SDA del Arduino.
- Los cables grises se usan para la salida analógica SCL. Se cortocircuitan las salidas SCL del transmisor y el receptor radio y se conectan al pin SCL del Arduino.
- El cable azul del CD4046B (salida de la patilla 10) se conecta con el pin analógico A0 del Arduino.
- OUT C (Amplificador de audio a través del condensador) se conecta con un cable amarillo a la entrada IN 14 (patilla 14) del PLL. Este cable conecta la salida de las dos etapas de amplificación a la entrada de enganche del PLL.
- La salida AUDIO del JACK OUT del receptor de radio se conecta con la entrada a las dos etapas de amplificación del TL082 (cable rojo soldado a un cable amarillo y aislados con un termorretráctil verde).

Se procede ahora a detallar los valores empleados para el montaje de los circuitos electrónicos auxiliares. Se recogen los valores para el circuito con el integrado CD4046B según el esquema de la Figura 3.4 (véase también Anexo C.2) en la Tabla 3.1.

Para el circuito que contiene el integrado TL082 se disponen los valores en la Tabla 3.2 siguiendo el esquema de la Figura 3.5 (véase también el Anexo C.1).

Por último se adjunta una imagen del montaje en la Figura 3.6. En la foto de la izquierda se puede apreciar la caja del avión abierta, mientras que en la foto de la derecha se aprecia la parte frontal de la caja cerrada, donde se encuentra el botón para comenzar la transmisión y la entrada de jack al transmisor. La caja también cuenta con dos antenas telescópicas. La antena frontal pertenece al transmisor y está atornillada a la caja, mientras que la otra antena pertenece al receptor y se puede conectar y extraer de la caja mediante un puerto jack. Junto a la antena del receptor se encuentra una salida jack de audio (de donde se puede escuchar qué está recibiendo el receptor). A la izquierda del jack de salida del receptor se localiza la alimentación de la caja (alimentación del Arduino MEGA2560 entre 6-12V).

CD4046B (PLL)	
L	1 mH
C1	10 nF
C2	1 nF
CF1	1 nF
CF2	100 nF
R1	47 K Ω
R2	82 K Ω
R3	100 K Ω
R4	68 K Ω

Tabla 3.1: Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.

TL082 (una etapa)	
C1	220 nF
C2	220 nF
R	1,2 K Ω
R1	10 K Ω
R2	220 K Ω

Tabla 3.2: Los valores RC empleados para el diseño del circuito con el integrado TL082.

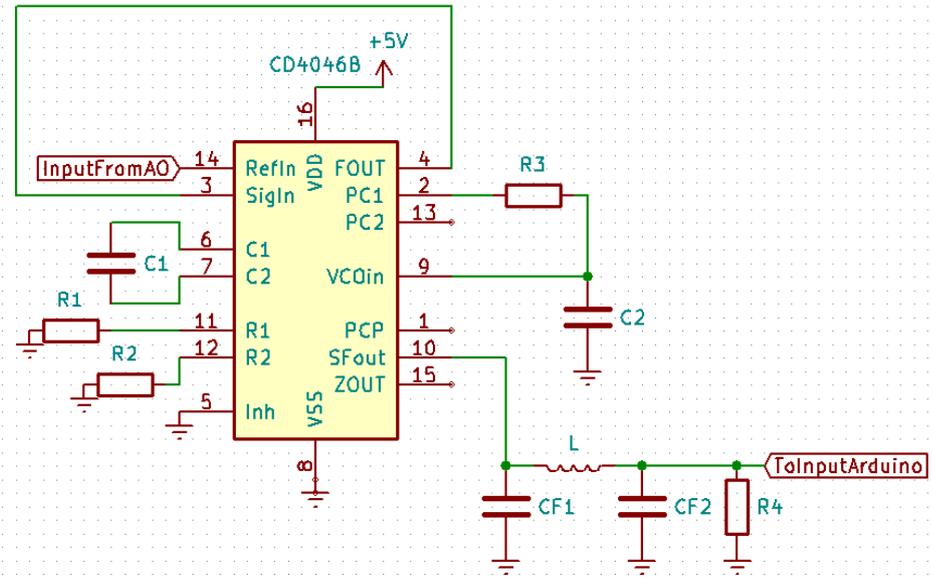


Figura 3.4: Esquema electrónico del conexionado del integrado CD4046B.

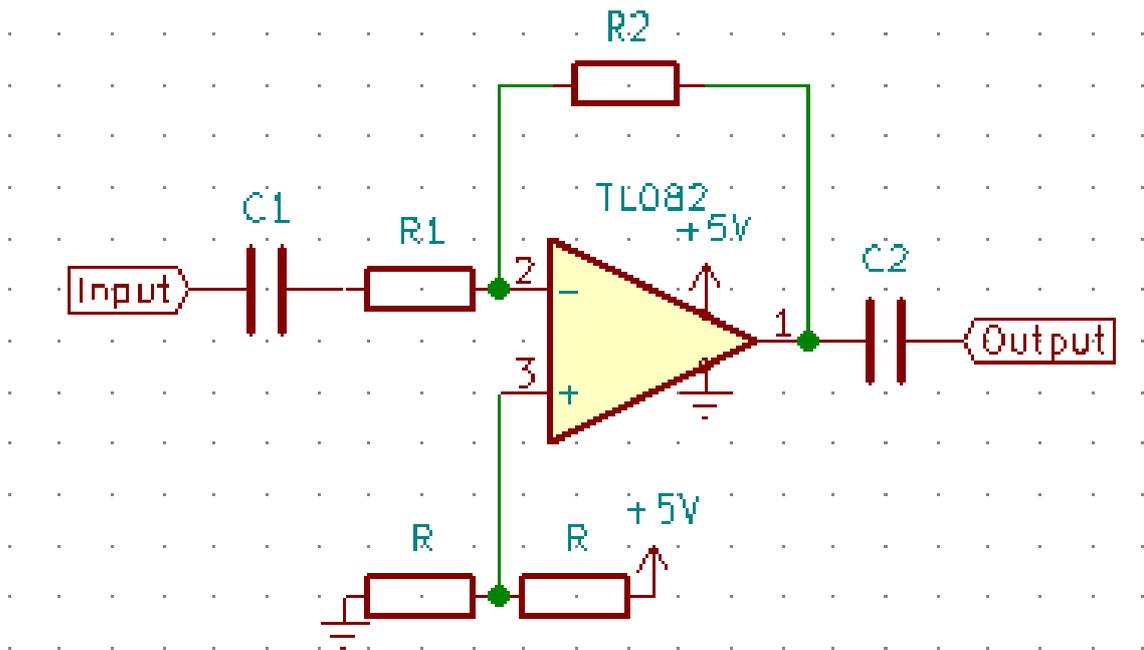


Figura 3.5: Montaje con una etapa de amplificación usando el TL082.

Con objetivo de clarificar las entradas y los elementos que componen el exterior de la caja se adjunta la Figura 3.7.

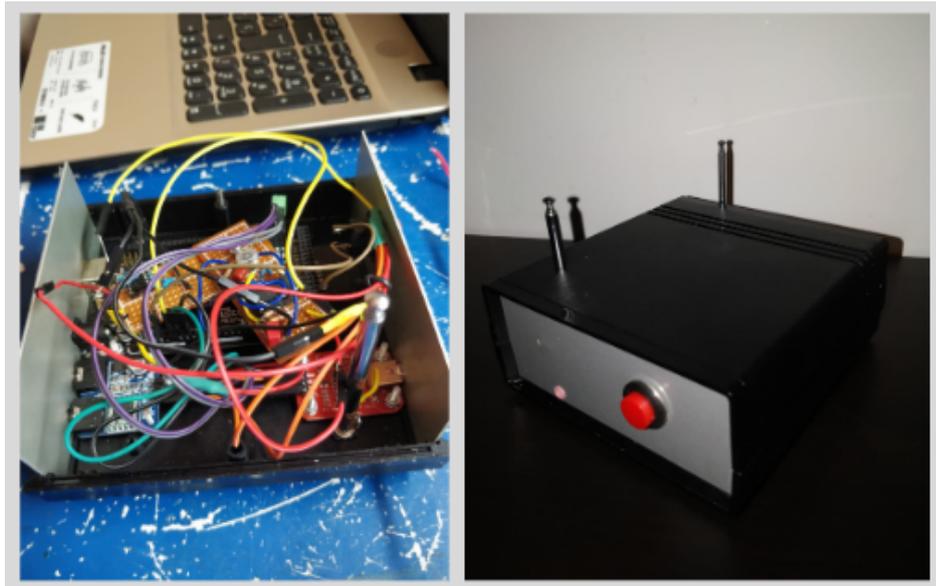


Figura 3.6: Fotos del montaje de la caja del avión.



Figura 3.7: Vista frontal y dorsal de la caja del avión.

3.2. Módulo boya

El módulo boya es el sistema con mayor relevancia en este proyecto, dado que es el nexo entre los dos extremos de la comunicación (avión y submarino). Este sistema implementa un transceptor radio, un transceptor acústico y un módulo GPS, lo que convierte a esta caja en el montaje más complejo del proyecto. Para comprender la complejidad que conlleva la tecnología implementada en este módulo se adjunta su diagrama de funcionamiento en la Figura 3.8.

cide con la cadena en ASCII "GPS", entonces la boya obtiene su ubicación del módulo **GPS** que integra. **Este método de petición GPS se mejora por una idea que involucra el haber recibido o no un tono de una frecuencia completamente distinta a la que se usa en las comunicaciones⁴, en este caso se emplea una frecuencia de 500 Hz.** De este modo, se consigue un sistema más fiable y más eficiente, dado que la boya no estará continuamente traduciendo los mensajes (idea que se opone a descargar a la boya de trabajo). Una vez obtenida la ubicación, el módulo **Arduino MEGA2560** se encarga de conmutar la comunicación de la transmisión procedente del avión con destino al submarino con la transmisión de la ubicación GPS de la boya solicitada por el submarino. En esta situación el **Arduino MEGA2560** se encarga de codificar las coordenadas extraída del módulo GPS a código Morse. Para ello se diseña un método que simplifica la notación con intención de evitar la necesidad de añadir nuevos símbolos para definir la coma decimal o el signo que pueden llevar las coordenadas GPS, tal y como se ilustra en la Figura 3.9. Una vez traducido a Morse, el **Arduino MEGA2560** pone un pin digital (en el caso de este proyecto se usa el pin 13) en alta (5V) o en baja (GND común del circuito proporcionado por el **Arduino MEGA2560**). La salida de este pin digital se usa para introducir un nivel de tensión en el **VCO** (integrado en el CD4046B, cuya información se encuentra en el Anexo C.2). Dado que el pin solo se puede escribir en alta(5V) o en baja(0V) se emplea un divisor resistivo que proporcione a la entrada del **VCO** (patilla 9 ó 10 del CD4046B) la tensión necesaria para que este proporcione a su salida (patilla 4) la frecuencia que se requiere para codificar el punto (4100 Hz) o la raya (7000 Hz). La salida de la patilla 4 será una señal cuadrada y por consiguiente cuenta con armónicos en los múltiplos impares de la frecuencia fundamental, los cuales caen fuera de la frecuencia de enganche de los PLL y no supondrán ningún problema.

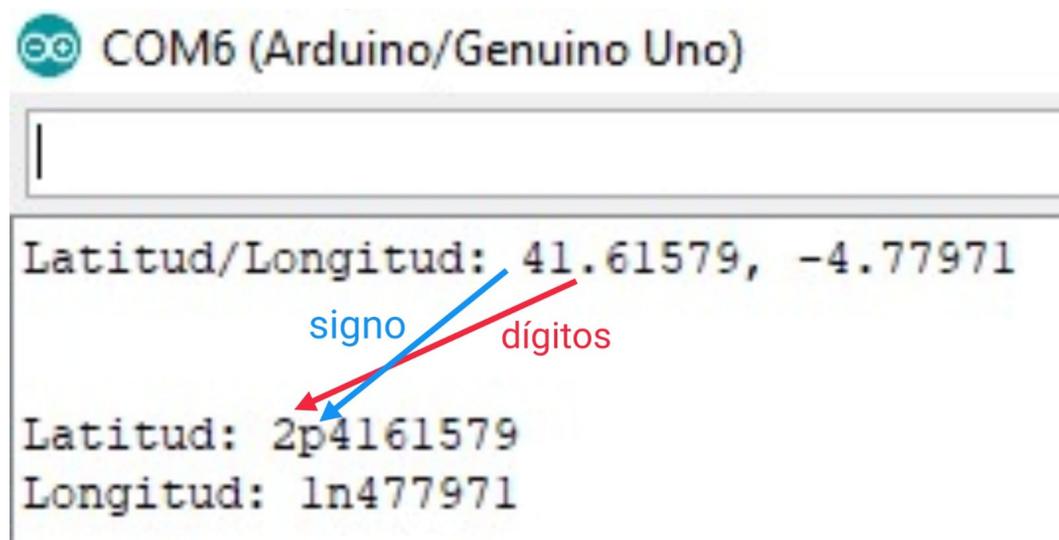


Figura 3.9: Se ilustra el método de codificación empleado. En **rojo**: se manda el número de dígitos antes de la coma decimal. En **azul**: se manda el signo de la coordenada, donde *p* equivale a positivo y *n* a negativo.

⁴Recordamos que son 1800 Hz, 2800 Hz, 4100 Hz y 7000 Hz

Para conmutar entre la transmisión del avión y la transmisión de las coordenadas GPS de la boya se empleará un relé, que siendo activado desde el **Arduino MEGA2560** podrá cortar la comunicación proveniente del avión mientras se tramite la respuesta de la ubicación GPS de la boya.

Se adjunta el diagrama de conexionado del módulo boya en la Figura 3.10 con la intención de clarificar el montaje.

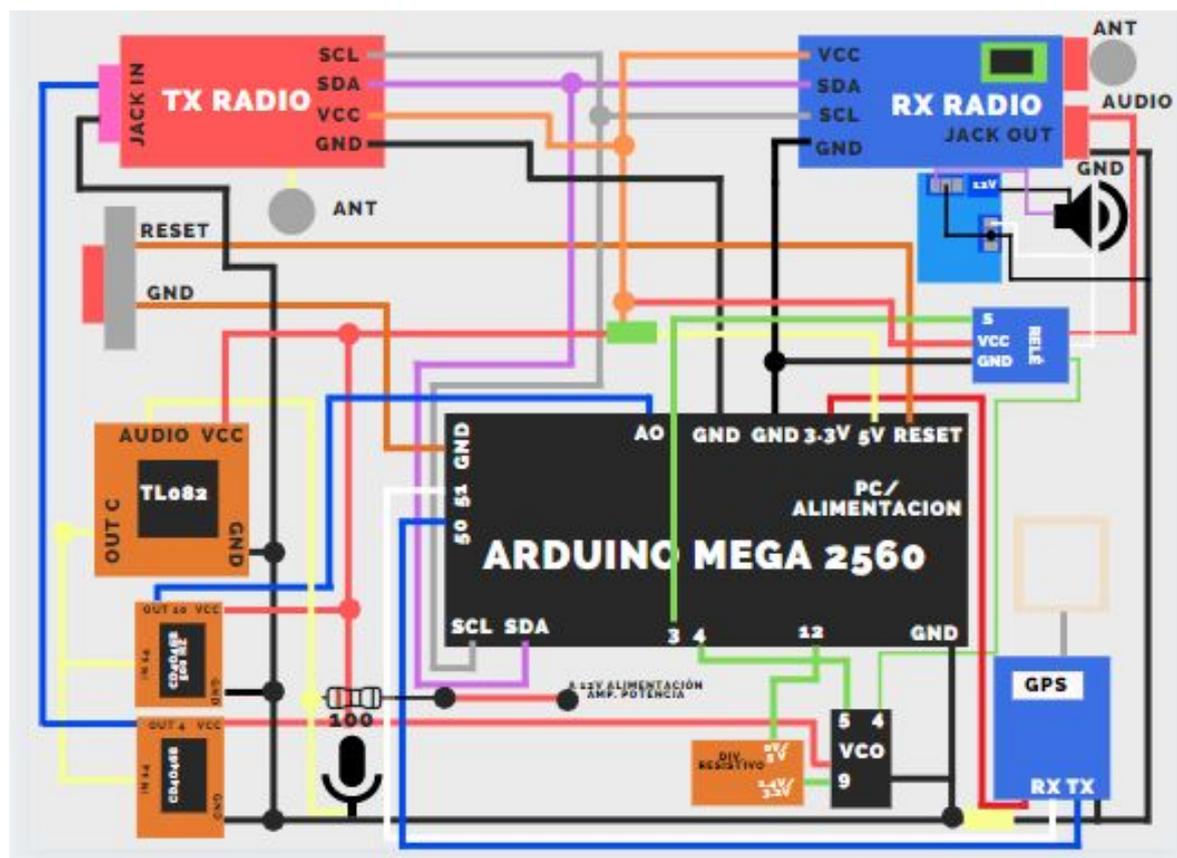


Figura 3.10: Diagrama de conexionado del módulo boya.

Siguiendo el diagrama se procede a explicar el conexionado:

- El cable amarillo (pin Arduino 5V) que se ramifica en cables naranjas para la alimentación de módulos de transmisión radio y en cables rojos para la alimentación del TL082, CD4046b y VCO. El resto de cables amarillos se usan para extraer el audio del hidrófono e introducirlo a las etapa de amplificación y para conectar las etapas de amplificación con los CD4046B.
- El cable negro (pin Arduino GND) se emplea como masa común para todo el circuito.
- Para el módulo GPS el cable rojo (pin Arduino 3.3V) se emplea para la alimentación. De este módulo salen además un cable azul (TX) conectado al pin 50 de Arduino y un cable blanco (RX) conectado al pin 51 de Arduino.

- Los cables marrones se emplean para el botón de inicio de la comunicación radio.
- Los cables morados se usan para la salida analógica SDA. Se cortocircuitan las salidas SDA del transmisor y el receptor radio y se conectan al pin SDA del Arduino. **Excepto** el cable que se usa para sacar el canal mono amplificado del amplificador de potencia hacia el altavoz.
- Los cables grises se usan para la salida analógica SCL. Se cortocircuitan las salidas SCL del transmisor y el receptor radio y se conectan al pin SCL del Arduino.
- El cable azul del CD4046B 500 Hz (salida de la patilla 10) se conecta con el pin analógico A0 del Arduino.
- El cable azul del CD4046B (salida de la patilla 4) se conecta con el transmisor radio para enviar al avión la comunicación del submarino. Se manda al avión una señal cuadrada de la frecuencia de la señal que el CD4046 recibe como entrada.
- OUT C (Amplificador de audio a través del condensador) se conecta con un cable amarillo a la entrada IN 14 (patilla 14) del PLL. Este cable conecta la salida de las dos etapas de amplificación a la entrada de enganche del PLL.
- El relé se conecta a Vcc por un cable rojo, GND por un cable negro y por un cable verde al pin que determinará cuando debe conmutar la comunicación.
- El VCO se alimenta con el cable rojo a 5V y con los cables verdes realiza el conexionado con los pines de control del Arduino y al relé. Pin 4 para habilitar o deshabilitar el VCO (a su patilla 5, para habilitar conectado a GND y para deshabilitar a 5V), pin 12 para introducir 5V o 0V en el divisor resistivo y la patilla 4 del VCO con una de las entradas de conmutación del relé.
- El cable blanco saliente del relé es el cable que conmuta las entradas al altavoz desde el VCO o desde el receptor.

El módulo boya está constituido por numerosos montajes básicos analizados a lo largo de este documento. La mayoría de estos montajes se basan en el integrado CD4046B y en el módulo boya se reúnen tres diseños electrónicos basados en este integrado. En la boya se emplean dos PLL y un VCO utilizando el CD4046B, dónde los dos PLL se usan para detectar los tonos provenientes del submarino y el VCO para generar los tonos que se transmitirán al submarino en el caso de que este solicite la ubicación GPS de la boya. El primer PLL que se analiza es el empleado para recibir los tonos provenientes del submarino. Estos tonos son de 1800 Hz y 2800 Hz, que son las frecuencias a las que se engancha el módulo avión en el otro extremo y a las cuales la boya debe ser sensible para poder introducir las en el transmisor radio (como el analizado en la sección 3.1). El segundo PLL se diseña para detectar la solicitud de GPS; lo que supone un diseño de enganche para el tono de 500 Hz (como el explicado en el punto 2.2.3). Se adjunta el esquema del montaje electrónico del PLL y sus respectivos valores en la Figura 3.11 y las tablas 3.3 para las frecuencias del avión y 3.4 para la detección del tono de 500 Hz de la ubicación GPS.

CD4046B (PLL)	
L	1 mH
C1	10 nF
C2	1 nF
CF1	1 nF
CF2	100 nF
R1	47 K Ω
R2	82 K Ω
R3	100 K Ω
R4	68 K Ω

Tabla 3.3: Los valores de los componentes electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.

CD4046B (para GPS)	
L	1 mH
C1	10 nF
C2	6,8 nF
CF1	10 nF
CF2	100 nF
R1	270 K Ω
R2	∞
R3	82 K Ω

Tabla 3.4: Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz.

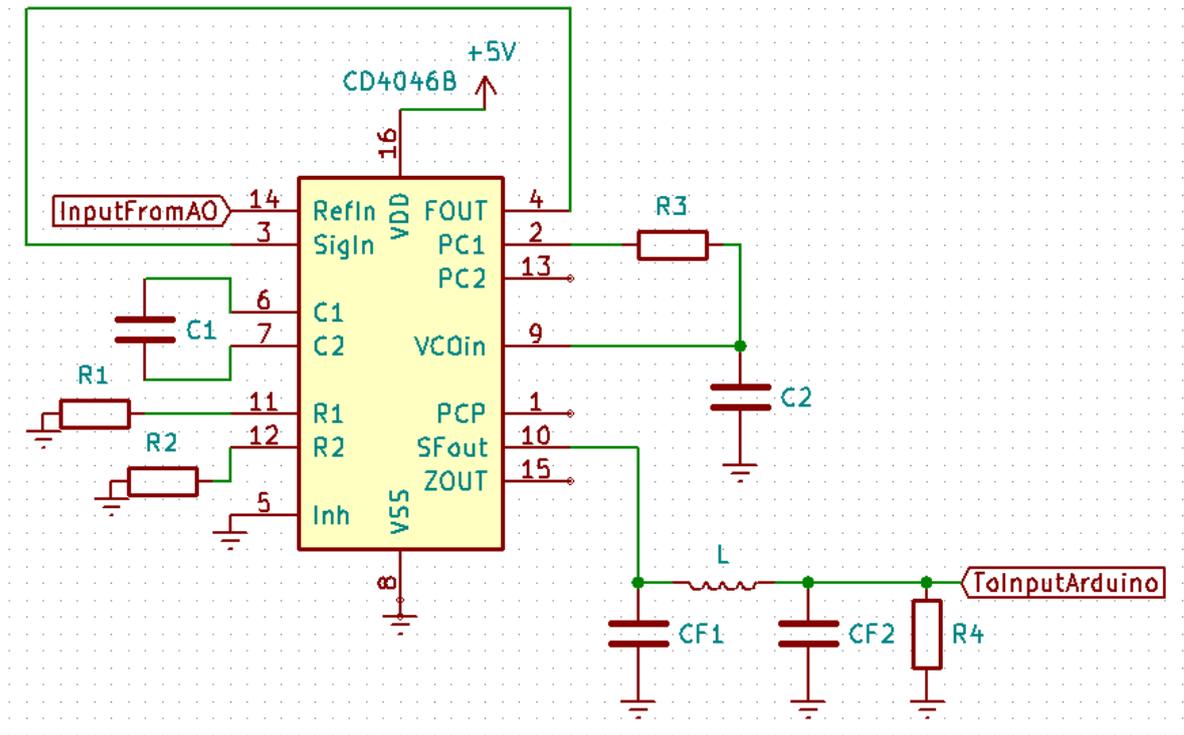


Figura 3.11: Esquema electrónico del conexionado del integrado CD4046B.

Explicados los PLL utilizados en el módulo boya se procede a analizar los valores del VCO empleado para generar los tonos que codifican la información de la ubicación GPS (como se explica en el punto 2.4.1). Estos tonos serán de 4100 Hz para el punto y 7000 Hz para la raya, frecuencias a las que la recepción acústica del submarino puede engancharse con su PLL. Se adjunta el esquema del VCO en la Figura 3.12 y los valores empleados en la tabla 3.5.

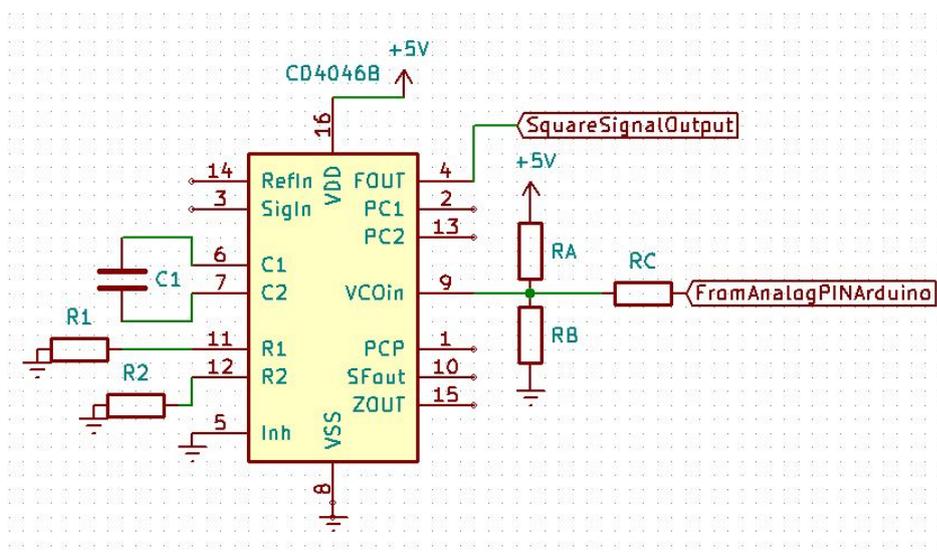


Figura 3.12: Esquema de conexionado del integrado CD4046B empleando su VCO.

CD4046B (VCO)	
C1	6,8 nF
R1	47 K Ω
R2	82 K Ω
RA	12,86 K Ω \rightarrow 12 K Ω
RB	10 K Ω
RC	10 K Ω

Tabla 3.5: Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.

Por último se analizan los módulos empleados para la amplificación en la boya. En el canal de bajada (boya-submarino) se emplea un amplificador de potencia comercial como el analizado en el punto 2.2.3. Para el canal de subida (boya-avión) se emplea una amplificación en dos etapas con un TL082 como se muestra en la Figura 3.13 con los valores de la tabla 3.6.

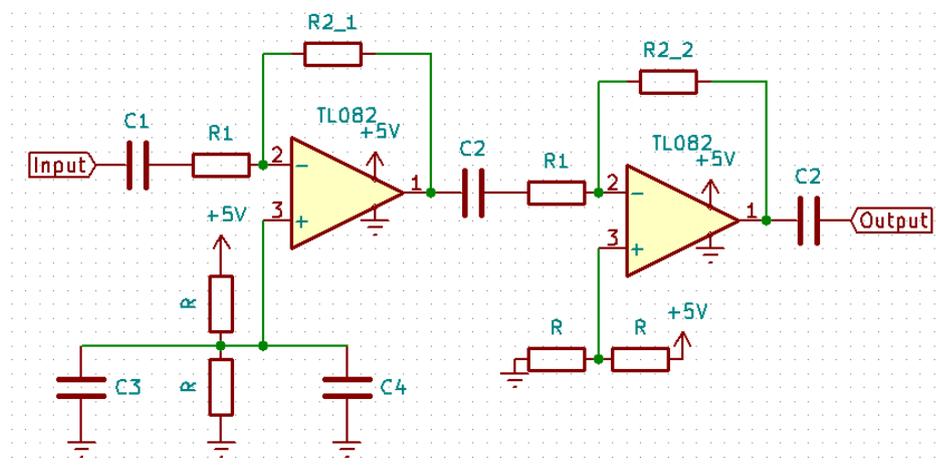


Figura 3.13: Montaje total de dos etapas de amplificación usando el TL082.

A diferencia de las otras cajas el módulo boya no dispone de un ordenador que la alimente. Se ha empleado una fuente de alimentación de 12V con 100 W de potencia de salida para abastecer al amplificador de potencia de la caja a 12 V y con un convertidor DC-DC el microcontrolador Arduino a 8V. Se muestra en la Figura 3.14 la fuente de alimentación empleada y en la Figura 3.15 el convertidor dc-dc.

TL082	
C1	100 nF
C2	220 nF
C3	680 pF
C4	1 μ F
R	1,2 K Ω
R1	10 K Ω
R2.1	220 K Ω
R2.2	47 K Ω

Tabla 3.6: Los valores RC empleados para el diseño del circuito con el integrado TL082.



Figura 3.14: Se muestra la fuente de alimentación de 12V empleada.

Para facilitar la comprensión al lector del montaje total se adjuntan las Figuras 3.16 y 3.17 para ilustrar la apariencia exterior e interior del módulo boya. Se puede observar que es la caja más voluminosa de las tres debido a que concentra las tecnologías presentes en las cajas del avión y el submarino a parte de la necesaria para implementar el módulo GPS. La caja cuenta con un botón de activación rojo para iniciar el módulo de transmisión FM al igual que la caja del avión. Otro dato relevante sobre la caja es que presenta todas las entradas en el frontal (el hidrófono y la alimentación del Arduino y del amplificador de potencia) y la salida (altavoz) en el dorsal junto con el regulador de potencia de esta.



Figura 3.15: Se muestra el convertidor dc-dc que reduce la tensión de 12V a 8V para alimentar el Arduino.



Figura 3.16: Fotos del montaje de la caja de la boya.

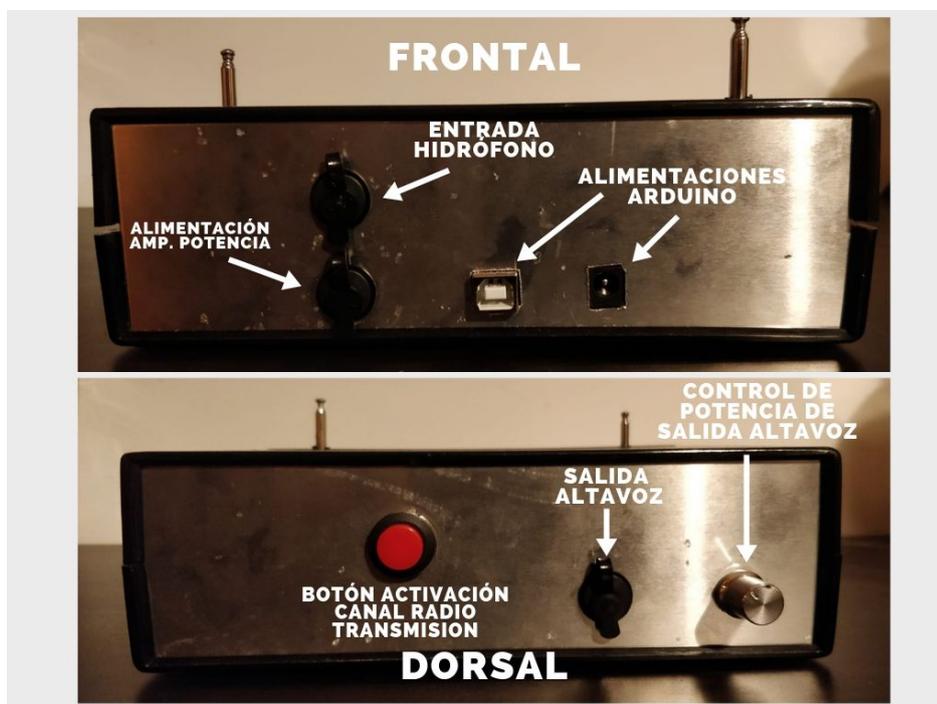


Figura 3.17: Fotos del exterior de la caja de la boya.

3.3. Módulo transceptor acústico submarino

El sistema integrado en el submarino presenta evidentes similitudes con el sistema aerotransportado explicado en la subsección 3.1. La única diferencia es que los sistemas de transmisión-recepción radio se sustituyen por sistemas acústicos. Se adjunta el diagrama de funcionamiento en la Figura 3.18, para una mejor comprensión de la evolución del sistema al transmitir o al recibir un mensaje.

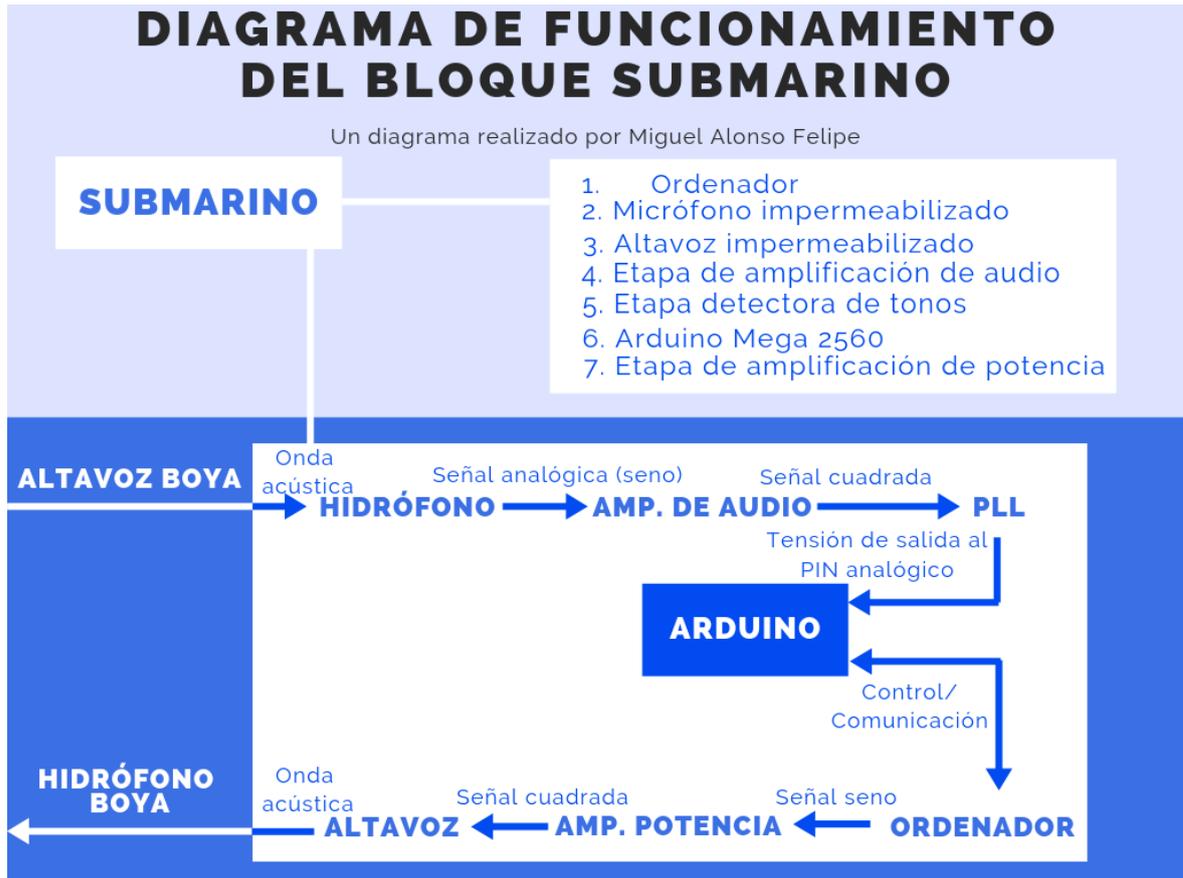


Figura 3.18: Diagrama de flujo explicativo de funcionamiento del bloque submarino.

Se comienza analizando la recepción de un mensaje (tonos analógicos transmitidos por ondas acústicas) en el **hidrófono** instalado en el submarino. Esta onda acústica llegará típicamente bastante atenuada, por consiguiente será necesario amplificarla con una doble etapa de amplificación basada en el integrado TL082 (**AMP. DE AUDIO**, véase Anexo C.1). Tras ser amplificada y saturada se obtiene una señal cuadrada que entrará en la patilla 14 del **PLL**. Si todo ha ido correctamente y la frecuencia es la adecuada, el PLL se enganchará y generará a la salida (patilla 9 ó 0.7V menos en la patilla 10) un nivel de tensión proporcional a la frecuencia de la señal cuadrada introducida. Este nivel de tensión se introduce en el pin analógico A0 del **Arduino** y cuya lectura servirá para interpretar si se ha recibido un punto o una raya. Las frecuencias que se reciben en este módulo son las generadas por el transmisor radio del módulo del avión o bien por la boya al transmitir su localización GPS, es decir, $\approx 4,1$ KHz y ≈ 7 KHz. Una vez realizada la traducción del mensaje vía software como se

recoge en el Anexo F.3, se muestra por la pantalla del **Ordenador** el mensaje recibido.

Si el mensaje se genera desde el **Ordenador** de abordo del submarino, el proceso comienza con la generación vía software del mensaje Morse acústico mediante tonos de distinta frecuencia (tal y como se recoge en el Anexo A y en el diagrama funcionamiento del Anexo D.1). Tras generarse el tono analógico se introduce en un amplificador de potencia (**AMP. POTENCIA**) que consigue un nivel de señal lo suficientemente alto como para que el **altavoz** genere ondas acústicas que alcancen sin problemas y con la calidad suficiente al **hidrófono** instalado en la boya.

Se adjunta el diagrama de conexionado del módulo submarino en la Figura 3.19 con intención de clarificar el montaje.

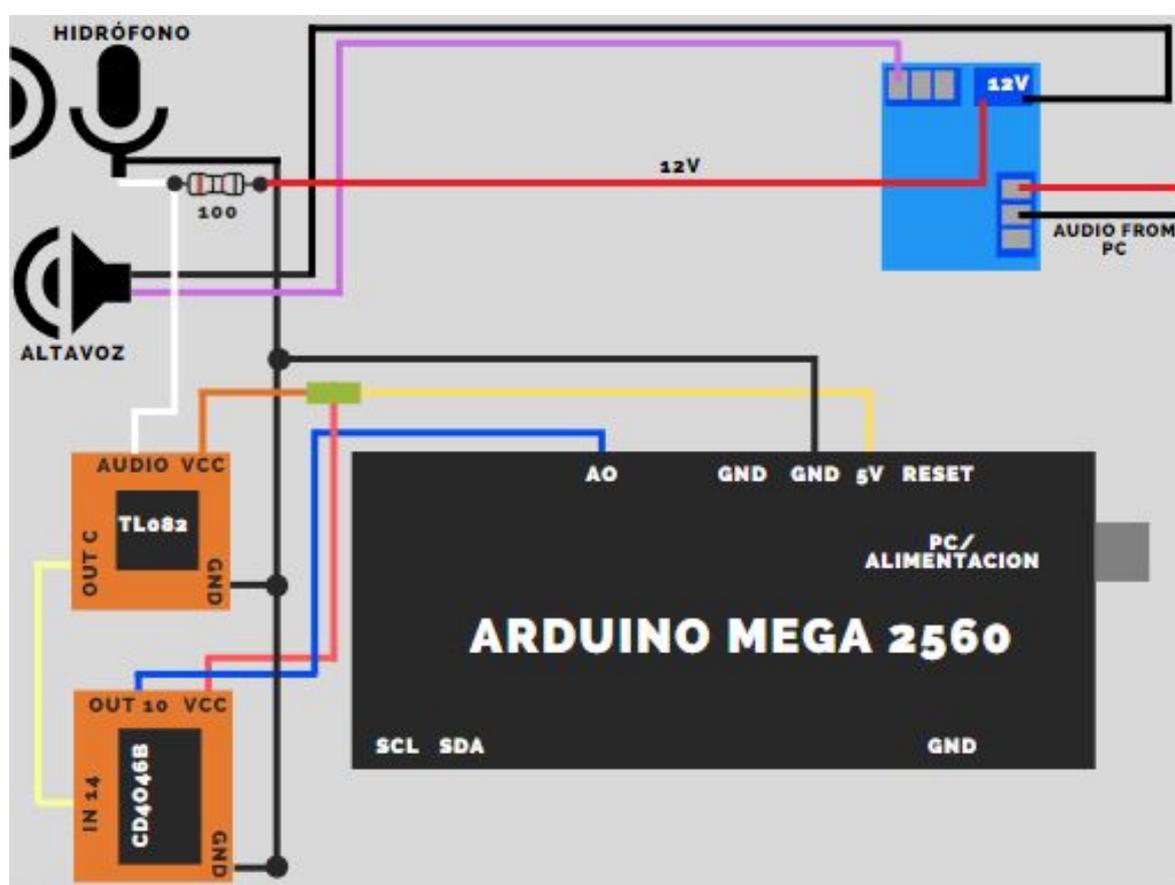


Figura 3.19: Diagrama de conexionado del módulo submarino.

Siguiendo el diagrama se procede a explicar el conexionado:

- El cable rojo del Arduino (PIN de 5V de salida) se usa para alimentar las etapas de amplificación (TL082), el CD4046B y polarizar el micrófono. Al TL082 sale el cable naranja y al CD4046B el cable rojo desde el empalme con termorretráctil verde.
- El cable negro (PIN GND del Arduino) se utiliza para proporcionar una tierra común a todos los elementos electrónicos del circuito.

- El cable azul del CD4046B (salida de la patilla 10) se conecta con el pin analógico A0 del Arduino.
- OUT C (Amplificador de audio a través del condensador) se conecta con un cable amarillo a la entrada IN 14 (patilla 14) del PLL. Este cable conecta la salida de las dos etapas de amplificación a la entrada de enganche del PLL.
- El cable blanco se utiliza para extraer la señal de audio del hidrófono e introducirlo en las etapas de amplificación implementadas con el TL082.
- El cable morado se utiliza para introducir la salida del amplificador de potencia en el altavoz.
- El cable rojo que sale del amplificador de potencia se utiliza para alimentar el micrófono a 12V y así otorgarle mayor sensibilidad.

Se procede ahora a detallar los valores empleados para el montaje de los circuitos electrónicos auxiliares. Se recogen los valores para el circuito con el integrado CD4046B según el esquema de la Figura 3.20 (véase también Anexo C.2) en la Tabla 3.7.

CD4046B (PLL)	
L	1 mH
C1	6,8 nF
C2	1 nF
C3	100 nF
R1	33 K Ω
R2	47 K Ω
R3	100 K Ω
R4	56 K Ω

Tabla 3.7: Los valores de los componentes electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B.

Para el circuito que contiene el integrado TL082 se disponen los valores en la Tabla 3.8 siguiendo el esquema de la Figura 3.21 (véase también el Anexo C.1).

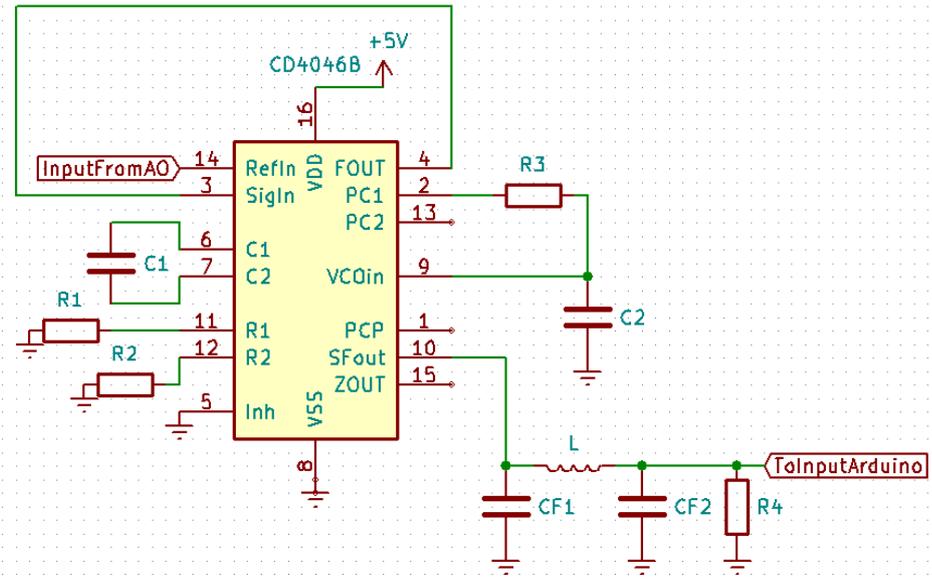


Figura 3.20: Esquema electrónico del conexionado del integrado CD4046B.

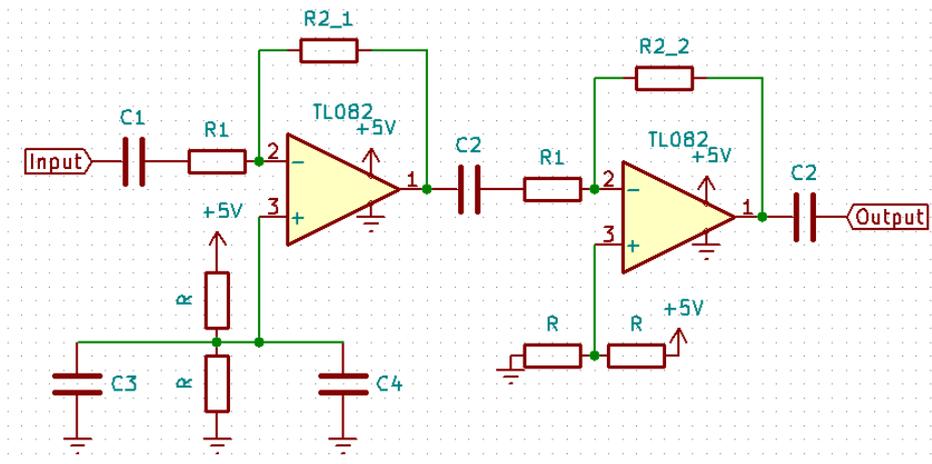


Figura 3.21: Montaje total de dos etapas de amplificación usando el TL082.

Para ilustrar el trabajo realizado se adjuntan imágenes del montaje (Figura 3.22, Figura 3.23), tanto del interior como del exterior de la caja. Se etiqueta también las entradas y salidas, donde se ha seguido como patrón situar todas las entradas en el frontal de la caja y la salida en el dorsal.

TL082	
C1	100 nF
C2	220 nF
C3	680 pF
C4	1 μ F
R	1,2 K Ω
R1	10 K Ω
R2.1	220 K Ω
R2.2	47 K Ω

Tabla 3.8: Los valores RC empleados para el diseño del circuito con el integrado TL082.



Figura 3.22: Fotos del montaje de la caja del submarino.



Figura 3.23: Fotos del exterior de la caja del submarino.

Capítulo 4

Pruebas y resultados

En este apartado se recogen los experimentos realizados con el prototipo y los resultados que estos arrojan. Para su mejor comprensión se adjuntan, tanto en este documento como en el CD, fotos y vídeos que ilustran su funcionamiento.

Dado la gran cantidad de subsistemas que componen el sistema global, se detallarán las pruebas individuales de cada elemento y después las pruebas de comunicación desde el avión al submarino y viceversa. Por último, la prueba de geolocalización entre submarino y boya.

4.1. Pruebas de subsistemas

4.1.1. Pruebas de generación de tonos mediante software

En esta subsección se llevarán a cabo las pruebas de generación de tonos con los distintos programas desarrollados en Octave, MatLab y C++ (Arduino). Finalmente se acaba empleando y afinando el código de Octave, pero se adjuntan las distintas pruebas comprobando la correcta ejecución del código en otros lenguajes por si fuese de utilidad en una futura implementación. Todo el código desarrollado se encuentra analizado con más detalle en la sección 2.4.

Generación de tonos empleando Octave

Empleando el código que se recoge en el Anexo A.1 se transmite el mensaje "*Hola Mundo*" reproduciendo los tonos del código morse resultantes de la traducción del programa con un altavoz la salida del puerto jack del ordenador. El mensaje introducido por el usuario es traducido carácter a carácter a código Morse y seguidamente a tonos para interpretar el punto y la raya. La secuencia de traducción se va imprimiendo por pantalla y los tonos se escuchan en el altavoz correctamente. Se muestra en la Figura 4.1 una fotografía extraída de la prueba de funcionamiento.

En el transcurso del TFG se añade una funcionalidad que mejora la eficiencia del montaje. Esta mejora trata de poder solicitar la ubicación GPS al sistema boya mediante un tono de 500 Hz. Todo esto implica una modificación en el código y se ilustra

la prueba de funcionamiento en la Figura 4.2.

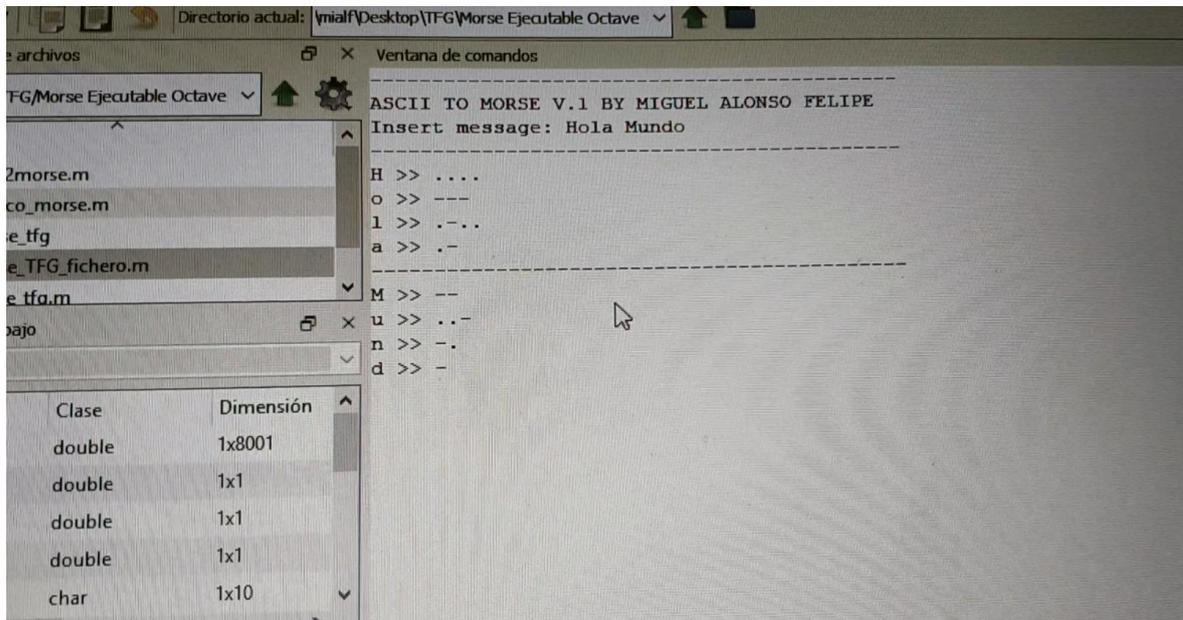


Figura 4.1: Fotografía de la prueba de ejecución del código Octave.

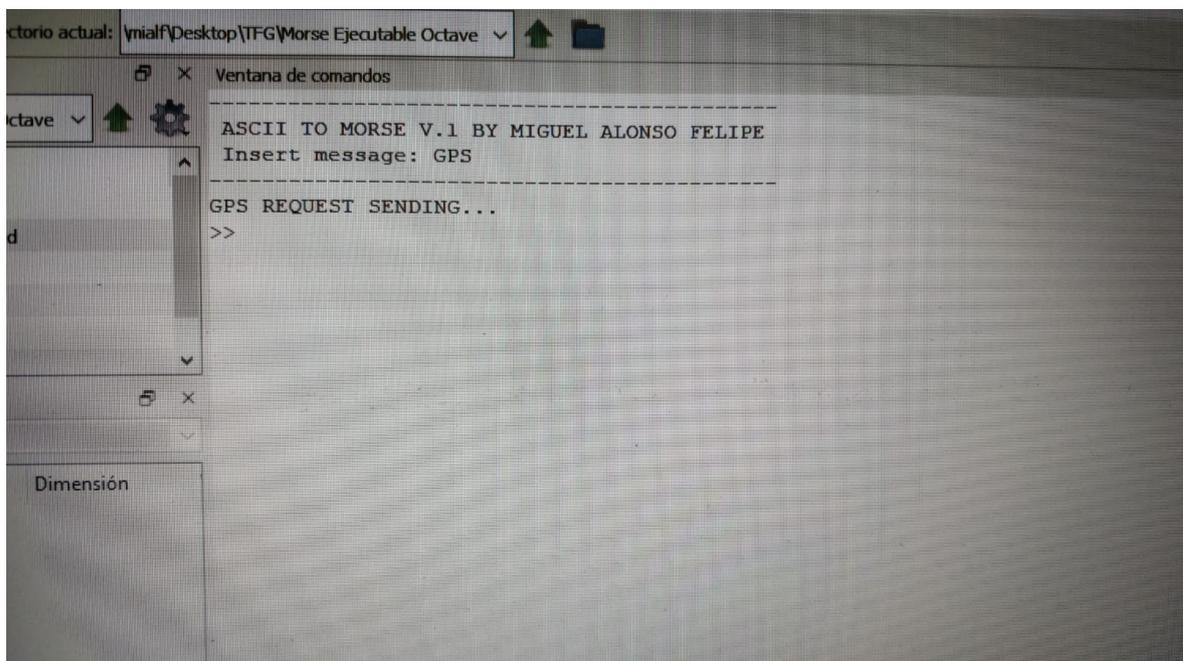


Figura 4.2: Fotografía de la prueba de la petición GPS vía software.

Generación de tonos empleando MatLab

Se comprueba el funcionamiento del entorno gráfico desarrollado para implementar el traductor de ASCII a tonos mediante código Morse. Se introduce el mensaje y se pulsa en "Send". El mensaje introducido se muestra en todo momento y en la casilla de "Ascii process" se muestra el transcurso de la traducción, del mismo modo que en la casilla "Morse process" se muestra el proceso de reproducción de cada carácter mediante su correspondiente traducción.

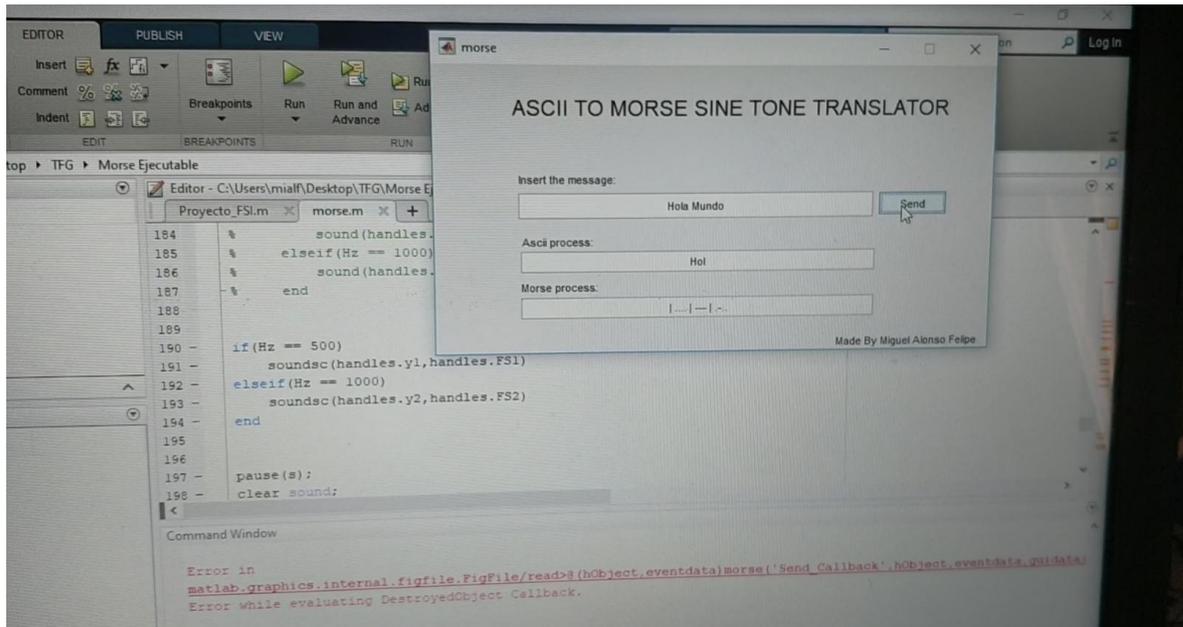


Figura 4.3: Fotografía de la prueba de ejecución del código MatLab.

Generación de tonos empleando C++ con Arduino y un *Buzzer*

Se implementa el sistema generador de tonos en un microcontrolador de Arduino MEGA2560 y se reproduce la traducción empleando un *Buzzer* (un zumbador electroacústico). El funcionamiento es muy similar al código implementado en Octave y MatLab dado que es una extrapolación del código en lenguaje M desarrollado.

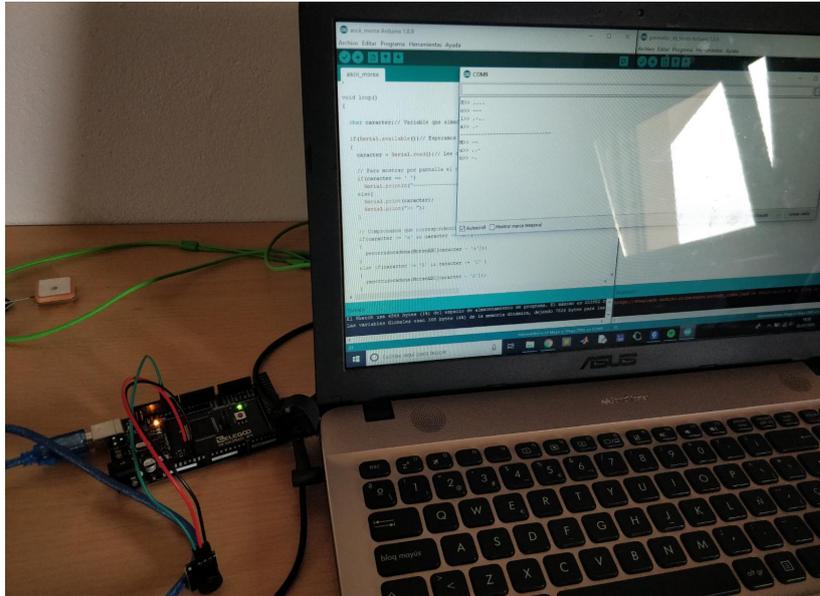


Figura 4.4: Fotografía de la prueba de ejecución del código en C++ implementado en el microcontrolador de Arduino.

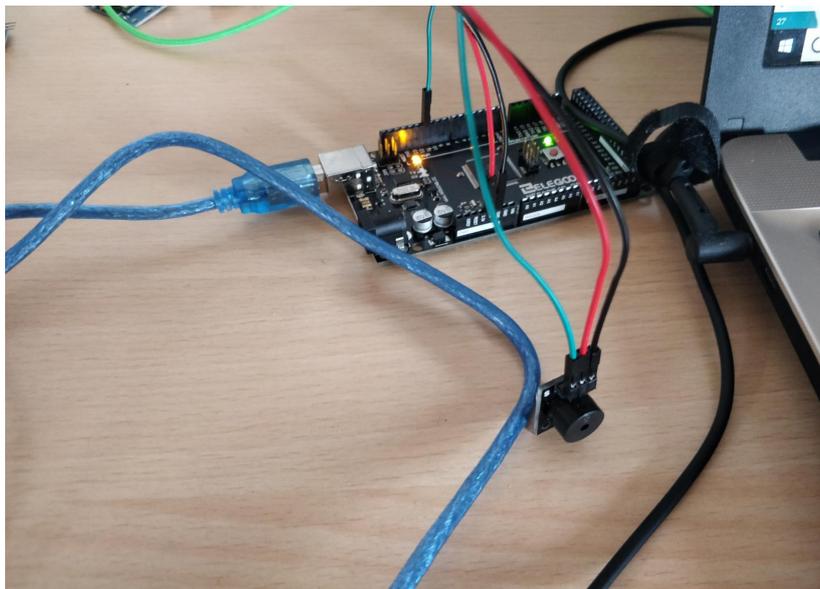


Figura 4.5: Fotografía del conexionado del *buzzer* con el Arduino.

4.1.2. Transmisión-Recepción radio en la banda de 2,4 GHz a 2,5 GHz

Para probar el funcionamiento de estos módulos basados en el integrado NRF24L01 se usa el código del programa, recogido en el Anexo B.3 y explicado en el apartado 2.2.2. Se realiza una comunicación empleando dos microcontroladores Arduino MEGA, lo cual no es necesario ya que se puede probar perfectamente con uno solo, pero es útil para analizar el funcionamiento en un caso real con dos puntos distantes.

Se realiza una prueba en la que los módulos intercambian continuamente el rol de transmisor y receptor (por consiguiente no es una comunicación *full-duplex*) transmitiendo una cadena ASCII que identifica a cada módulo. El transceptor desde el que se observa la comunicación a través del *Serial* de Arduino emite "TXRX-1" y el otro transceptor emite "TXRX-2". En un momento de la comunicación se desconecta la alimentación de uno de los módulos y se puede observar en la Figura 4.6 como el transceptor que mantiene la comunicación abierta lo detecta. En la Figura se adjunta el montaje.

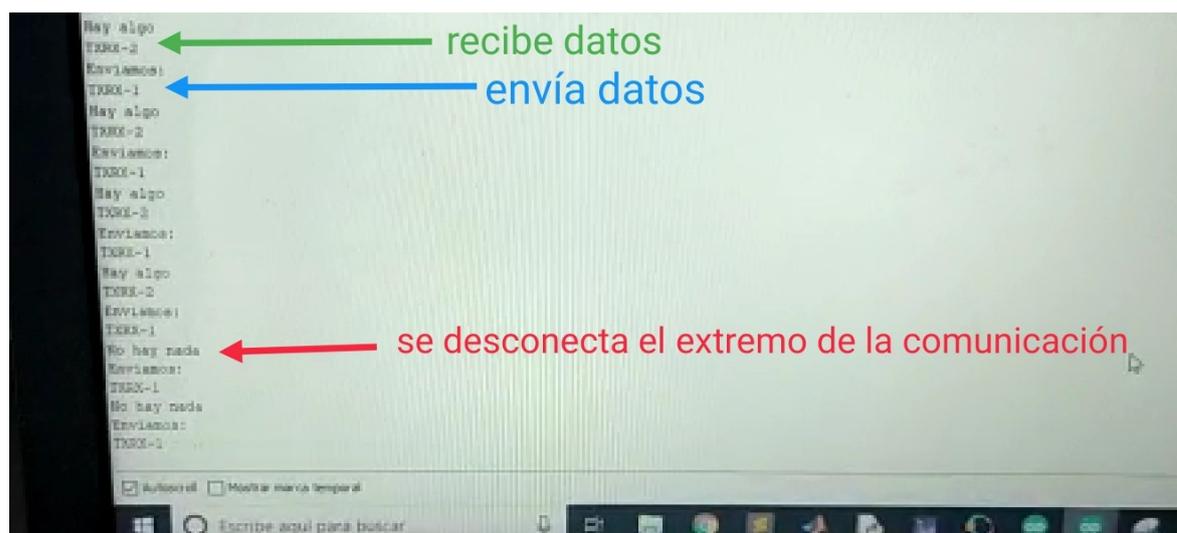


Figura 4.6: Fotografía de la prueba realizada con los módulos de transmisión radio en la banda de 2,4 GHz a 2,5 GHz. Se señala en verde la recepción de datos en el transceptor 1, en azul cuando el transceptor 1 toma el rol de emisor y en rojo el momento en el que el transceptor 2 abandona la comunicación.

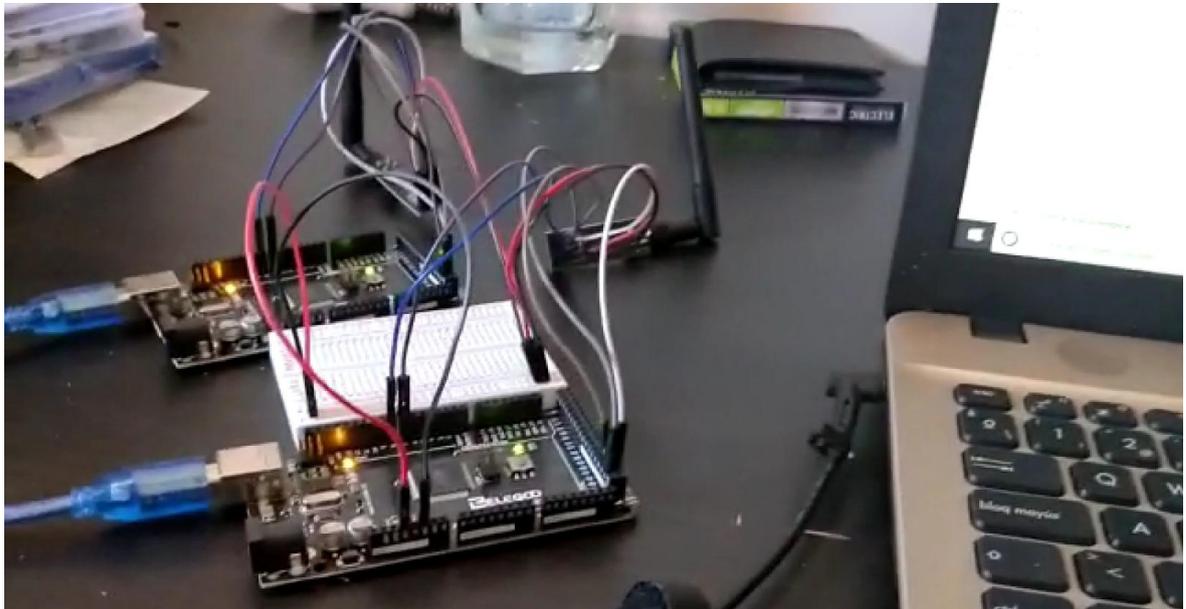


Figura 4.7: Fotografía del montaje de los módulos transceptores radio en la banda de 2,4 GHz a 2,5 GHz.

4.1.3. Transmisión-Recepción radio en la banda de FM

En esta prueba se explica el funcionamiento básico de la transmisión de tonos por el canal radio en la banda de FM comercial, generados mediante el código de Octave desarrollado. En la Figura 4.8 se observa el conexionado de los módulos radio que componen el transceptor, explicado anteriormente en el apartado 2.2.2 y en la Figura 3.3. Además, se puede observar la antena provisional del transmisor de FM cuyas dimensiones se calcularon como se explica en la Introducción 1.

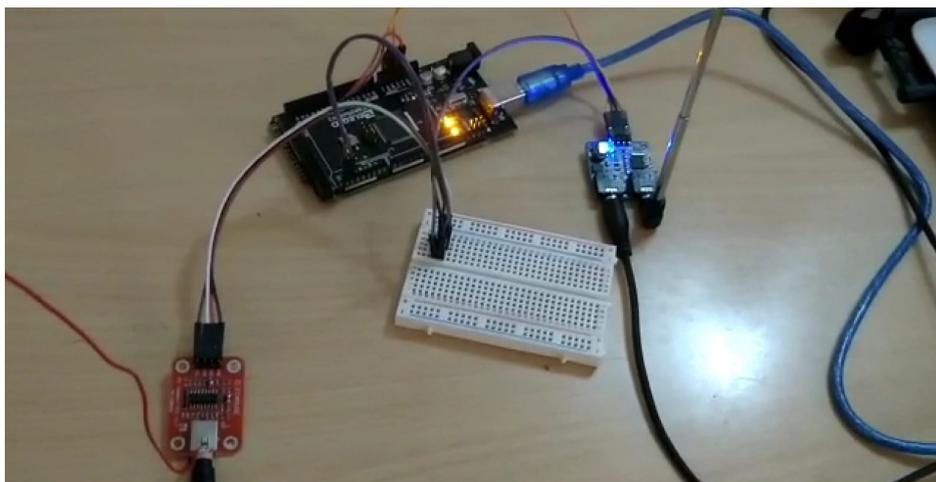


Figura 4.8: Fotografía de la prueba realizada con los módulos de transmisión radio en la banda de FM y la antena provisional del transmisor de FM.

La transmisión se realizó desde un módulo transmisor KT0803K hacia el receptor TEA5767, ambos controlados por un Arduino MEGA y presentados en el apartado 2.2.2. Se enviaron los tonos desde el ordenador a la entrada del transmisor FM y a la salida del receptor FM fueron visualizados correctamente en el osciloscopio y reproducidos en un altavoz.

4.1.4. Pruebas sobre el funcionamiento del enlace radio entre el avión y la boya, incluyendo la detección de tonos en el avión

Los módulos involucrados en el experimento son: ordenador con la generación de tonos en Octave, transmisor de FM KT0803K de la caja de la boya, Receptor FM TEA5767 de la caja del avión, amplificador de una sola etapa basado en TL082 (conectado a la salida del receptor FM) en el avión, y decodificador de tonos (punto y raya) basado en un PLL construido con el circuito integrado CD4046B en el avión.

Para probar el funcionamiento se comienza transmitiendo los tonos desde el ordenador tal y como se explica en la subsección 2.4.2. Los tonos se generan utilizando un programa desarrollado con Octave en el que al introducir el mensaje (en el caso de la demostración "Hola Mundo"). Se traduce a código Morse y reproducen los tonos correspondientes de la frecuencia asignada al punto y a la raya, en este caso se asigna 500 Hz para el punto y 1000 Hz para la raya. En la prueba se usó un PLL al cual finalmente se le subiría su rango de frecuencias antes de introducirlo en la caja del avión, para captar 1800 Hz para el punto y 2800 Hz para la raya del sistema final. Estos tonos salen por el puerto jack del ordenador y se introducen en el transmisor de FM que los modulará sobre la misma portadora el receptor del avión sintoniza. En el caso del funcionamiento real que se simula en esta prueba, los tonos que salen del transmisor de la boya no son generados por un ordenador sino que llegan a través del enlace acústico con el submarino (a través del agua). Los tonos que se van reproduciendo se muestran en el osciloscopio (Figura 4.9), se pasan por una sola etapa de amplificación usando un TL082 (con una etapa es suficiente ya que el objetivo es que el PLL decodificador que le sigue reconozca la frecuencia de la señal y para ello no es preciso amplificar mucho el tono desmodulado) y se introducen en la patilla 14 del PLL. Por último el osciloscopio (conectado a la patilla 4 del PLL, Figura 4.10) muestra como este se engancha a las distintas frecuencias que recibe.



Figura 4.9: Se muestra uno de los tonos transmitidos desde el ordenador (500 Hz).

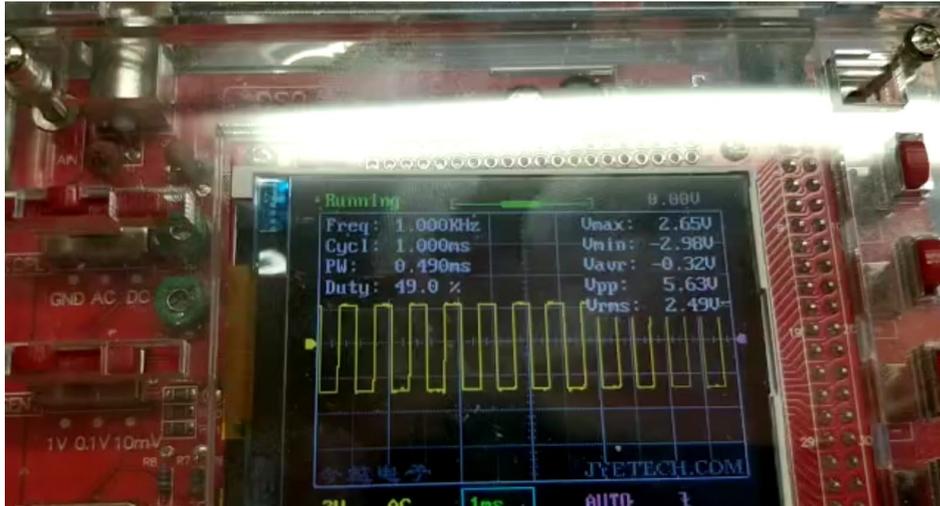


Figura 4.10: Se muestra el enganche del PLL a uno de los tonos (1000 Hz).

Los resultados obtenidos fueron satisfactorios y los voltajes medidos a la salida de la patilla 10 del CD4046B servirán para que el Arduino reconozca el punto o la raya recibida desde la boya y se lo comunique al PC del puesto de control del avión.

4.1.5. Amplificación de señales de audio recogidas con el micrófono

Los subsistemas involucrados en este experimento son: generador de tonos acústicos (un móvil en este caso), micrófono de carbón, amplificador de 2 etapas basado en TL082 y PLL decodificador de los tonos transmitidos. Se simula una comunicación entre la boya y el submarino mediante señal acústica, aunque en este caso en medio aéreo. Según la teoría presentada en el capítulo 2, la comunicación a través del agua (situación real) sería aún más favorable por realizarse en un medio más denso. De todos modos también se realizará, posteriormente, un experimento para comprobar la transmisión de señales acústicas a través del agua.

En esta prueba se ilustra como el receptor acústico consigue captar un tono de audio generado con una aplicación desde un teléfono móvil¹. Tras captar el tono, la salida de este micrófono se introduce en un amplificador de dos etapas basado en un integrado TL082 explicado en el apartado 2.2.3 y el Anexo C.1. Una vez obtenida la señal con un nivel de tensión adecuado se introduce en la patilla 14 de un PLL. La figura ilustra la sinusoide en amarillo (tono reproducido desde el teléfono móvil captado por el micrófono) y en azul la salida de la patilla 4 del PLL enganchado a la frecuencia de 800 Hz. Esta fue una de las pruebas más relevantes del proyecto, ya que arrojó resultados muy prometedores a favor de la recepción acústica empleando el micrófono extraído del teléfono antiguo.



Figura 4.11: Aplicación móvil de Android generando un tono puro de 800 Hz.

¹Se comprobó previamente con el osciloscopio que la salida del teléfono móvil fuese un tono puro.

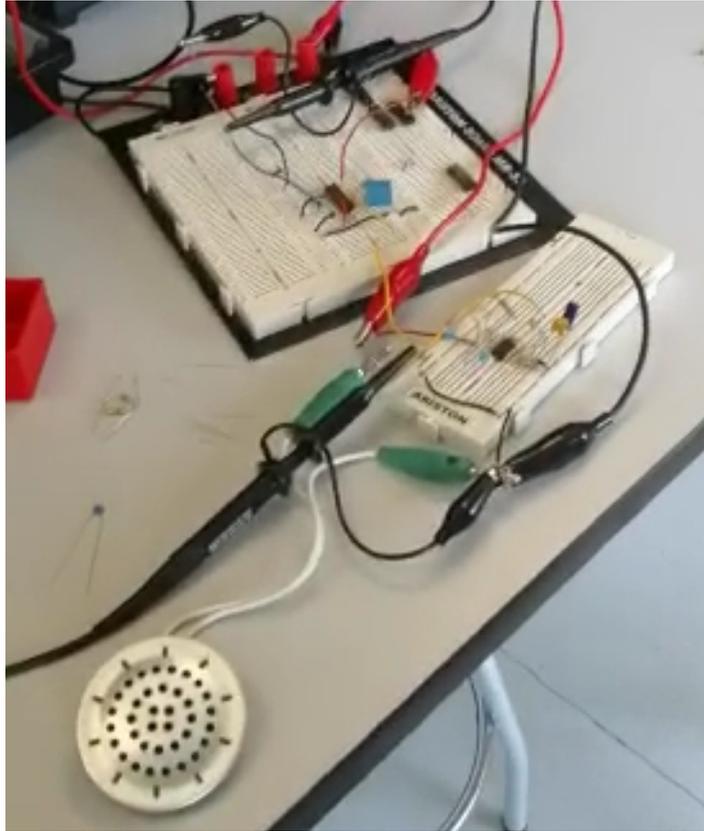


Figura 4.12: Se muestra el montaje empleado para la recepción, amplificación y enganche de un tono de 800 Hz empleando un receptor acústico.

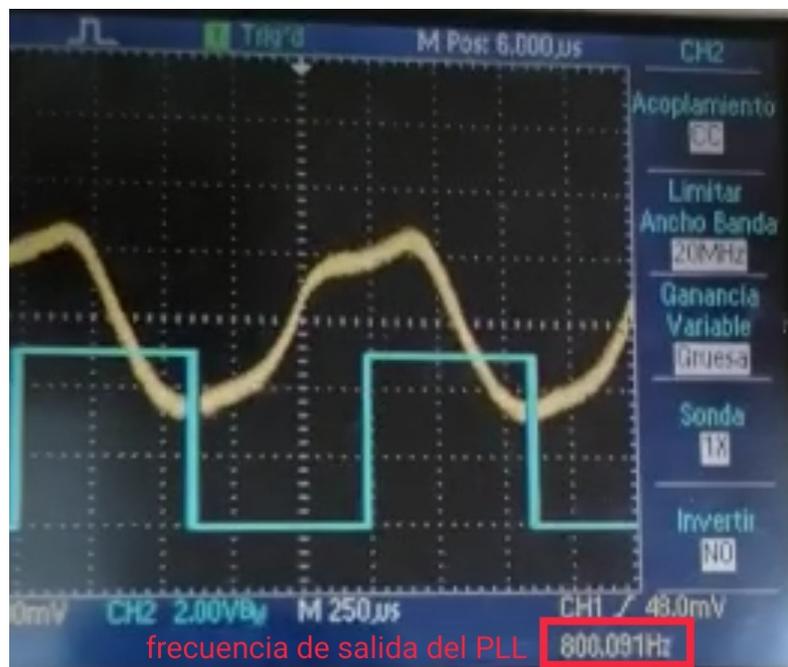


Figura 4.13: Se muestra la pantalla del osciloscopio del laboratorio de docencia, donde se grafica en amarillo el tono de 800 Hz captado por el micrófono y en azul la salida de la patilla 14 del PLL. Se recuadra en **rojo** como la frecuencia de enganche del PLL se corresponde con la transmitida desde el teléfono móvil.

4.1.6. Transmisión y recepción subacuática de tonos

En esta prueba se lleva a cabo la transmisión acústica subacuática de un tono de 500 Hz extraído del generador de funciones. Se pretende demostrar la viabilidad de dicha transmisión, aún cuando ya ha sido teóricamente justificada con anterioridad en el apartado 2.2.1. Además, también se comprueba la eficiencia de los altavoces y micrófonos elegidos para el prototipo (aunque a distancias cortas o moderadas). El tono se introduce en el altavoz sumergido y este comienza a generar una oscilación de esta misma frecuencia en su membrana mediante un electroimán que ejerce un campo magnético variable con la corriente que le atraviesa. Esta oscilación se traslada al revestimiento plástico empleado para impermeabilizarlo y de este pasa al agua. La onda se propaga transmitiendo la oscilación a través de las partículas del agua y es recogida por un micrófono de carbón impermeabilizado. Las ondas acústicas presionan la cápsula de carbón, donde los gránulos que encierra se comprimen y expanden a causa de las variaciones de las paredes de la cápsula. La concentración de los gránulos de carbón se traduce en variaciones de la resistencia de la cápsula y por consiguiente en variaciones de la corriente que la atraviesa. Por último se visualiza en el osciloscopio la salida del micrófono, donde se observa el efecto sobre la señal eléctrica de salida de la variación de presión acústica en el agua captada por la cápsula de gránulos de carbón. Se adjuntan y explican a continuación fotografías tomadas del experimento.



Figura 4.14: Se muestra el generador de funciones fijado a una frecuencia de 500 Hz para una onda sinusoidal. El tono se genera sin offset (el cursor está desviado) y a máxima amplitud.



Figura 4.15: Se muestran el transmisor y el receptor acústico impermeabilizados estableciendo el enlace de transmisión del tono de 500 Hz bajo el agua.

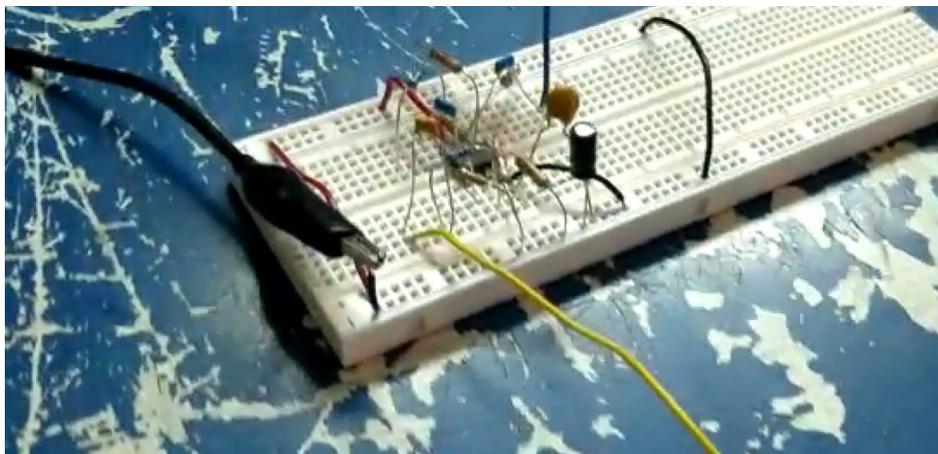


Figura 4.16: Se muestran la doble etapa de amplificación basada en el integrado TL082.

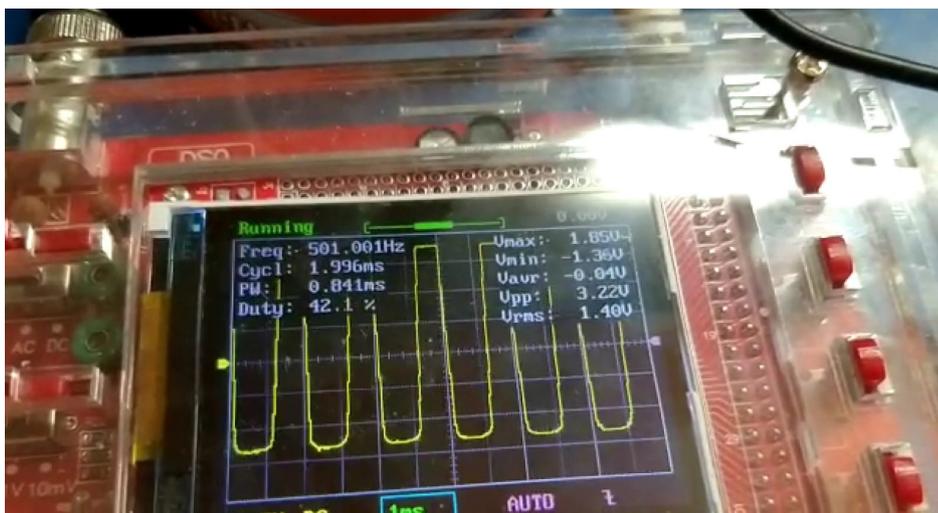


Figura 4.17: Se muestran la frecuencia de 500 Hz extraída del receptor acústico y amplificada con la doble etapa de amplificación basada en el integrado TL082.

4.1.7. Prueba con el generador de tonos basado en VCO

Para demostrar el funcionamiento de este subsistema, que es capaz de generar dos tonos de diferente frecuencia controlado por una salida digital del Arduino (0V o 5V), alimentamos el VCO expuesto en el apartado 2.4.1 (Anexo C.3) y aplicamos manualmente a la red resistiva que se conectaría a la salida digital del Arduino los voltajes de 0V y de 5V. En el experimento se midieron con un multímetro las salidas de voltaje esperadas (para las cuales fue diseñada la red de tres resistencias) y de ello pudo observarse también en el osciloscopio las correspondientes salidas digitales de onda cuadrada de (muy aproximadamente) 4,1 KHz y de 7 KHz que el VCO debía ser capaz de generar. Se adjuntan las imágenes del montaje (Figura 4.18) y de la prueba de generación de tonos (Figura 4.19 y Figura 4.20).

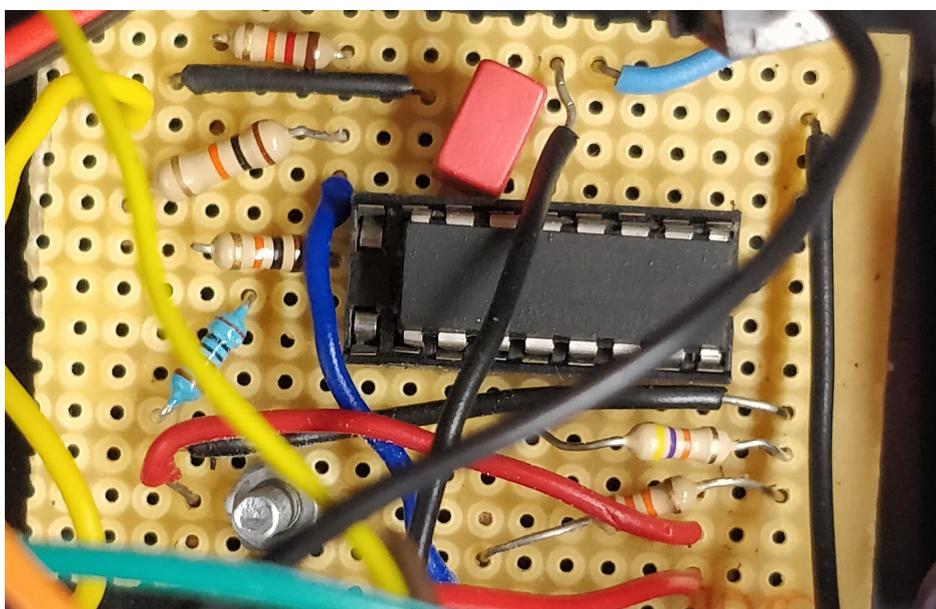


Figura 4.18: Se muestra el VCO insertado en el zócalo con su red resistiva a la entrada de la patilla 9.

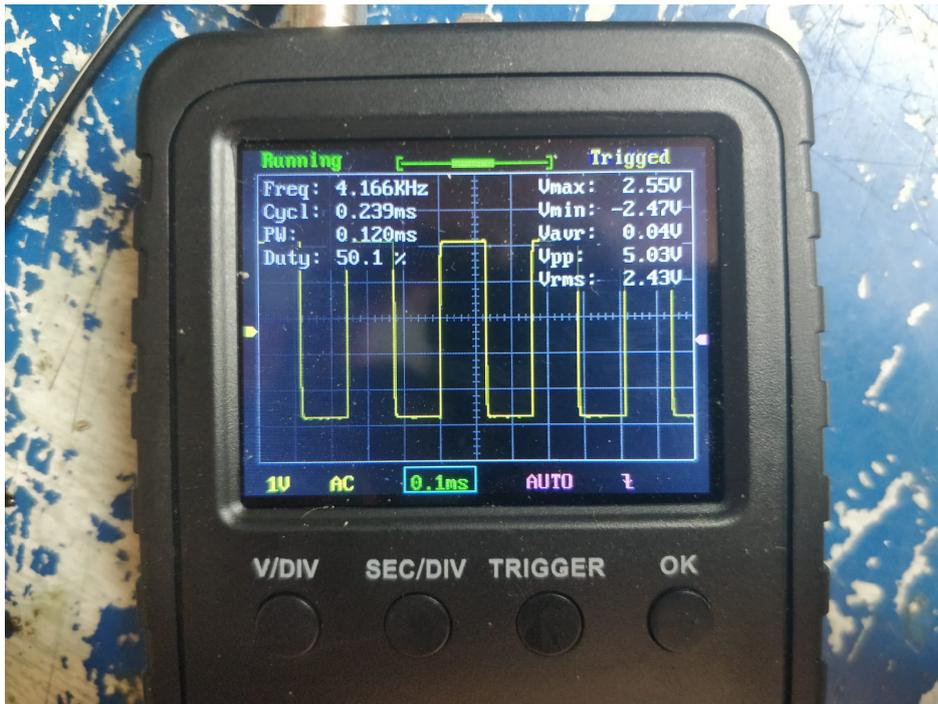


Figura 4.19: Se muestra en el osciloscopio la salida de la patilla 4 del VCO al aplicar una tensión de 5V a la entrada de la red resistiva.

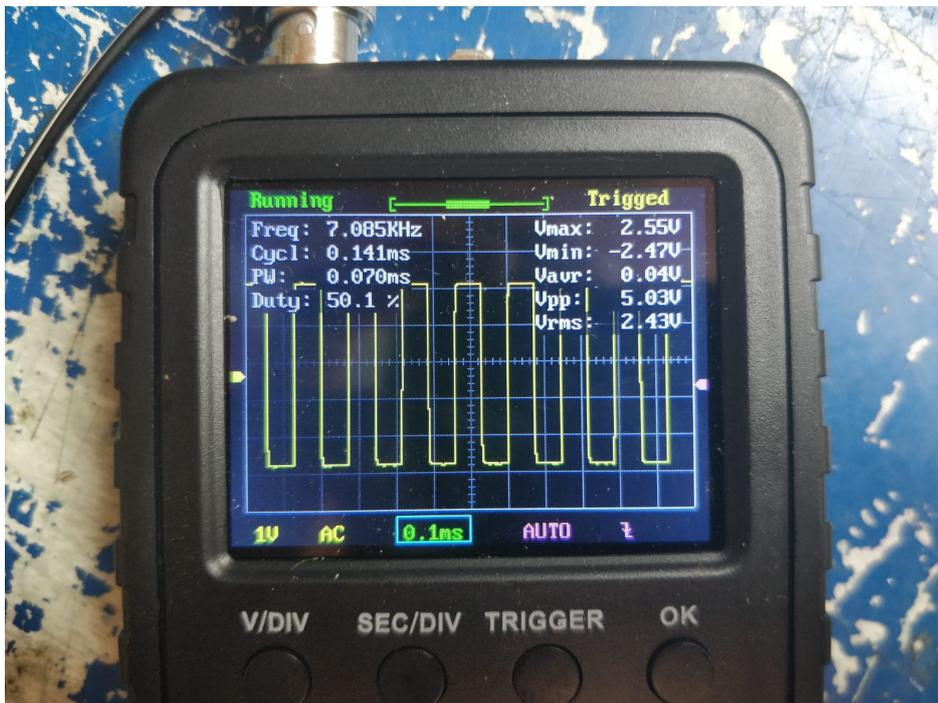


Figura 4.20: Se muestra en el osciloscopio la salida de la patilla 4 del VCO al aplicar una tensión de 0V a la entrada de la red resistiva.

4.2. Pruebas de sistema

Estos experimentos se realizaron para comprobar el funcionamiento conjunto de los subsistemas montados en las 3 cajas: aerotransportada, boya y submarina.

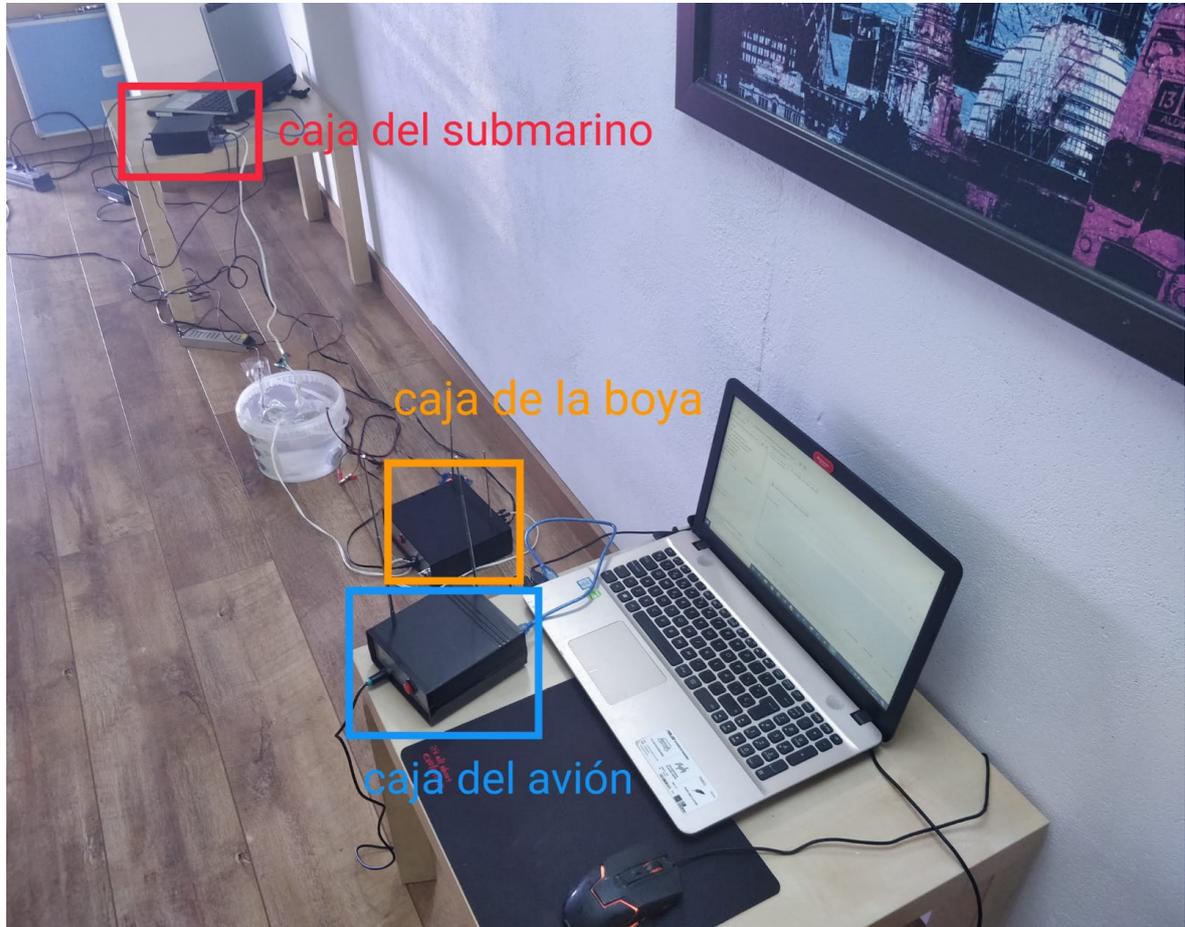


Figura 4.21: Se muestra una visión general del montaje total del sistema.

4.2.1. Prueba de recepción en submarino de mensaje del avión

Los subsistemas que están involucrados en este experimento son los siguientes:

En la caja del avión

1. Ordenador y software Octave.
2. Transmisor FM KT0803K.

En la caja de la boya

3. Receptor FM TEA5767.
4. Amplificador de potencia.

5. Altavoz impermeabilizado.
En la caja del submarino
6. Micrófono impermeabilizado.
7. Amplificador de dos etapas basado en el AO TL082.
8. Detector de tonos de 4100 Hz y de 7000 Hz basado en PLL.
9. Arduino MEGA2560.
10. Ordenador y software en C++ (Arduino).

El sistema se montó completamente y se alimentó a través del USB del PC la caja del avión, de una alimentación de 12V externa la boya, la caja del submarino, a través del USB del PC de control del submarino y de la fuente externa de 12V. Se presenta una foto del sistema totalmente montado en la fig xx, y con una cubeta conteniendo el micrófono y el altavoz impermeabilizados simulando el medio acuático.

Se procedió como sigue: tras arrancar los correspondientes programas en los dos PC y habilitar la transmisión mediante el botón de reset se escribe un mensaje en el PC del avión y se envía. Se comienza a escuchar el sonido de los tonos procedente del agua de la cubeta y se puede observar la recepción correcta del mensaje en PC el submarino.

4.2.2. Prueba de recepción en avión de mensaje del submarino

Los subsistemas que están involucrados en este experimento son los siguientes:

En la caja del submarino

1. Ordenador y software Octave.
2. Amplificador de potencia.
3. Altavoz impermeabilizado.

En la caja de la boya

4. Micrófono impermeabilizado.
5. Amplificador de dos etapas basado en el AO TL082.
6. Detector de tonos de 1800 Hz y de 2800 Hz basado en PLL.
7. Transmisor FM KT0803K.

En la caja del avión

8. Receptor FM TEA5767.
9. Amplificador de una etapa basado en el AO TL082.
10. Detector de tonos de 1800 Hz y de 2800 Hz basado en PLL.

11. Arduino MEGA2560.

12. Ordenador y software en C++ (Arduino).

El montaje y el procedimiento para realizar la prueba es análogo al explicado en el apartado 4.2.1 a excepción de la dirección de la transmisión, dado que en este caso se emplea el canal de subida (desde el submarino hacia el avión). Se observa el mensaje recibido en el ordenador conectado a la caja del avión a través del *Serial* de Arduino.

4.2.3. Prueba de obtención de posición GPS

Los subsistemas que están involucrados en este experimento son los siguientes:

En la caja del submarino

1. Ordenador y software Octave.
2. Amplificador de potencia.
3. Altavoz impermeabilizado.

En la caja de la boya

4. Micrófono impermeabilizado.
5. Amplificador de dos etapas basado en el AO TL082.
6. Detector del tono de 500 Hz basado en PLL.
7. Arduino MEGA2560.
8. VCO para generar tonos de 4100 Hz y 7000 Hz.
9. Amplificador de potencia.
10. Altavoz impermeabilizado.

En la caja del submarino

11. Micrófono impermeabilizado.
12. Amplificador de dos etapas basado en el AO TL082.
13. Detector de tonos de 4100 Hz y de 7000 Hz basado en PLL.
14. Arduino MEGA2560.
15. Ordenador y software en C++ (Arduino).

El montaje es análogo al explicado en el apartado 4.2.1, sin embargo, la caja del avión no interfiere en esta prueba. Se introduce el mensaje en ASCII "GPS" en el software desarrollado en Octave y se genera el tono de 500 Hz asociado a esta petición. La petición de coordenadas GPS se recibe en la boya y el Arduino conmuta las comunicaciones procedentes del avión a las de la transmisión de la ubicación GPS de la boya. Las coordenadas GPS se reciben en el ordenador del submarino y se muestran por el *Serial* de Arduino.

Capítulo 5

Conclusiones

Las telecomunicaciones siempre encuentran nuevos retos que resolver y en todos ellos se busca un diseño eficiente que pueda cumplir eficazmente con las especificaciones. En el transcurso de la realización de este proyecto, se ha podido comprobar que es posible establecer un sistema de comunicación **full-duplex aire-agua** empleando una combinación asequible y económica de tecnologías radio y acústicas. Se ha conseguido ofrecer una solución que consigue un **enlace bidireccional** frente al enlace unidireccional que lograban las tecnologías coetáneas al desarrollo de este TFG, gracias a un estudio exhaustivo de los medios de propagación sobre los que se lleva a cabo la comunicación, las tecnologías existentes que se pueden emplear y del comportamiento de las ondas acústicas y electromagnéticas en función del medio en el que se encuentran.

Gracias a este TFG se han conseguido consolidar los conocimientos y disciplinas adquiridos durante la carrera que recogen el desarrollo de software, la construcción de circuitos electrónicos, el manejo de instrumentación de laboratorio y el empleo de protocolos de comunicación. Además de la ampliación de conocimientos sobre sistemas de radiofrecuencia y física sobre propagación acústica y electromagnética.

Con todo esto, damos por cumplidos los objetivos propuestos en la sección 1.2.

Capítulo 6

Líneas de mejora

Mejoras hardware

En el proyecto, con objetivo de no recargar la boya se utiliza un sistema que conmuta la transmisión de datos del avión hacia el submarino con la transmisión de la ubicación GPS cuando el submarino la solicita. De este modo, se corta la comunicación *full-duplex* cuando la boya transmite sus coordenadas GPS. Para solucionar este inconveniente se puede añadir otro receptor acústico (hidrófono) en el submarino y otro altavoz impermeable en la boya. De esta forma bastaría con transmitir a frecuencias distintas a las del enlace avión-submarino y submarino-avión diseñando el VCO (integrado CD4046B, véase el Anexo C.2) con otro recorrido de frecuencias y seleccionar dos frecuencias¹ que no estén en uso en los enlaces *full-duplex* (si se usan las de este proyecto, distintas a 1800 Hz, 2800 Hz, 4100 Hz y 7000 Hz).

Una mejora hardware evidente sería construir físicamente una boya que flote e incluya las funcionalidades de la caja ya montada. Una posible opción sería realizarla mediante impresión 3D, buscar una solución económica, etc... Esta estructura debe incluir el montaje de forma estanca y prevenir cualquier hueco por el que pueda filtrarse el agua.

Buscar un hardware más adecuado para la transmisión subacuática en el cuál se analice el efecto de la presión del agua en los equipos en su trayecto a la superficie. Además, se puede realizar una búsqueda de un recorrido de frecuencias óptimo para la transmisión subacuática (frecuencias a las que menos afecte la atenuación), ya que en este proyecto se ha tenido que adaptar a la respuesta frecuencial de los altavoces y micrófonos.

Otra mejora posible es construir más boyas que solo implementen el sistema GPS y la tecnología de transmisión acústica. De esta forma se puede establecer un método de geolocalización mediante triangulación que daría una precisión mucho mayor al implementado en este proyecto con una sola boya.

Realizar una colaboración con la armada para diseñar una boya acorde a sus nece-

¹**NOTA:** Recordando que si se usa el CD4046B se genera una señal cuadrada y hay que tener en cuenta los armónicos.

sidades. Podría además contar con un sistema basado en placas solares para alargar la vida útil de la boya, operar en bandas militares, etc...

Mejoras software

Invirtiendo más tiempo se puede realizar un software más robusto y sofisticado que incluya además un sistema de identificación para que los extremos de la comunicación puedan verificar la seguridad del enlace. Esta ampliación del código es trivial con respecto al ya desarrollado en este TFG. La idea consiste en que el submarino contiene una tabla de códigos que identifica unívocamente a cualquier nodo aliado que intenta iniciar el intercambio de información. Si el submarino recibe a través de la boya una solicitud de comunicación, este se encarga de comprobar que el identificador es correcto. En el caso de que sea correcto, el submarino contestará si este lo cree oportuno (p.e: si se encuentra en una situación que pueda comprometer su seguridad no responderá, a pesar de que el otro nodo se haya identificado correctamente). Si la identificación es errónea el submarino permanece en el anonimato y no responde. La comunicación se establecerá según el sistema *full-duplex* implementado en el momento en el que el submarino responda al avión. La boya será un nodo transparente que actuando como un sistema de escucha avisa al submarino en todo momento de que ha recibido una señal radio (ya sea aliada o enemiga). En un sistema de múltiples boyas, solo una de ellas se encargará de mantener el enlace avión-submarino y todas contestarán a las peticiones GPS.

En la Figura 6.1 se muestra una de las situaciones que se pueden dar en el enlace avión-boya empleando este diseño. Se puede observar como en un primer momento el **AVIÓN ALIADO** manda una solicitud de conexión (**C.REQ**) en la que adjunta su identificador (**ID = 012X4TY9**). Esta solicitud es captada por la **BOYA** y retransmitida hacia el **SUBMARINO**. Una vez recibida en el **SUBMARINO**, este comprueba en sus tablas si el ID que acompaña la solicitud (**C.REQ**) es válido. Tras confirmar el identificador se procede a responder a la solicitud de conexión (**ACK SUB.**) y se establece el enlace *full-duplex* implementado en este proyecto. Más adelante, aparece en el rango de alcance radio del sistema **BOYA** un nuevo avión, en este caso un **AVIÓN ENEMIGO** que pretende suplantar a un avión aliado en la comunicación. Su transmisión es captada y retransmitida al **SUBMARINO**, donde se decidirá si seguir transmitiendo corriendo el riesgo de que se la señal pueda ser interferida o abortar la transmisión.

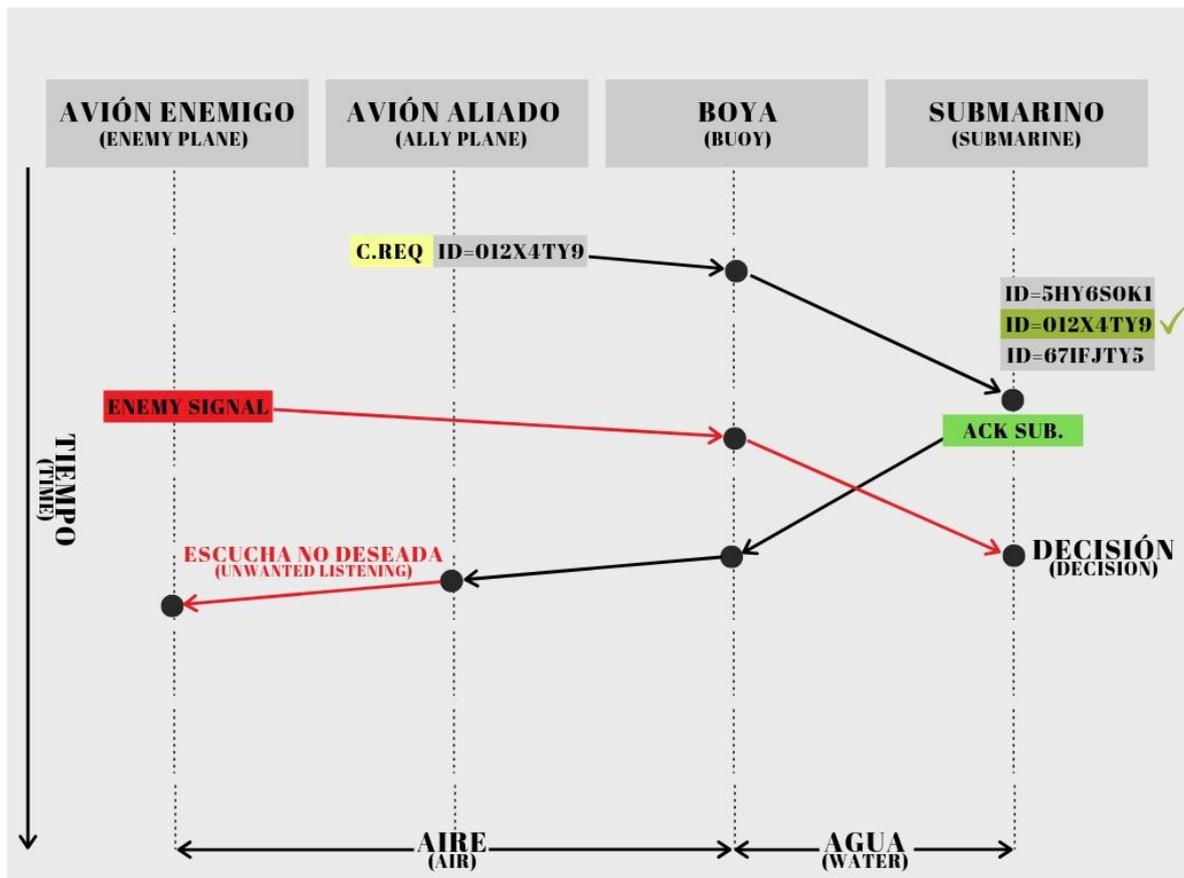


Figura 6.1: Ejemplo de una de las situaciones del protocolo de comunicación donde hay un avión aliado y uno enemigo. **C.REQ**: Communication Request, Petición de establecer conexión; **ACK SUB.:** ACKnowledge, Asentimiento de conexión por parte del submarino; **ENEMY SIGNAL**: Señal enemiga

Anexos

Anexos A

Código para la generación de tonos (transmisión del mensaje desde avión y desde el submarino)

En este Anexo se recoge el código empleado para la traducción de un mensaje ASCII a Morse mediante tonos acústicos de distintas frecuencias. Este código se emplea en los ordenadores de abordo tanto del submarino como del avión con la **única diferencia de las frecuencias de los tonos generados para el punto y la raya.**

A.1. Octave

```
1 function ascii2morse()
2 clc;
3 clear all;
4 % Mostramos el mensaje para que el usuario introduzca la
   informacion a transmitir
5 display('-----');
6 display(' ASCII TO MORSE V.1 BY MIGUEL ALONSO FELIPE');
7 cadena = input(' Insert message: ','s');
8 display('-----');
9 % Array con datos ya escritos ,que son la traduccion del
   abecedario en codigo Morse.
10 MorseABC = {'.-','-...','-.-.','-..','.','.-.-','--.','....','
   ..','-.---','-.-','.-..','--','-.','---','-.-','.-.-','.-.',
   ...','-','.-','...-','.--','-.-','-.--','--..'};
11
12 % Array con datos ya escritos, que son los numeros en codigo
   Morse
13 Numeros = {'-----','-.---','..---','...--','....-','.-.-.',
   -....','--...','----.','----.'};
14
15 % Declaramos el caracter para extraer del array uno a uno
16 caracter = '';
17
```

```

18 % Declaramos la cadena que almacenara todos los caracteres
    segun se mandan
19 completado_ascii = '';
20 completado_morse = '';
21
22 % Comprobamos que posicion ocupa el caracter en nuestra cadena
    de codigo morse
23     for i=1:length(cadena)
24         caracter = cadena(i);
25         if(caracter >= 'a' && caracter <= 'z' )
26             recorridoCadena(MorseABC{caracter - 'a' + 1},
                caracter);
27         elseif(caracter >= 'A' && caracter <= 'Z' )
28             recorridoCadena(MorseABC{caracter - 'A' + 1},
                caracter);
29         elseif(caracter >= '0' && caracter <= '9' )
30             recorridoCadena(Numeros{(caracter - '0') + 1},
                caracter);
31         elseif(caracter == ' ' )
32             display('
                -----');
33             pause(1);
34         end
35     end
36     display('-----');
37     display(' Message Successfully Transmitted!');
38     display('-----');
39     end
40
41 % Esta funcion se encarga de desglosar el caracter traducido a
    morse para poder procesarlo como sonido
42 function recorridoCadena(char_morse,caracter)
43 printf([' ',caracter,' >> ']);
44 % Barremos todos los elementos del caracter en morse
45     for i=1:length(char_morse)
46         if(char_morse(i) == '.')
47             printf(char_morse(i));
48             tono(1800,0.8);% Para el submarino sera
                de 4100
49         elseif(char_morse(i) == '-')
50             printf(char_morse(i));
51             tono(2800,0.8);% Para el submarino sera
                de 7000
52         end
53     end
54 pause(2); % Tiempo entre caracteres
55 display('');
56 end
57
58 % Esta funcion recibe la orden de reproducir el punto o la raya

```

```

59 function tono(Hz,tiempo_espera)
60 % Generamos los tonos para transmitir el punto y la raya
61 fs=8000; % Frecuencia de muestreo
62 duracion=tiempo_espera; % Duracion del tono
63 freq1=1800;freq2=2800; % Frecuencias de los tonos a reproducir
    (4100 y 7000 para el submarino)
64 val=0:1/fs:duracion; % Creamos un vector temporal para la
    reproduccion del sonido con el numero de muestras necesario
65
66 % Generamos ambos tonos
67 a1=sin(2*pi* freq1*val);
68 a2=sin(2*pi* freq2*val);
69
70 % Distinguimos segun el la frecuencia el sonido a reproducir
71     if(Hz == 1800)% 4100 para el submarino
72         sound(a1,fs);
73     elseif(Hz == 2800)% 7000 para el submarino
74         sound(a2,fs);
75     end
76 end

```

A.2. Matlab

```

1 % Esta funcion se genera automaticamente por Matlab al usar GUI
2
3 % Esta funcion NO se modifica y declara los parametros basicos
4 % de la interfaz grafica.
5 function varargout = morse_tfg(varargin)
6 % Begin initialization code - DO NOT EDIT
7 gui_Singleton = 1;
8 gui_State = struct('gui_Name',      mfilename, ...
9 'gui_Singleton',  gui_Singleton, ...
10 'gui_OpeningFcn', @morse_tfg_OpeningFcn, ...
11 'gui_OutputFcn',  @morse_tfg_OutputFcn, ...
12 'gui_LayoutFcn',  [], ...
13 'gui_Callback',   []);
14 if nargin && ischar(varargin{1})
15     gui_State.gui_Callback = str2func(varargin{1});
16 end
17
18 if nargin
19     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
20 else
21     gui_mainfcn(gui_State, varargin{:});
22 end
23 % End initialization code - DO NOT EDIT
24 end

```

```

23
24 % Esta funcion se genera automaticamente por Matlab al usar GUI
25
26 % En esta funcion se declaran las variables globales definidas
27 % como handles de forma que puedan ser usadas y modificadas en
28 % las funciones del fichero a las que se les pase como
29 % parametro.
30 function morse_tfg_OpeningFcn(hObject, eventdata, handles,
31     varargin)
32 % -----Declaramos los handles-----
33 handles.cadena = ''; % Declaramos la cadena que almacenara el
34 % mensaje escrito por el usuario.
35
36 % Se generan los tonos
37 amp=10; % Amplitud del tono
38 handles.fs=8000; % Frecuencia de muestreo (suficiente para
39 % cumplir Nyquist y buena calidad de reproduccion).
40 duration=5; % Duracion del tono
41 freq=[500,1000]; % Frecuencias del tono
42
43 val=0:1/handles.fs:duration; % Valores del tono
44
45 % Se declaran los handles que portan los tonos
46 handles.x1=amp*sin(2*pi* freq(1)*val);
47 handles.x2=amp*sin(2*pi* freq(2)*val);
48
49 %----fin de las modificaciones realizadas en esta funcion-----
50
51 % Choose default command line output for morse_tfg
52 handles.output = hObject;
53
54 % Update handles structure
55 guidata(hObject, handles);
56 end
57
58 % Esta funcion se genera automaticamente por Matlab al usar GUI
59 % La salida de esta funcion se devuelven a la linea de comandos
60
61 function varargout = morse_tfg_OutputFcn(hObject, eventdata,
62     handles)
63 varargout{1} = handles.output;
64 end
65
66 % Esta funcion se genera automaticamente por Matlab al declarar
67 % un bloque de texto donde se introduce el mensaje a traducir.
68 function ascii_Callback(hObject, eventdata, handles)
69 handles.cadena=get(hObject,'String'); % Recogemos el mensaje y
70 % lo almacenamos en el handle.
71 guidata(hObject,handles); % Grabamos las modificaciones
72 % realizadas en el handle para que esten presentes en el resto

```

```

        de funciones que usen el handle despues.
61 end
62
63 % Esta funcion se genera automaticamente por Matlab al declarar
    un bloque de texto. Esta define la apariencia de este bloque
    .
64 function ascii_CreateFcn(hObject, eventdata, handles)
65 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
66 set(hObject,'BackgroundColor','white');
67 end
68 end
69
70
71 % Esta funcion se genera automaticamente por Matlab al declarar
    el boton "Send" usado para comenzar la traduccion.
72 function Send_Callback(hObject, eventdata, handles)
73 ascii2morse(handles);% Al pulsar el boton se llama a la funcion
    principal del programa y se le pasan los handles para que
    pueda usarlos.
74 end
75
76 % Esta funcion se genera automaticamente por Matlab al declarar
    un bloque de texto donde se ira escribiendo que caracter se
    esta traduciendo en cada momento (en ASCII)
77 function completado_Callback(hObject, eventdata, handles)
78 end
79
80 % Esta funcion se genera automaticamente por Matlab al declarar
    el bloque de texto definido encima. Esta define la
    apariencia de este bloque.
81 function completado_CreateFcn(hObject, eventdata, handles)
82 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
83 set(hObject,'BackgroundColor','white');
84 end
85 end
86
87 % Esta funcion se genera automaticamente por Matlab al declarar
    un bloque de texto donde se ira escribiendo que caracter se
    esta traduciendo en cada momento (en Morse), es decir los
    puntos y las rayas.
88 function morse_completado_Callback(hObject, eventdata, handles)
89 end
90
91 % Esta funcion se genera automaticamente por Matlab al declarar
    el bloque de texto definido encima. Esta define la
    apariencia de este bloque.
92 function morse_completado_CreateFcn(hObject, eventdata, handles
    )

```

```

93  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
      defaultUicontrolBackgroundColor'))
94  set(hObject,'BackgroundColor','white');
95  end
96  end
97
98  % Se define la funcion principal del programa.
99  function ascii2morse(handles)
100 % Array con datos ya escritos ,que son la traduccion del
      abecedario en codigo Morse.
101 MorseABC = {'.-','-...','-.-.','-..','.','...-','--.','....','
      ..','-.---','-.-','-...','--','-.','---','-.---','--.-','-.','
      ...','-','-.-','-...-','--','-.-','-.-','--..'};
102
103 % Array con datos ya escritos, que son los numeros en codigo
      Morse
104 Numeros = {'-----','-----','-...--','-...--','....-','.....','
      -....','--...','---..','----.'};
105
106 % Declaramos el caracter para extraer del array uno a uno
107 caracter = '';
108
109 % Declaramos la cadena que almacenara todos los caracteres
      segun se mandan
110 completado_ascii = '';
111 completado_morse = '';
112
113 for i=1:length(handles.cadena)% Se usa un bucle para recorrer
      todos los caracteres de la cadena
114 caracter = handles.cadena(i);% Se extrae el i-esimo caracter de
      la cadena
115 completado_ascii = [completado_ascii,caracter];% Se va
      concatenando el caracter actual para irlo mostrando por al
      cuadro de texto
116 set(handles.completado,'String',completado_ascii);% Se muestra
      la cadena concatenada en el cuadro de texto
117
118 % Conjunto de sentencias que evaluan en que rango se encuentra
      el caracter para compararlo con su valor ASCII
119 if(caracter >= 'a' && caracter <= 'z' )
120 completado_morse = [completado_morse,' | ',MorseABC{caracter -
      'a' + 1}];% Se concatena el progreso de puntos y rayas
121 set(handles.morse_completado,'String',completado_morse);% Se
      muestra la cadena concatenada en el cuadro de texto
122
123 recorridoCadena(handles,MorseABC{caracter - 'a' + 1});% Se
      llama a la funcion recorridoCadena() y se le pasan los
      parametros necesarios para su funcionamiento. Se evalua en la
      cadena la posicion que le corresponde restando al valor
      ASCII el valor ASCII correspondiente al primer caracter, de

```

```

forma que si es la 'a' - 'a' = 0 y dado que en Matlab el
indice 0 no existe en los array se le suma 1. De esta forma
se elige de la cadena de Morse el conjunto de puntos y rayas
que corresponden al caracter 'a'.
124
125 elseif(caracter >= 'A' && caracter <= 'Z' )
126 completado_morse = [completado_morse,' | ',MorseABC{caracter -
    'A' + 1}];
127 set(handles.morse_completado,'String',completado_morse);
128
129 recorridoCadena(handles,MorseABC{caracter - 'A' + 1});
130
131 elseif(caracter >= '0' && caracter <= '9' )
132 completado_morse = [completado_morse,' | ',MorseABC{caracter -
    '0' + 1}];
133 set(handles.morse_completado,'String',completado_morse);
134
135 recorridoCadena(handles,MorseABC{(caracter - '0') + 1});
136
137 elseif(caracter == ' ' )% Acaba una palabra y comienza otra
138 completado_morse = [completado_morse,' | ','//'];% Se introduce
    una doble barra para separar las palabras
139 set(handles.morse_completado,'String',completado_morse);
140
141 pause(3);% Espera entre palabras en segundos, en este caso 3
    seg.
142 end
143 end
144 end
145
146 % Funcion que recorre los puntos y las rayas de cada caracter
    en Morse para traducirlo a sonidos.
147 function recorridoCadena(handles,char_morse)
148 % Se recorre con un bucle todos los simbolos de cada caracter
    traducido a Morse.
149 for i=1:length(char_morse)
150 if(char_morse(i) == '.')% Si es punto
151 tono(handles,500,1);% Se manda hacer sonar un tono de 500Hz
152 elseif(char_morse(i) == '-')% Si es raya
153 tono(handles,1000,1);% Se manda hacer sonar un tono de 1000Hz
154 end
155 pause(0.3);% Espera entre puntos y rayas
156 end
157 end
158
159 % Funcion que reproduce el tono cuya frecuencia recibe como
    parametro y durante el tiempo que se le ordena.
160 function tono(handles,HZ,tiempo_espera)
161 % Distinguimos segun la variable HZ la frecuencia el sonido a
    reproducir

```

```

162 if(Hz == 500)
163 sound(handles.x1,handles.fs);% Se reproduce el tono a la
    frecuencia de muestreo que se genero
164 elseif(Hz == 1000)
165 sound(handles.x2,handles.fs);
166 end
167 pause(tiempo_espera);% Tiempo que estara sonando el tono
168 clear sound;% Tras la espera se silencia el tono por completo
169 end

```

A.3. Arduino

```

1 #define SERIAL_BUFFER_SIZE 256 // Para evitar desbordamiento
    del buffer
2
3 const int pinBuzzer = 11;// Declaramos el pin al que esta
    conectado el Buzzer
4
5 //Array con el abecedario en codigo Morse
6 char* MorseABC[] = {".-","-...","-.-","-..",".",".-.-","--.",
    "...","..",".---","-.-","-...","--","-.", "---", ".---", "--.-",
    ".-.", "...", "-", ".-.", "...-", ".--", "-.-.", "-.-.", "-.-."};
7
8 //Array con los numeros en codigo Morse
9 char* Numeros[] = {"-----", ".-----", ".-.-.-", "...--", "....-",
    ".....", "-.....", "--....", "---...", "----."};
10
11 void setup()
12 {
13 Serial.begin(9600);// Declaramos la velocidad de simbolos por
    segundo del monitor serie
14 }
15
16 void loop()
17 {
18
19 char caracter;// Variable que almacena el caracter
20
21 if(Serial.available()){// Esperamos a que haya datos que leer
    del monitor serie
22 caracter = Serial.read();// Lee caracter a caracter del monitor
    serie
23
24 // Para mostrar por pantalla el caracter actual
25 if(caracter == ' ')
26 Serial.println("-----");
27 else{

```

```

28 Serial.print(caracter);
29 Serial.print(">> ");
30 }
31
32 // Comprobamos que correspondencia tiene el caracter con el
    codigo Morse
33 if(caracter >= 'a' && caracter <= 'z' )
34 {
35 recorridocadena(MorseABC[caracter - 'a']);
36 }
37 else if(caracter >= 'A' && caracter <= 'Z' )
38 {
39 recorridocadena(MorseABC[caracter - 'A']);
40 }
41 else if(caracter >= '0' && caracter <= '9' )
42 {
43 recorridocadena(MorseABC[caracter - '0']);
44 }
45 else if(caracter == ' ' )
46 {
47 delay(3000); // Espera entre palabras en ms
48 }
49 }
50
51 }
52
53 //Funcion que recorre las posiciones de las cadenas usando como
    variable un puntero de tipo char.
54 void recorridocadena(char* posicion)
55 {
56 Serial.println(posicion);
57 int i = 0; // Variable contador para recorrer los valores del
    array
58 while (posicion[i] != NULL){ // Hasta recorrer todos los
    elementos (puntos y rayas)
59 if(posicion[i] == '.') {
60 tone(pinBuzzer, 500, 500); // Genera tono de 500Hz durante 500
    ms
61 delay(1000);
62 noTone(pinBuzzer);
63 }
64 else if(posicion[i] == '-') {
65 tone(pinBuzzer, 1000, 500); // Genera tono de 1000Hz durante 500
    ms
66 delay(1000);
67 noTone(pinBuzzer);
68 }
69 i++;
70 }
71 delay(1000); // Espera entre caracteres en ms

```


Anexos B

Código de control de los módulos de Arduino

B.1. Transmisor en la banda de FM comercial. FM Transmitter v2.0

```
1 #include <FMTX.h>// Libreria para controlar el transmisor FM
2 float fm_freq = 108; // Frecuencia de transmision
3 void setup(){
4   Serial.begin(9600);// Inicializa el puerto serie a 9600 baudios
   (valor por defecto)
5   Serial.print("FM-TX TFG\r\n");// Imprime por el monitor serie
6
7   fmtx_init(fm_freq, EUROPE);// Para transmitir en el rango de
   frecuencias europeas
8
9   Serial.print("Canal: ");
10  Serial.print(fm_freq,1);
11  Serial.println("MHz");
12  }
13 void loop(){
14  // Podria usarse para controlar constantemente el puerto serie
   y variar en tiempo real la frecuencia de transmision, por
   ejemplo. Dado que para nuestro sistema esto no tiene mayor
   relevancia se deja simplificado el codigo.
15 }
```

B.2. Receptor en la banda de FM comercial. Receptor basado en el módulo de radio TEA5767

```
1 #include <radio.h>
2 #include <TEA5767.h>
```

```

3
4 #define FIX_BAND RADIO_BAND_FM // Sintonizamos FM
5 #define FIX_STATION 10800 // Definimos la estacion a 108.00 Mhz
6
7 TEA5767 radio; // Creamos un objeto radio
8
9 void setup(){
10 Serial.begin(57600); // Establecemos a 57600 baudios por ser el
    valor que necesita el modulo FM para comunicarse con el
    Arduino. De esta forma podremos imprimir por pantalla la
    informacion.
11
12 radio.init(); // Inicializar el objeto radio
13 radio.setBandFrequency(FIX_BAND, FIX_STATION); // Seleccionamos
    la emisora 108.00
14 radio.setVolume(15); // De 0 a 15
15 radio.setMono(true); // Para recibir en mono, ya que es el tipo
    que se utiliza en este sistema
16 }
17
18 void loop(){
19 // Se puede emplear el loop para imprimir informacion acerca de
    la banda en la que esta fijado el receptor, si por ejemplo
    se implementa un sistema para cambiar de cadena con un
    potenciómetro.
20 }

```

B.3. Módulo transceptor nRF24L01 (2,4GHz-2,5GHz)

B.3.1. Módulo 1

```

1 /*MODULO TRANSCCEPTOR 1*/
2
3 #include <SPI.h>
4 #include <nRF24L01.h>
5 #include <RF24.h>
6
7
8 RF24 radio(7, 8); // CE, CSN
9 const byte addresses[][6] = {"00001", "00002"}; // Declaramos
    las dos direcciones
10
11 void setup() {
12     Serial.begin(9600);
13     radio.begin();// Inicializamos el objeto con el metodo
        begin()
14     radio.openWritingPipe(addresses[1]); // 00002

```

```

15     radio.openReadingPipe(1, addresses[0]); // 00001, se
        usa el ID de tubería 1 para que no haya conflicto con
        openWritingPipe() que usa el ID = 0
16     radio.setPALevel(RF24_PA_MIN); // Se selecciona la
        sensibilidad del amplificador de potencia, en este
        caso se ha seleccionado la mínima
17 }
18 void loop() {
19     delay(5);
20     radio.stopListening();// Se para la escucha del puerto
21     const char text[] = "TXRX-1";// Se inicializa el
        mensaje a transmitir (se podría poner un Serial.read
        () y que fuese introducido por el usuario en tiempo
        real)
22     Serial.println("Enviamos:");// Se indica por pantalla
        que cadena se va a transmitir
23     Serial.println(text);
24     radio.write(&text, sizeof(text));// Se escribe en la
        tubería ID = 1, en el puerto de escucha del otro
        módulo el 00002
25     delay(1000);
26     radio.startListening();// Se comienza a escuchar por el
        puerto prefijado
27     if (radio.available()) { //Comprobamos si hay datos
        para recibir
28         Serial.println("Hay algo");// Se imprime por
        pantalla un mensaje que nos indique que se ha
        entrado en este estado
29         delay(1000);
30         char text[32] = ""; // Se inicializa un tamaño
        máximo de lectura que se espera recibir.
31         radio.read(&text, sizeof(text));// Se almacena
        la lectura en la cadena text
32         Serial.println(text); // Se imprime por la
        pantalla el mensaje recibido.
33     }
34     else{
35         Serial.println("No hay nada");// En el caso de
        no haber datos esperando a ser leídos se
        muestra un mensaje que lo indique
36         delay(1000);
37     }
38 }

```

B.3.2. Módulo 2

```

1  /*MODULO TRANSCCEPTOR 2*/
2

```

```

3  #include <SPI.h>
4  #include <nRF24L01.h>
5  #include <RF24.h>
6
7
8  RF24 radio(7, 8); // CE, CSN
9  const byte addresses[][6] = {"00001", "00002"};
10
11 void setup() {
12     radio.begin();
13     //Como las direcciones de escucha y emision deben estar
14     //intercambiadas con el otro transceptor...
15     radio.openWritingPipe(addresses[0]); // 00001
16     radio.openReadingPipe(1, addresses[1]); // 00002
17     radio.setPALevel(RF24_PA_MIN);
18 }
19 void loop() {
20     delay(5);
21     radio.startListening();
22     if (radio.available()) { //Comprobamos si hay datos
23         para recibir
24         char text[32] = ""; // Se inicializa un tamaño
25         //máximo de lectura que se espera recibir.
26         radio.read(&text, sizeof(text)); // Se almacena
27         //la lectura en la cadena text
28         Serial.println(text); // Se imprime por la
29         //pantalla el mensaje recibido.
30     }
31     delay(5);
32     radio.stopListening();
33     const char text[] = "TXRX-2";
34     radio.write(&text, sizeof(text));
35     delay(1000);
36 }

```

Anexos C

Esquemas electrónicos y módulos auxiliares

C.1. Esquema de amplificador operacional con alimentación simple

Para las dos etapas de amplificación se repite el esquema que se muestra en la Figura C.1. Se recurre a un montaje con alimentación simple para reducir el número de fuentes en el sistema; de forma que quede un montaje más sencillo y se pueda alimentar el mayor número de elementos posibles con la misma fuente (esto es de gran utilidad para poder alimentar con la salida de Arduino de 5V el sistema de amplificación al completo).

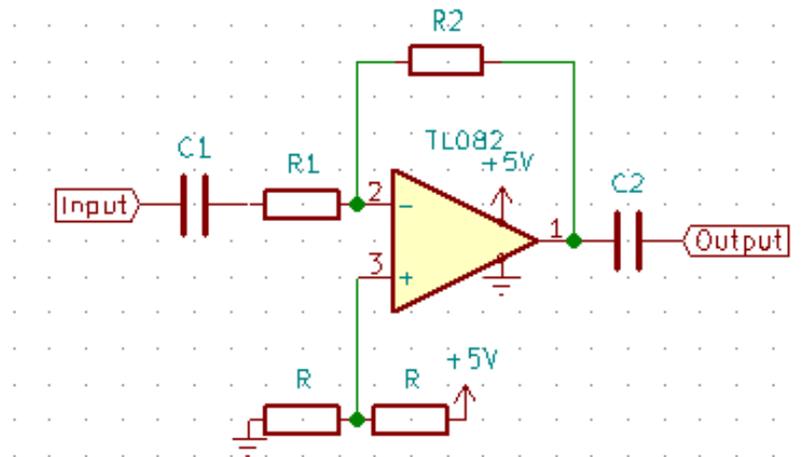


Figura C.1: Montaje de una etapa de amplificación usando el TL082.

Para este esquema la ganancia se puede definir como:

$$G = -\frac{R_2}{R_1} \quad (\text{C.1})$$

Se ilustra en la Figura C.2 el montaje total con ambas etapas, donde se añaden los

condensadores C3 y C4 para evitar que en la referencia se introduzca ruido de alta frecuencia.

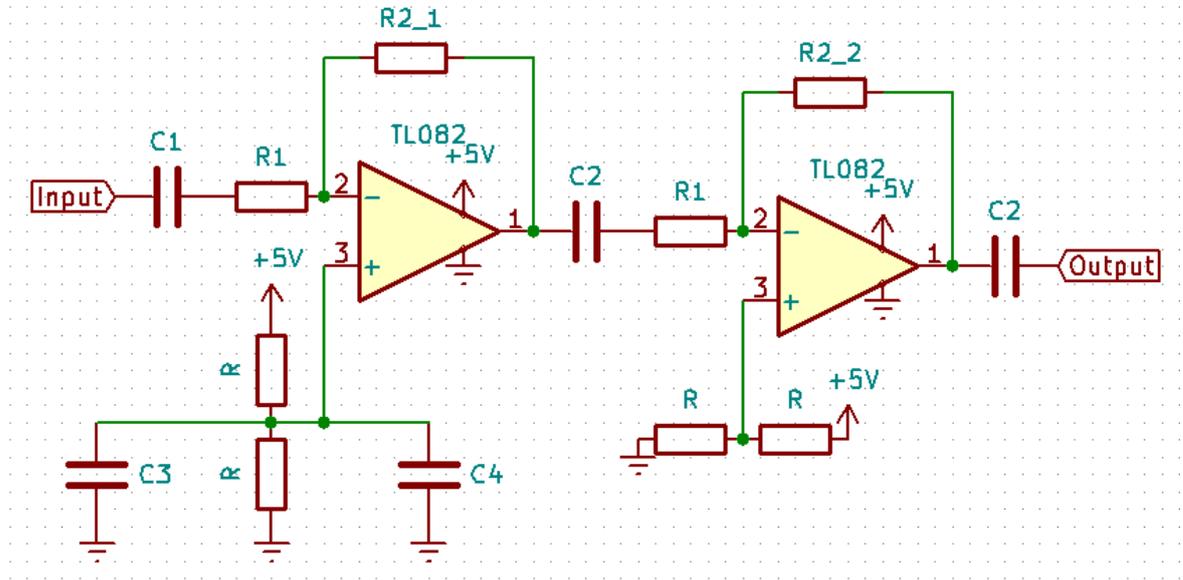


Figura C.2: Montaje total de dos etapas de amplificación usando el TL082.

Se utiliza una amplificación de dos etapas para controlar mejor el nivel de offset de la señal. Amplificar el offset de la señal puede provocar que se pierda la información que esta transmite. Si el nivel de tensión de offset satura tras ser amplificado (alcanza el nivel de tensión de la alimentación, en este caso 5V) recortaría por completo la señal analógica que transmite la información.

C.1.1. Valores de la etapa de amplificación basada en el integrado TL082 empleada en la caja del avión

Se adjuntan los valores empleados en la caja del avión para amplificar la señal recibida en el receptor radio y de este modo acondicionar el nivel de la señal antes de introducirla al montaje PLL.

TL082 (una etapa)	
C1	220 nF
C2	220 nF
R	1,2 K Ω
R1	10 K Ω
R2	220 K Ω

Tabla C.1: Los valores RC empleados para el diseño del circuito con el integrado TL082.

C.1.2. Valores de la etapa de amplificación doble basada en el integrado TL082 empleada en la caja de la boya y del submarino

Se adjuntan los valores empleados para la generación de dos etapas de amplificación implementadas en la caja de la boya y del submarino. Este montaje se utiliza para acondicionar la señal en recepción a la entrada de sus respectivos PLL.

TL082	
C1	100 nF
C2	220 nF
C3	680 pF
C4	1 μ F
R	1,2 K Ω
R1	10 K Ω
R2.1	220 K Ω
R2.2	47 K Ω

Tabla C.2: Los valores RC empleados para el diseño del circuito con el integrado TL082.

C.2. Esquema electrónico del conexionado del CD4046B (PLL). Sistemas tipo 1 de orden 2

El integrado CD4046B[12] tiene una gran importancia en el funcionamiento del sistema, dado que es una pieza fundamental para poder interpretar los tonos en la recepción y también para generar tonos (señales cuadradas) de la frecuencia que nosotros queramos transmitir sin la necesidad de usar otro tipo de módulos o un ordenador. Dicho esto, queda por comentar la función de los elementos electrónicos pasivos que se utilizan y se pueden observar en la Figura C.3.

El CD4046B incluye un bloque interno para construir el oscilador controlado por voltaje (VCO) requerido en un PLL. El rango de frecuencias de funcionamiento del VCO se ajusta con tres componentes externos, R1, R2 y C1. Fijando el valor de C1 es posible establecer la frecuencia mínima de oscilación del VCO calculando R1 y la máxima calculando R2 según las especificaciones del fabricante¹. Entonces, a menor R1 tendremos una mayor frecuencia de salida máxima y a menor R2 tendremos una mayor frecuencia de salida mínima².

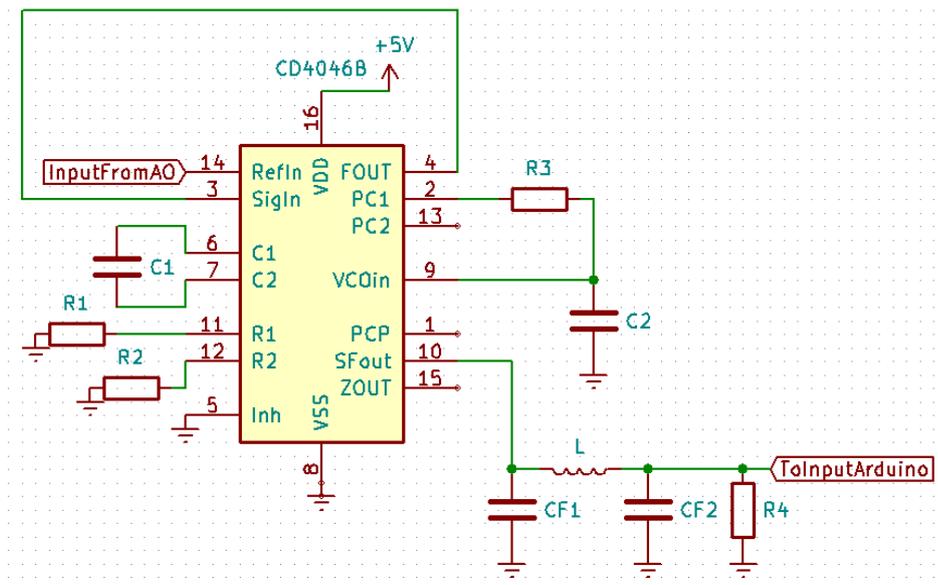


Figura C.3: Esquema electrónico del conexionado del integrado CD4046B.

Queda entonces por comentar la función del resto de elementos electrónicos pasivos. En primer lugar se analiza el conjunto R3 y C2 (formando un filtro paso bajo) conectado entre las patillas 2 (salida del comparador de fase) y 9 (entrada al VCO). Este conjunto RC (cuya respuesta frecuencial queda definida por la función de transferencia $F(s)$ (C.2)) fija dos parámetros fundamentales del integrado, que son: la w_n

¹R2 tiene una influencia de segundo orden sobre la frecuencia máxima, lo que implica que afecta en menor medida que R1.

²Si dejamos la patilla 12 sin conectar a tierra a través de una resistencia R2, es decir, en alta impedancia; el recorrido comienza en 0 Hz.

(pulsación natural) y la ξ (coeficiente de amortiguamiento). Además este bloque RC integra la salida pulsada del comparador digital de fase (puerta XOR) ³ de forma que se otorga un nivel de tensión idealmente continuo⁴ a la entrada del VCO. Se observa la localización de este bloque en el diagrama de la Figura C.4.

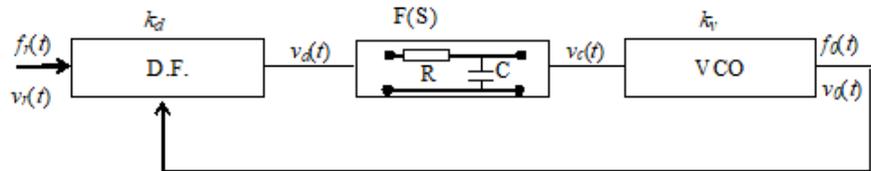


Figura C.4: Diagrama de bloques del PLL.

$$F(s) = \frac{1}{1 + sRC} \quad (C.2)$$

Estos parámetros afectan al tiempo que tarda el PLL en engancharse a una frecuencia de entrada (que esté en su recorrido de frecuencias), además de definir la función de transferencia del PLL $H(s)$ (C.3). Esta función de transferencia es tipo paso bajo (como se ilustra en la Figura C.5) y presenta un codo para pulsación natural ω_n . Para frecuencias de entrada superiores a ω_n (y tratando casos de frecuencias dentro del margen de operación) la salida del PLL se verá atenuada por un factor 40dB/década, es decir, el PLL conseguirá seguir a la entrada, pero a medida que esta frecuencia se aleje de la pulsación natural, la señal de salida sufrirá cada vez una mayor atenuación.

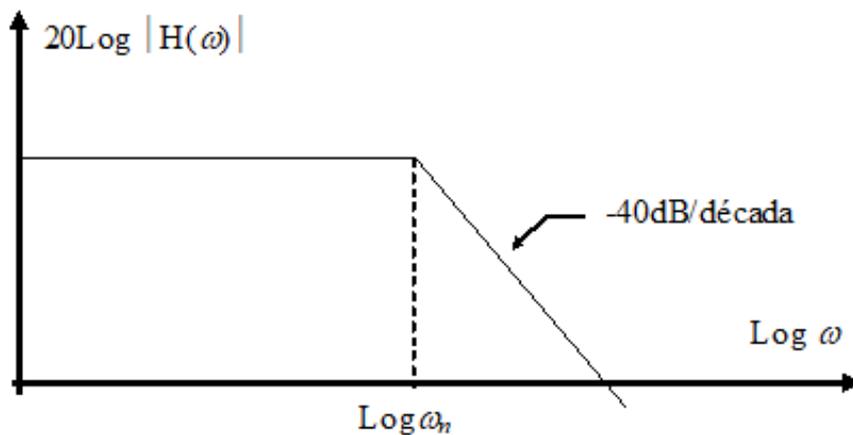


Figura C.5: Función de transferencia del PLL.

³Una señal de pulsos cuadrados que indica cómo de diferente es la fase de la señal de entrada con respecto la fase de la señal de referencia.

⁴En la práctica se puede observar que presenta un rizado.

$$H(s) = \frac{\phi_0(s)}{\phi_r(s)} = \frac{KF(s)}{s + KF(s)}, \text{ donde } K = 2\pi k_d k_v^5 \quad (\text{C.3})$$

Siguiendo las fórmulas (C.4) se puede determinar que a menor pulsación natural w_n , es decir, cuanto más bajo sea la función de transferencia del PLL (Figura C.5) implicará una respuesta más lenta por parte del PLL a la hora de engancharse a una frecuencia que esté dentro de su recorrido. También se puede extraer la conclusión de que a mayor coeficiente de amortiguamiento ξ , el PLL se enganchará antes a la frecuencia de entrada. Este fenómeno se puede observar en la Figura C.6, donde se suele considerar que el PLL está enganchado para valores entorno a $\pm 0,1$ del valor unidad $\theta_0(t) = 1$.

$$\tau = RC; \quad \omega_n = \sqrt{\frac{K}{\tau}}; \quad \xi = \frac{1}{2\tau\omega_n} \quad (\text{C.4})$$

Si ahora se sustituye (C.3) en (C.4):

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (\text{C.5})$$

Por último, se comenta el bloque con dos condensadores CF1, CF2 y L. Su función principal es filtrar por completo el rizado del nivel de continua que aparece por la patilla 10 (que será 0.7V menor que la de la patilla 9 por ser la misma salida tras un seguidor de emisor). Este rizado ocasiona errores en la medición por parte del Arduino dado que la excursión del nivel de tensión llega a ser muy significativa y alcanza valores de tensión que se alejan mucho del esperado para la frecuencia a la que se encuentra enganchado el PLL.

⁵Donde k_d es la pendiente de la función de transferencia del comparador de fase en (V/rad) y k_v la pendiente de la función de transferencia del VCO en (Hz/V).

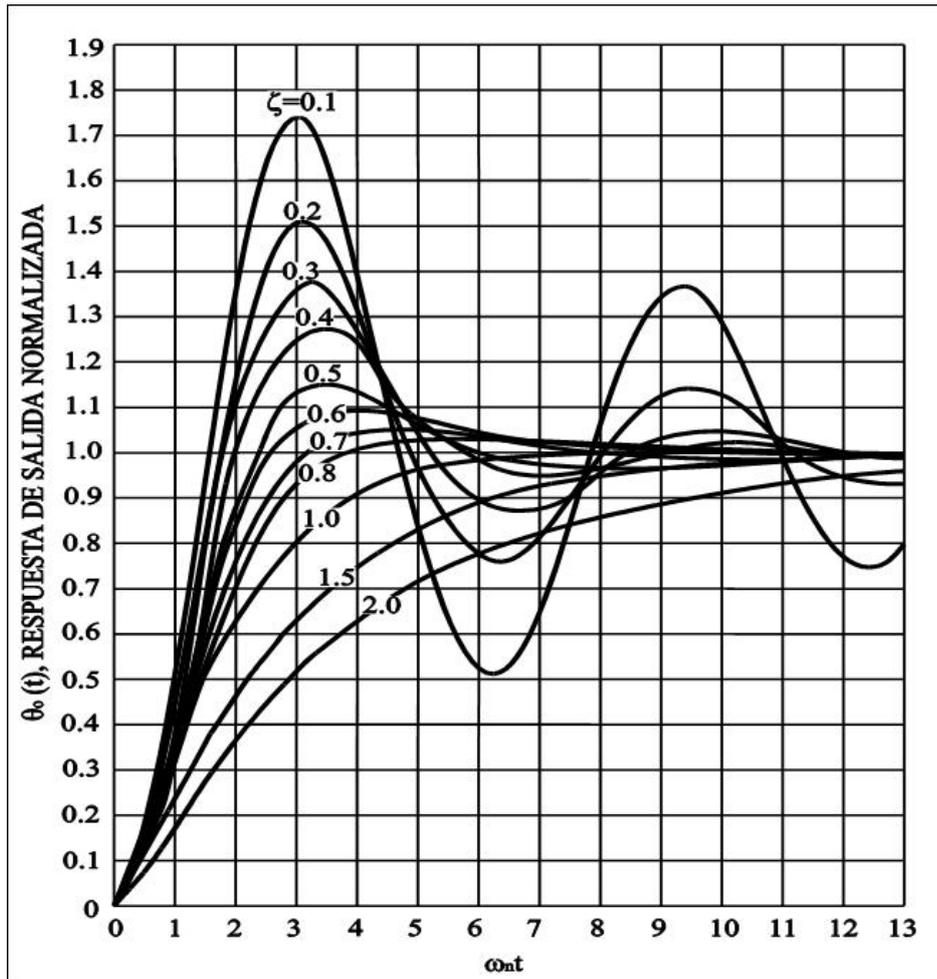


Figura C.6: Respuesta al escalón de un sistema tipo I de orden II.

C.2.1. Valores de los esquemas electrónicos basados en PLL

En esta subsección se recogen todos los valores empleados en los montajes que incluyen un esquema electrónico con PLL.

Valores de PLL en la caja del avión

Este diseño se emplea para detectar las frecuencias de 1800 Hz y 2800 Hz. Estas frecuencias son las utilizadas por el canal de subida (submarino-avión).

CD4046B (PLL - AVIÓN)	
L	1 mH
C1	10 nF
C2	1 nF
CF1	1 nF
CF2	100 nF
R1	47 K Ω
R2	82 K Ω
R3	100 K Ω
R4	68 K Ω

Tabla C.3: Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja del avión.

Valores de PLL en la caja de la boya

Este diseño se emplea para detectar las frecuencias de 1800 Hz y 2800 Hz. Estas frecuencias son las utilizadas por el canal de subida (submarino-avión). Este diseño se emplea para detectar la frecuencia de 500 Hz empleada para la solicitud de GPS del submarino a la boya.

CD4046B (PLL - BOYA)	
L	1 mH
C1	10 nF
C2	1 nF
CF1	1 nF
CF2	100 nF
R1	47 K Ω
R2	82 K Ω
R3	100 K Ω
R4	68 K Ω

Tabla C.4: Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja de la boya.

CD4046B (BOYA - para GPS)	
L	1 mH
C1	10 nF
C2	6,8 nF
CF1	10 nF
CF2	100 nF
R1	270 K Ω
R2	∞
R3	82 K Ω

Tabla C.5: Los valores RC empleados para el diseño del PLL basado en el circuito con el integrado CD4046B para detectar el tono de 500 Hz en la caja de la boya.

Valores de PLL en la caja del submarino

Este diseño se emplea para detectar las frecuencias de 4100 Hz y 7000 Hz. Estas frecuencias son las utilizadas por el canal de bajada (avión-submarino).

CD4046B (PLL)	
L	1 mH
C1	6,8 nF
C2	1 nF
C3	100 nF
R1	33 K Ω
R2	47 K Ω
R3	100 K Ω
R4	56 K Ω

Tabla C.6: Los valores de los elementos electrónicos pasivos empleados para el diseño del circuito con el integrado CD4046B en la caja del submarino.

C.3. Integrado CD4046 (VCO)

En esta configuración, al aplicar una tensión de entrada en la patilla 9 se obtendrá una señal cuadrada a la salida de la patilla 4 del CD4046B con una frecuencia proporcional a esa tensión de entrada. Se ilustra el conexionado en la Figura C.7.

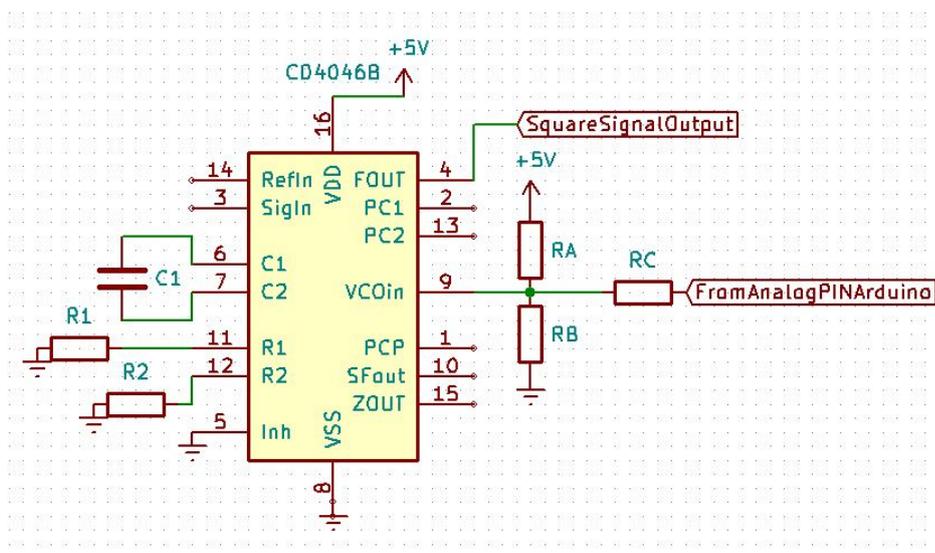


Figura C.7: Esquema de conexionado del integrado CD4046B empleando su VCO.

C.3.1. Valores del VCO empleados en la caja de la boya

Estos valores se emplean para la generación de tonos de 4100 Hz (punto) y 7000 Hz (raya) (canal de bajada avión-submarino). Este VCO se emplea para la generación de la respuesta a la petición de la ubicación GPS por parte del submarino mediante los tonos correspondientes al código Morse de las coordenadas.

CD4046B (VCO)	
C1	6,8 nF
R1	47 K Ω
R2	82 K Ω
RA	12,86 K Ω \rightarrow 12 K Ω
RB	10 K Ω
RC	10 K Ω

Tabla C.7: Los valores RC empleados para el diseño del VCO basado en el circuito con el integrado CD4046B.

C.4. Amplificador de potencia

El amplificador de potencia empleado en la transmisión acústica. Para amplificar la señal de audio antes de transmitirla por el altavoz a través del agua, se utiliza un amplificador de potencia de clase D. Que sea de clase D implica que utiliza el modo conmutado de los transistores (corte y saturación) para regular la entrega de potencia, esto hace que se pierda menos energía y consigue que el sistema de amplificación sea más eficiente. De esta forma, dado que irán integrados en la boya (además de en el submarino), se conseguirá alargar la vida útil de esta. Este módulo cuenta con un potenciómetro que regula la amplificación de la salida, una entrada de alimentación, una entrada de señal de audio mediante un puerto jack y salidas mediante un conector de 4 pines para obtener una amplificación mono (empleando solo una salida) o estéreo (empleando las dos salidas). Se ilustra el módulo empleado en la Figura C.8.

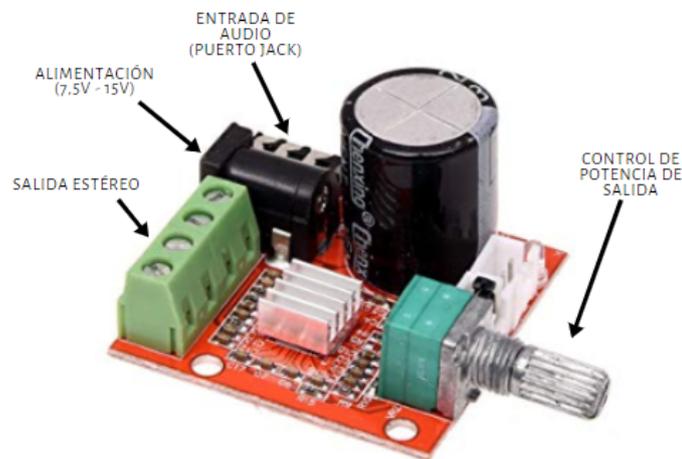


Figura C.8: Módulo amplificador de potencia.

Otras características de este módulo son: una potencia nominal de salida de 10W, unas frecuencias de operación de 20 Hz a 50 KHz (más que suficiente para el rango que se emplea en nuestro sistema), con un voltaje de funcionamiento de entre 7.5V y 15V (el fabricante recomienda alimentarlo con fuentes de alimentación de 9V ó 12V) y una corriente de reposo de 20 mA.

En el proyecto se acaba empleando un módulo de características de conexionado similares, con la mitad de potencia (5W) y con una alimentación de 1.8V a 12V. Se escoge definitivamente este módulo ya que facilita la integrabilidad en las cajas y cumple perfectamente con los niveles de amplificación necesarios para el altavoz. Se adjunta en la Figura C.9.

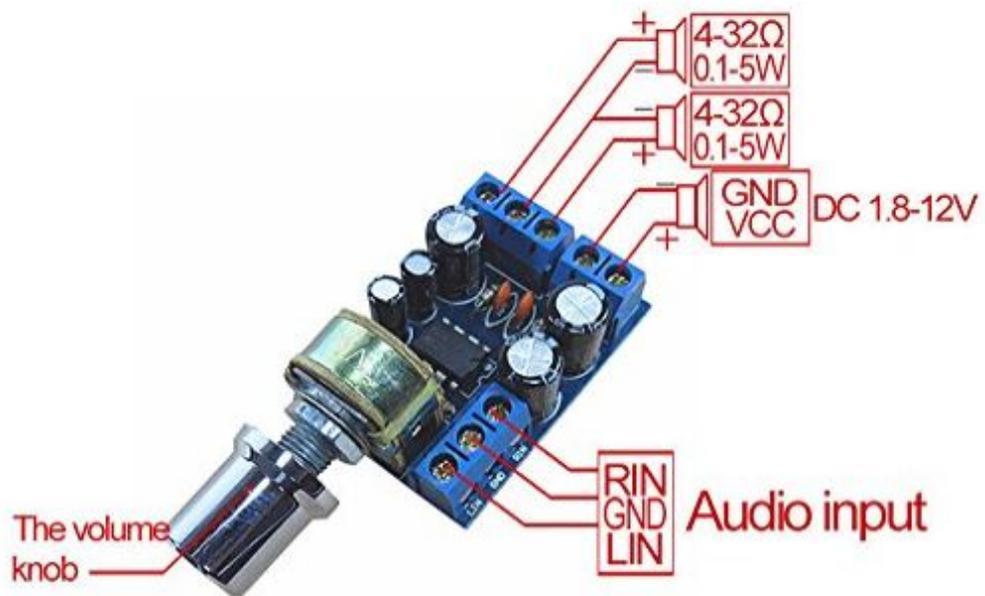


Figura C.9: Módulo amplificador de potencia empleado en los montajes.

Anexos D

Diagramas de flujo

D.1. Diagrama de flujo de generación de código Morse



Figura D.1: Diagrama de flujo explicativo de la traducción de código ASCII a Morse mediante tonos de distinta frecuencia.

D.4. Diagrama de flujo del bloque submarino

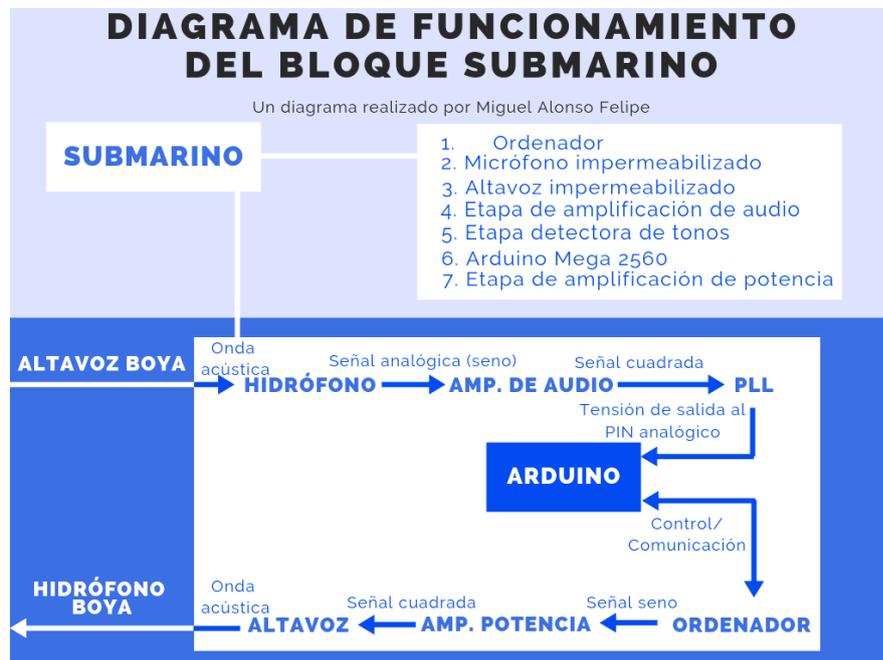


Figura D.4: Diagrama de flujo explicativo del funcionamiento del bloque submarino.

Anexos E

Información sobre submarinos

E.1. Distribución de antenas por el casco del submarino

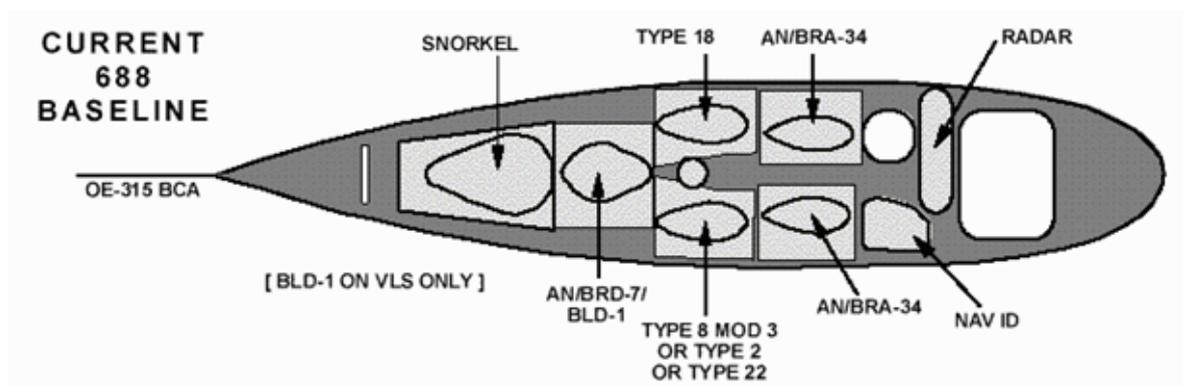


Figura E.1: Distribución de antenas en un submarino clase 688.

E.2. Tablas de los sistemas de antenas usados por los submarinos

Table A-2. Existing Submarine Communication Antenna Systems (Page 1 of 2)

Antenna	Frequency Range	Mode/ Data Rate (bps)	Application		
			SSN 688	SSN 21	SSBN 726
OE-315(V)/BRC	10 Hz - 400 MHz	Data, Voice 50 - 75	X	X	X
AN/BRA-34	VLF/LF 5 kHz - 150 kHz	Data 50	X	X	
	MF/HF 2 MHz - 30 MHz	Data, Voice 50 - 2250	X	X	
	UHF 225 MHz - 400 MHz	Data, Voice 75 - 9600	X	X	
	UHF 240 MHz - 315 MHz	Data, Voice 75 - 9600	X	X	
	IFF 950 MHz - 1150 MHz	N/A	X	X	
	GPS 1227 MHz and 1575 MHz	N/A	X	X	
OE-207/BR	VLF/LF 10 kHz - 160 kHz	Data 50			X
	MF/HF 2 MHz - 30 MHz	Data, Voice 50 - 75			X
	UHF 225 MHz - 400 MHz	Data, Voice 75 - 2400			X

Figura E.2: Tipo de sistemas de antenas usado en un submarino (1 de 2).

Table A-2. Existing Submarine Communication Antenna Systems (Page 2 of 2)

Antenna	Frequency Range	Mode/ Data Rate (bps)	Application		
			SSN 688	SSN 21	SSBN 726
OE-207/BR (Continued)	UHF 240 MHz - 320 MHz	Data, Voice 75 - 2400			X
	IFF 950 MHz - 1150 MHz	N/A			X
	GPS 1227 MHz and 1575 MHz	N/A			X
AT-441/MRC	MF/HF 2 MHz - 30 MHz	Data, Voice 50 - 75	X	X	X
AT-774/UR	MF/HF 2 MHz - 30 MHz	Data, Voice 50, 75	X	X	X
AN/BRR-6	VLF-HF 10 MHz - 160 MHz 2 MHz - 30 MHz	Data 50			X
Type 18 (RO)	MF-UHF 5 kHz-500 MHz	Data, Voice 75 - 9600	X	X	
	VHF-UHF 32 kHz - 500 MHz	Data, Voice 75 - 9600	X	X	
Type 15 (RO)	VLF-UHF 0.5 kHz - 500 MHz	Data, Voice 75 - 2400			X
Type 8 Mod 3 (RO)	VLF-UHF 12 kHz - 500 MHz	Data, Voice 75 - 2400			X
Type 8 Mod 3 EHF Antenna	EHF 43.5 - 45.5 GHz Uplink 20.2 - 21.2 GHz Downlink	Data, Voice 75 - 2400	X	X	X

Figura E.3: Tipo de sistemas de antenas usado en un submarino (2 de 2).

E.3. Tabla de frecuencias usadas por los submarinos y su utilidad

Table A-1. Submarine Communications Circuits

LINK	FREQUENCY	LINK TYPE	MODULATION	DATA RATE (bps)	CRYPTO	INFORMATION TYPE
ELF	40-80 Hz	R	MSK	LDR	KG-V-11	Record
VLF/LF VERDIN (Tactical)	14-60 kHz	R	FSK/MSK/CW	50	KG-38	Record/Tactical
VLF/LF VERDIN (Strategic)	14-60 kHz	R	MSK	LDR	KG-38	Record
VLF/LF FSK (Verdin) (Strategic)	14-60 kHz	R	FSK	50	N/A	Record
VLF/LF VALLOR	14-150 kHz	R	FSK	50	KWR-46	Record/Tactical
VLF/LF VALLOR (NATO)	14-60 kHz	R	FSK/MSK/CW	50	KWR-46	Record
VLF/LF DOUBLE DECRYPT (Verdin)	14-60 kHz	R	MSK	50	KG-38/ KWR-46	Record
VLF/LF CW/FSK	14-60 kHz	R	CW/FSK	N/A	N/A	Audio (Morse)
HF SPECIAL COMMUNICATIONS	2-30 MHz	T	Special	Special	Off-Line	Record
HF SECURE TTY	2-30 MHz	HD T/R	FSK	75	KG-84C	Record/Tactical
HF VALLOR	2-30 MHz	R	FSK	50	KWR-46	Record/Tactical
HF LINK 11	2-30 MHz	HD T/R	ISB	2250	KG-40	Tactical
HF NARROWBAND SECVOX	2-30 MHz	HD T/R	SSB	N/A	KYV-5	Voice
HF CW	2-30 MHz	HD T/R	CW	N/A	N/A	Audio (Morse)
HF CLEAR VOICE	2-30 MHz	HD T/R	SSB	N/A	N/A	Voice
HF SCANNING	2-30 MHz	R	Multiple	N/A	N/A	Audio
SLOT BUOY	168-174 MHz	T	AM	N/A	Off-Line	Audio
LDR SATCOM	240-320 MHz	HD T	DPSK	75	KG-84C	Record/Tactical
UHF OTCIXS	240-320 MHz	HD T/R	DPSK	2400	KG-84A	Record/Tactical
UHF SSIXS	240-320 MHz	HD T/R	DPSK	2400/4800	KG-84A	Record/Tactical
UHF BGIXS	240-320 MHz	HD T/R	DPSK	2400/4800	KG-84A	Record/Tactical
UHF LINK 11	225-400 MHz	HD T/R	FM	2250	KG-40	Tactical
UHF TADIXS A	240-320 MHz	R	DPSK	2400	KG-84A	Tactical
UHF TADIXS B	240-320 MHz	R	QPSK	9600	KGR-96	Tactical
UHF SATCOM BUOY	290-320 MHz	T	DPSK	2400/4800	KG-84A	Record
UHF SATCOM SECVOX	240-320 MHz	HD T/R	DPSK	2400	KYV-5	Voice
UHF VINSON	225-400 MHz	HD T/R	AM or FM	N/A	KY-58	Voice
UHF CLEAR VOICE	225-400 MHz	HD T/R	AM	N/A	N/A	Voice
UHF SECURE TTY	225-400 MHz	HD T/R	AFTS AM/FSK	75-2400	KG-84A	Record/Tactical
UHF AFSATCOM	225-400 MHz	HD T/R	FSK	75	Off-Line	Record
EHF SECURE TTY	20.2-21.2 GHz R 43.5-45.5 GHz T	FD T/R	FSK/DPSK FSK	75-2400	KG-84A	Record/Tactical
EHF SPECIAL COMMUNICATIONS	20.2-21.2 GHz R 43.5-45.5 GHz T	FD T/R	FSK/DPSK FSK	75	Off-Line	Record
EHF SECVOX	20.2-21.2 GHz R 43.5-45.5 GHz T	FD T/R	FSK/DPSK FSK	2400	KYV-5	Voice

R - Receive Only
T - Transmit Only
HD - Half Duplex
FD - Full Duplex
T/R - Transmit/Receive

bps - bits per second
MSK - Minimum Shift Keying
FSK - Frequency Shift Keying
CW - Continuous Wave
ISB - Independent Sideband
SSB - Single Sideband

DPSK - Differential Phase Shift Keying
FM - Frequency Modulation
AM - Amplitude Modulation
AFTS - Audio Frequency Tone Shift
N/A - Not Applicable
QPSK - Quadrature Phase Shift Keying

Figura E.4: Bandas de frecuencia utilizadas para la comunicación en un submarino.

Anexos F

Código de las cajas del avión, la boya y el submarino

En este Anexo se recoge el código utilizado para la recepción e interpretación del mensaje Morse (mediante tonos) que ha generado uno de los extremos de la comunicación (avión o submarino). El código utilizado para la **transmisión** se recoge en el Anexo A. En este Anexo también se localiza el código empleado en la boya para transmitir al submarino su localización GPS y recibir las peticiones que genere el submarino.

F.1. Código de la caja del avión (recepción)

```
1 //-----Bibliotecas TX-FM-----
2 #include <FMTX.h>
3
4 //-----Bibliotecas RX-FM-----
5 #include <radio.h>
6 #include <TEA5767.h>
7
8 #define FIX_BAND RADIO_BAND_FM // Sintonizamos FM
9 #define FIX_STATION 10800 // En la freq de 108MHz
10
11 TEA5767 radio; // Creamos una instancia de radio
12
13 float fm_freq = 87.50; //Frecuencia FM inicial
14
15 // Array con el abecedario en codigo Morse
16 int MorseABC[] =
17     {12,2111,2121,211,1,1121,221,1111,11,1222,212,1211,22,21,222,
18     1221,2212,121,111,2,112,1112,122,2112,2122,2211};
19
20 // Array con los numeros en codigo Morse
21 int Numeros[] =
22     {22222,12222,11222,11122,11112,11111,21111,22111,22211,22221};
```

```

21
22 // Array vacia para almacenar la palabra
23 String cadena;
24
25 // Array que contiene el mensaje final
26 String mensaje;
27
28 // Declaramos una cadena auxiliar (un valor entero)
29 int cadenaux;
30
31 // Declaramos variables auxiliares
32 int flag = 0; int cont = 0; int inicio = 0; int paso = 0; int
    barra = 0;
33
34 void setup() {
35 // Parte de RX
36 radio.init(); // Inicializar el objeto radio
37 radio.setBandFrequency(FIX_BAND, FIX_STATION); // Seleccionamos
    la emisora 87.50
38 radio.setVolume(15); // De 0 a 15
39 radio.setMono(true);
40 Serial.begin(9600); //Inicializa Serial con 9600 baud rate(
    valor por defecto)
41 Serial.println("MODULO AVION: ");
42
43 // Parte de TX
44 fmtx_init(fm_freq, EUROPE); //Para transmitir en el rango de
    frecuencias europeas, inicializa el transmisor para enviar el
    mensaje generado por el ordenador
45 }
46
47 void loop() {
48 // Leemos la entrada analogica del pin A0
49 int sensorValue = analogRead(A0);
50 // Convertimos la entrada analogica (que va de 0 -
    1023) a un voltaje (0 - 5V):
51 float voltage = sensorValue * (5.0 / 1023.0);
52 // Se introduce un retardo para la coordinacion entre
    el transmisor y el receptor
53 delay(625);
54
55 // Comenzamos detectando si es un tono u otro
56 if(voltage<= 0.50 && inicio == 1){ //Detectamos el tono
    , el punto (vemos que nivel de tension se le asigna)
57     cadena = cadena + "1"; // Almacenamos un punto
    en la i-esima posicion de la cadena
58     Serial.print(".");
59     flag = 1; // Indica que se han recibido datos
60     cont = 0; // Se reinicia el contador al recibir
    datos para no pasar al siguiente caracter

```

```

61         barra = 0; // Se reinicia la barra para comenzar
           el nuevo mensaje
62     }
63     else if(voltage >= 1 && inicio == 1){ // Detectamos el
           tono, la raya
64         cadena = cadena + "2"; // Almacenamos una raya
           en la i-esima posicion de la cadena
65         Serial.print("-");
66         flag = 1; // Indica que se han recibido datos
67         cont = 0; // Se reinicia el contador al recibir
           datos para no pasar al siguiente caracter
68         barra = 0; // Se reinicia la barra para comenzar
           el nuevo mensaje
69     }
70     else if(flag == 1){ // Si la flag = 1, quiere decir que
           ya se han comenzado a recibir datos
71         cont++;
72         if(cont > 3){ // Cuenta el numero de ciclos que pasan sin
           recibir un punto o una raya (si pasan mas de 3 ciclos
           se considera que se ha acabado de transmitir el
           caracter)
73             Serial.print("/"); // Se imprime un separador
           entre los caracteres en Morse
74             cont = 0; // Se reinicia el contador
75             paso = 1; // Se tiene en cuenta que ha pasado de
           caracter para que pueda entrar a evaluar el
           codigo recogido con el correspondiente codigo
           ASCII
76             if(barra == 1){ // Si la barra = 1 quiere decir
           que han pasado 6 ciclos sin recibir ningun
           dato y se considera que se ha acabado el
           mensaje
77                 barra = 0; // Se reinicia la
                   barra
78                 Serial.println(mensaje); // Se
                   imprime el mensaje total
                   recogido
79                 mensaje = " "; // Se limpia la
                   cadena
80                 flag = 0; // Se reinicia el flag
                   para esperar la siguiente
                   llegada de datos
81             }
82             barra = 1; // Se pone a 1 tras el paso
                   de 3 ciclos sin recibir datos
83         }
84     }
85
86     inicio = 1; // Para comenzar la recepcion de datos sin
           que se imprima basura por el serial antes de que se

```

```

      quiera iniciar el programa
87
88     if(paso == 1){// Si paso = 1 quiere decir que se
      termino el caracter Morse transmitido y se pasa a
      evaluar su correspondencia
89     paso = 0;
90     cadenaux = "";
91     cadenaux = cadena.toInt(); // Convertimos a entero la
      cadena para evaluarla
92     cadena = ' ';
93
94     // Comparamos los disntitos casos
95     if(MorseABC[0] == cadenaux){
96         mensaje = mensaje + 'a';
97     }
98     else if(MorseABC[1] == cadenaux){
99         mensaje = mensaje + 'b';
100    }
101    else if(MorseABC[2] == cadenaux){
102        mensaje = mensaje + 'c';
103    }
104    else if(MorseABC[3] == cadenaux){
105        mensaje = mensaje + 'd';
106    }
107    else if(MorseABC[4] == cadenaux){
108        mensaje = mensaje + 'e';
109    }
110    else if(MorseABC[5] == cadenaux){
111        mensaje = mensaje + 'f';
112    }
113    else if(MorseABC[6] == cadenaux){
114        mensaje = mensaje + 'g';
115    }
116    else if(MorseABC[7] == cadenaux){
117        mensaje = mensaje + 'h';
118    }
119    else if(MorseABC[8] == cadenaux){
120        mensaje = mensaje + 'i';
121    }
122    else if(MorseABC[9] == cadenaux){
123        mensaje = mensaje + 'j';
124    }
125    else if(MorseABC[10] == cadenaux){
126        mensaje = mensaje + 'k';
127    }
128    else if(MorseABC[11] == cadenaux){
129        mensaje = mensaje + 'l';
130    }
131    else if(MorseABC[12] == cadenaux){
132        mensaje = mensaje + 'm';

```

```

133     }
134     else if(MorseABC[13] == cadenaux){
135         mensaje = mensaje + 'n';
136     }
137     else if(MorseABC[14] == cadenaux){
138         mensaje = mensaje + 'o';
139     }
140     else if(MorseABC[15] == cadenaux){
141         mensaje = mensaje + 'p';
142     }
143     else if(MorseABC[16] == cadenaux){
144         mensaje = mensaje + 'q';
145     }
146     else if(MorseABC[17] == cadenaux){
147         mensaje = mensaje + 'r';
148     }
149     else if(MorseABC[18] == cadenaux){
150         mensaje = mensaje + 's';
151     }
152     else if(MorseABC[19] == cadenaux){
153         mensaje = mensaje + 't';
154     }
155     else if(MorseABC[20] == cadenaux){
156         mensaje = mensaje + 'u';
157     }
158     else if(MorseABC[21] == cadenaux){
159         mensaje = mensaje + 'v';
160     }
161     else if(MorseABC[22] == cadenaux){
162         mensaje = mensaje + 'w';
163     }
164     else if(MorseABC[23] == cadenaux){
165         mensaje = mensaje + 'x';
166     }
167     else if(MorseABC[24] == cadenaux){
168         mensaje = mensaje + 'y';
169     }
170     else if(MorseABC[25] == cadenaux){
171         mensaje = mensaje + 'z';
172     }
173     else if(Numeros[0] == cadenaux){
174         mensaje = mensaje + '0';
175     }
176     else if(Numeros[1] == cadenaux){
177         mensaje = mensaje + '1';
178     }
179     else if(Numeros[2] == cadenaux){
180         mensaje = mensaje + '2';
181     }
182     else if(Numeros[3] == cadenaux){

```

```

183         mensaje = mensaje + '3';
184     }
185     else if(Numeros[4] == cadenaux){
186         mensaje = mensaje + '4';
187     }
188     else if(Numeros[5] == cadenaux){
189         mensaje = mensaje + '5';
190     }
191     else if(Numeros[6] == cadenaux){
192         mensaje = mensaje + '6';
193     }
194     else if(Numeros[7] == cadenaux){
195         mensaje = mensaje + '7';
196     }
197     else if(Numeros[8] == cadenaux){
198         mensaje = mensaje + '8';
199     }
200     else if(Numeros[9] == cadenaux){
201         mensaje = mensaje + '9';
202     }
203 }
204 }

```

F.2. Código de la caja de la boya(transmisión y recepción)

```

1  #include <SoftwareSerial.h>// Incluimos la libreria
   SoftwareSerial que nos permite declarar ciertos pines como
   puertos serie
2  #include <TinyGPS.h>// Incluimos la libreria TinyGPS que nos
   ayuda a tener una extraccion de datos mas sencilla del modulo
   GPS
3  #define vcoONOFF 4// Definimos el PIN 4 para habilitar el VCO
4  #define vcotono 12// Definimos el PIN 12 para dar las tensiones
   que generaran los tonos en el VCO (puntos y rayas) -> en
   alta se genera un punto y en baja la raya
5
6  //-----Bibliotecas TX-FM-----
7  #include <FM_TX.h>
8
9  //-----Bibliotecas RX-FM-----
10 #include <radio.h>
11 #include <TEA5767.h>
12
13 #define FIX_BAND RADIO_BAND_FM // Sintonizamos FM
14 #define FIX_STATION 10800 // En la freq de 108MHz
15

```

```

16 TEA5767 radio; // Creamos una instancia de radio
17
18 float fm_freq = 108.00; //Frecuencia FM inicial
19
20 TinyGPS gps;// Declaramos el objeto gps de la clase TinyGPS
21 SoftwareSerial serialgps(50,51);// Declaramos el pin 50 Tx y 51
    Rx (nuevos puertos serie para Arduino MEGA2560)
22
23 //Declaracion de variables globales
24 // Array para almacenar el mensaje en Morse
25 String GPS_Morse_lon="";
26 String GPS_Morse_lat="";
27
28 // Array para almacenar la ubicacion
29 int ubicacion[20];
30 char signo = "";
31
32 // Variables para almacenar la ubicacion
33 float latitude = 0, longitude = 0;
34 String cadena_latitud[9];String latitud_rot[9]="";
35 String cadena_longitud[9];String longitud_rot[9]="";
36
37 // Array vacia para almacenar la palabra
38 String cadena;
39
40 // Array que contiene el mensaje final
41 String mensaje;
42
43 // Array para comparar
44 String num_sig[]={ "0","1","2","3","4","5","6","7","8","9","n",""
    p"};
45
46 // Declaramos una cadena auxiliar (un valor entero)
47 int cadenaux;
48
49 // Declaramos variables auxiliares
50 int flag = 0; int cont = 0; int inicio = 0; int paso = 0; int
    barra = 0;
51 float numero = 0; float numero_aux = 0; int j = 0; int i = 0;
    int k = 0; float division = 10; double doble = 0; int entero
    = 0; String numero2 = "";
52
53 // La variable para comenzar con la obtencion GPS
54 int solicitud_gps = 0;
55
56 void setup(){
57 Serial.begin(115200);// Iniciamos el puerto serie a 115200
    baudios para imprimir por el monitor serie la informacion
    extraida

```

```

58  serialgps.begin(9600); // Iniciamos el puerto serie por el cual
    se comunican el modulo GPS y TX-RX con el Arduino
59
60  // Parte de RX
61  radio.init(); // Inicializar el objeto radio
62  radio.setBandFrequency(FIX_BAND, FIX_STATION); // Seleccionamos
    la emisora 108.00
63  radio.setVolume(15); // De 0 a 15
64  radio.setMono(true);
65  Serial.begin(9600); // Inicializa Serial con 9600 baud rate(
    valor por defecto)
66  Serial.println("MODULO BOYA: ");
67
68  // Parte de TX
69  fmtx_init(fm_freq, EUROPE); // Para transmitir en el rango de
    frecuencias europeas
70
71  }
72
73  void loop(){
74  // Leemos la entrada analogica del pin A0
75  int sensorValue = analogRead(A0);
76  // Convertimos la entrada analogica (que va de 0 - 1023) a un
    voltaje (0 - 5V):
77  float voltage = sensorValue * (5.0 / 1023.0);
78
79  //-----
80  //-----OBTENCION UBICACION GPS-----
81  //-----
82
83  // Detectamos si llega un tono de 500 Hz que se corresponde con
    la petición de GPS, para ello comprobamos el nivel de
    tension
84  // a la entrada del pin analogico 0
85
86  if(voltage > 3.3){ // Si se detecta una solicitud de GPS se pasa
    a obtener la ubicacion
87  solicitud_gps = 0; // Se reinicia la variable
88  while(latitude == 0 && longitude == 0){
89  while(serialgps.available()) // Mientras haya datos disponibles
    en el puerto serie declarado por software...
90  {
91  int c = serialgps.read(); // Extraemos la informacion del puerto
    serie
92
93  if(gps.encode(c)){ // Cada byte de datos NEMA debe darse a
    TinyGPS mediante el uso de encode().
94  // Devuelve "true" cuando los nuevos datos se han descodificado
    completamente y se pueden utilizar.

```

```

95  gps.f_get_position(&latitude, &longitude);// Obtenemos la
    latitud y la longitud
96  // Se imprime por el monitor serie del ordenador la informacion
    obtenida
97  Serial.print("Latitud/Longitud: ");
98  Serial.print(latitude,5);
99  Serial.print(", ");
100 Serial.println(longitude,5);
101 }
102 }
103 }
104 // Tras obtener la ubicacion saldra del bucle while
105 // Convertimos a Morse y las tensiones que se introduzcan en
    el pin analogico para introducir las al VCO
106 // Se comienza con un bucle para convertir la latitud y la
    longitud
107 for(j=0;j<2;j++){// El bucle se ejecuta dos veces (Una para la
    latitud y otra para la longitud)
108 if(j == 0){// Si j == 0, quiere decir que es la primera
    iteracion y se trabaja la latitud
109 numero = latitude;
110 }
111 else if(j == 1){// Si el j == 1, se esta en la segunda
    iteracion y se trabaja la longitud
112 numero = longitude;
113 }
114
115 if(numero<0){ //Se comprueba si el numero es positivo o
    negativo y se le adjunta un prefijo
116 signo = 'n';
117 numero = -numero;// En caso de ser negativo se convierte a
    positivo para trabajar con el
118 }
119 else{
120 signo = 'p';
121 }
122
123 // Hay que calcular la parte entera de la mantisa del numero en
    coma flotante
124 numero_aux = numero;// Se almacena el numero en una variable
    auxiliar para no perder el valor real con las operaciones
125
126 while((int)numero_aux != 0){// Se divide el numero hasta que su
    parte entera sea 0, es decir se desplaza la coma hasta que
    el numero quede de la forma 0.[numero]
127 numero_aux = numero_aux/division;
128 k++;// Contamos el numero de digitos que tiene el entero
129 }
130

```

```

131  numero = numero*100000;// Para coger 5 decimales se multiplica
      por 10^5 de forma que se corre la coma hasta ese punto de los
      decimales
132
133  while((int)numero != 0){
134  // Se almacena el numero en la string para poder transformarlo
      a entero
135  numero2 = numero;
136
137  if(j == 0){
138  // Nos quedamos con el resto matematico y extraemos uno a uno
      los numeros que componen la coordenada de derecha a izquierda
139  doble = (int)(numero2.toInt()%10); // Se almacena en un doble
      para conservar la precision y evitar redondeos
140  cadena_latitud[i] = (int)doble;
141  }
142  else if(j == 1){
143  // Nos quedamos con el resto matematico y extraemos uno a uno
      los numeros que componen la coordenada de derecha a izquierda
144  doble = numero2.toInt()%10; // Se almacena en un doble para
      conservar la precision y evitar redondeos
145  cadena_longitud[i] = (int)doble;
146  }
147
148  // Se corre la coma en el flotante
149  numero = numero/division;
150  i++;
151  }
152
153  //Distinguimos las coordenadas
154  if(j == 0){
155  cadena_latitud[i] = signo;
156  cadena_latitud[i+1] = k;
157  }
158  else if(j == 1){
159  cadena_longitud[i] = signo;
160  cadena_longitud[i+1] = k;
161  }
162
163  k = 0; // Reiniciamos el contador de entero
164  i = 0; // Reiniciamos el recorrido de la cadena
165  }
166
167  //-----SE ROTAN LAS CADENAS
      -----
168  // Ahora rotamos las cadenas para dejar la ubicacion bien
      ordenada
169  int m = 0;
170  Serial.print("Latitud: ");

```

```

171 for (int u = 9 - 1; u >= 0; u--){// Recorremos decrementando la
    cadena
172 latitud_rot[m] = cadena_latitud[u];// Vamos concatenando en la
    auxiliar la cadena al revés
173 Serial.print(latitud_rot[m]);
174 m++;
175 }
176 m = 0;
177 Serial.println("");
178 Serial.print("Longitud: ");
179 for (int u = 9 - 1; u >= 0; u--){// Recorremos decrementando la
    cadena
180 longitud_rot[m] = cadena_longitud[u];// Vamos concatenando en
    la auxiliar la cadena al revés
181 Serial.print(longitud_rot[m]);
182 m++;
183 }
184 //
-----

185 // Comparamos nuestra cadena con todos los números para
    traducirlo a Morse y con los caracteres n y p
186 for(m = 0;m<9;m++){
187 //-----LONGITUD-----
188 if(num_sig[0] == longitud_rot[m]){
189 GPS_Morse_lon += "-----/";
190 }
191 else if(num_sig[1] == longitud_rot[m]){
192 GPS_Morse_lon += ".----/";
193 }
194 else if(num_sig[2] == longitud_rot[m]){
195 GPS_Morse_lon += "..---/";
196 }
197 else if(num_sig[3] == longitud_rot[m]){
198 GPS_Morse_lon += "...--/";
199 }
200 else if(num_sig[4] == longitud_rot[m]){
201 GPS_Morse_lon += "....-/";
202 }
203 else if(num_sig[5] == longitud_rot[m]){
204 GPS_Morse_lon += "...../";
205 }
206 else if(num_sig[6] == longitud_rot[m]){
207 GPS_Morse_lon += "-..../";
208 }
209 else if(num_sig[7] == longitud_rot[m]){
210 GPS_Morse_lon += "--.../";
211 }
212 else if(num_sig[8] == longitud_rot[m]){
213 GPS_Morse_lon += "---../";

```

```

214 }
215 else if(num_sig[9] == longitud_rot[m]){
216 GPS_Morse_lon += "----./";
217 }
218 else if(num_sig[10] == longitud_rot[m]){
219 GPS_Morse_lon += "-./";
220 }
221 else if(num_sig[11] == longitud_rot[m]){
222 GPS_Morse_lon += ".--./";
223 }
224 }
225 for(m = 0;m<9;m++){
226 //-----LATITUD-----
227 if(num_sig[0] == latitud_rot[m]){
228 GPS_Morse_lat += "----/";
229 }
230 else if(num_sig[1] == latitud_rot[m]){
231 GPS_Morse_lat += ".----/";
232 }
233 else if(num_sig[2] == latitud_rot[m]){
234 GPS_Morse_lat += "..---/";
235 }
236 else if(num_sig[3] == latitud_rot[m]){
237 GPS_Morse_lat += "...--/";
238 }
239 else if(num_sig[4] == latitud_rot[m]){
240 GPS_Morse_lat += "....-/";
241 }
242 else if(num_sig[5] == latitud_rot[m]){
243 GPS_Morse_lat += "...../";
244 }
245 else if(num_sig[6] == latitud_rot[m]){
246 GPS_Morse_lat += "-..../";
247 }
248 else if(num_sig[7] == latitud_rot[m]){
249 GPS_Morse_lat += "--.../";
250 }
251 else if(num_sig[8] == latitud_rot[m]){
252 GPS_Morse_lat += "---../";
253 }
254 else if(num_sig[9] == latitud_rot[m]){
255 GPS_Morse_lat += "----./";
256 }
257 else if(num_sig[10] == latitud_rot[m]){
258 GPS_Morse_lat += "-./";
259 }
260 else if(num_sig[11] == latitud_rot[m]){
261 GPS_Morse_lat += ".--./";
262 }
263 }

```

```

264
265 // Se concatena como marcador final una 'f'
266 GPS_Morse_lat += "f";
267 GPS_Morse_lon += "f";
268
269 // Para comprobar que se ha formado bien el codigo se imprime
    por pantalla
270 Serial.println("");
271 Serial.println("Latitud Morse:");
272 Serial.print(GPS_Morse_lat);
273 Serial.println("");
274 Serial.println("Longitud Morse:");
275 Serial.print(GPS_Morse_lon);
276
277 //-----ATENCION-----
278 //ANTES DE COMENZAR LA TRANSMISION DEBE CORTAR LA COMUNICACION
279 //-----
280 digitalWrite(vcopin, LOW); // Se pone el pin 4 en bajo, de forma
    que se habilita la comunicacion
281
282 Serial.println("");
283 Serial.println("Transmitiendo latitud...");
284 // Una vez obtenido el array con codigo Morse, se traduce a
    niveles de tension para sacar por el pin analogico de Arduino
285 //-----SE TRANSMITE LA
    LATITUD
    -----
286 for(m = 0; m < GPS_Morse_lat.length(); m++){
287     if(GPS_Morse_lat[m] == '.') {
288         // Se escribe un nivel de tension para generar el tono que
            representa el punto
289         Serial.print("1");
290     }
291     else if(GPS_Morse_lat[m] == '-') {
292         // Se escribe un nivel de tension para generar el tono que
            representa la raya
293         Serial.print("-");
294         digitalWrite(vcotono, LOW); // Tras el divisor resistivo -> 3,2
            V
295         digitalWrite(vcoONOFF, LOW); // Se habilita el VCO
296         delay(800); // Tiempo que dura la generacion del tono
297         digitalWrite(vcoONOFF, HIGH); // Se deshabilita el VCO
298     }
299     else if(GPS_Morse_lat[m] == '/') { // Se finaliza el primer
        caracter y se espera para transmitir el siguiente
300         Serial.print("/");
301         delay(2000); // Se introduce el delay correspondiente para que
            el receptor entienda que se cambia de caracter
302     }

```

```

303 else if(GPS_Morse_lat[m] == 'f'){// Se finaliza la transmision
      de la latitud
304 Serial.print("//Fin");
305 delay(4000);
306 break;
307 }
308 }
309 Serial.println("");
310 Serial.println("Transmitiendo longitud...");
311 //Se repite el mismo proceso para transmitir la longitud
312 //-----SE TRANSMITE LA
      LONGITUD
      -----
313 for(m = 0;m<GPS_Morse_lon.length();m++){
314 if(GPS_Morse_lon[m] == '.'){
315 // Se escribe un nivel de tension para generar el tono que
      representa el punto
316 Serial.print(".");
317 digitalWrite(vcotono, HIGH); // Tras el divisor resistivo ->
      1,4V -> 4000 Hz
318 digitalWrite(vcoONOFF, LOW); // Se habilita el VCO
319 delay(800);// Tiempo que dura la generacion del tono
320 digitalWrite(vcoONOFF, HIGH); // Se deshabilita el VCO
321 }
322 else if(GPS_Morse_lon[m] == '-'){
323 // Se escribe un nivel de tension para generar el tono que
      representa la raya
324 Serial.print("-");
325 digitalWrite(vcotono, LOW); // Tras el divisor resistivo -> 3,2
      V -> 7000 Hz
326 digitalWrite(vcoONOFF, LOW); // Se habilita el VCO
327 delay(800); // Tiempo que dura la generacion del tono
328 digitalWrite(vcoONOFF, HIGH); // Se deshabilita el VCO
329 }
330 else if(GPS_Morse_lon[m] == '/'){// Se finaliza el primer
      caracter y se espera para transmitir el siguiente
331 Serial.print("/");
332 delay(2000);// Se introduce el delay correspondiente para que
      el receptor entienda que se cambia de caracter
333 }
334 else if(GPS_Morse_lon[m] == 'f'){// Se finaliza la transmision
      de la latitud
335 Serial.print("//Fin");
336 delay(4000);
337 break;
338 }
339 }
340 //
      -----

```

```

341 // Cuando se termine la comunicacion se deshabilita el VCO
342 digitalWrite(vcopin, HIGH); // Se pone el pin 4 en alto, de esta
    forma se pondra la patilla 5 del CD4046B en alto y se
    deshabilita
343 }
344 }

```

F.3. Código de la caja del submarino (recepción)

```

1 // Array con el abecedario en codigo Morse
2 int MorseABC[] =
    {12,2111,2121,211,1,1121,221,1111,11,1222,212,1211,22,21,222,1221,2212
3
4 // Array con los numeros en codigo Morse
5 int Numeros[] =
    {22222,12222,11222,11122,11112,11111,21111,22111,22211,22221};
6
7 // Array vacia para almacenar la palabra
8 String cadena;
9
10 // Array que contiene el mensaje final
11 String mensaje;
12
13 // Declaramos una cadena auxiliar (un valor entero)
14 int cadenaux;
15
16 // Declaramos variables auxiliares
17 int flag = 0; int cont = 0; int inicio = 0; int paso = 0; int
    barra = 0;
18
19 void setup() {
20 Serial.begin(9600); //Inicializa Serial con 9600 baud rate(
    valor por defecto)
21 Serial.println("MODULO SUBMARINO: ");
22 }
23
24 void loop() {
25 // Leemos la entrada analogica del pin A0
26 int sensorValue = analogRead(A0);
27 // Convertimos la entrada analogica (que va de 0 - 1023) a un
    voltaje (0 - 5V):
28 float voltage = sensorValue * (5.0 / 1023.0);
29 // Se introduce un retardo para la coordinacion entre el
    transmisor y el receptor
30 delay(625);

```

```

31
32 // Comenzamos detectando si es un tono u otro
33 if(voltage<= 0.9 && inicio == 1){ //Detectamos el tono, el
    punto (vemos que nivel de tension se le asigna)
34 cadena = cadena + "1"; // Almacenamos un punto en la i-esima
    posicion de la cadena
35 Serial.print(".");
36 flag = 1; // Indica que se han recibido datos
37 cont = 0; // Se reinicia el contador al recibir datos para no
    pasar al siguiente caracter
38 barra = 0; // Se reinicia la barra para comenzar el nuevo
    mensaje
39 }
40 else if(voltage>=2.1 && inicio == 1){ // Detectamos el tono, la
    raya
41 cadena = cadena + "2"; // Almacenamos una raya en la i-esima
    posicion de la cadena
42 Serial.print("-");
43 flag = 1; // Indica que se han recibido datos
44 cont = 0; // Se reinicia el contador al recibir datos para no
    pasar al siguiente caracter
45 barra = 0; // Se reinicia la barra para comenzar el nuevo
    mensaje
46 }
47 else if(flag == 1){ // Si la flag = 1, quiere decir que ya se
    han comenzado a recibir datos
48 cont++;
49 if(cont>3){ // Cuenta el numero de ciclos que pasan sin recibir
    un punto o una raya (si pasan mas de 3 ciclos se considera
    que se ha acabado de transmitir el caracter)
50 Serial.print("/"); // Se imprime un separador entre los
    caracteres en Morse
51 cont = 0; // Se reinicia el contador
52 paso = 1; // Se tiene en cuenta que ha pasado de caracter para
    que pueda entrar a evaluar el codigo recogido con el
    correspondiente codigo ASCII
53 if(barra == 1){ // Si la barra = 1 quiere decir que han pasado 6
    ciclos sin recibir ningun dato y se considera que se ha
    acabado el mensaje
54 barra = 0; // Se reinicia la barra
55 Serial.println(mensaje); // Se imprime el mensaje total recogido
56 mensaje = " "; // Se limpia la cadena
57 flag = 0; // Se reinicia el flag para esperar la siguiente
    llegada de datos
58 }
59 barra = 1; // Se pone a 1 tras el paso de 3 ciclos sin recibir
    datos
60 }
61 }
62

```

```

63 inicio = 1; // Para comenzar la recepcion de datos sin que se
    imprima basura por el serial antes de que se quiera iniciar
    el programa
64
65 if(paso == 1){ // Si paso = 1 quiere decir que se termino el
    caracter Morse transmitido y se pasa a evaluar su
    correspondencia
66 paso = 0;
67 cadenaux = "";
68 cadenaux = cadena.toInt(); // Convertimos a entero la cadena
    para evaluarla
69 cadena = ' ';
70
71 // Comparamos los disntitos casos
72 if(MorseABC[0] == cadenaux){
73 mensaje = mensaje + 'a';
74 }
75 else if(MorseABC[1] == cadenaux){
76 mensaje = mensaje + 'b';
77 }
78 else if(MorseABC[2] == cadenaux){
79 mensaje = mensaje + 'c';
80 }
81 else if(MorseABC[3] == cadenaux){
82 mensaje = mensaje + 'd';
83 }
84 else if(MorseABC[4] == cadenaux){
85 mensaje = mensaje + 'e';
86 }
87 else if(MorseABC[5] == cadenaux){
88 mensaje = mensaje + 'f';
89 }
90 else if(MorseABC[6] == cadenaux){
91 mensaje = mensaje + 'g';
92 }
93 else if(MorseABC[7] == cadenaux){
94 mensaje = mensaje + 'h';
95 }
96 else if(MorseABC[8] == cadenaux){
97 mensaje = mensaje + 'i';
98 }
99 else if(MorseABC[9] == cadenaux){
100 mensaje = mensaje + 'j';
101 }
102 else if(MorseABC[10] == cadenaux){
103 mensaje = mensaje + 'k';
104 }
105 else if(MorseABC[11] == cadenaux){
106 mensaje = mensaje + 'l';
107 }

```

```

108     else if(MorseABC[12] == cadenaux){
109         mensaje = mensaje + 'm';
110     }
111     else if(MorseABC[13] == cadenaux){
112         mensaje = mensaje + 'n';
113     }
114     else if(MorseABC[14] == cadenaux){
115         mensaje = mensaje + 'o';
116     }
117     else if(MorseABC[15] == cadenaux){
118         mensaje = mensaje + 'p';
119     }
120     else if(MorseABC[16] == cadenaux){
121         mensaje = mensaje + 'q';
122     }
123     else if(MorseABC[17] == cadenaux){
124         mensaje = mensaje + 'r';
125     }
126     else if(MorseABC[18] == cadenaux){
127         mensaje = mensaje + 's';
128     }
129     else if(MorseABC[19] == cadenaux){
130         mensaje = mensaje + 't';
131     }
132     else if(MorseABC[20] == cadenaux){
133         mensaje = mensaje + 'u';
134     }
135     else if(MorseABC[21] == cadenaux){
136         mensaje = mensaje + 'v';
137     }
138     else if(MorseABC[22] == cadenaux){
139         mensaje = mensaje + 'w';
140     }
141     else if(MorseABC[23] == cadenaux){
142         mensaje = mensaje + 'x';
143     }
144     else if(MorseABC[24] == cadenaux){
145         mensaje = mensaje + 'y';
146     }
147     else if(MorseABC[25] == cadenaux){
148         mensaje = mensaje + 'z';
149     }
150     else if(Numeros[0] == cadenaux){
151         mensaje = mensaje + '0';
152     }
153     else if(Numeros[1] == cadenaux){
154         mensaje = mensaje + '1';
155     }
156     else if(Numeros[2] == cadenaux){
157         mensaje = mensaje + '2';

```

```
158 }
159 else if(Numeros[3] == cadenaux){
160     mensaje = mensaje + '3';
161 }
162 else if(Numeros[4] == cadenaux){
163     mensaje = mensaje + '4';
164 }
165 else if(Numeros[5] == cadenaux){
166     mensaje = mensaje + '5';
167 }
168 else if(Numeros[6] == cadenaux){
169     mensaje = mensaje + '6';
170 }
171 else if(Numeros[7] == cadenaux){
172     mensaje = mensaje + '7';
173 }
174 else if(Numeros[8] == cadenaux){
175     mensaje = mensaje + '8';
176 }
177 else if(Numeros[9] == cadenaux){
178     mensaje = mensaje + '9';
179 }
180 }
181 }
```

Referencias

- [1] ELECTRONIC DESIGN, "Conquering Underwater-Communication Challenges with Electro-optics, Radar", último acceso 3/6/2019.
- [2] MIT MEDIA LAB, FRANCESCO TONOLINI, FADEL ADIB, "Networking across Boundaries: Enabling Wireless Communication through the Water-Air Interface", <http://www.mit.edu/~fadel/papers/TARF-paper.pdf>, último acceso 14/08/2019.
- [3] ARDUINO COMPANY, "SoftwareSerial Library", <https://www.arduino.cc/en/Reference/SoftwareSerial>, último acceso 23/6/2019.
- [4] GITHUB INC, <https://github.com/mikalhart/TinyGPS/blob/master/TinyGPS.h>, último acceso 27/6/2019.
- [5] MONOLITHIC DIGITAL STEREO FM TRANSMITTER. "Radio-Station-on-a-Chip. KTMicro", <https://datasheetspdf.com/pdf/705403/KTMicro/KT0803K/1>, último acceso 29/6/2019.
- [6] ARDUINO COMPANY, "Advanced I/O", <https://www.arduino.cc/reference/en/#functions>, último acceso 1/7/2019.
- [7] NORDIC SEMICONDUCTOR, "Single chip 2.4 GHz Transceiver nRF24L01", https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf, último acceso 4/7/2019.
- [8] NXP SEMICONDUCTORS, "Low-power FM stereo radio for handheld applications TEA5767", <https://www.sparkfun.com/datasheets/Wireless/General/TEA5767.pdf>, último acceso 29/7/2019.
- [9] DEPARTAMENTO DE ING. ELÉCTRICA, ELECTRÓNICA Y AUTOMÁTICA, UNIVERSIDAD DE EXTREMADURA, JOAQUÍN APARICIO, ENRIQUE GARCÍA, ANA JIMÉNEZ, FERNANDO ÁLVAREZ, JESÚS UREÑA, "Modelos de Propagación de Señales Acústicas en Entornos Subacuáticos", http://geintra-uah.org/system/files/modelos_de_propagacion_de_senales_acusticas_en_entornos_s_vdef.pdf, último acceso 16/08/2019.
- [10] IEEE COMMUNICATIONS MAGAZINE, LIANE GROBE, ANAGNOSTIS PARASKEVOPOULOS, JONAS HILT, DOMINIC SCHULZ, FRIEDRICH LASAK, FLORIAN HARTLIEB, CHRISTOPH KOTTKE, VOLKER JUNGNICHEL, KLAUS-DIETER LANGER, "High-Speed Visible Light Communication Systems",

<https://ieeexplore.ieee.org/abstract/document/6685758/authors#authors>, último acceso 16/08/2019.

- [11] GITHUB INC, RF24 CLASS REFERENCE, <http://maniacbug.github.io/RF24/classRF24.html>, último acceso 16/08/2019.
- [12] TEXAS INSTRUMENTS, DAVID K. MORGAN, "CD4046B Phase-Locked Loop: A Versatile Building Block for Micropower Digital and Analog Applications", <http://www.ti.com/lit/an/scha002a/scha002a.pdf>, último acceso 18/08/2019.
- [13] IEEE PHOTONICS JOURNAL, CHAO WANG, HONG-YI YU, YI-JUN ZHU, "A long distance underwater visible light communication system with single photon avalanche diode", <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7551212>, último acceso 19/08/2019.
- [14] SUBMARINE FORCE STRATEGIC PLAN, SUBMARINE COMMUNICATIONS PROGRAM MANAGER (SPACE AND NAVAL WARFARE SYSTEMS COMMAND [SPAWAR] PMW 173) FOR THE ASSISTANT CHIEF OF NAVAL OPERATIONS (ACNO), SUBMARINE WARFARE DIVISION (N87) AND THE DIRECTOR, SPACE AND ELECTRONIC WARFARE (N6), "Submarine Communications Master Plan, 1995", <https://fas.org/man/dod-101/navy/docs/scmp/intro.htm>, último acceso 19/08/2019.
- [15] NEWNES, IAN POOLE, "Newnes Guide to Radio and Communications Technology, 1th Edition, 2003", última consulta 22/08/2019.
- [16] IEE, MILICA STOJANOVIC, "1995, Underwater acoustic communications In Electro/95 International Professional Program Proceedings", última consulta 22/08/2019.
- [17] UNIVERSITY OF CALIFORNIA, SAN DIEGO VISIBILITY LABORATORY OF THE SCRIPPS INSTITUTION OF OCEANOGRAPHY LAJOLLA, CALIFORNIA 92093, ROSWELL W. AUSTIN, GEORGE HALIKAS, "The Index of Refraction of Seawater", http://misclab.umeoce.maine.edu/education/VisibilityLab/reports/SIO_76-1.pdf, último acceso 30/08/2019.