



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería Informática

Universitat Politècnica de València

Tesina de Máster

Máster Universitario en Ingeniería de Computadores, 2013-2014

## Localización y posicionamiento de equipos subacuáticos mediante ultrasonidos

**Autor:**

D. Raúl Sáez Cañete

**Tutor:**

D. Pedro Yuste Pérez

**Co-tutor:**

D. Àngel Perles Ivars

Valencia, septiembre 2014

# Índice de contenido

1. Introducción .....	4
2. Objetivo.....	5
3. Estado del arte .....	5
3.1. Técnicas de localización tradicionales .....	6
3.2. Técnicas de localización para redes inalámbricas de sensores .....	10
3.2.1. Técnicas de localización centralizadas .....	11
3.2.2. Técnicas de localización distribuidas .....	13
4. Desarrollo.....	16
4.1. El módem ITACA-S1000.....	18
4.2. El puente de comunicaciones .....	20
4.3. API de comunicación IWCA .....	21
4.3.1. Tipos de mensajes .....	22
4.3.2. Adaptación a la aplicación .....	25
4.3.3. Protocolo de comunicación .....	28
4.3.4. Implementación .....	28
4.3.5. Aplicación de test .....	31
4.4. API de comunicación IUMOTE .....	32
4.4.1. Tipos de mensajes .....	32
4.4.2. Protocolo de comunicación .....	33
4.4.3. Implementación .....	34
4.4.4. Aplicación de test .....	35
4.5. Protocolo de acceso al medio.....	36
4.6. Aplicación de obtención de tiempos de vuelo.....	37
4.6.1. Visión general .....	37
4.6.2. Implementación del protocolo TDMA .....	40
4.6.3. Obtención del tiempo de vuelo .....	41
4.6.4. Implementación de la aplicación .....	43
5. Experimentación y resultados .....	50
5.1.1. Caracterización de los tiempos para la obtención del tiempo de vuelo. ....	51
5.1.2. Ajuste de la ranura TDMA .....	60
5.1.3. Campaña de obtención de tiempos de vuelo .....	62
6. Conclusiones y trabajo futuro .....	69
7. Referencias.....	72

## Índice de figuras

Figura 1. Localización en USBL .....	7
Figura 2. Localización en iUSBL.....	8
Figura 3. Localización en SBL .....	8
Figura 4. Localización en LBL .....	9
Figura 5. Localización en GIB .....	10
Figura 6. Solución propuesta para mejorar el sistema de navegación de un AUV .....	17
Figura 7. Geometría del haz del transductor acústico .....	19
Figura 8. Sistema de comunicación basado en el módem ITACA-S1000 .....	20
Figura 9. Implementación en capas de la API IWCA en el módem .....	29
Figura 10. Implementación en capas de la API IWCA en el puente .....	30
Figura 11. Interfaz gráfica de la aplicación de demostración de la API IWCA.....	32
Figura 12. Secuencia de eventos del protocolo de comunicación IUMOTE.....	34
Figura 13. Implementación en capas de la API IUMOTE.....	34
Figura 14. Interfaz gráfica de la aplicación de demostración de la API IUMOTE .....	35
Figura 15. Intercambio de mensajes en la obtención del tiempo de vuelo. ....	38
Figura 16. Implementación en capas de la aplicación de obtención de tiempos de vuelo .....	44
Figura 17. Diagrama de estados de la aplicación en el sonar .....	45
Figura 18. Diagrama de flujo del proceso de obtención de tiempos de vuelo.....	47
Figura 19. Diagrama de estados de la aplicación en la baliza .....	48
Figura 20. Montaje en el acuario para realizar los experimentos.....	51
Figura 21. Envío del mensaje serie obtenido en el osciloscopio.....	52
Figura 22. Campaña de prueba .....	62
Figura 23. Resultados del experimento 1 de la campaña de pruebas. ....	63
Figura 24. Resultados del experimento 2 de la campaña de pruebas. ....	64
Figura 25. Resultados del experimento 3 de la campaña de pruebas. ....	65
Figura 26. Resultados del experimento 4 de la campaña de pruebas. ....	66
Figura 27. Resultados del experimento de obtención de distancias con osciloscopio .....	69

## Índice de tablas

Tabla 1. Órdenes de la API IWCA.....	23
Tabla 2. Definiciones de la API IWCA.....	26
Tabla 3. Tiempo de envío de los mensajes WAIT .....	52
Tabla 4. Tiempo de envío de los mensajes WSN_NOTIFICATION.....	53
Tabla 5. Tiempo de procesamiento del mensaje de activación inalámbrico .....	54
Tabla 6. Tiempo de procesamiento del mensaje de datos inalámbrico .....	54
Tabla 7. Tiempo de envío del mensaje de datos inalámbrico.....	55
Tabla 8. Tiempo de procesamiento $T_{módem\_wait}$ .....	56
Tabla 9. Tiempo de procesamiento del mensaje serie WAIT.....	57
Tabla 10. Tiempo de procesamiento del mensaje serie WSN_NOTIFICATION .....	58
Tabla 11. Resultados de la sincronización manual de relojes .....	60
Tabla 12. Resultados de medida de distancias del experimento 1 .....	63
Tabla 13. Resultados de medida de distancias del experimento 2 .....	64
Tabla 14. Resultados de medida de distancias del experimento 3 .....	65
Tabla 15. Resultados de medida de distancias del experimento 4 .....	66
Tabla 16. Tiempo de vuelo del mensaje de activación inalámbrico .....	67
Tabla 17. Tiempo de vuelo del mensaje de datos inalámbrico.....	68
Tabla 18. Tiempo de vuelo promedio de ambos mensajes .....	68
Tabla 19. Resultados de medida de distancias con osciloscopio .....	69

# 1. Introducción

Durante las últimas décadas, las redes inalámbricas de sensores subacuáticos han tenido un gran auge debido a su gran variedad de aplicaciones en el medio subacuático, desde monitorización de instalaciones de acuicultura marina hasta minería y extracción subacuáticas pasando por seguimiento de especies marinas, predicción de seísmos, arqueología marina, etc. Estas redes se componen de pequeños sensores inalámbricos de bajo coste y bajo consumo que se comunican con el fin de llevar a cabo una tarea concreta. Esta característica permite el desarrollo de sistemas de monitorización flexibles que cubren grandes áreas y no tienen necesidad de mantenimiento. Por otra parte, el desarrollo de vehículos autónomos subacuáticos ha demostrado ser de gran interés ya que proporcionan una manera eficiente de observar las profundidades del océano y recolectar datos del entorno. En la actualidad, los sistemas de navegación y control de estos vehículos están siendo investigados por un gran número de investigadores en todo el mundo.

La información de localización es de gran interés en las redes inalámbricas de sensores subacuáticos puesto que permiten aportar un valor añadido a la monitorización. Mediante esta información, se puede detectar y localizar exactamente las propiedades de un fenómeno en un área de interés. Por ejemplo, se pueden detectar las fugas en una instalación de extracción petrolífera y utilizar la información de localización para repararla rápidamente. Por otro lado, la información de localización permite que la red inalámbrica sea capaz de asistir a la navegación de un vehículo subacuático. Esto abre la puerta al desarrollo de sistemas de monitorización completamente autónomos en los que la red de sensores monitoriza un fenómeno en particular y los vehículos subacuáticos recopilan datos del entorno y de la propia red de sensores.

En 2011, un equipo de investigación del instituto ITACA de la Universitat Politècnica de Valencia junto con el grupo VICOROB de la Universitat de Girona iniciaron un proyecto coordinado subvencionado por el Ministerio de Ciencia e Innovación, el proyecto Automatización de la iNspección y el renDimiento Remotos de Explotaciones de Acuicultura marina (ANDREA). El objetivo del proyecto es desarrollar un sistema completamente automatizado con control remoto de los parámetros biológicos e inspecciones de las instalaciones de acuicultura marina. Para lograr alcanzar dicho objetivo, el proyecto se divide en dos subproyectos, el subproyecto SABINA (Sensorización Ambiental suBacuática para la Inspección y moNitorización de explotaciones de Acuicultura marina) y el subproyecto RAIMON (Robot Autónomo submarino para la Inspección y MONitorización de explotaciones de acuicultura marina).

El objetivo del subproyecto SABINA es desarrollar una red inalámbrica de sensores subacuáticos para la monitorización de una instalación de acuicultura marina. Mediante este subproyecto, se ha conseguido mejorar y refinar un sistema de comunicaciones subacuática de bajo consumo para la creación de redes inalámbricas subacuáticas. Por su parte, el subproyecto RAIMON tiene

como objetivo mejorar el robot autónomo submarino (AUV) del grupo VICOROB para inspeccionar las redes submarinas mediante procesamiento de imagen. Uno de los objetivos a cumplir en este subproyecto es mejorar la navegación del AUV mediante una red inalámbrica de transpondedores subacuáticos construida con el sistema de comunicaciones conseguido en el subproyecto SABINA.

Para aportar una solución a dicho objetivo, se ha planteado la realización de una tesis doctoral en localización y posicionamiento subacuático.

## **2. Objetivo**

Esta tesina está enfocada como una introducción a dicha tesis. Por lo tanto, el objetivo es realizar una primera aproximación a las técnicas y los algoritmos utilizados en el medio subacuático para localización y posicionamiento, al mismo tiempo que se desarrolla un sistema básico para dar soporte al posicionamiento de equipos subacuáticos.

Para alcanzar este objetivo, en primer lugar se ha realizado una búsqueda bibliográfica acotada de artículos relacionados con técnicas y algoritmos de localización y posicionamiento subacuático. Usando los resultados obtenidos, se ha realizado un primer estado del arte para la tesis.

En segundo lugar, se ha desarrollado un sistema básico para dar soporte al posicionamiento de equipos subacuáticos utilizando el módem de bajo consumo diseñado por el grupo de investigación, el módem ITACA-S1000 de Antonio Sánchez *et al.* [1]. Por una parte, el objetivo de desarrollar dicho sistema es aportar una primera solución básica al objetivo del proyecto coordinado de mejorar el algoritmo de navegación utilizado por el AUV de VICOROB. Por otra parte, con este desarrollo se pretende identificar las ventajas y los inconvenientes de utilizar el actual módem ITACA-S1000 en sistemas de posicionamiento y, al mismo tiempo, identificar los problemas propios del desarrollo de dichos sistemas. De esta forma se pretende conseguir un punto de apoyo para el comienzo del desarrollo de la tesis.

## **3. Estado del arte**

Como una primera introducción a las técnicas de localización y posicionamiento en el medio subacuático se ha realiza una búsqueda bibliográfica en las bibliotecas digitales *IEEE Xplore Digital Library* ([ieeexplore.ieee.org](http://ieeexplore.ieee.org)) y *ACM Digital Library* ([dl.acm.org](http://dl.acm.org)). Durante la búsqueda se han intentado seguir las referencias de aquellos artículos que resultaban de interés para intentar encauzar y acotar la búsqueda. Debido a que el tiempo disponible para realizar la búsqueda y posteriormente analizar los artículos era limitado, el estado del arte obtenido es incompleto y necesita ser ampliado a corto plazo volviendo a realizar la búsqueda en ambas bibliotecas digitales y ampliándola a nuevas fuentes.

Las técnicas de localización y posicionamiento analizadas pueden dividirse en dos grandes categorías, las técnicas de localización tradicionales y las técnicas de localización para redes inalámbricas de sensores (UWSN). Las técnicas de localización tradicionales se han utilizado en oceanografía para recuperar equipamiento subacuático después de realizar experimentos y para llevar a cabo el seguimiento de especies marinas. Las técnicas de localización para UWSN han tenido un gran auge en los últimos años debido a la necesidad de incorporar información de localización en el funcionamiento de las UWSN para dar soporte a diferentes aplicaciones como vigilancia y rescate, monitorización y minería subacuática.

### **3.1. Técnicas de localización tradicionales**

Las técnicas de localización tradicionales se han utilizado en oceanografía durante mucho tiempo. Tradicionalmente, el objetivo de estas técnicas ha sido realizar un seguimiento de los equipos subacuáticos durante una campaña de experimentos y localizarlos para su posterior recuperación. Además, estas técnicas también se han utilizado para realizar el seguimiento de especies marinas y observar su comportamiento [2]. En los últimos años, estas técnicas han vuelto a tener un gran auge debido a su facilidad de uso y su gran utilidad para los vehículos subacuáticos. Actualmente, los sistemas de navegación de los AUV, ROV y otros equipamientos subacuáticos propulsados utilizan una combinación de alguna de estas técnicas junto con sus instrumentos de navegación inercial para reducir el error durante la navegación [3] [4].

- **USBL / SSBL**

Ultra Short Baseline (USBL), también conocido como Super Short Baseline (SSBL), es una técnica de localización en la cual un barco realiza el seguimiento de un equipo subacuático mediante un emisor de corta distancia. En esta técnica, el barco dispone de un emisor y un receptor de señales subacuáticas. El receptor está compuesto por una malla de transductores separados por una distancia equidistante de unos pocos centímetros. De esta forma se permite que el barco pueda medir el ángulo de llegada de la señal subacuática. Por su parte, el equipo subacuático dispone de un transpondedor que responde a la señal generada por el barco. Así, cuando el barco necesita conocer la localización del equipo subacuático lanza una señal acústica de interrogación que es recibida por el transpondedor. A continuación, el transpondedor responde con su señal subacuática la cual es recibida por el receptor del barco. La localización del equipo subacuático se calcula entonces mediante el ángulo de la señal recibida y la distancia relativa obtenida con el tiempo de vuelo de la señal.

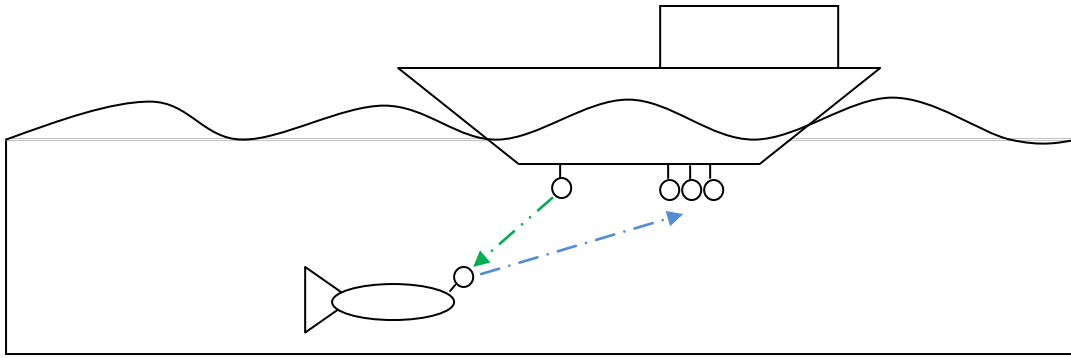


Figura 1. Localización en USBL

En esta técnica, el tiempo de vuelo se calcula mediante telemetría bidireccional, por lo tanto, no es necesaria una sincronización fuerte de los dispositivos de comunicación. Basta con que los relojes tengan una frecuencia parecida y el tiempo de procesamiento del transpondedor sea cercano a una constante [5].

Por otra parte, el error de localización obtenido por los sistemas basados en esta técnica es bastante elevado. Por este motivo, en [6] se presenta una mejora para reducir el error de localización. En esta mejora se permite el envío de datos en los pulsos de la señal subacuática de forma que el barco y el equipo subacuático, en este caso un AUV, pueden compartir información. Así, el AUV envía información relativa a su navegación (información inercial, profundidad, velocidad del sonido, etc) para permitir que el barco pueda realizar una localización más precisa. Por su parte, el barco envía la última localización calculada del AUV para que éste pueda corregir el error producido por su sistema de navegación inercial.

Los principales inconvenientes de esta técnica de localización son su corto alcance y el coste debido a la necesidad de usar un barco.

- **iUSBL / iSSBL**

Inverse Ultra Short Baseline (iUSBL), también conocido como inverse Super Short Baseline (iSSBL), es una técnica de posicionamiento derivada de USBL en la cual la posición del emisor, el receptor y el transpondedor se invierten. Aquí, el emisor y el receptor se localizan en el equipo subacuático y el transpondedor en el barco. De esta forma, se permite que el equipo subacuático sea capaz de obtener su posición relativa al barco. En [7] se utiliza esta técnica para permitir que un AUV pueda obtener su localización y se propone una mejora que permite el posicionamiento de múltiples AUVs. Esta mejora se basa en permitir que el barco envíe una señal constantemente con su posición GPS y su velocidad. El AUV recibe esta señal y calcula su posición mediante la información recibida del barco y el ángulo de llegada de la señal. Así, el AUV no tiene que enviar señales y, por lo tanto, no es necesario ningún protocolo de arbitraje de las comunicaciones en el posicionamiento de múltiples AUV.



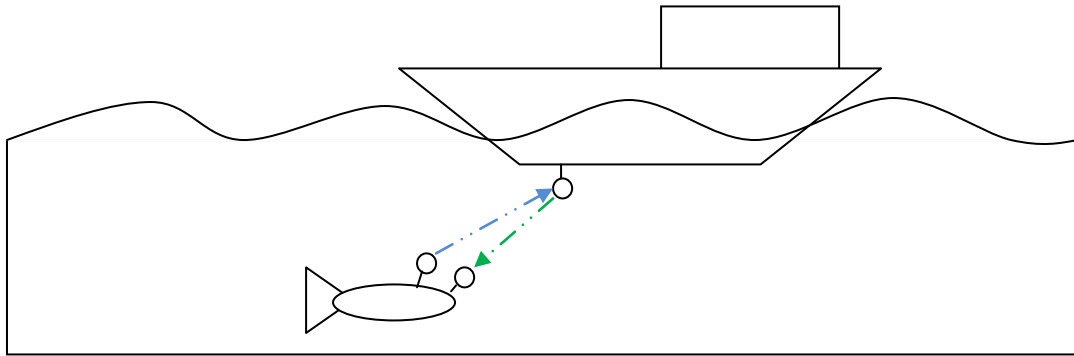


Figura 2. Localización en iUSBL

- **SBL**

Short Baseline (SBL) es una técnica de localización en la cual un barco realiza el seguimiento de un equipo subacuático. En esta técnica, el barco dispone de un emisor y un conjunto de tres receptores. Por su parte, el equipo subacuático está equipado con un transpondedor que responde a las señales provenientes del barco. Cuando el barco necesita conocer la localización del equipo subacuático envía una señal acústica de interrogación que es recibida por el transpondedor. A continuación, el transpondedor envía una señal que es recibida por los receptores del barco. La localización del equipo subacuático se calcula entonces mediante triangulación utilizando las distancias calculadas por cada uno de los receptores. Por lo tanto, al igual que en USBL, las distancias se calculan utilizando el tiempo de vuelo obtenido mediante telemetría bidireccional. Como consecuencia no es necesaria una sincronización de los dispositivos de comunicación.

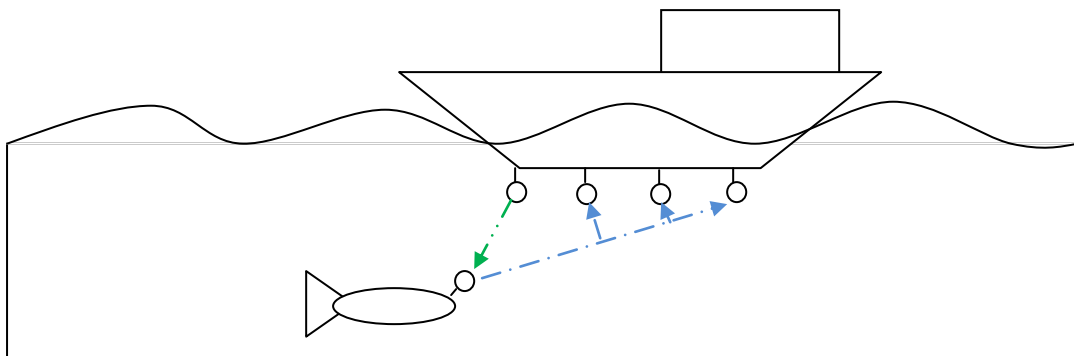


Figura 3. Localización en SBL

En [8] se realiza un estudio de los principales factores que causan errores en la localización calculada por los sistemas basados en esta técnica, llegando a la conclusión de que la incertidumbre en la velocidad del sonido es la principal causa.

Aunque tradicionalmente se ha utilizado un barco como infraestructura para esta técnica se pueden utilizar otras soluciones gracias a que el alcance conseguido es mayor que en USBL. Por ejemplo, en [9] se utiliza un dique para construir un sistema de posicionamiento para AUVs basado en esta técnica.

Aún así, el coste de despliegado de la infraestructura es el principal inconveniente.

- **LBL**

Long Baseline (LBL) es una técnica de posicionamiento que permite a los equipos subacuáticos conocer su localización dentro de un área de interés. Esta técnica consiste en un conjunto de transpondedores desplegados en los vértices del área de interés y amarrados al fondo oceánico en una posición fija conocida. El equipo subacuático está equipado con un emisor y un receptor de largo alcance de manera que puede enviar y recibir señales subacuáticas de los transpondedores. Cuando el equipo subacuático necesita conocer su localización envía una señal subacuática de largo alcance. Los transpondedores que reciben la señal envían entonces una señal de vuelta que es recibida por el equipo subacuático. Así, el equipo subacuático calcula su localización mediante triangulación con las distancias estimadas a, al menos, 3 de los transpondedores. Por lo tanto, al igual que en la técnica anterior, las distancias se calculan utilizando el tiempo de vuelo obtenido mediante telemetría bidireccional. Como consecuencia no es necesaria una sincronización de los dispositivos de comunicación.

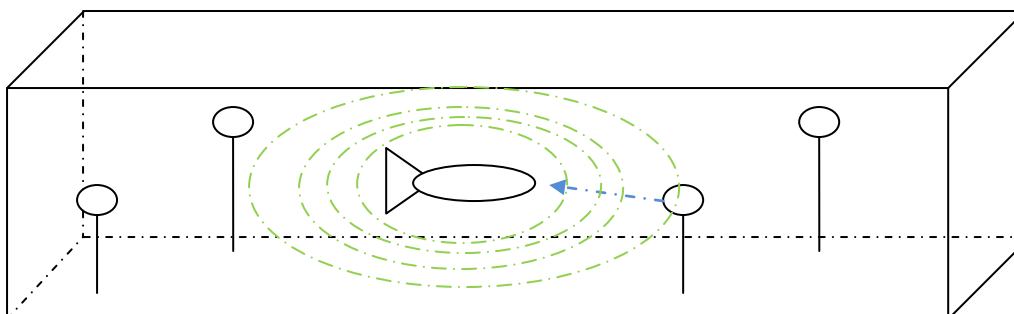


Figura 4. Localización en LBL

Esta técnica necesita de un conocimiento preciso del perfil de la velocidad del sonido del área de interés y realizar estimaciones del movimiento de los amarres para reducir el error en la localización. Sin embargo, una vez que se superan estos problemas se convierte en una técnica de localización muy fiable y robusta que puede utilizarse en situaciones críticas, como en operaciones bajo el hielo [10], y que es capaz de ampliar el tamaño del área de interés hasta kilómetros [11].

En [12] se propone una mejora para intentar adaptar esta técnica a la localización de múltiples AUV. En este caso, las distancias se calculan utilizando el tiempo de vuelo obtenido mediante telemetría unidireccional. Gracias a la telemetría unidireccional, el barco difunde una señal con su posición GPS y los AUV pueden calcular su localización de forma pasiva al recibir dicha señal. Sin embargo, es necesario que los relojes de los sistemas de comunicación estén fuertemente sincronizados para conseguir un tiempo global y, además, tienen que ser muy precisos para evitar posibles derivas durante la operación del sistema.

Los principales inconvenientes de esta técnica son el alto coste de despliegado de los transpondedores, ya que necesitan un amarre en el fondo del mar, y el consumo de energía generado por el envío de señales de largo alcance.

- **GIB**

GPS Intelligent Buoys [13] es una técnica de localización que permite a una estación base realizar el seguimiento de equipos subacuáticos. Esta técnica consiste en un conjunto de boyas de superficie capaces de obtener su localización mediante GPS y escuchar las señales provenientes de los equipos subacuáticos. En esta técnica, los equipos subacuáticos envían una señal cada cierto periodo de tiempo la cual es recibida por las boyas de superficie. Entonces las boyas obtienen el tiempo de llegada de la señal y la envían a una estación base junto con su localización GPS. De esta forma, la estación base puede calcular la distancia del equipo subacuático a cada una de las boyas y realizar el seguimiento. Por lo tanto, en esta técnica la obtención del tiempo de vuelo se realiza mediante telemetría unidireccional, siendo necesaria una fuerte sincronización de los relojes de los dispositivos de comunicación.

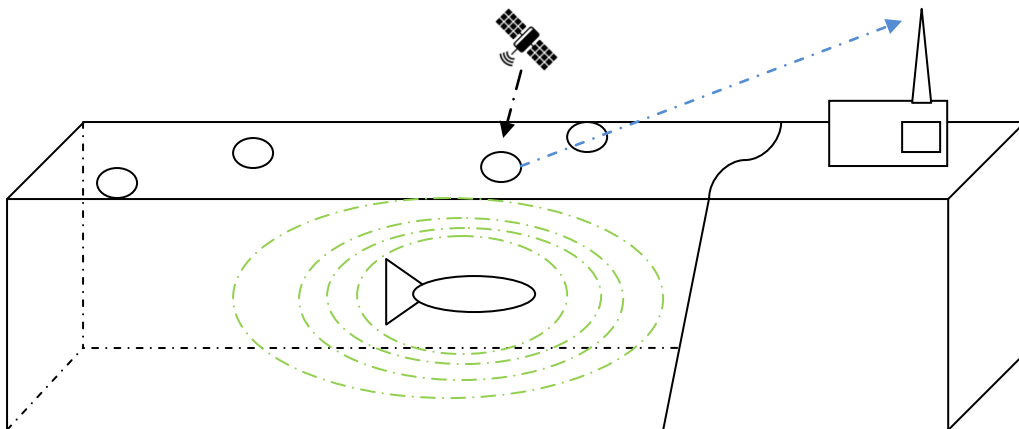


Figura 5. Localización en GIB

El principal inconveniente de esta técnica es la necesidad de coordinar los equipos subacuáticos para evitar la colisión de sus señales. Además, es importante encontrar un equilibrio entre el consumo de energía y el número de boyas desplegadas. Si se despliegan pocas boyas es necesario enviar las señales mucho más lejos y, por lo tanto, el consumo de energía aumenta. Sin embargo, si las comunicaciones son a corta distancia es necesario aumentar el número de boyas para dotar de cobertura la misma área de interés.

### **3.2. Técnicas de localización para redes inalámbricas de sensores**

Las redes inalámbricas de sensores han tenido un gran auge en los últimos años debido a su escalabilidad, su bajo coste de mantenimiento y su flexibilidad. Estas características permiten dar soporte a un gran número de aplicaciones subacuáticas tales como monitorización, vigilancia y rescate,

minería y prospecciones subacuáticas, etc. En estas aplicaciones, la información de localización de los nodos sensor aporta una valiosa información, desde el área exacta donde está ocurriendo un fenómeno monitorizado hasta la localización exacta de posibles fugas en una plataforma de extracción en alta mar. Además, permitir que los nodos sensor conozcan su localización, ya sea relativa o absoluta, permite dar soporte a protocolos de encaminamiento geográfico [14] y dar soporte a la navegación de vehículos subacuáticos.

Las técnicas de localización para UWSN analizadas pueden clasificarse dependiendo de dónde se realice el cálculo de la localización de los nodos. Así, estas técnicas se dividen en técnicas de localización centralizadas y técnicas de localización distribuidas.

### **3.2.1. Técnicas de localización centralizadas**

En las técnicas de localización centralizadas, los nodos sensor envían la información de localización a un sumidero para que pueda ser procesada y, así, permitir el cálculo de la localización de cada nodo de la red. En este caso, los nodos sensor no conocen su localización a menos que el sumidero la envíe explícitamente.

Las técnicas de localización centralizadas pueden clasificarse dependiendo de la forma en la que obtienen la información de localización. Así, estas técnicas se dividen en técnicas que hacen uso de la telemetría para calcular la distancia y técnicas que no la utilizan.

- **Cálculo de la distancia mediante telemetría**

- **MASL**

Motion-Aware Self Localization (MASL) [15] es una técnica de localización para redes de sensores subacuáticos móviles. En esta técnica, los nodos sensor obtienen información de localización de sí mismo y de sus vecinos mediante un intercambio constante de mensajes. Cada nodo calcula las distancias a sus vecinos en la recepción de los mensajes que estos envían. Por lo tanto, el cálculo se realiza utilizando una estimación de la velocidad del sonido y obteniendo los tiempos de vuelo de los mensajes mediante telemetría unidireccional. Como consecuencia, los nodos sensor tienen que estar sincronizados y es necesario un algoritmo de sincronización para el correcto funcionamiento de esta técnica en misiones de larga duración.

La localización de cada nodo sensor se calcula después en una fase de post-procesamiento en una estación central. Las distancias obtenidas de los nodos sensor se utilizan como entrada de un algoritmo de estimación iterativo. En cada iteración, el algoritmo obtiene la localización de los nodos sensor en una malla mediante la probabilidad de que se encuentren en un área en concreto. Esta localización se refina después en la próxima iteración teniendo en cuenta las corrientes oceánicas durante el proceso. De esta forma, se va refinando la posición para cada nodo sensor.

La principal ventaja de esta técnica es la reducción de la sobrecarga computacional en los nodos sensor, permitiendo que su hardware sea más barato. Además, no es necesaria ningún tipo de infraestructura a parte de los nodos sensor desplegados. Sin embargo, esta técnica tiene una sobrecarga de comunicación y un consumo de energía elevados debido al constante envío de la estimación de distancias por parte de los nodos sensor al sumidero. Además, no es una técnica adecuada para aplicaciones que necesitan información de localización en tiempo real, como la monitorización in situ, debido a su fase de post-procesamiento.

- **EERS**

Energy-Efficient Ranging Scheme (EERS) [16] es una técnica de localización enfocada a localizar un conjunto de nodos sensor flotantes. En esta técnica, los nodos sensor flotan a la deriva con las corrientes oceánicas mientras estiman la distancia a un subconjunto de nodos vecinos periódicamente. La estimación de distancias se realiza en dos pasos. En primer lugar, todos los nodos se sincronizan mediante envíos de mensajes haciendo uso de un algoritmo de sincronización. Después, se selecciona un subconjunto de los nodos para que difundan mensajes de localización. Así, cada nodo puede estimar su distancia a los nodos del conjunto seleccionado cuando recibe un mensaje. La estimación de la distancia se realiza haciendo uso del tiempo de vuelo del mensaje y una estimación de la velocidad del sonido obtenida mediante un sensor. Por lo tanto, este algoritmo hace uso de la telemetría unidireccional. Finalmente, la localización de cada nodo sensor se calcula en una estación base después de recuperar los nodos sensor del mar.

La principal ventaja de esta técnica es la reducción de la sobrecarga computacional en los nodos sensor, permitiendo que su hardware sea más barato. Además, no es necesario ningún tipo de infraestructura a parte de los nodos sensor desplegados. Sin embargo, esta técnica tiene una sobrecarga de comunicación y un consumo de energía elevados debido a la difusión realizada por los nodos sensor en ambas fases del algoritmo. Por último, esta técnica no es adecuada para aplicaciones que necesitan información de localización en tiempo real ya que la obtención de la localización se realiza posteriormente a la recogida de los nodos sensor.

- **Cálculo de la distancia mediante otros métodos**

- **ALS**

Area-based Localization Scheme (ALS) [17] es una técnica de localización de redes inalámbricas de sensores terrestres [18] adaptada al medio subacuático. En esta técnica, un conjunto de nodos especiales (llamados anclas) ubicados en posiciones conocidas dividen la región de interés en pequeñas áreas enviando mensajes mientras varían su potencia de transmisión. En estos mensajes se incluye la potencia de transmisión utilizada para evitar que los nodos sensor tengan que medir la potencia y puedan añadir error en la localización. Por su parte, los nodos sensor escuchan pasivamente estos mensajes y generan una lista con las anclas al alcance y sus

respectivos niveles de potencia. Esta información es entonces enviada a un sumidero que conoce la posición exacta de las anclas. El sumidero calcula la localización de cada nodo sensor estimando el área donde se encuentra desplegado. Por lo tanto, esta técnica no proporciona una localización exacta de los nodos sensor. Además, no hace uso de métodos de telemetría evitando así la necesidad de sincronización.

La principal ventaja de esta técnica es la reducción de la sobrecarga computacional en los nodos sensor, permitiendo que su hardware sea más barato. Además, no necesita ningún tipo de sincronización. Sin embargo, esta técnica tiene una sobrecarga de comunicación y un consumo de energía elevados debido al constante envío de la lista de anclas por parte de los nodos sensor al sumidero. Por último, esta técnica no es adecuada para aplicaciones que necesitan precisión en la localización ni información de localización en tiempo real.

- **3D-MALS**

3D multi-power area localization scheme (3D-MALS) [19] es una extensión de ALS que permite obtener la localización de los nodos sensor en una red 3D móvil subacuática. En esta técnica, las anclas se convierten en boyas flotantes capaces de sumergirse hasta una cierta profundidad. Mientras se sumergen, estas boyas envían mensajes a profundidades fijas variando la potencia de transmisión. Estos mensajes incluyen la potencia de transmisión utilizada y la posición GPS de la boya. Por su parte, los nodos sensor escuchan pasivamente estos mensajes y generan una lista con la posición GPS de las boyas al alcance y sus respectivos niveles de potencia. Esta información es entonces enviada a un sumidero para que pueda calcular la localización de cada nodo sensor estimando el área donde se encuentra desplegado. Por lo tanto, al igual que ALS, esta técnica no proporciona una localización exacta de los nodos sensor. Además, no hace uso de métodos de telemetría evitando así la necesidad de sincronización.

La precisión de la localización obtenida con esta técnica es menor que la de ALS. Además, el uso de boyas flotantes capaces de sumergirse aumenta el coste total del sistema y genera una dependencia del tiempo total de localización con la velocidad de inmersión de las boyas flotantes.

### **3.2.2. Técnicas de localización distribuidas**

En las técnicas de localización distribuida, los nodos sensor comparten información de localización con sus vecinos para permitir que cada nodo de la red estime su propia localización.

Las técnicas de localización distribuidas pueden clasificarse dependiendo de la forma en la que obtienen la información de localización. Así, estas técnicas se dividen en técnicas que hacen uso de la telemetría para calcular la distancia y técnicas que no la utilizan.

- **Cálculo de la distancia mediante telemetría**

- **AAL**

AUV-Aided Localization (AAL) [20] es una técnica de localización donde un vehículo autónomo subacuático (AUV) da soporte en la localización de los nodos sensor de una red inalámbrica subacuática. El AUV es capaz de conocer su propia localización mediante sus instrumentos inerciales de abordaje después de obtener su localización mediante GPS en la superficie. Durante la navegación, el AUV difunde señales subacuáticas periódicamente que reciben los nodos sensor. Cuando un sensor recibe la señal, envía una petición al AUV y éste envía una respuesta con su localización. De esta manera, el nodo sensor estima su distancia al AUV utilizando el tiempo de vuelo obtenido mediante telemetría bidireccional y una estimación de la velocidad del sonido. Por lo tanto, no es necesaria una sincronización fuerte de los dispositivos de comunicación. Finalmente, el nodo sensor calcula su propia localización mediante trilateración [21] en el momento en el que obtiene las distancias al AUV en tres localizaciones co-planares diferentes.

Para evitar la acumulación del error de localización generado por los instrumentos inerciales durante la navegación, el AUV sube periódicamente a la superficie para calibrar su posición mediante GPS. Por lo tanto, el error de localización de esta técnica depende de la tasa de calibración de la posición del AUV. Además, la energía del AUV es limitada así que es necesario gestionar bien el proceso de calibración para encontrar un equilibrio entre el error de localización y el consumo de energía.

- **UPS**

Underwater positioning Scheme (UPS) [22] está basado en una técnica de localización presentada para redes inalámbricas terrestres en [23]. En esta técnica, cuatro nodos ancla en una posición fija y conocida difunden señales subacuáticas secuencial y periódicamente. En un primer momento, una de las anclas es seleccionada como la ancla maestra y empieza el proceso de envío de señales subacuáticas enviando su señal. Cuando otra ancla escucha la señal responde con su propia señal. Ubicando los nodos ancla en una geometría concreta se consigue que las anclas respondan secuencialmente una detrás de otra. Una vez que los nodos sensor han escuchado las señales de las cuatro anclas, calculan la diferencia de tiempo de llegada (TDoA) entre las anclas usando su propio reloj local. A continuación, calculan las diferencias de distancia entre anclas utilizando una estimación de la velocidad del sonido. Finalmente, calculan su propia localización mediante trilateración haciendo uso de las posiciones de las anclas y las diferencias de distancia calculadas.

La principal ventaja de esta técnica es el bajo consumo y la baja sobrecarga de comunicación debido a que los nodos sensor no envían mensajes. Además, esta técnica no necesita sincronización aunque se utiliza telemetría unidireccional gracias al uso de TDoA. Sin embargo, sólo los nodos encerrados dentro del área de las cuatro balizas pueden obtener su

localización. Por lo tanto, hay que tener presente la geometría que forman las balizas. Por otro lado, no todos los nodos en el área son capaces de obtener una localización única. Los nodos cercanos a los nodos ancla necesitan de cinco anclas para obtener su localización. Por este motivo, los autores proponen una mejora de esta técnica llamada Wide Coverage Positioning (WPS) [24] en la cual se utilizan cuatro nodos ancla para obtener una localización única excepto en el caso en el que no sea posible, en el cual se utiliza un nodo ancla adicional.

- **LSHL**

Large-Scale Hierarchical Localization (LSHL) [25] es una técnica de localización enfocada a localizar nodos sensor con comunicaciones de corto alcance en redes inalámbricas subacuáticas de gran escala. En esta técnica, la fase inicial consiste en un conjunto de nodos ancla que difunden mensajes periódicamente para que los nodos sensor al alcance puedan obtener su localización mediante UPS. A continuación, se realiza una fase iterativa donde se selecciona un conjunto de nodos sensor localizados para que se conviertan en nodos de referencia. Estos nodos de referencia actúan como anclas UPS para otros nodos sensor vecinos. De esta forma, la mayor parte de los nodos sensor de la red pueden obtener su localización mediante UPS. Finalmente, se realiza una fase complementaria en la que los nodos que todavía no han conseguido su localización envían una petición de localización. Esta petición permite que se seleccione un nuevo conjunto de nodos de referencia para que el nodo pueda obtener su localización mediante UPS.

La principal ventaja de esta técnica es la ampliación de la cobertura conseguida inicialmente por UPS. El inconveniente de conseguir esta ampliación es el aumento del consumo de energía y la sobrecarga de las comunicaciones debido al envío de mensajes por parte de los nodos sensor en las fases iterativa y complementaria.

- **Cálculo de la distancia mediante otros métodos**

- **UDB**

Using Directional Beacons (UDB) [26] es una técnica de localización similar a AAL. En esta técnica también se utiliza un AUV para dar soporte en la localización de los nodos sensor de una red inalámbrica subacuática. En este caso, el AUV navega por encima de la red difundiendo mensajes acústicos con un emisor direccional. En estos mensajes se incluye la localización del AUV y el ángulo de su emisor. Al igual que en AAL, el AUV obtiene su localización mediante GPS y sus instrumentos inerciales de abordaje realizando emersiones para calibrar periódicamente su localización. Por su parte, los nodos sensor escuchan estos mensajes desde que el AUV entra en su rango de comunicación hasta que sale. En ese momento calculan su localización utilizando la posición del AUV al entrar y salir de su rango de comunicación y el ángulo recibido en los mensajes. De esta forma, la localización se calcula sin necesidad de obtener el tiempo de vuelo.



Esta técnica es más eficiente en energía que AAL ya que los nodos sensor no emiten mensajes y no es necesario ningún tipo de sincronización. Sin embargo, a los inconvenientes de AAL hay que sumarle que en esta técnica el AUV siempre tiene que navegar por encima de la red, lo cual podría no ser posible. Además, la precisión de la localización depende de los intervalos en los que el AUV difunde sus mensajes.

- **RSMB**

Range-free Schemes with Mobile Beacons (RSMB) [27] es una técnica de localización donde un nodo ancla móvil se mueve por la superficie del mar a una velocidad constante siguiendo diferentes trayectorias a destinos aleatorios. Mientras se mueve, el nodo ancla difunde mensajes con su localización GPS a intervalos regulares. Por su parte, los nodos sensor disponen de un sensor de presión que les permite obtener su profundidad. Cuando un nodo sensor recibe suficientes mensajes del nodo ancla después de entrar y salir de su rango de comunicación varias veces, selecciona tres mensajes tras identificarlos como los mensajes recibidos a la entrada y a la salida del rango de comunicación. A continuación, el nodo sensor estima su localización mediante la posición del ancla recibida en esos mensajes y su profundidad.

En esta técnica no se necesita sincronización y, además, el consumo de energía de los nodos sensor es bajo debido a que no realizan envíos de mensajes. Sin embargo, el nodo ancla consume mucha energía debido a la navegación.

Por último, la precisión de la localización conseguida con esta técnica depende del intervalo de envío y de las trayectorias seguidas por la ancla móvil.

## **4. Desarrollo**

Como parte del proyecto ANDREA se pretende mejorar el sistema de navegación del vehículo autónomo submarino (AUV) del grupo VICOROB mediante una red de transpondedores. En una primera aproximación para dar solución a este objetivo, se propone simplificar el problema sustituyendo la red de transpondedores por un conjunto de balizas en posiciones fijas y conocidas. Las balizas formarán una malla a lo largo de una región de interés por donde navegará el vehículo. Cuando el vehículo necesite obtener su localización, transmitirá una señal acústica en todas direcciones. Las balizas que escuchen esta señal transmitirán un mensaje con su profundidad. Calculando el tiempo de vuelo y utilizando la profundidad recibida, el vehículo será capaz de estimar su distancia a cada una de las balizas y, por lo tanto, podrá obtener su posición mediante trilateración ya que las balizas tienen una posición fija y conocida.

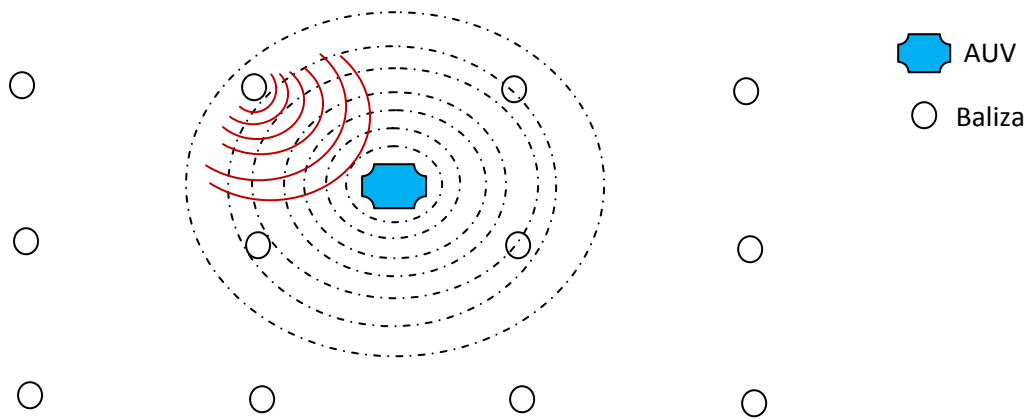


Figura 6. Solución propuesta para mejorar el sistema de navegación de un AUV

Del estado del arte se puede concluir que la obtención del tiempo de vuelo y la estimación de la velocidad del sonido son factores críticos a la hora de estimar las distancias y, por tanto, de realizar una estimación precisa de la localización de un objeto. La velocidad del sonido puede obtenerse mediante sensores. Con sensores de presión, salinidad y temperatura se puede calcular teóricamente la velocidad del sonido de una masa de agua mediante las diferentes fórmulas que pueden encontrar en la literatura tales como [28] y [29]. Además, se puede obtener la velocidad del sonido directamente con un sensor DVL [30]. Por lo tanto, la estimación de la velocidad del sonido puede realizarla el AUV con sus sensores de a bordo. Sin embargo, la obtención del tiempo de vuelo tiene que realizarse mediante un algoritmo y es importante realizarla de la forma más precisa posible. Como consecuencia, aunque el AUV ejecuta sus tareas mediante un sistema operativo de tiempo real es interesante que el tiempo de vuelo lo obtenga un sistema independiente dedicado exclusivamente a esa función. Por este motivo se pretende desarrollar un sistema para dar soporte al posicionamiento mediante la obtención del tiempo de vuelo. Este sistema no sólo puede aplicarse al AUV sino que puede aplicarse a cualquier equipo subacuático que necesite conocer su localización y tenga capacidad de procesamiento para realizar la trilateración.

El sistema en cuestión consta de dos componentes, el sonar y la baliza. La baliza es un dispositivo autónomo que escucha el medio subacuático a la espera de una señal acústica. Cuando la baliza recibe la señal acústica, envía un mensaje de datos con su profundidad. Por su parte, el sonar es un dispositivo que envía señales acústicas cuando se lo ordena un dispositivo maestro y espera recibir respuestas de las balizas. Cada vez que el sonar recibe una respuesta, calcula el tiempo de vuelo a partir del tiempo transcurrido desde el envío de la señal acústica hasta la recepción del mensaje de datos y comunica el resultado al dispositivo maestro.

Dado que ambos componentes tienen que enviar y recibir mensajes en el medio subacuático, necesitan de un sistema de comunicación subacuático para su funcionamiento. En este caso, el sistema de comunicación que se utilizará es el módem ITACA-S1000.

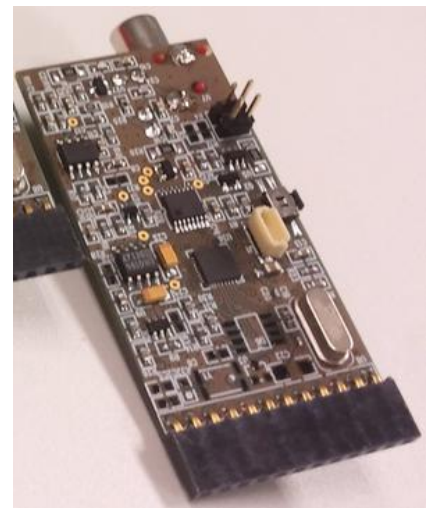
El resto de este apartado se organiza como sigue. En el punto 4.1 y el punto 4.2 se explica el sistema de comunicación basado en el módem ITACA-S1000 que permitirá al sonar y a las balizas comunicarse inalámbricamente. En el punto 4.3 se explica la comunicación entre los componentes del sistema de comunicación para introducir los mensajes y el protocolo de comunicación utilizados en la implementación de la aplicación ejecutada por el sonar y las balizas. En el punto 4.4 se explica la comunicación entre el sonar y el AUV que la aplicación del sonar tendrá que implementar. En el punto 4.5 se explica el protocolo de acceso al medio seleccionado para gestionar el funcionamiento de las balizas. Por último, en el punto 4.6 se explica detalladamente la implementación de la aplicación desarrollada tanto en el sonar como en las balizas para dar funcionalidad al sistema descrito.

#### 4.1. El módem ITACA-S1000

El módem ITACA-S1000 [1] es un módem flexible, de bajo consumo y bajo coste diseñado específicamente para dar soporte a protocolos de comunicación eficientes y a aplicaciones de larga duración usando redes inalámbricas de sensores subacuáticas.

Las especificaciones técnicas del módem son las siguientes:

- Tamaño: 3 cm x 6.5 cm
- Transductor: Hummimbird XP 9 20
- Ángulo del haz: 60 °
- Velocidad de transmisión: 1000 kbps (FSK)
- Frecuencia central: 85 kHz
- Alcance máximo: 100 m
- Interfaz: RS232
  - Consumo:
    - MAX. 300 mW en transmisión
    - 36 mW en recepción
    - 10  $\mu$ W con detección de activación
    - 3  $\mu$ W en bajo consumo
- Alimentación:
  - 3.3 V DC
  - 3-30 V DC (amplificador de potencia)



El diseño del módem gira en torno a un microcontrolador de bajo consumo, el C8051F920 de Silicon Labs [31]. El resto de componentes están diseñados específicamente para cumplir con restricciones de coste, consumo y complejidad. Así, el amplificador de potencia ha sido diseñado como una solución personalizada que incorpora elementos hardware y software. Esta solución llamada Push-Pull class-B-based Digitally controlled amplifier (D-PP-B) [32] maximiza la potencia que es transferida a la carga mientras minimiza el consumo de energía. Por otra parte, el amplificador de recepción es una solución completamente analógica que incorpora amplificadores operacionales y potenciómetros digitales que pueden ser variados por el microcontrolador para ajustar la ganancia en la recepción.

Además, el módem incorpora un sistema de activación asíncrono (AAW-U) [33] basado en la recepción de una señal subacuática que permite reducir el consumo de los protocolos y las aplicaciones subacuáticas. Mediante este sistema, un sistema puede pasar a modo de bajo consumo cuando está ocioso sin perder ningún mensaje. Simplemente el emisor tiene que enviar una señal acústica antes del envío de cada mensaje. De esta manera, el módem puede volver a activar el sistema receptor antes de recibir el mensaje.

El sistema de activación asíncrono está construido con un hardware comercial diseñado con restricciones de bajo consumo, el AS3933 de AMS [34]. El diseño de este sistema incorpora una red adaptación de impedancias capaz de utilizar el mismo transductor que el del sistema de comunicación. Por lo tanto, el módem sólo necesita un transductor para realizar todas sus funciones.

Teniendo en cuenta las restricciones de costes, el módem está diseñado para utilizar un transductor piezoeléctrico direccional mucho más barato que un hidrófono. El transductor utilizado es el Hummimbird XP 9 20 el cual dispone de dos posibles bandas de frecuencias, una de 200 kHz y otra de 85kHz. Con el fin de conseguir el máximo alcance y la mayor robustez posible ante la atenuación del agua, el diseño del módem utiliza la frecuencia de 85 kHz. Esta frecuencia ofrece un ángulo de haz de  $60^\circ$  con la siguiente geometría:

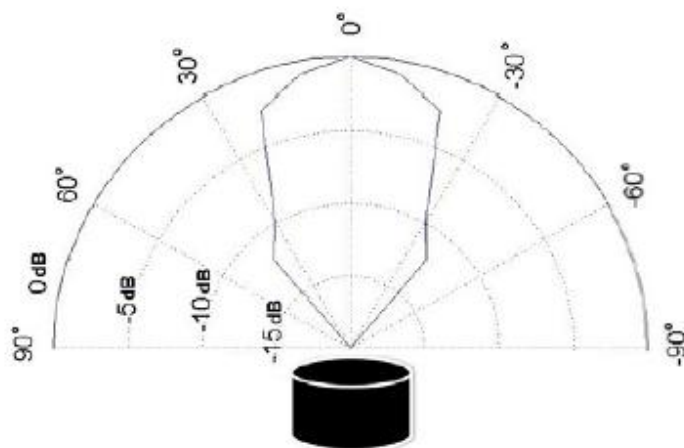


Figura 7. Geometría del haz del transductor acústico

En cuanto a la transmisión de datos, el módem utiliza una modulación binaria FSK coherente, siendo ésta la primera solución diseñada para utilizar este tipo de modulación en el ámbito subacuático [32]. El microcontrolador realiza íntegramente el algoritmo de modulación utilizando sus contadores internos. Sin embargo, el algoritmo de demodulación se ha diseñado como una solución híbrida entre software y hardware debido que utiliza un bucle de enganche de fase (PLL) con un filtro paso bajo. En esta solución, el microcontrolador implementa el oscilador controlado del PLL mientras que el filtro y el detector de fase son hardware. De esta forma se obtiene una solución eficiente con bajo consumo y bajo coste de computación. Con este diseño, el módem es capaz de proporcionar una velocidad de transmisión de 1kbps con un ancho de banda de 1kHz usando una frecuencia portadora de 85 kHz.

Debido al diseño del algoritmo de modulación y a la utilización de un microcontrolador de bajo consumo, el módem utiliza todo su tiempo de procesamiento durante la transmisión y la recepción de datos. Por lo tanto no es viable implementar una aplicación compleja sobre el hardware del módem. Como consecuencia es necesario utilizar otro dispositivo capaz de ejecutar la aplicación haciendo uso del módem para realizar las transmisiones de datos. Además, si fuera necesario, este dispositivo también tiene que hacerse cargo de la gestión de la sensorización y de la comunicación con otros dispositivos haciendo uso de diferentes protocolos. Por ello se ha decidido llamar a este dispositivo adicional, el puente.

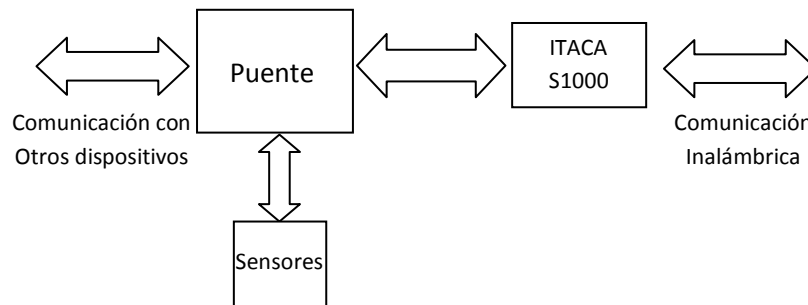


Figura 8. Sistema de comunicación basado en el módem ITACA-S1000

## 4.2. El puente de comunicaciones

Dado que el puente de comunicaciones se va a encargar de ejecutar todas las aplicaciones, gestionar la sensorización y, además, hacer de puente entre la comunicación inalámbrica del módem ITACA-S1000 y otros dispositivos, el hardware con el que debe contar tiene que ser lo suficientemente potente. En la actualidad los microcontroladores Cortex-M [35] de ARM están teniendo un gran auge en los sistemas empujados debido a su alto rendimiento, su bajo consumo, su baja relación precio/rendimiento y a su facilidad de uso gracias a sus bibliotecas estándar. Por lo tanto, como hardware del puente se ha decidido utilizar un Cortex-M4, el más potente de la familia Cortex-M. Con este microcontrolador se asegura que el hardware sirva para cualquier tipo de aplicación que se quiera implementar sobre el puente. De esta forma, todo el esfuerzo en aprendizaje sobre el funcionamiento del hardware servirá para futuros proyectos.

Ya que el módem es un dispositivo pensado para que tenga un bajo coste, el puente debería también tener un coste bajo. En la actualidad, la compañía ST dispone de placas de evaluación con Cortex-M4 a un precio de alrededor de 15\$ con las cuales se puede aprovechar toda la funcionalidad del microcontrolador. Por lo tanto, se ha decidido utilizar la placa de evaluación STM32F4-DISCOVERY [36] como hardware del puente.

Las especificaciones técnicas de la placa STM32F4-DISCOVERY y de su Cortex-M4 se describen a continuación.

- Microcontrolador STM32F407VGT6:
  - Núcleo de 32-bit ARM Cortex-M4F con DSP.
  - 1 MB Flash.
  - 192 KB RAM.
  - Controlador de memoria externa.
  - 17 contadores y un RTC.
  - Interfaz para LCD.
  - Interfaces de comunicación:
    - USART
    - I<sup>2</sup>C
    - SPI
    - CAN
    - SDIO
    - USB
  - DMA de propósito general.
  - Conversores A/D y D/A.



- Depurador en placa: ST-LINK/V2.
- Alimentación: USB o externa (5V).
- Salidas de alimentación: 3 V y 5 V.
- Acelerómetro LIS302DL o LIS3DSH de 3 ejes.
- Sensor de audio omni-direccional MP45DT02.
- DAC de audio CS43L22.
- 8 LEDs: 4 de ellos programables por el usuario.
- Dos pulsadores: uno de usuario y uno de reset.
- USB OTG FS con micro-usb.
- Puertos de extensión de todos los pines de entrada / salida.

### 4.3. API de comunicación IWCA

Debido a que el módem no tiene ningún pin de entrada / salida libre para compartir el reloj o para generar interrupciones externas, toda la interacción entre el módem y el puente tiene que realizarse mediante un intercambio asíncrono de mensajes serie. Como consecuencia, es necesaria una API de comunicaciones para que la aplicación pueda configurar y gestionar el módem desde el puente.

Esta API utiliza un esquema maestro –esclavo en el cual el puente es el dispositivo maestro que controla al módem. La API permite que el puente pueda activar y desactivar el módem, configurar sus parámetros, guardar y borrar su configuración y, lo que es más importante, enviar y recibir mensajes inalámbricos a través de él.

Aunque la API ha sido diseñada pensando en la comunicación entre el puente y el módem, en realidad puede utilizarse en cualquier sistema de comunicación basado en un esquema maestro – esclavo, donde el maestro necesita utilizar al dispositivo esclavo para transmitir mensajes inalámbricos a través de una comunicación serie. Por este motivo se ha decidido llamar a esta API, ITACA WIRELESS COMMUNICATION API (IWCA).

La API IWCA ha sido diseñada para conseguir la mayor flexibilidad, adaptabilidad y portabilidad posibles con el fin de utilizarla en distintos dispositivos y dar soporte a un número de aplicaciones lo más amplio posible. Para lograr este cometido, la API no sólo define el formato de los mensajes serie, el protocolo comunicación serie y el tipo y la función de cada mensaje válido sino que, además, define el formato de los mensajes inalámbricos, el protocolo de comunicación inalámbrico y la transformación entre los mensajes serie y los mensajes inalámbricos. De esta forma se pretende independizar la comunicación inalámbrica que percibe el dispositivo maestro del hardware de comunicación del dispositivo esclavo. Así sea cual sea el hardware de comunicación inalámbrica, los mensajes de datos que recibe el dispositivo maestro siempre tendrán el mismo formato y la misma semántica en cada campo. Además, esto previene futuros errores de programación e interpretación por parte de los desarrolladores puesto que todo lo relacionado con el API está bien definido.

#### **4.3.1. Tipos de mensajes**

La API IWCA define 2 tipos distintos de mensajes:

- **Los mensajes de datos**

Los mensajes de datos se utilizan para intercambiar datos entre dos pares de nodos. La API define dos tipos de mensajes de datos relacionados entre sí, los mensajes de datos serie y los inalámbricos.

Los mensajes de datos serie se intercambian entre el puente y el módem en la comunicación serie. Cuando el puente necesita transmitir datos a un nodo remoto, envía un mensaje de datos serie al módem. Entonces el módem transforma este mensaje en un mensaje de datos inalámbrico y lo envía.

Los mensajes de datos inalámbricos se intercambian entre los módems en la comunicación inalámbrica. Cuando el módem recibe un mensaje inalámbrico lo transforma en un mensaje de datos serie y se lo envía al puente. De esta forma se consigue que las aplicaciones de ambos nodos puedan intercambiar datos.

- **Las órdenes**

Las órdenes son mensajes especiales cuya función es configurar y gestionar el comportamiento del módem. Además, también dan soporte a la iteración entre ambos dispositivos enviando reconocimientos de mensajes y gestionando el flujo de la comunicación. Para realizar estas funciones, la API define dos tipos de órdenes, las órdenes locales y las órdenes remotas.

Las órdenes locales son mensajes serie con una función definida. La Tabla 1 muestra todas las órdenes locales definidas por la API y la posibilidad de que puedan utilizarse como órdenes remotas.

Las órdenes remotas son mensajes inalámbricos. Cuando el módem recibe una orden remota serie la transforma en un mensaje inalámbrico y lo envía al módem remoto. Por lo tanto estas órdenes permiten que la aplicación pueda solicitar una función específica de un nodo remoto.

Tipo	Uso	Remoto
ACK	Control	SÍ
RESET	Control	NO
RECEIVER	Control	NO
SLEEP	Control	NO
WAKEUP	Control	SÍ
WAKEUP_NOTIFICATION	Control	NO
WAIT	Control	NO
READ_VALUE	Control	NO
VALUE	Control	NO
SAVE_CONF	Control	NO
ERASE_CONF	Control	NO
WSN_NOTIFICATION	Control	NO
WAKEUP_CONF	Configuración	NO
ADDRESS	Configuración	NO

Tabla 1. Órdenes de la API IWCA.

A continuación se describe brevemente la función de cada una de las órdenes:

- La orden ACK

Esta orden puede ser enviada tanto por el módem como por el puente. Se utiliza como reconocimiento a los mensajes recibidos por ambos dispositivos. La API define una bandera en el formato de todos sus mensajes para indicar cuando un mensaje necesita reconocimiento.

- La orden RESET

Esta orden sólo la envía el puente e indica que el módem debe reiniciarse. Una vez reiniciado, el módem envía una orden WAIT para indicar que está listo.

- La orden RECEIVER

Esta orden sólo la envía el puente y se utiliza para gestionar el receptor del módem, activándolo o desactivándolo. De esta forma se permite que la aplicación pueda ahorrar energía mientras no necesite recibir mensajes.

- La orden SLEEP

Esta orden sólo la envía el puente e indica que el módem tiene que desactivarse pasando a modo de bajo consumo. De esta forma se permite que el puente desactive el módem cuando ya no necesite utilizarlo.



- La orden WAKEUP

Esta orden puede ser tanto local como remota y siempre la envía el puente. Si la orden es local, indica que el módem debe activarse saliendo del modo de bajo consumo. De esta forma se permite que el puente pueda activar el módem en el momento en que necesite utilizarlo.

Si la orden es remota, el puente solicita el envío de una señal acústica o de un mensaje de activación dependiendo de si el módem dispone o no de un hardware específico para el envío de señales subacuáticas.

- La orden WAKEUP\_NOTIFICATION

Esta orden sólo la envía el módem cuando recibe una señal acústica o un mensaje de activación. Su función es la de avisar al puente para que pueda activarse en caso de que estuviera en modo de bajo consumo, permitiendo así que el puente pueda recibir mensajes de los nodos remotos mientras ahorra energía.

- La orden WAIT

Esta orden puede ser enviada tanto por el módem como por el puente. Se utiliza como un mensaje de control de flujo con el cual el dispositivo que la envía solicita la detención o la reanudación del flujo de mensajes.

- La orden READ\_VALUE

Esta orden sólo la envía el puente y se utiliza para obtener los parámetros de configuración del módem. Además, también permite obtener el valor RSSI tanto de la última transmisión realizada como del canal de comunicación.

- La orden VALUE

Esta orden sólo la envía el módem y se utiliza para responder con el valor apropiado a la orden READ\_VALUE enviada por el puente.

- La orden SAVE\_CONF

Esta orden sólo la envía el puente e indica que el módem tiene que guardar su configuración en memoria no volátil. Durante el proceso, el módem no puede recibir mensajes. Por este motivo el módem utiliza la orden WAIT para detener el flujo de mensajes hasta que termina el proceso. A continuación, vuelve a enviar otra orden WAIT para reanudar el flujo de mensajes.

- La orden ERASE\_CONF

Esta orden sólo la envía el puente e indica que el módem tiene que borrar su configuración de la memoria no volátil. Durante el proceso, el módem no puede recibir mensajes. Por este motivo el módem utiliza la orden WAIT para

detener el flujo de mensajes hasta que termina el proceso. A continuación, vuelve a enviar otra orden WAIT para reanudar el flujo de mensajes.

- La orden WSN\_NOTIFICATION

Esta orden sólo la envía el módem e indica que se ha recibido un mensaje de datos remoto de otro nodo. Por lo tanto, ante la recepción de esta orden, el puente tiene que prepararse para recibir el mensaje de datos.

Esta orden puede usarse como una orden de activación si el puente está en modo de bajo consumo pero el módem sigue activo. Esta situación puede darse cuando la aplicación tiene restricciones de energía pero el módem no dispone de un hardware de activación específico basado en la recepción de señales subacuáticas.

- La orden WAKEUP\_CONF

Esta orden sólo la envía el puente y se utiliza para configurar el hardware de activación basado en señales subacuáticas del módem.

En el caso de que el módem no disponga de tal hardware, la configuración se utiliza como un identificador para el mensaje de activación. De este modo, el módem sólo enviará la orden WAKEUP\_NOTIFICATION si el identificador recibido en el mensaje WAKEUP remoto coincide con la configuración de activación.

- La orden ADDRESS

Esta orden sólo la envía el puente y se utiliza para configurar la dirección de red del módem. En la actualidad, esta dirección coincide con la dirección MAC. Cuando el módem recibe este mensaje, configura su dirección, guarda su configuración y se reinicia para que los cambios surtan efecto.

#### **4.3.2. Adaptación a la aplicación**

La posibilidad de solicitar el reconocimiento de los mensajes y las órdenes por sí mismas permiten a la API adaptarse dinámicamente a las necesidades de la aplicación pero de forma muy limitada. Algunas aplicaciones, sobre todo las más complejas, necesitan que la API se adapte de una manera mucho más profunda a sus necesidades. Así, algunas aplicaciones pueden necesitar que los mensajes se reenvíen automáticamente hasta que se reciba su reconocimiento mientras que otras necesitan todo lo contrario, los mensajes no deben reconocerse ni reenviarse sin el control de la aplicación. Por otro lado, algunas aplicaciones pueden necesitar hacer uso del bajo consumo incluso cuando el módem no dispone hardware de activación basado en señales subacuáticas capaz de activar el puente cuando está en modo de bajo consumo.

Para permitir que la API se adapte más adecuadamente a las aplicaciones más complejas y a las particularidades del hardware de comunicación, la implementación incluye varias definiciones que añaden o eliminan código con ciertas funcionalidades. Obviamente, en este caso la adaptación de la API es estática y necesita que su código sea recompilado.

La API dispone de dos tipos de definiciones, las definiciones de configuración y las definiciones de comportamiento. Las definiciones de configuración permiten configurar parámetros de la API que pueden ser críticos para ciertas aplicaciones. Por ejemplo, el tamaño de las colas tanto de recepción como de envío, los valores por defecto de la configuración del módem, etc. Por su parte, las definiciones de comportamiento permiten modificar el comportamiento de la API ante ciertas circunstancias para adaptarse mejor a las aplicaciones. La Tabla 2 muestra todas las definiciones que pueden utilizarse antes de compilar la API para una aplicación en concreto.

Definición	Tipo	Valor por defecto
THIS_DEVICE_ADDRESS	Configuración	{0x11,0x22,0x33,0x44}
WIRELESS_PORT	Configuración	0xFF
WIRELESS_WAKEUP_LENGTH	Configuración	1
USART_BAUDRATE	Configuración	57600
SERIAL_QUEUE_SENDS	Configuración	2
SERIAL_QUEUE_PERIOD	Configuración	1000
WIRELESS_MAX_RESENDS	Configuración	2
WIRELESS_QUEUE_PERIOD	Configuración	1000
WIRELESS_WAKEUP_ENABLED	Comportamiento	
RESET_TIMESTAMP_ENABLE	Comportamiento	
WIRELESS_BLOQUING_RECEPTION	Comportamiento	
WIRELESS_BLOQUING_SENDING	Comportamiento	
WIRELESS_NET_ENABLE	Comportamiento	
USE_WSN_TX_QUEUE	Comportamiento	
USE_COM_TX_QUEUE	Comportamiento	
DEBUG	Comportamiento	

Tabla 2. Definiciones de la API IWCA

A continuación se explica brevemente el cambio de comportamiento ocasionado por cada una de las definiciones de comportamiento sobre el código de la API.

- WIRELESS\_WAKEUP\_ENABLED

El uso de esta definición indica que el módem dispone de un hardware específico de activación basado en señales inalámbricas.

Cuando se usa esta definición, la orden remota WAKEUP envía una señal inalámbrica a otros nodos y permite que el módem pueda recibirlas. Cuando se recibe una señal inalámbrica, el módem se activa, sólo en el caso de que

estuviera en modo de bajo consumo, y envía una orden WAKEUP\_NOTIFICATION al puente.

Cuando no se usa esta definición, la orden remota WAKEUP envía un mensaje de activación inalámbrico a otros nodos. Cuando se recibe un mensaje de activación, el módem envía una orden WAKEUP\_NOTIFICATION al puente. En este caso, el módem no puede usar el modo de bajo consumo ya que no podría recibir ningún mensaje de activación inalámbrico.

- RESET\_TIMESTAMP\_ENABLE

El uso de esta definición reinicia el reloj del sistema del módem cuando se recibe un mensaje de activación inalámbrico. Esta definición sólo puede usarse cuando no se utiliza la definición WIRELESS\_WAKEUP\_ENABLED.

- WIRELESS\_BLOQUING\_RECEPTION

El uso de esta definición indica que el módem no es capaz de realizar ningún procesamiento cuando su receptor está activado. Esta definición añade un estado de recepción en el módem al cual se entra cuando se activa el receptor inalámbrico. La recepción de cualquier mensaje serie saca automáticamente al módem del estado de recepción. Por lo tanto, el puente tiene que activar de nuevo el receptor inalámbrico después de enviar un reconocimiento de un mensaje recibido.

- WIRELESS\_BLOQUING\_SENDING

El uso de esta definición indica que el módem no es capaz de realizar ningún procesamiento cuando está enviando un mensaje. Esta definición añade el envío de órdenes de control de flujo antes y después del envío de un mensaje inalámbrico. De este modo el módem avisa al puente de cuándo puede reanudar el flujo de mensajes.

- WIRELESS\_NET\_ENABLE

El uso de esta definición indica que el módem forma parte de una red. Esta definición añade el reenvío automático de mensajes inalámbricos a otros nodos de la red cuando no están dirigidos al módem.

- USE\_WSN\_TX\_QUEUE

El uso de esta definición indica que el módem hace uso de una cola de envío para los mensajes inalámbricos. Esta definición añade el reenvío automático de mensajes inalámbricos cuando es necesario un reconocimiento para el mensaje. Cuando no se hace uso de esta definición, los mensajes inalámbricos nunca solicitan un reconocimiento.

- USE\_COM\_TX\_QUEUE

El uso de esta definición indica que el módem hace uso de una cola de envío para los mensajes serie. Esta definición añade el reenvío automático de mensajes serie cuando es necesario un reconocimiento para el mensaje.

- DEBUG

El uso de esta definición indica que el módem está en modo depuración. Esta definición deshabilita el temporizador *watchdog* para evitar reinicios durante la depuración.

#### **4.3.3. Protocolo de comunicación**

El protocolo de comunicación definido por la API IWCA define una configuración por defecto que puede ser modificada mediante las definiciones de configuración y que permite utilizar el módem sin necesidad de utilizar las órdenes. Por este motivo, el uso de las órdenes es opcional a excepción de la orden RECEIVER.

El protocolo define que el emisor debe decidir si desea recibir o no un reconocimiento cuando va a enviar un mensaje, estableciendo adecuadamente el valor de la bandera de reconocimiento del mensaje. En cuanto a la recepción, el protocolo define que el receptor está obligado a enviar un reconocimiento si la bandera de reconocimiento del mensaje está activa. En el caso de los mensajes de datos inalámbricos y las órdenes remotas se define, además, que antes de poder recibir uno de estos mensaje se debe activar el receptor del módem mediante una orden RECEIVER.

#### **4.3.4. Implementación**

La implementación de la API IWCA se divide en dos partes. Por un lado la implantación en el puente como una biblioteca de funciones. Por el otro, la implementación en el módem como una aplicación. En ambos casos la implementación se ha realizado mediante un esquema de capas para asegurar la máxima portabilidad posible del código.

- **Implementación en el módem**

La implementación en el módem es una aplicación que desarrolla todo el comportamiento definido por la API, desde la traducción entre mensajes serie y mensajes inalámbricos hasta la funcionalidad de cada una de las órdenes.

La implementación está diseñada en 5 capas, la capa núcleo, la capa de protocolo serie, la capa de protocolo inalámbrico, la capa de abstracción y la capa hardware.

Núcleo		
Protocolo serie		Protocolo inalámbrico
Abstracción serie	Abstracción núcleo	Abstracción inalámbrica
Hardware		

Figura 9. Implementación en capas de la API IWCA en el módem

En la capa hardware se implementan las funciones necesarias para gestionar el hardware del módem. Típicamente esta capa consiste en funciones distribuidas por el fabricante del hardware. En este caso, sin embargo, la mayoría de las funciones han sido desarrolladas por el grupo de investigación.

En la capa de abstracción se definen todas aquellas funciones que tienen una dependencia con el hardware. La capa de abstracción se divide en tres subcapas para permitir una portabilidad más flexible:

- La capa de abstracción serie

En esta subcapa se definen las funciones que dependen del hardware de comunicación serie del módem. Aquí se implementan las funciones de inicialización y configuración de la comunicación serie y las funciones de envío y recepción de las tramas serie.

Esta subcapa permite cambiar el hardware de comunicación serie sin necesidad de modificar toda la capa de abstracción. Únicamente hay que volver a implementar las funciones de esta subcapa para gestionar el nuevo hardware. De esta forma se permite cambiar fácilmente entre diversas opciones de comunicación serie tales como UART TTL, SPI o I2C.

- La capa de abstracción inalámbrica

En esta subcapa se definen las funciones que dependen del hardware de comunicación inalámbrico del módem. Aquí se implementan las funciones de inicialización y configuración de la comunicación inalámbrica y las funciones de envío y recepción de las tramas inalámbricas.

Esta subcapa permite cambiar el hardware de comunicación inalámbrico sin necesidad de modificar toda la capa de abstracción. Únicamente hay que volver a implementar las funciones de esta subcapa para gestionar el nuevo hardware. De esta forma se permite cambiar fácilmente entre diversas opciones de comunicación inalámbrica tales como WIFI, ZIG-BEE, BLUETOOTH, 868 MHz, ultrasonidos, etc.

- La capa de abstracción núcleo

En esta subcapa se definen las funciones que dependen del hardware de procesamiento del módem, es decir del microcontrolador. Aquí se implementan las funciones para gestionar el bajo consumo, la memoria flash y los temporizadores.

Si se decide cambiar de microcontrolador será necesario volver a implementar toda la capa de abstracción, es decir, las tres subcapas.

En la capa de protocolo serie se implementan las funciones de encapsulamiento de los datos en mensajes serie. Además es aquí donde se define el formato de los mensajes serie y donde se implementan las funciones de procesamiento de los mensajes recibidos.

En la capa de protocolo inalámbrico se implementan las funciones de encapsulamiento de los datos en mensajes inalámbricos. Además es aquí donde se define el formato de los mensajes inalámbricos y donde se implementan las funciones de procesamiento de los mensajes inalámbricos recibidos.

Por último, la capa núcleo implementa todas las funciones necesarias para dar funcionalidad a la API. En esta capa se realizan las transformaciones de mensajes serie a mensajes inalámbricos y viceversa. Además se implementa la funcionalidad de cada una de las órdenes y la gestión de la configuración del módem.

- **Implementación en el puente**

La implementación en el puente es una biblioteca de funciones que permite a los desarrolladores abstraerse del envío y de la recepción de mensajes serie cuando utilizan la API. Cada una de las funciones se ocupa de todo el proceso de envío del mensaje serie así como de la recepción de las respuestas a dichos mensajes.

La implementación está diseñada en cuatro capas, la capa núcleo, la capa de protocolo, la capa de abstracción y la capa hardware.

Núcleo
Protocolo
Abstracción
Hardware

Figura 10. Implementación en capas de la API IWCA en el puente

En la capa hardware se implementan las funciones necesarias para gestionar el hardware de comunicación serie. Típicamente esta capa consiste en funciones distribuidas por el fabricante del hardware.

En la capa de abstracción se definen todas aquellas funciones que tienen una dependencia con el hardware. En ella se implementan las funciones de inicialización y configuración de la comunicación serie y las funciones de envío y recepción de las tramas serie. Esta capa es la única que hay que volver a implementar si se piensa cambiar el hardware de comunicación serie.

En la capa de protocolo se implementan las funciones de encapsulamiento de los datos en mensajes serie. Además es aquí donde se define el formato de los mensajes serie y donde se implementan las funciones de procesamiento de los mensajes recibidos.

Por último, la capa núcleo implementa todas las funciones necesarias para que el desarrollador pueda gestionar la configuración y el comportamiento del módem mediante órdenes y realizar el envío y la recepción de los mensajes de datos.

#### **4.3.5. Aplicación de test**

Además de la implementación de la API de comunicaciones, también se ha desarrollado una aplicación de test para PC con el fin de evaluar la implementación realizada. Esta aplicación ha sido desarrollada en Qt [37] con el objetivo de facilitar su portabilidad y añadir una interfaz gráfica con el menor esfuerzo posible.

La aplicación en cuestión permite enviar los mensajes de datos y las órdenes definidas por la API IWCA mediante una interfaz amigable a través de un puerto de comunicación serie. La interfaz permite seleccionar el puerto de comunicación serie al cual está conectado el módem mediante una lista desplegable. Por otra parte dispone también de un área de texto donde se muestran todos los mensajes de datos enviados, todos los mensajes de datos recibidos y todas las órdenes enviadas así como sus respuestas. Por lo tanto, la aplicación puede utilizarse como una herramienta para configurar el módem y para obtener sus valores de configuración. Además la interfaz también permite enviar un mensaje repetidamente cada cierto periodo de tiempo, facilitando así la realización de pruebas para evaluar la API de comunicaciones.



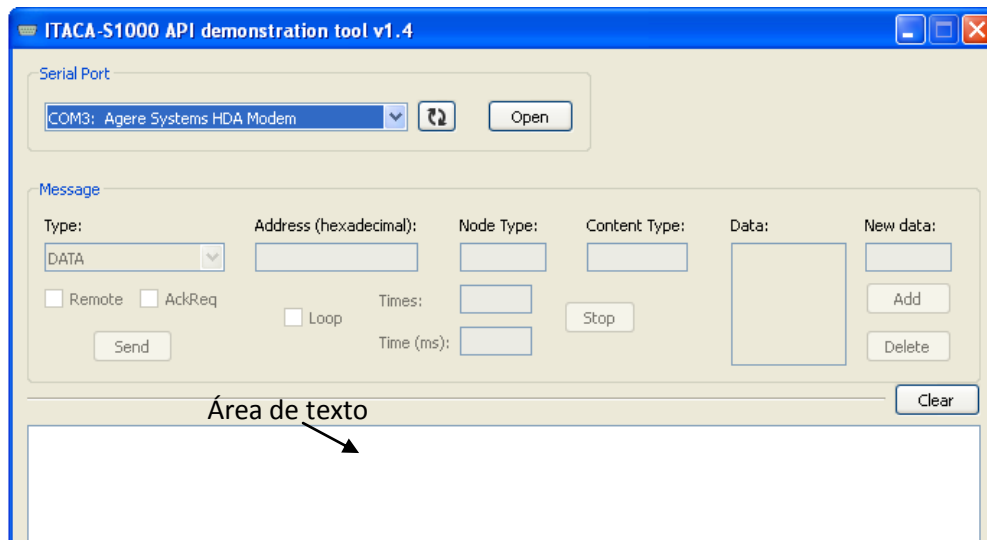


Figura 11. Interfaz gráfica de la aplicación de demostración de la API IWCA

Finalmente, la aplicación incluye registros automáticos tanto de errores como de respuestas recibidas. En el registro de errores se registran todos aquellos datos inesperados y todos aquellos mensajes erróneos enviados por el módem junto con su instante de recepción, el cual se obtiene del reloj del sistema del PC. En el registro de respuestas se registran todos aquellos mensajes válidos enviados por el módem, esto incluye tanto mensaje de datos como respuestas a las órdenes. Al igual que en el registro de errores, en el registro de respuestas también se incluye el instante de recepción de los mensajes.

#### 4.4. API de comunicación IUMOTE

Tal y como se ha comentado al principio de este apartado, el equipo subacuático necesita comunicarse con el sonar para solicitar el envío de la señal subacuática y recibir los tiempos de vuelo calculados. Para asegurarse de que el sonar es capaz de comunicarse con una amplia variedad de dispositivos, la comunicación se ha realizado mediante un canal serie gestionado por una API de comunicaciones llamada IUMOTE.

##### 4.4.1. Tipos de mensajes

La API IUMOTE define 5 tipos de mensajes para dar soporte a la aplicación de obtención de tiempos de vuelo:

- El mensaje START

Utilizado por el equipo subacuático para activar el sonar.

- El mensaje STOP

Utilizado por el equipo subacuático para desactivar el sonar. Tras la recepción de este mensaje, el sonar debe entrar en modo de bajo consumo.

- El mensaje CONF.

Utilizado por el equipo subacuático para indicar el tiempo máximo en el cual espera recibir los tiempos de vuelo de las diferentes balizas.

- El mensaje ACK

Utilizado por el sonar para indicar la correcta recepción de los mensajes enviados por el equipo subacuático.

- El mensaje DATA

Utilizado por el sonar para enviar la profundidad y el tiempo de vuelo de cada baliza que ha respondido a la señal subacuática.

- El mensaje DEBUG

Utilizado por el sonar para enviar datos útiles con los cuales depurar su funcionamiento.

#### **4.4.2. Protocolo de comunicación**

El protocolo de comunicación definido por la API IUMOTE es un protocolo basado en reconocimientos. Los mensajes START, STOP y CONF necesitan ser reconocidos tras su correcta recepción. Sin embargo, los mensajes DATA y DEBUG no necesitan ser reconocidos ya que el sonar no puede reenviar los mensajes. Esto es debido a que estos mensajes se generan en el proceso de obtención de los tiempos de vuelo y, para conseguir la máxima precisión posible, el sonar utiliza todo su tiempo de cómputo en recibir los mensajes de cada baliza y realizar el cálculo del tiempo de vuelo. La Figura 12 muestra la secuencia de mensajes.

El protocolo define la siguiente secuencia de eventos para realizar la obtención de los tiempos de vuelo. En primer lugar el equipo subacuático debe enviar un mensaje START para activar el sonar. Una vez activado, el sonar envía un reconocimiento y espera un mensaje CONF. Entonces el equipo subacuático envía un mensaje CONF para indicar el tiempo máximo que estará esperando para recibir los tiempos de vuelo. Tras recibir un mensaje CONF, el sonar envía un reconocimiento y empieza el proceso de obtención de tiempos de vuelo. Cada vez que el sonar termina el cálculo del tiempo de vuelo de una baliza, genera y envía un mensaje DATA con la profundidad de la baliza, el valor del tiempo de vuelo y el instante en el que se realizó el cálculo en referencia al tiempo máximo indicado por el equipo subacuático. Durante la obtención de los tiempos de vuelo, el equipo subacuático puede enviar en cualquier momento un mensaje CONF para reconfigurar el tiempo máximo o un mensaje STOP para terminar el proceso. Por último, cuando el equipo subacuático termina la espera para recibir los tiempos de vuelo puede optar por volver a enviar un mensaje CONF para obtener más tiempos de vuelo, o bien enviar un mensaje STOP para desactivar el sonar. Ante la recepción de un

mensaje STOP, el sonar envía un reconocimiento y pasa a modo de bajo consumo.

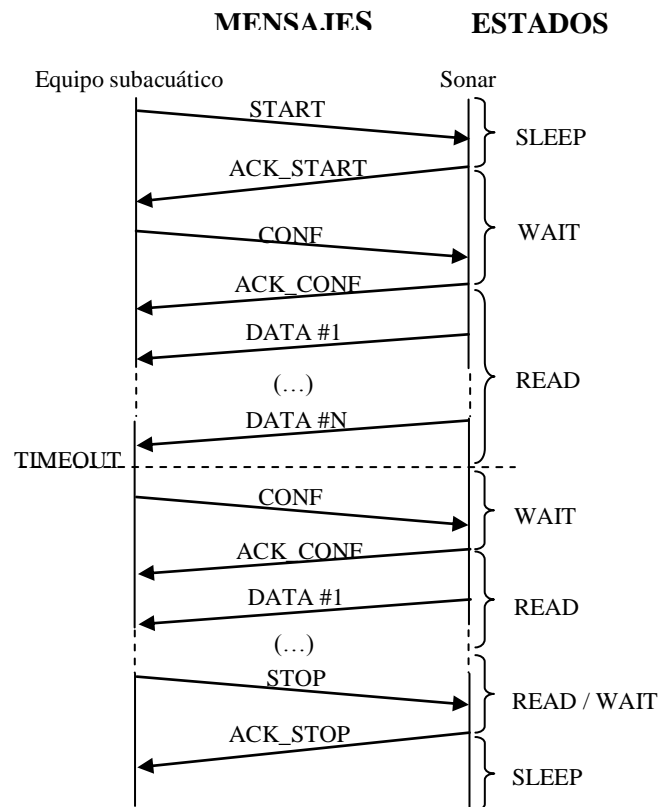


Figura 12. Secuencia de eventos del protocolo de comunicación IUMOTE

#### 4.4.3. Implementación

Al igual que la API IWCA, la API IUMOTE ha sido diseñada para ser independiente del hardware de comunicación utilizado. La implementación de esta API consiste en tres capas: la capa de protocolo, la capa de abstracción y la capa hardware.

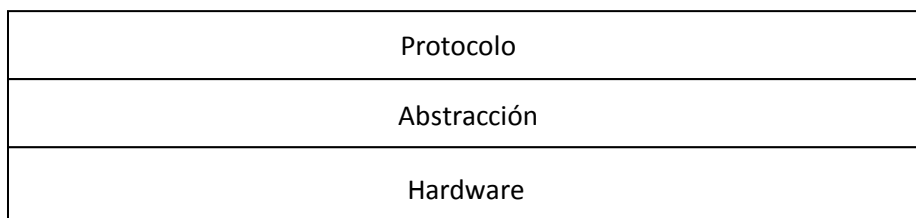


Figura 13. Implementación en capas de la API IUMOTE

En la capa hardware se implementan las funciones necesarias para gestionar el hardware de comunicación. Típicamente esta capa consiste en funciones distribuidas por el fabricante del hardware.

En la capa de abstracción se definen todas aquellas funciones que tienen una dependencia con el hardware serie. En esta capa se implementan las funciones de inicialización y configuración de la comunicación serie y las

funciones de envío y recepción de tramas serie. Esta capa es la única que hay que volver a implementar si se cambia el hardware de comunicación serie.

Por último, en la capa de protocolo se define el formato de los mensajes serie y se implementan todas las funciones necesarias para el funcionamiento del protocolo definido por la API. En esta capa se encuentran las funciones de encapsulamiento de los datos en mensajes y las funciones de procesamiento de los mensajes recibidos.

#### 4.4.4. Aplicación de test

Además de la implementación de la API IUMOTE, también se ha desarrollado una aplicación de test para PC con el fin de apoyar la evaluación de la aplicación de obtención de tiempos de vuelo. Esta aplicación ha sido desarrollada en Qt con el objetivo de facilitar su portabilidad y añadir una interfaz gráfica con el menor esfuerzo posible.

La aplicación en cuestión permite enviar todos los mensajes definidos por la API IUMOTE mediante una interfaz amigable a través de un puerto de comunicación serie. La interfaz permite seleccionar el puerto de comunicación serie al cual está conectado el sonar mediante una lista desplegable. Por otra parte dispone también de un área de texto donde se muestran todos los mensajes enviados y todos los mensajes del sonar recibidos. Además la interfaz permite enviar un mensaje repetidamente cada cierto periodo de tiempo, facilitando así la realización de pruebas para evaluar la aplicación de obtención de tiempos de vuelo.

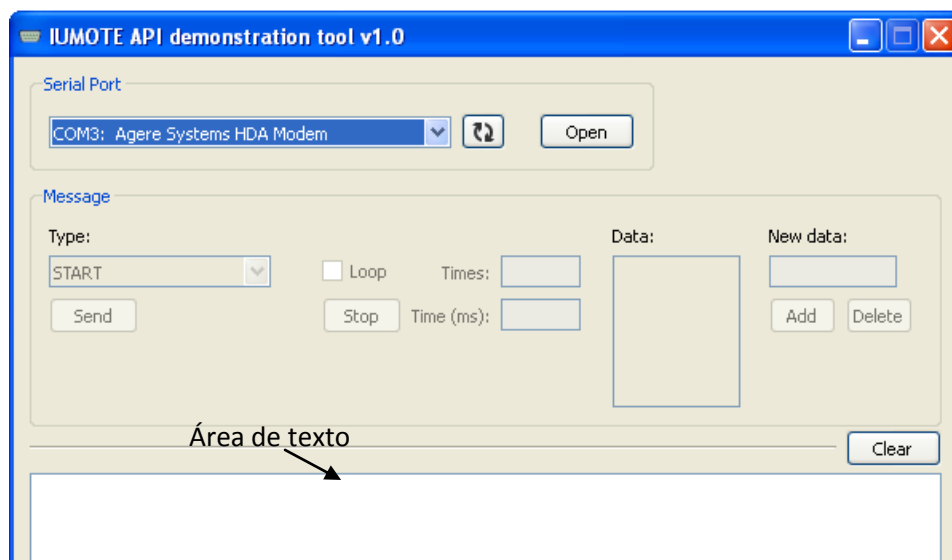


Figura 14. Interfaz gráfica de la aplicación de demostración de la API IUMOTE

Finalmente, la aplicación incluye registros automáticos tanto de errores como de respuestas recibidas. En el registro de errores se registran todos aquellos datos inesperados y todos aquellos mensajes erróneos enviados por el sonar junto con su instante de recepción, el cual se obtiene del reloj del

sistema del PC. En el registro de respuestas se registran todos aquellos mensajes DATA y DEBUG válidos enviados por el sonar incluyendo también su instante de recepción.

#### **4.5. Protocolo de acceso al medio**

En el sistema de obtención de tiempos de vuelo, el equipo subacuático enviará una señal acústica en todas direcciones cuando necesite conocer su localización. Todas las balizas que reciban esta señal intentarán enviar su mensaje de respuesta. Por lo tanto existe la posibilidad de colisiones entre transmisiones de las balizas y es necesario utilizar un protocolo de acceso al medio para arbitrar las comunicaciones.

En el medio subacuático, el uso de los protocolos clásicos de acceso al medio utilizados en redes inalámbricas terrestres basados en acceso múltiple por división de tiempo (TDMA), acceso múltiple por división de frecuencia (FDMA) y acceso múltiple por división de código (CDMA) no son adecuados [38]. Los protocolos basados en TDMA ven reducido su rendimiento como consecuencia del aumento del tamaño de sus ranuras de tiempo debido al gran retardo de propagación de las señales acústicas en el agua. Por otro lado, los protocolos basados en FDMA no funcionan adecuadamente ya que el ancho de banda disponible en las comunicaciones suele ser bastante bajo. Finalmente, los protocolos basados en CDMA reducen la velocidad de transmisión alcanzada por los dispositivos de comunicación que, en el caso de los sistemas subacuáticos, ya de por sí es bastante baja [1]. Como consecuencia, para evitar en lo posible las colisiones y mejorar la eficiencia de las comunicaciones, varios autores han propuesto protocolos adaptados al medio subacuático basados en intercambio de mensajes RTS/CTS y en CSMA. Ejemplos de estos protocolos son FAMA-U [39] y T-Lohi [40].

Como el módem ITACA-S1000 dispone de una velocidad de transmisión de tan sólo 1 Kbps, no parece adecuado disminuir todavía más su velocidad utilizando protocolos basados en división de código. Por otro lado, los protocolos basados en división de frecuencia no se pueden utilizar debido a que el módem está diseñado para trabajar en una única frecuencia. Además, los protocolos basados en escuchar mensajes de los nodos vecinos, como los basados en RTS/CTS o los basados en detección de portadora, tampoco se pueden utilizar a priori debido al ángulo de 60° del transductor utilizado por el módem. La única manera de utilizar estos protocolos es incorporar varios módems en las balizas de modo que cubran al menos 240°. De esta forma, las balizas podrían escuchar a todos los vecinos que se encuentren delante de ella y en sus laterales.

Como los protocolos basados en RTS/CTS y en CSMA necesitan de varios sistemas de comunicación trabajando en conjunto, es previsible que aparezcan problemas debido a la coordinación de estos sistemas. Ya que el sistema a desarrollar es una primera solución, se ha decidido que por el momento es más adecuado realizar una prueba de concepto con un protocolo más simple y, en un futuro, utilizar protocolos más complejos. Por este motivo, se ha decidido utilizar un protocolo TDMA basado en ranuras. Así, cada baliza tendrá una

ranura de tiempo asignada y, por lo tanto, cuando reciba la señal subacuática esperará a su ranura para transmitir su mensaje evitando así la posibilidad de colisiones.

#### **4.6. Aplicación de obtención de tiempos de vuelo**

Tal y como se ha explicado al principio de este apartado, el sistema de obtención del tiempo de vuelo desarrollado pretende dar soporte a la localización de equipos subacuáticos con capacidad de procesamiento tales como AUV, ROV y cualquier dispositivo que disponga de un microcontrolador.

El sistema consiste en un conjunto de balizas que responden a una señal subacuática generada por un sonar gestionado por el equipo subacuático. Cuando el equipo subacuático necesita conocer su localización, se comunica con el sonar por medio de la API de comunicaciones IUMOTE para solicitar la obtención del tiempo de vuelo de las balizas al alcance durante un cierto periodo de tiempo. Entonces, el sonar envía una señal subacuática y espera durante un tiempo recibir respuesta de las balizas al alcance. Cuando la espera termina, el sonar vuelve a enviar una señal subacuática siempre y cuando no haya vencido el tiempo indicado por el equipo subacuático.

Cuando las balizas reciben la señal subacuática generan un mensaje de datos con su profundidad y lo envían siguiendo un protocolo de acceso al medio TDMA basado en ranuras para evitar colisiones. La generación del mensaje no es inmediata y las balizas necesitan contar el tiempo de procesamiento que tardan en enviar el mensaje de datos desde que reciben la señal subacuática. Este tiempo lo añaden al mensaje de datos en una marca de tiempo que define la API de comunicaciones IWCA. Así el sonar calcula el tiempo de vuelo como la mitad de la resta entre el tiempo transcurrido desde el envío de la señal subacuática hasta la obtención del mensaje de datos, y el tiempo de procesamiento de la baliza obtenido de la marca de tiempo del mensaje recibido.

Tanto el sonar como las balizas consisten en un sistema maestro-esclavo donde un dispositivo llamado puente gestiona el módem ITACA-S1000 por medio de la API de comunicaciones IWCA para enviar y recibir mensajes subacuáticos. Por lo tanto, la aplicación implementada para desarrollar la funcionalidad del sistema descrito consiste en una aplicación distribuida donde una parte se ejecuta sobre el puente del sonar y la otra sobre el puente de la baliza.

A continuación se explica detalladamente todo el proceso de obtención del tiempo de vuelo de una baliza para facilitar la comprensión de la implementación realizada.

##### **4.6.1. Visión general**

En el estado inicial del sistema, el sonar permanece activo a la espera de órdenes del equipo subacuático mientras que las balizas permanecen en modo de bajo consumo a la espera de señales subacuáticas. Una vez que el sistema

lleva un tiempo funcionando, se espera que el equipo subacuático desactive el sonar para ahorrar energía. Por lo tanto, el sonar también permanecerá en modo de bajo consumo a la espera de órdenes.

La Figura 15 muestra el flujo del intercambio de mensajes entre los dispositivos que intervienen en el proceso de obtención del tiempo de vuelo. A continuación se explica paso por paso todo el proceso para obtener el tiempo de vuelo de una baliza teniendo en cuenta que inicialmente tanto el sonar como la baliza están en modo de bajo consumo.

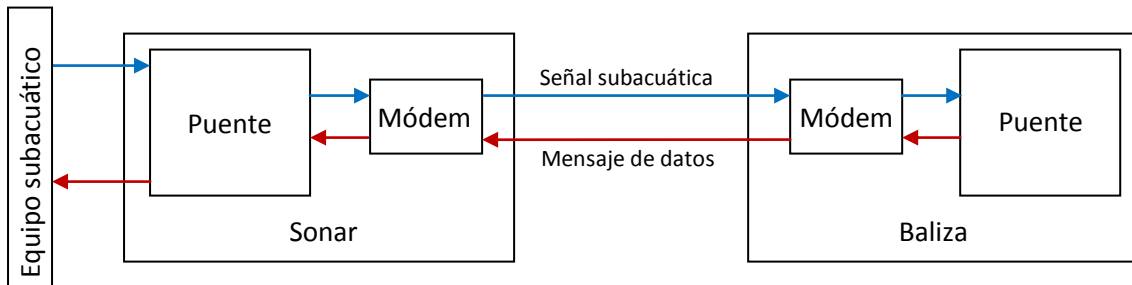


Figura 15. Intercambio de mensajes en la obtención del tiempo de vuelo.

1. El equipo subacuático envía un mensaje START al puente del sonar precedido de un byte de activación.
2. Al recibir el byte de activación, el puente del sonar sale del modo de bajo consumo y recibe el mensaje START. Entonces envía un reconocimiento al equipo subacuático y queda a la espera de órdenes.
3. El equipo subacuático envía un mensaje CONF al puente del sonar con el tiempo en el cual espera recibir los tiempos de vuelo.
4. El puente del sonar envía un reconocimiento al equipo subacuático, configura un temporizador con el tiempo recibido y empieza el proceso de obtención de tiempos de vuelo.
5. El puente del sonar envía un mensaje WAKEUP remoto con solicitud de reconocimiento al módem del sonar para generar una señal subacuática.
6. El módem del sonar envía un reconocimiento al puente del sonar. A continuación, envía un mensaje WAIT para indicar que va a estar ocupado en la generación de la señal subacuática. Entonces genera y envía la señal. Cuando termina el envío de la señal subacuática, envía otro mensaje WAIT al puente del sonar para indicarle que está listo de nuevo para recibir más mensajes.
7. En el momento en el que el puente del sonar recibe el último mensaje WAIT, configura un temporizador para esperar las respuestas de las balizas teniendo en cuenta las ranuras de tiempo TDMA. A continuación, envía un mensaje RECEIVER con solicitud de reconocimiento al módem del sonar para activar el receptor y poder recibir las respuestas.

8. El módem del sonar envía un reconocimiento al puente del sonar y entra en el estado de recepción.
9. El módem de la baliza recibe la señal subacuática saliendo del modo de bajo consumo. A continuación, avisa al puente de la baliza mediante el envío de un mensaje WAKEUP\_NOTIFICATION precedido por un byte de activación.
10. Al recibir el byte de activación, el puente de la baliza sale del modo de bajo consumo y recibe el mensaje WAKEUP\_NOTIFICATION. Entonces, espera a que llegue su ranura de tiempo TDMA antes de enviar el mensaje de datos.
11. Al comienzo de su ranura TDMA, el puente de la baliza comprueba el canal de comunicación inalámbrico para asegurarse de que está libre. Para ello envía un mensaje READ\_VALUE con solicitud de reconocimiento al módem de la baliza para obtener el valor RSSI del canal.
12. El módem de la baliza envía un reconocimiento seguido de un mensaje VALUE con el valor RSSI del canal al puente de la baliza.
13. El puente de la baliza comprueba el valor RSSI para asegurarse de que el canal está libre. Si el canal está ocupado vuelve a realizar la secuencia a partir del punto 11. Si el canal está libre obtiene su profundidad y la envía en un mensaje de datos serie con solicitud de reconocimiento al módem de la baliza.
14. El módem de la baliza envía un reconocimiento al puente de la baliza y transforma el mensaje de datos serie en un mensaje de datos inalámbrico.
15. El módem de la baliza envía un mensaje WAIT al puente de la baliza para indicar que va a estar ocupado con el envío del mensaje de datos inalámbrico. A continuación envía el mensaje. Cuando termina el envío del mensaje inalámbrico, envía otro mensaje WAIT al puente de la baliza para indicarle que está listo de nuevo para recibir más mensajes.
16. En el momento en el que el puente de la baliza recibe el último mensaje WAIT, envía un mensaje SLEEP con solicitud de reconocimiento al módem de la baliza para desactivarlo.
17. El módem de la baliza envía un reconocimiento al puente de la baliza y entra en modo de bajo consumo.
18. Al recibir el reconocimiento, el puente de la baliza entra también en modo de bajo consumo.
19. Cuando el módem del sonar recibe el mensaje de datos de la baliza envía un mensaje WSN\_NOTIFICATION al puente del sonar. A



continuación, transforma el mensaje de datos inalámbrico en un mensaje de datos serie y lo envía al puente del sonar.

20. Al recibir el mensaje WSN\_NOTIFICATION, el puente del sonar espera recibir el mensaje de datos de la baliza y, una vez recibido, calcula el tiempo de vuelo para esa baliza.
21. El puente del sonar genera un mensaje DATA con el instante de tiempo en el cual se ha calculado el tiempo de vuelo contando desde la recepción del mensaje CONF del punto 4, el valor del tiempo de vuelo y la profundidad de la baliza. A continuación se envía el mensaje al equipo subacuático.

De esta forma se obtiene el tiempo de vuelo de una baliza. A partir de este punto, el sonar continúa recibiendo respuestas de cada una de las balizas siguiendo con el proceso desde el punto 9. Una vez que el sonar termina la espera de respuestas iniciada en el punto 7, vuelve a seguir el proceso desde el punto 5 para enviar de nuevo la señal subacuática si el temporizador del punto 4 no ha vencido, es decir, si el equipo subacuático todavía espera recibir más tiempos de vuelo. Si no es así, el puente del sonar envía un mensaje RECEIVER al módem del sonar para desactivar su receptor y queda a la espera de recibir más órdenes del equipo subacuático.

En este punto, el equipo subacuático puede decidir continuar con la obtención de los tiempos de vuelo o terminar el proceso. Si decide continuar con el proceso de obtención de tiempos de vuelo, el proceso empieza de nuevo a partir del punto 3. En caso contrario, el equipo subacuático envía un mensaje STOP al puente del sonar. Al recibir el mensaje, el puente del sonar envía un reconocimiento al equipo subacuático y envía un mensaje SLEEP con solicitud de reconocimiento al módem del sonar para desactivarlo. Cuando el módem del sonar recibe el mensaje, envía un reconocimiento al puente del sonar y entra en modo de bajo consumo. Finalmente, al recibir el reconocimiento, el puente del sonar también entra en modo de bajo consumo volviendo así al estado inicial de partida.

#### **4.6.2. Implementación del protocolo TDMA**

En un protocolo TDMA basado en ranuras, el tamaño y la asignación de las ranuras son cuestiones críticas. El tamaño de la ranura tiene que ser lo suficientemente grande como para permitir que el mensaje de la baliza llegue al equipo subacuático después de recibir la señal subacuática. Pero, a la vez, tiene que ser lo suficientemente ajustada como para que no se pierda tiempo después de que el equipo subacuático reciba el mensaje. Cuanto más tiempo se pierda por ranura, mayor será el tiempo total que el equipo subacuático está a la escucha de respuestas. Por lo tanto, mayor será el consumo de energía y de recursos en el sistema de comunicación del equipo subacuático.

Teniendo en cuenta el intercambio de mensajes entre el sonar y una baliza explicado en el punto anterior, la ranura TDMA empieza cuando la baliza recibe la señal subacuática. El tiempo de ranura se divide entonces en el tiempo de

procesamiento necesario para que la baliza envíe el mensaje de datos después de la recepción de la señal subacuática, el tiempo de transmisión del mensaje y el tiempo de vuelo del mensaje.

$$T_{ranura} = T_{procesamiento} + T_{transmisión} + T_{vuelo}$$

Ecuación 1. Tamaño de una ranura de tiempo TDMA

Por su parte, la asignación de la ranura también tiene un impacto en el consumo del sistema de comunicación del equipo subacuático. A mayor número de ranuras mayor será el tiempo que el equipo subacuático permanece a la escucha de respuestas. Por lo tanto, el número de ranuras tiene que ser lo más reducido posible. Las balizas que no están al alcance unas de otras pueden reutilizar ranuras ya asignadas. Como consecuencia, durante la asignación de ranuras sólo se necesita tener en cuenta a aquellas balizas con posibilidad de colisión.

El problema de asignación de ranuras es equivalente al problema de coloración de grafos [41], el cual es un problema NP completo conocido. Por lo tanto, para resolver este problema se necesitan algoritmos específicos con los cuales llegar a una solución aproximada, como por ejemplo los algoritmos presentados en [42] y [43]. En este caso, las ranuras se asignarán manualmente ya que se espera que el número total de balizas sea reducido y el número de balizas con posibilidad de colisión sea como máximo de 4.

La asignación de la ranura se ha implementado utilizando el último byte de la dirección MAC de la baliza, de forma que este último byte indica el número de ranura utilizada. Por consiguiente, el resto de bytes se utilizan para conseguir una dirección única para la baliza. De esta forma, durante el despliegado de la malla de balizas se puede asignar manualmente la ranura TDMA de cada baliza al mismo tiempo que se configura su dirección MAC. Por ejemplo, en una red de 4 balizas donde tres de ellas están en colisión mutua y la cuarta está en colisión con la tercera, la primera puede tener la dirección 0xA0000001, la segunda 0xA0000003, la tercera 0xA0000002 y la cuarta la dirección 0xA0000101. De esta forma la primera y la cuarta baliza comparten la ranura 1.

#### 4.6.3. Obtención del tiempo de vuelo

Como se ha explicado anteriormente, el cálculo del tiempo de vuelo se basa en el intercambio de mensajes subacuáticos entre el sonar y una baliza para obtener el tiempo de ida y vuelta a partir del envío de una señal subacuática. Como simplificación, se asume que el tiempo de ida de la señal subacuática es el mismo que el tiempo de vuelta de la respuesta por parte de la baliza. Así, el tiempo de vuelo se calcula como la mitad de la resta entre el tiempo de ida y vuelta y el tiempo de procesamiento de la baliza.

$$T_{vuelo} = \frac{T_{RTT} - T_{proc\_baliza}}{2}$$

Tal y como se observa en la Figura 15, en la obtención del tiempo de vuelo intervienen 4 dispositivos distintos. Dado que la comunicación entre estos dispositivos se realiza mediante un intercambio de mensajes asíncrono, cada uno de ellos es un dispositivo totalmente independiente con su propia visión del paso del tiempo. Por lo tanto, en este sistema existen cuatro relojes distintos.

Para que el cálculo del tiempo de vuelo pueda realizarse adecuadamente, todos los relojes que intervienen en el proceso tienen que estar sincronizados. Con el fin de simplificar el proceso de sincronización y reducir el error en el cálculo hay que intentar reducir el número de relojes a dos, un reloj para el sistema de comunicación del sonar y un reloj para el sistema de comunicación de la baliza. Además, la obtención del valor de los relojes tiene que realizarse lo más cerca posible del envío y la recepción de los mensajes para reducir el error cometido en el cálculo. De esta forma se pretende reducir el tiempo que no se tiene en cuenta y que se añade al tiempo de vuelo calculado.

Teniendo en cuenta los requisitos previamente mencionados, la opción más adecuada es utilizar el reloj del módem en el proceso de obtención del tiempo de vuelo. Sin embargo, hay que tener presente la limitación de procesamiento del actual módem ITACA-S1000. En el caso de la baliza, el tiempo se obtiene desde que llega la señal subacuática hasta que se envía el mensaje de datos, por lo tanto no hay ningún problema. Sin embargo en el caso del sonar, el tiempo se obtiene desde que se envía la señal subacuática hasta que se recibe el mensaje de datos de la baliza. Es decir, el módem tiene que contar el tiempo mientras recibe el mensaje. No obstante, como ya se comentó en el punto 4.1, el módem no es capaz de realizar ninguna otra tarea durante el proceso de recepción

Para resolver este inconveniente se ha optado por utilizar el reloj del módem en el caso de la baliza y el reloj del puente en el caso del sonar. Así, el tiempo en el sonar empieza a contarse justo después de que el puente envíe el mensaje serie WAKEUP remoto al módem y termina en el momento en el que el puente recibe un mensaje WSN\_NOTIFICATION. En la baliza, el tiempo empieza a contarse en el momento en el que el módem recibe la señal subacuática y termina justo antes de enviar el mensaje de datos con su profundidad. Finalmente, el tiempo de vuelo se calcula como la mitad de la resta entre el tiempo contado por el puente del sonar y el tiempo contado por el módem de la baliza.

$$T_{vuelo} = \frac{T_{puente\_sonar} - T_{módem\_baliza}}{2}$$

Este proceso de obtención del tiempo de vuelo no es capaz de tener en cuenta todo el tiempo involucrado en la obtención del tiempo de vuelo. En el sonar, el tiempo empieza a contar en el momento en el que el puente recibe el mensaje WAIT que indica que la señal subacuática se ha enviado. Sin embargo, el tiempo tiene que empezar a contarse desde que el módem termina el envío de la señal subacuática. Por lo tanto, al tiempo contado por el puente hay que sumarle el tiempo que necesita el módem para enviar el mensaje WAIT después de enviar la señal subacuática, el tiempo de envío del mensaje

WAIT y el tiempo que necesita el puente para procesar el mensaje WAIT. Además, el puente termina de contar cuando se recibe el mensaje WSN\_NOTIFICATION del módem. No obstante, la cuenta tiene que terminar cuando el módem recibe el mensaje de datos de la baliza. Por lo tanto, al tiempo contado por el puente hay que restarle el tiempo que tarda el módem en procesar el mensaje de datos recibido y empezar con el envío del mensaje WSN\_NOTIFICATION, el tiempo de envío del mensaje WSN\_NOTIFICATION y el tiempo que tarda el puente en procesar el mensaje WSN\_NOTIFICATION.

$$T_{puente\_sonar} = T_{cuenta\_puente} + T_{módem\_wait} + T_{wait} + T_{puente\_wait} - T_{módem\_not} - T_{notificación} - T_{puente\_not}$$

En el caso de la baliza, el tiempo contado tampoco se corresponde con el tiempo que realmente tarda en responder a la señal subacuática. En este caso, el módem empieza a contar cuando tiene constancia de que se ha recibido una señal subacuática. Sin embargo, tiene que empezar a contar justo cuando recibe la señal subacuática. Por lo tanto, al tiempo contado por el módem hay que sumarle el tiempo que necesita el módem para procesar la señal subacuática recibida. Además, el módem termina de contar justo antes de enviar el mensaje de datos de la baliza pero la cuenta debería involucrar también el envío del mensaje de datos. Por lo tanto, al tiempo contado por el módem hay que sumarle el tiempo de envío del mensaje de datos.

$$T_{módem\_baliza} = T_{cuenta\_módem} + T_{módem\_datos} + T_{msg\_datos}$$

De este modo, el cálculo del tiempo de vuelo que tiene que realizar el puente del sonar viene dado en la siguiente fórmula:

$$T_{vuelo} = \frac{1}{2} * (T_{cuenta\_puente} + T_{módem\_wait} + T_{wait} + T_{puente\_wait} - T_{módem\_not} - T_{notificación} - T_{puente\_not} - (T_{cuenta\_módem} + T_{módem\_datos} + T_{msg\_datos}))$$

Ecuación 2. Cálculo del tiempo de vuelo

#### 4.6.4. Implementación de la aplicación

Como se ha comentado anteriormente, la aplicación desarrollada es una aplicación distribuida que se ejecuta en el puente del sonar y en el puente de las balizas. Por lo tanto, la implementación de la aplicación se divide en dos partes, la implementación en el sonar y la implementación en la baliza.

Ambas implementaciones hacen uso de las API de comunicaciones vistas anteriormente. En el caso del sonar, la API IUMOTE se utiliza tanto para comunicarse con el equipo subacuático como para depuración. En el caso de la baliza, la API IUMOTE sólo se utiliza para enviar mensajes de depuración. En ambos casos, la API IWCA se utiliza en la comunicación con el módem para enviar y recibir los mensajes subacuáticos.

La implementación de la aplicación asume que el módem y el equipo subacuático son capaces de comunicarse mediante una comunicación serie basada en una UART TTL a 57600 baudios con palabras de 8 bits sin paridad y sin control de flujo hardware. Además, también asume que el módem no realiza ningún tipo de reenvío automático de mensajes y dispone de un hardware de activación asíncrono basado en la recepción de señales subacuáticas. Por lo tanto, la implementación de la API IWCA en el módem tiene que compilarse usando únicamente las definiciones WIRELESS\_WAKEUP\_ENABLED, WIRELESS\_BLOQUING\_SENDING y WIRELESS\_BLOQUING\_RECEPTION. Además, la capa de abstracción serie de la API IWCA tiene que implementarse de forma que haga uso de una UART TTL con la configuración mencionada.

Al igual que las implementaciones de las API de comunicaciones, la implementación de la aplicación ha sido diseñada para conseguir la máxima portabilidad posible. Así, la implementación está dividida en 7 capas distintas tal y como se muestra en la siguiente figura.

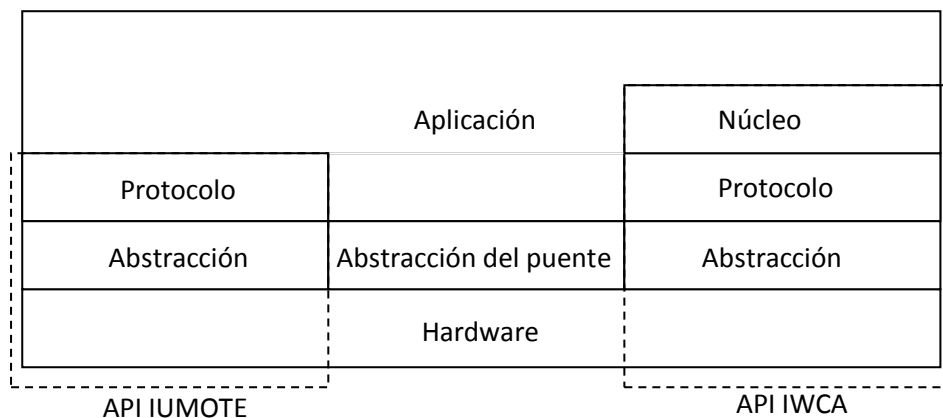


Figura 16. Implementación en capas de la aplicación de obtención de tiempos de vuelo

Como se puede observar, la mayoría de las capas son heredadas de las API de comunicaciones. Sin embargo la aplicación incluye dos nuevas capas, la capa de aplicación y la capa de abstracción del puente.

En la capa de abstracción del puente se definen todas aquellas funciones que dependen del hardware del puente. En esta capa se implementan las funciones de inicialización y configuración del hardware, las funciones de gestión del modo de bajo consumo y las funciones de gestión de los temporizadores y contadores necesarios para realizar la aplicación. Por lo tanto, esta capa tiene que implementarse de nuevo junto con las capas de abstracción de las APIs de comunicaciones si en algún momento se decide cambiar el hardware del puente.

Finalmente, en la capa de aplicación se implementa la aplicación desarrollada para la obtención de tiempos de vuelo. A continuación se explica la implementación de esta capa tanto en el sonar como en la baliza.

- **Implementación en el sonar**

La implementación de la aplicación en el sonar espera recibir órdenes del equipo subacuático. Cuando el equipo subacuático necesita obtener su localización solicita al sonar la obtención de los tiempos de vuelo de las balizas al alcance. Entonces, el sonar obtiene los tiempos de vuelo mediante el envío de señales subacuáticas. Para realizar esta tarea, la implementación sigue el diagrama de estados mostrado a continuación.

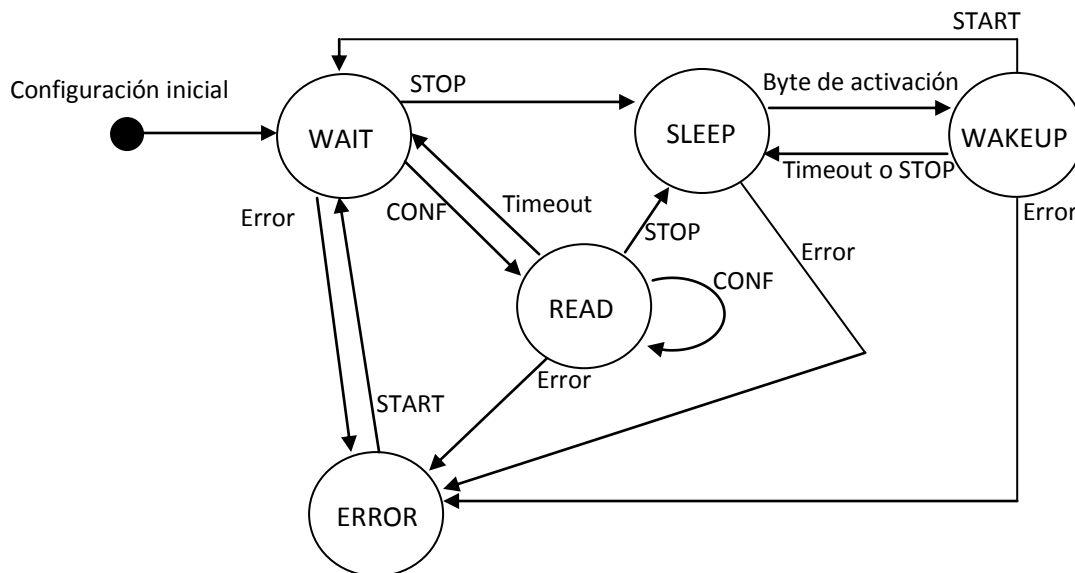


Figura 17. Diagrama de estados de la aplicación en el sonar

En el estado WAIT, la aplicación espera recibir órdenes del equipo subacuático. Si recibe un mensaje START, la aplicación envía un reconocimiento al equipo subacuático. Si recibe un mensaje STOP, desactiva el módem enviando una orden SLEEP, envía un reconocimiento al equipo subacuático y cambia al estado SLEEP. Si recibe un mensaje CONF, la aplicación pone en marcha un temporizador de recepción con el tiempo recibido en el mensaje. A continuación, envía un reconocimiento al equipo subacuático y cambia al estado READ donde se realiza el proceso de obtención de tiempos de vuelo.

En el estado SLEEP, la aplicación configura el protocolo serie de la API IUMOTE como una fuente de activación y pasa el puente al modo de bajo consumo. Como se ha comentado anteriormente, cuando el equipo subacuático quiere activar el sonar envía un byte de activación seguido de un mensaje START. Al recibir el byte de activación, el puente sale del modo de bajo consumo y la aplicación vuelve a configurar el protocolo serie para poder recibir mensajes del equipo subacuático. A continuación, la aplicación pone en marcha un temporizador de activación y cambia al estado WAKEUP.

En el estado WAKEUP, la aplicación espera recibir un mensaje START. Si lo recibe, activa el módem enviando una orden WAKEUP, envía un reconocimiento al equipo subacuático y cambia al estado WAIT. Si en lugar de recibir un mensaje START recibe un mensaje STOP, envía un reconocimiento

al equipo subacuático y cambia al estado SLEEP. Si no recibe ninguno de los dos mensajes antes de que venza el temporizador de activación, la aplicación cambia al estado de SLEEP.

En el estado READ, la aplicación realiza el proceso de obtención de los tiempos de vuelo de las balizas. En cualquier momento durante el proceso de obtención, la aplicación puede recibir órdenes del equipo subacuático. Si recibe un mensaje STOP, la aplicación desactiva el módem enviando una orden SLEEP, envía un reconocimiento al equipo subacuático y cambia al estado SLEEP. Si recibe un mensaje CONF, reinicia el temporizador de recepción con el tiempo recibido en el mensaje. A continuación, envía un reconocimiento al equipo subacuático y reinicia el proceso de obtención de tiempos de vuelo.

Si en cualquiera de los estados anteriores se produce un error, la aplicación cambia al estado ERROR donde espera recibir un mensaje START. Si lo recibe, la aplicación activa el módem enviando una orden WAKEUP, envía un reconocimiento al equipo subacuático y cambia al estado WAIT.

A continuación, se explica detalladamente el proceso de obtención de tiempos de vuelo realizado en el estado READ. La Figura 18 muestra un diagrama del proceso.

En primer lugar, la aplicación comprueba el temporizador de recepción para asegurarse de que queda tiempo para obtener los tiempos de vuelo. Si no es así, la aplicación desactiva el receptor del módem enviando una orden RECEIVER y cambia al estado WAIT. Si todavía queda tiempo para obtener los tiempos de vuelo, la aplicación comprueba si tiene que enviar la señal subacuática bien sea porque todavía no la ha enviado o porque se ha acabado el tiempo de espera para recibir respuestas de las balizas de un envío anterior.

Si la aplicación no tiene que enviar la señal subacuática significa que ya se ha enviado previamente. Por lo tanto, tiene que comprobar si ha recibido un mensaje de alguna baliza, es decir, si ha recibido un mensaje WSN\_NOTIFICATION del módem. Si es así, obtiene el valor del contador de tiempo de vuelo y espera el mensaje de datos. En cuanto recibe el mensaje de datos, la aplicación calcula el tiempo de vuelo para la baliza mediante la Ecuación 2 descrita en el apartado anterior. A continuación, envía un mensaje DATA al equipo subacuático con la dirección de la baliza, su profundidad, el tiempo de vuelo calculado y una marca de tiempo con el valor del temporizador de recepción para que el equipo subacuático pueda ordenar los distintos tiempos de vuelo recibidos. Finalmente, el proceso vuelve a ejecutarse desde el principio.

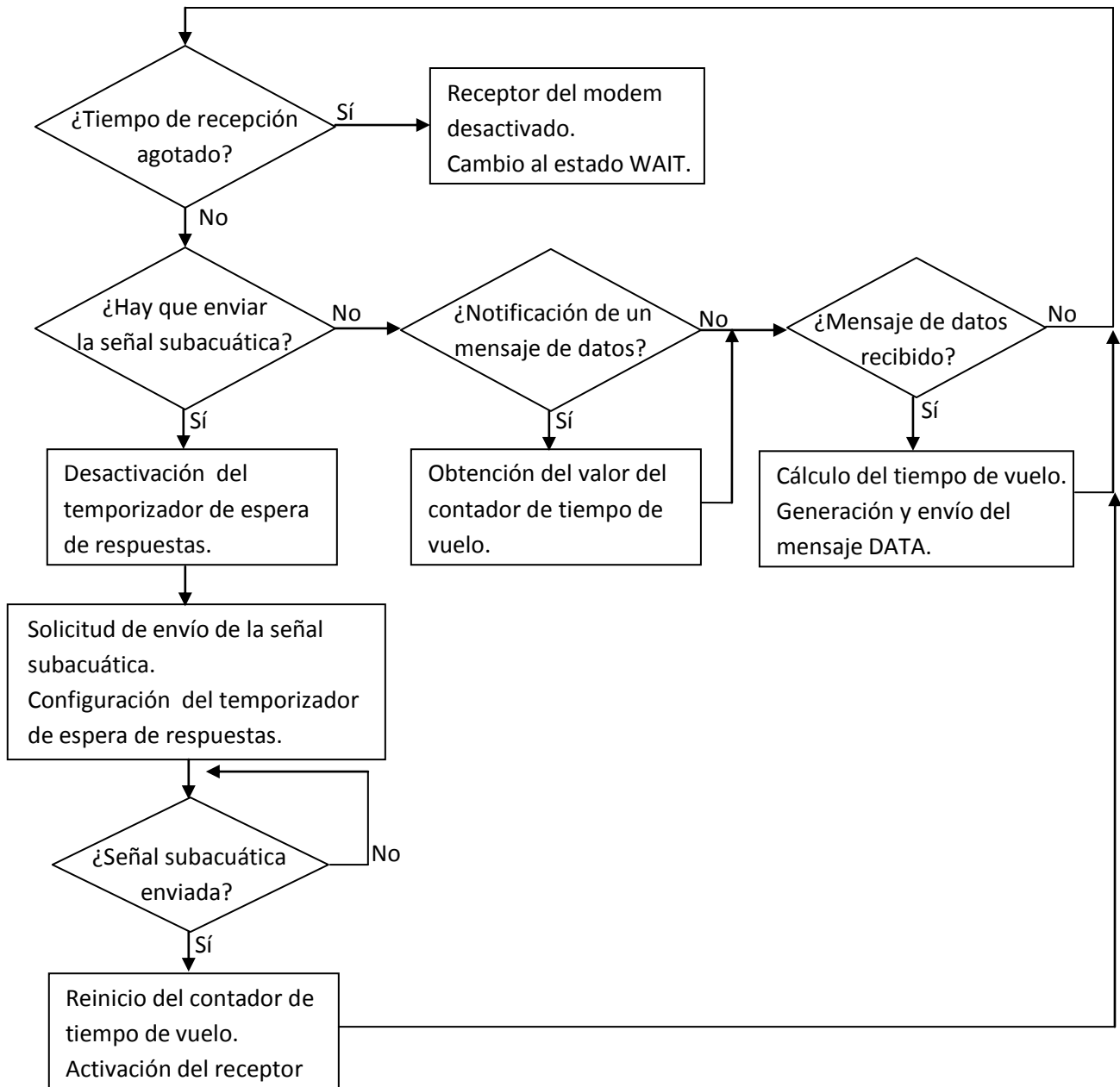


Figura 18. Diagrama de flujo del proceso de obtención de tiempos de vuelo

Si por el contrario la aplicación tiene que enviar la señal subacuática, desactiva el temporizador de espera de respuestas y envía una orden WAKEUP remota al módem para enviar la señal. A continuación, configurar el temporizador de espera de respuestas teniendo en cuenta el tamaño de la ranura TDMA y el número máximo de balizas que pueden colisionar entre ellas. Después espera recibir un mensaje WAIT del módem indicando que la señal ha sido enviada. En el momento en el que recibe el mensaje, la aplicación reinicia el contador de tiempo de vuelo y activa el receptor del módem para recibir las respuestas de las balizas enviando una orden RECEIVER. Después, envía un mensaje DEBUG al equipo subacuático indicando que se ha enviado la señal subacuática sólo en el caso de que la depuración esté activada. Finalmente, el proceso vuelve a ejecutarse desde el principio.



- **Implementación en la baliza**

La implementación de la aplicación en la baliza espera recibir una señal subacuática para, posteriormente, esperar su ranura TDMA asignada y enviar el mensaje de datos con su profundidad al sonar del equipo subacuático. Para realizar esta tarea, la implementación sigue el diagrama de estados mostrado a continuación.

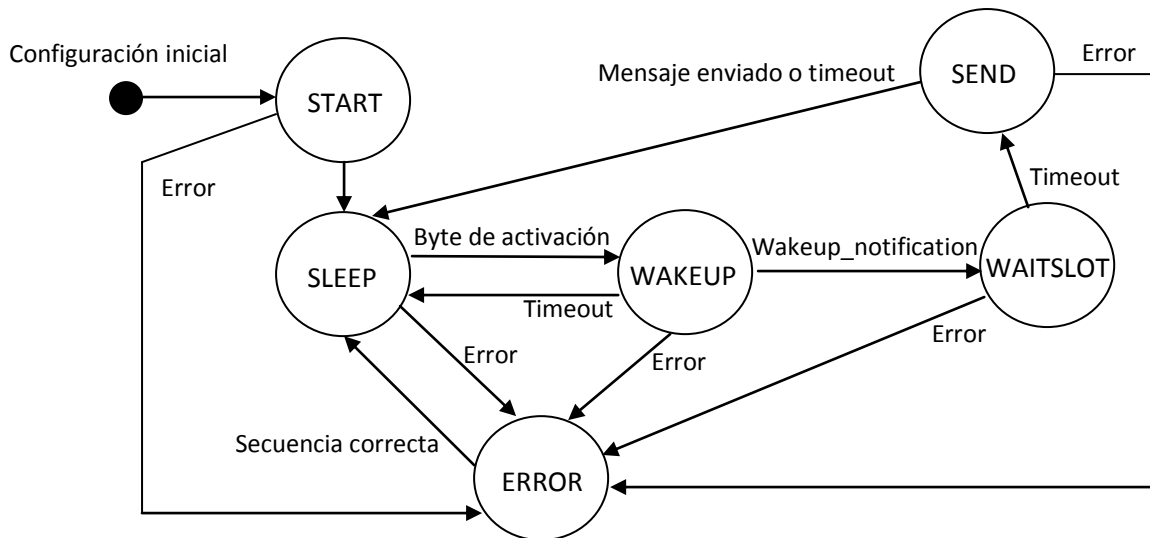


Figura 19. Diagrama de estados de la aplicación en la baliza

El estado START es el estado inicial. En este estado la aplicación obtiene la dirección MAC del módem para conocer su ranura TDMA. A continuación, desactiva el módem enviando una orden SLEEP y cambia al estado SLEEP.

En el estado SLEEP, la aplicación configura el protocolo serie de la API IWCA como una fuente de activación y pasa el puente al modo de bajo consumo. Como se ha comentado anteriormente, cuando el módem recibe una señal subacuática envía un byte de activación seguido de una orden WAKEUP\_NOTIFICATION. Al recibir el byte de activación, el puente sale del modo de bajo consumo y la aplicación vuelve a configurar el protocolo serie para poder recibir los mensajes del módem. A continuación, la aplicación pone en marcha un temporizador de activación y cambia al estado WAKEUP.

En el estado WAKEUP, la aplicación espera recibir una orden WAKEUP\_NOTIFICATION del módem antes de que venza el temporizador de activación. Si no la recibe, vuelve a cambiar al estado SLEEP. En el caso de recibir la orden, la aplicación envía un mensaje DEBUG mediante la API IUMOTE indicando que el puente se ha despertado sólo si la depuración está activada. A continuación, pone en marcha un temporizador con el tiempo necesario para esperar su ranura TDMA calculándolo mediante el último byte de la dirección MAC obtenida en el estado START. Acto seguido, cambia al estado WAIT SLOT.

En el estado WAIT SLOT, la aplicación espera a que venza el temporizador de espera de la ranura. Una vez acabada la espera, la aplicación pone en marcha un temporizador de envío con el tiempo máximo permitido para enviar el mensaje de datos y cambia al estado SEND.

En el estado SEND, la aplicación comprueba que el canal está libre obteniendo del módem el valor RSSI del canal mediante una orden READ\_VALUE. Si el canal está libre, la aplicación obtiene la profundidad de la baliza y la envía en un mensaje de datos al módem para que éste la envíe al sonar mediante un mensaje inalámbrico. Como en este caso no se dispone de un sensor de presión, la profundidad se obtiene a modo de secuencia empezando por cero. De esta forma se da soporte a la depuración permitiendo identificar si se pierde algún mensaje en la comunicación.

Si se consigue enviar el mensaje de datos, la aplicación envía un mensaje DEBUG mediante la API IUMOTE indicando la profundidad (que en este caso actúa como número de secuencia), desactiva el módem enviando una orden SLEEP y cambia al estado SLEEP. Si finalmente vence el temporizador de envío sin enviar ningún mensaje, la aplicación desactiva el módem enviando una orden SLEEP y cambia al estado SLEEP.

Tal y como se ha comentado anteriormente, la aplicación de la baliza no necesita contar el tiempo que tarda en enviar el mensaje de datos ya que de ello se ocupa el módem. Cada vez que el módem sale del modo de bajo consumo, su reloj del sistema se reinicia. De esta forma, la marca de tiempo que incluye el módem en el primer mensaje de datos inalámbrico, tal y como lo define la API IWCA, es en realidad el tiempo que tarda la baliza desde que recibe la señal subacuática hasta que envía el mensaje de datos.

Por último, si en cualquiera de los estados anteriores se produce un error, la aplicación cambia al estado ERROR donde intenta volver a un estado estable conocido. Para ello, la aplicación intenta activar el módem enviando una orden WAKEUP y, a continuación, intenta dormirlo enviando una orden SLEEP. Si la secuencia tiene éxito, la aplicación cambia al estado SLEEP.

- **Problemas con el bajo consumo**

Aunque la aplicación está preparada para utilizar el sistema de activación asíncrono del módem, hay que tener en cuenta el bajo consumo a la hora de obtener los tiempos de vuelo. Cuando el microcontrolador y todos sus periféricos entran en modo de bajo consumo, el tiempo sólo puede medirse mediante un reloj externo. Como ya se ha comentado anteriormente, el módem no dispone de ningún pin de entrada / salida libre para añadir un reloj externo, por lo tanto se debería usar su RTC interno. Sin embargo, el RTC interno del microcontrolador tiene demasiada dispersión y se debería sincronizar con el reloj del puente del sonar para evitar que añada un error variable en la obtención del tiempo de vuelo. Por este motivo se ha decidido no hacer uso del sistema de activación asíncrono del módem en esta primera implementación de la aplicación. Esta decisión tiene consecuencias en la implementación de la aplicación en la baliza.

La implementación de la aplicación en la baliza tiene que tener en cuenta que el módem no puede entrar en modo de bajo consumo con el fin de poder recibir la orden de activación inalámbrica. Por este motivo se ha incorporado la

definición WIRELESS\_WAKEUP\_SIGNAL en la implementación de la aplicación. Cuando se hace uso de esta definición, la aplicación asume que el módem hace uso del sistema de activación asíncrono y, por lo tanto, gestionará el modo de bajo consumo del módem tal y como se ha explicado anteriormente. En el caso de que no se haga uso de ella, el estado ERROR no puede volver a un estado estable y, por lo tanto, se convierte en un bucle infinito del que sólo se puede salir al reiniciar la baliza. Además en lugar de desactivar y activar el módem, la aplicación cambia el estado de su receptor mediante el envío de una orden RECEIVER. Así, la aplicación enciende el receptor del módem en lugar de desactivarlo y apaga el receptor en lugar de activarlo.

## 5. Experimentación y resultados

Para afinar la implementación desarrollada y reducir el error en la obtención de los tiempos de vuelo, es necesario realizar experimentos para obtener los distintos parámetros temporales involucrados en el proceso de obtención del tiempo de vuelo y en la definición del tamaño de las ranuras TDMA. Además, es importante realizar una campaña de pruebas para poder evaluar la bondad de la implementación realizada.

En el laboratorio se dispone de un acuario de 2,4 metros de largo y 3 sistemas de comunicación basados en el módem ITACA-S1000, es decir, 3 puentes STM32F4DISCOVERY, 3 módems ITACA-S1000 y 3 transductores piezoeléctricos Hummimbird XP 9 20.

Para realizar los experimentos se ha construido un montaje en el acuario en el cual dos de estos sistemas actúan como balizas y el tercero como sonar. Los transductores de las balizas se han montado uno encima del otro en un lado del acuario y el transductor del sonar se ha montado en el lado contrario a 2 m de distancia. Además, los módems de las balizas y del sonar se han configurado mediante la aplicación de demostración de la API IWCA para que tengan la siguiente configuración:

- Baliza 1:
  - Dirección MAC: 0xA0000001
  - Ranura TDMA asignada: ranura 1
  - Patrón de activación: 0xFF
  
- Baliza 2:
  - Dirección MAC: 0xA0000002
  - Ranura TDMA asignada: ranura 2
  - Patrón de activación: 0xFF
  
- Sonar:
  - Dirección MAC: 0xA0000101
  - Sin patrón de activación. Sistema de activación inalámbrica desactivado.

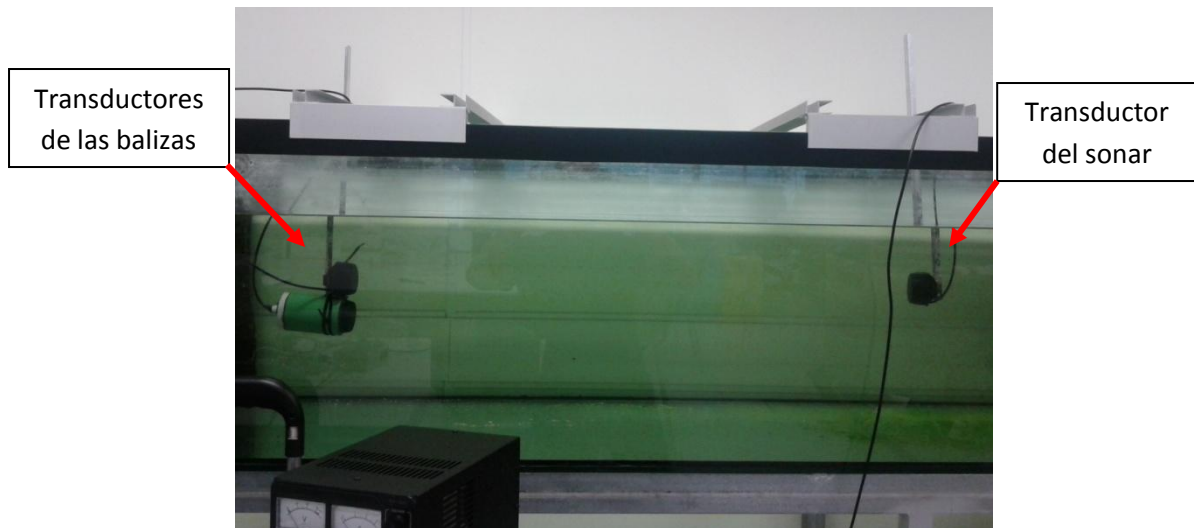


Figura 20. Montaje en el acuario para realizar los experimentos

A continuación se describen los experimentos realizados para obtener los distintos parámetros temporales involucrados en el proceso de obtención del tiempo de vuelo.

#### **5.1.1. Caracterización de los tiempos para la obtención del tiempo de vuelo.**

La sincronización de los relojes del puente del sonar y el módem de la baliza es un parámetro crítico a la hora de obtener el tiempo de vuelo. Además, también es crítico caracterizar los intervalos de tiempo que la implementación actual no es capaz de tener en cuenta durante el proceso de obtención del tiempo de vuelo. La caracterización de estos intervalos no puede realizarse en función de ninguno de los relojes del sistema de comunicación, ya que si así fuera la implementación podría tenerlos en cuenta. Por otro lado, la sincronización de los relojes va a realizarse manualmente ya que primero es necesario investigar qué algoritmos de sincronización se utilizan en el medio subacuático, tarea que se plantea como trabajo futuro. Por lo tanto, en ambos casos hay que utilizar un reloj global de referencia para realizar dichas tareas.

Para realizar este conjunto de experimentos se dispone de un osciloscopio AGILENT DSO6014A el cual se va a utilizar como reloj global. En cada experimento se va a modificar el firmware de los dispositivos para permitir que se utilice uno de los pines de entrada/salida para obtener los tiempos con el osciloscopio. En el caso del módem, el pin a utilizar es el pin TX de la comunicación serie. En el caso del puente, el pin a utilizar es el pin 7 del puerto B de la placa STM32F4DISCOVERY.

A continuación se detallan los experimentos realizados para obtener cada parámetro temporal involucrado en el proceso de obtención de los tiempos de vuelo.

- **Obtención del tiempo necesario para que el módem envíe el mensaje serie WAIT ( $T_{wait}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que envíe continuamente el mensaje serie WAIT cada 10 milisegundos. Utilizando el osciloscopio puede obtenerse el tiempo total del envío mediante el pin TX del módem. Hay que tener en cuenta que el bit de stop de la comunicación serie es a nivel alto y, por lo tanto, en el último byte se confunde con el estado *idle* de la línea.

**Primer start bit**

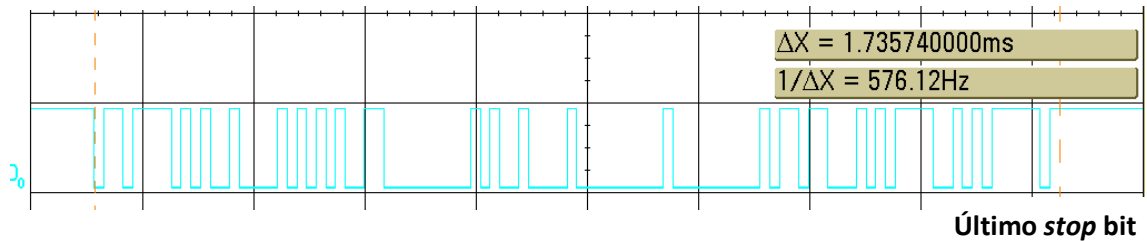


Figura 21. Envío del mensaje serie obtenido en el osciloscopio.

El experimento consiste en conectar cada uno de los módems y obtener el tiempo con el osciloscopio.

Los resultados del envío del mensaje WAIT se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	1735,8	1	1735,9	1	1735,8
2	1735,5	2	1735,6	2	1735,8
3	1735,5	3	1735,6	3	1735,5
4	1735,8	4	1735,6	4	1735,8
5	1735,8	5	1735,9	5	1735,8

Tabla 3. Tiempo de envío de los mensajes WAIT

Realizando la media para cada caso y redondeando los decimales, los resultados coinciden con los resultados teóricos que se debería obtener. Así, el tiempo de envío de un mensaje WAIT es de 1736 microsegundos.

- **Obtención del tiempo necesario para que el módem envíe el mensaje serie WSN\_NOTIFICATION ( $T_{notificación}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que envíe continuamente el mensaje serie WSN\_NOTIFICATION cada 10 milisegundos. Utilizando el osciloscopio puede obtenerse el tiempo total del envío mediante el pin TX del módem. Hay que tener en cuenta que el bit de stop de la comunicación serie es a nivel alto y, por lo tanto, en el último byte se confunde con el estado *idle* de la línea tal y como se muestra en la Figura 21.

El experimento consiste en conectar cada uno de los módems y obtener el tiempo con el osciloscopio.

Los resultados del envío del mensaje WSN\_NOTIFICATION se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	1562,4	1	1562,3	1	1562,4
2	1562,3	2	1562,3	2	1562,4
3	1562,3	3	1562,2	3	1562,4
4	1562,4	4	1562,3	4	1562,4
5	1562,4	5	1562,3	5	1562,4

Tabla 4. Tiempo de envío de los mensajes WSN\_NOTIFICATION.

Realizando la media para cada caso y redondeando los decimales, los resultados coinciden con los resultados teóricos que se debería obtener. Así, el tiempo de envío de un mensaje WSN\_NOTIFICATION es de 1562 microsegundos.

- **Obtención del tiempo necesario para que el módem procese el mensaje de activación inalámbrico recibido ( $T_{módem\_not}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que el pin TX de la comunicación serie pueda utilizarse para medir el tiempo de procesamiento del mensaje de activación inalámbrico recibido. Para ello, en primer lugar se establece el pin TX a nivel bajo justo al comienzo del estado de recepción. Una vez que se recibe el mensaje de activación, el pin TX se establece a nivel alto. Finalmente, vuelve a establecerse a nivel bajo cuando el mensaje de activación ha sido procesado y justo antes de que se envíe el mensaje WAKUP\_NOTIFICATION. De esta forma se puede medir con el osciloscopio el tiempo necesario para procesar el mensaje de activación midiendo el tiempo que el pin TX permanece a nivel alto.

El experimento consiste en utilizar la aplicación de demostración de la API IWCA para activar el receptor del módem a evaluar y, a continuación, enviar un mensaje de activación mediante otro módem. El experimento se realiza con cada uno de los módems disponibles conectando el osciloscopio al pin TX.

Los resultados del experimento se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	436	1	426	1	434
2	436	2	426	2	436
3	436	3	426	3	436
4	436	4	426	4	434
5	435	5	426	5	436

Tabla 5. Tiempo de procesamiento del mensaje de activación inalámbrico

Realizando la media y redondeando los decimales, el tiempo de procesamiento del mensaje de activación inalámbrico en el módem es de 432 microsegundos.

- **Obtención del tiempo necesario para que el módem procese el mensaje de datos inalámbrico recibido ( $T_{módem\_datos}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que el pin TX de la comunicación serie pueda utilizarse para medir el tiempo de procesamiento del mensaje de datos inalámbrico recibido. Para ello, en primer lugar se establece el pin TX a nivel bajo justo al comienzo del estado de recepción. Una vez que se recibe el mensaje de datos, el pin TX se establece a nivel alto. Finalmente, vuelve a establecerse a nivel bajo cuando el mensaje de datos ha sido procesado y justo antes de que se envíe el mensaje WSN\_NOTIFICATION. De esta forma se puede medir con el osciloscopio el tiempo necesario para procesar el mensaje de datos midiendo el tiempo que el pin TX permanece a nivel alto.

El experimento consiste en utilizar la aplicación de demostración de la API IWCA para activar el receptor del módem a evaluar y, a continuación, enviar un mensaje de datos inalámbrico mediante otro módem. El experimento se realiza con cada uno de los módems disponibles conectando el osciloscopio al pin TX.

Los resultados del experimento se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	420	1	412	1	420
2	420	2	412	2	420
3	420	3	412	3	420
4	418	4	412	4	420
5	420	5	412	5	420

Tabla 6. Tiempo de procesamiento del mensaje de datos inalámbrico

Realizando la media y redondeando los decimales, el tiempo de procesamiento del mensaje de datos inalámbrico en el módem es de 417 microsegundos.

- **Obtención del tiempo necesario para que el módem envíe el mensaje de datos inalámbrico ( $T_{msg\_datos}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que envíe continuamente un mensaje de datos inalámbrico cada 500 milisegundos permitiendo, además, hacer uso del pin TX para que se pueda medir el tiempo total de envío de cada mensaje. Para ello, en primer lugar el pin TX se establece a nivel bajo al comienzo de la ejecución. A continuación, se

establece a nivel alto justo antes de enviar el mensaje de datos inalámbrico. Una vez que el mensaje de datos ha sido enviado, el pin TX vuelve a establecerse a nivel bajo. Acto seguido, vuelve a establecerse a nivel alto durante 250 milisegundos cuando la ejecución vuelve al programa principal después de ejecutar la función de envío. Finalmente se mantiene a nivel bajo otros 250 milisegundos antes de volver a enviar el mensaje de datos. De esta forma se puede obtener el tiempo de envío del mensaje de datos midiendo el tiempo que el pin TX permanece a nivel alto por primera vez.

El experimento consiste en conectar cada uno de los módems y obtener el tiempo con el osciloscopio.

Los resultados del experimento se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	200220	1	200220	1	200220
2	200220	2	200220	2	200220
3	200220	3	200220	3	200217
4	200220	4	200220	4	200217
5	200220	5	200220	5	200220

Tabla 7. Tiempo de envío del mensaje de datos inalámbrico.

Realizando la media y redondeando los decimales, el tiempo de envío del mensaje de datos inalámbrico enviado por el módem es de 200220 microsegundos.

- **Obtención del tiempo necesario para que el módem empiece el envío del mensaje WAIT después de enviar el mensaje de activación inalámbrico ( $T_{módem\_wait}$ ).**

Para realizar este experimento se ha modificado el firmware del módem para que envíe continuamente un mensaje de activación inalámbrico cada 500 milisegundos permitiendo, además, hacer uso del pin TX para que se pueda medir el tiempo que el módem necesita para empezar el envío del mensaje WAIT después de enviar cada mensaje. Para ello, en primer lugar el pin TX se establece a nivel bajo al comienzo de la ejecución. A continuación, se establece a nivel alto justo antes de enviar el mensaje de activación inalámbrico. Una vez que el mensaje de activación ha sido enviado, el pin TX vuelve a establecerse a nivel bajo. Acto seguido, vuelve a establecerse a nivel alto durante 250 milisegundos cuando la ejecución vuelve al programa principal después de ejecutar la función de envío. Finalmente se mantiene a nivel bajo otros 250 milisegundos antes de volver a enviar el mensaje de activación. De esta forma se puede obtener el tiempo necesario para empezar el envío del mensaje WAIT, midiendo el tiempo que el pin TX permanece a nivel bajo después de permanecer a nivel alto durante el envío del mensaje de activación.



El experimento consiste en conectar cada uno de los módems y obtener el tiempo con el osciloscopio.

Los resultados del experimento se muestran a continuación.

Baliza 1		Baliza 2		Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	15992,6	1	15987,6	1	15985,2
2	15987,6	2	15990	2	15990
3	15985	3	15987,6	3	15987,6
4	15990	4	15992,6	4	15990
5	15992,4	5	15992,6	5	15990

Tabla 8. Tiempo de procesamiento  $T_{módem\_wait}$ .

Realizando la media y redondeando los decimales, el tiempo que necesita el módem para empezar el envío el mensaje WAIT después de enviar el mensaje de activación es de 15989 microsegundos.

- **Obtención del tiempo necesario para que el puente procese el mensaje WAIT recibido ( $T_{puente\_wait}$ ).**

Para realizar este experimento se ha modificado el firmware del puente para que se pueda medir el tiempo de procesamiento del mensaje serie WAIT recibido después de que el módem envíe el mensaje de activación. Para conseguirlo es necesario utilizar dos pines del puente al mismo tiempo, el pin 3 del puerto A y el pin 7 del puerto B. En primer lugar, se establece el pin 7 a nivel alto justo antes de enviar el mensaje WAKEUP remoto al módem. Una vez que el puente recibe el mensaje WAIT, el pin 7 se establece a nivel bajo. Utilizando el pin 3, el cual está conectado al pin TX del módem, se pueden ver con el osciloscopio todos los mensajes que el módem envía en el proceso de envío del mensaje de activación. Por lo tanto, se puede medir el tiempo que tarda el puente en procesar el mensaje WAIT midiendo el tiempo desde el final del último mensaje enviado por el módem hasta que el pin 7 pasa a nivel bajo. Hay que tener en cuenta que, en el pin 3, el bit de *stop* del último byte del último mensaje puede confundirse con el estado *idle* de la línea tal y como se muestra en la Figura 21.

El experimento consiste en conectar un módem al puente del sonar y el osciloscopio al pin 3 del puerto A y al pin 7 del puerto B del puente. A continuación se envía un mensaje de activación mediante un mensaje CONF haciendo uso de la aplicación de demostración de la API IUMOTE.

Los resultados del experimento se muestran a continuación.

Sonar	
Núm.	Tiempo(us)
1	7,6
2	9,6
3	8,6
4	7,6
5	7,6
6	7,6
7	8,6
8	9,6
9	8,6
10	7,6

Tabla 9. Tiempo de procesamiento del mensaje serie WAIT

Realizando la media y redondeando los decimales, el tiempo de procesamiento del mensaje WAIT en el puente es de 8 microsegundos.

- **Obtención del tiempo necesario para que el puente procese el mensaje WSN\_NOTIFICATION recibido ( $T_{puente\_not}$ ).**

Para realizar este experimento se ha modificado el firmware del puente para que se pueda medir el tiempo de procesamiento del mensaje serie WSN\_NOTIFICATION recibido después de que el módem reciba un mensaje de datos inalámbrico. Para conseguirlo es necesario utilizar dos pines del puente al mismo tiempo, el pin 3 del puerto A y el pin 7 del puerto B. En primer lugar, se establece el pin 7 a nivel bajo justo al comienzo de la ejecución de la aplicación. Una vez que el puente recibe el mensaje WSN\_NOTIFICATION, el pin 7 se establece a nivel alto. Utilizando el pin 3, el cual está conectado al pin TX del módem, se pueden ver con el osciloscopio todos los mensajes que el módem envía en la recepción de un mensaje de datos inalámbrico. Por lo tanto, se puede medir el tiempo que tarda el puente en procesar el mensaje WSN\_NOTIFICATION midiendo el tiempo desde el final del último mensaje enviado por el módem hasta que el pin 7 pasa a nivel alto. Hay que tener en cuenta que, en el pin 3, el bit de *stop* del último byte del último mensaje puede confundirse con el estado *idle* de la línea tal y como se muestra en la Figura 21.

El experimento consiste en conectar un módem al puente del sonar y otro al puente de una baliza. Después, se conecta el osciloscopio al pin 3 del puerto A y al pin 7 del puerto B del puente del sonar. A continuación se envía un mensaje de activación mediante un mensaje CONF haciendo uso de la aplicación de demostración de la API IUMOTE para obtener la respuesta de la baliza.

Los resultados del experimento se muestran a continuación.

Sonar	
Núm.	Tiempo(us)
1	8,6
2	7,6
3	5,6
4	9,6
5	11,6
6	7,6
7	9,6
8	7,6
9	7,6
10	8,6

Tabla 10. Tiempo de procesamiento del mensaje serie WSN\_NOTIFICATION

Realizando la media y redondeando los decimales, el tiempo de procesamiento del mensaje WSN\_NOTIFICATION en el puente es de 8 microsegundos.

Finalmente se explica detalladamente el experimento llevado a cabo para sincronizar manualmente los relojes del sonar y de la baliza.

- **Sincronización manual de los relojes del sonar y de la baliza**

Del estado del arte se ha concluido que la sincronización de los relojes en la obtención del tiempo de vuelo es un parámetro muy crítico. La implementación de un algoritmo de sincronización de relojes queda fuera del presente trabajo ya que no sólo supone realizar la implementación y probarla sino que, además, hay que realizar una búsqueda bibliográfica previa para conocer los algoritmos de sincronización utilizados en el medio subacuático. Como hasta ahora se han caracterizado los tiempos con el osciloscopio utilizándolo de reloj global, la sincronización va a realizarse de forma manual utilizando también el osciloscopio.

Para realizar la sincronización manual con el osciloscopio ha sido necesario modificar tanto el firmware del módem de la baliza como el firmware del puente del sonar. El firmware del módem se ha modificado de forma que el pin TX se establece a nivel bajo justo antes de obtener la marca de tiempo enviada en el mensaje de datos. Después, una vez que se ha obtenido la marca de tiempo, el pin TX se establece a nivel alto hasta que se copia en el mensaje de datos y, a continuación, vuelve a establecerse a nivel bajo. De esta forma se puede medir con el osciloscopio el tiempo que el módem debería enviar en su marca de tiempo midiendo el tiempo desde que se envía el byte de activación, el cual puede identificarse como una ristra de unos y ceros alternándose, hasta que pin TX se establece a nivel bajo.

Por otro lado, el firmware del puente se ha modificado de forma que el pin 7 del puerto B se establece a nivel bajo cuando inicia su ejecución. Después, cada vez que el puente envía un mensaje de activación, el pin 7 se establece a nivel alto justo después de que se active el contador de tiempo de vuelo. Por último, cuando se recibe un mensaje de datos, se obtiene el valor del contador y se establece el pin 7 a nivel bajo. De esta forma se puede medir con el osciloscopio el tiempo que debería contar el puente midiendo el tiempo que permanece el pin 7 a nivel alto.

El procedimiento de sincronización manual consiste en enviar un mensaje de activación mediante el sonar y esperar la respuesta de la baliza. Mediante el depurador incorporado en el hardware del puente (ST-LINK/V2) se obtiene la cuenta realizada por el sonar al recibir el mensaje de datos de la baliza. Por otro lado, mediante la marca de tiempo incorporada en el mensaje de datos de la baliza se obtiene la cuenta realizada por la baliza. Al mismo tiempo, mediante el osciloscopio se obtienen los tiempos que deberían contar tanto el sonar como la baliza. De esta forma, se puede corregir el desfase en la cuenta de los dispositivos obteniendo la relación entre los tiempos globales y los tiempos contados.

Los resultados del proceso se muestran a continuación.

Baliza 1			
Núm.	Tiempo global	Tiempo contado	Relación
1	355044	355121	1,000217
2	355636	355717	1,000228
3	354962	355042	1,000225
4	354992	355063	1,000200
5	354956	355019	1,000177
6	355010	355077	1,000189
7	354990	355062	1,000203
8	354956	355020	1,000180
9	354974	355042	1,000192
10	354990	355062	1,000203

Baliza 2			
Núm.	Tiempo global	Tiempo contado	Relación
1	355686	355766	1,000225
2	354992	355062	1,000197
3	354992	355060	1,000192
4	355946	356019	1,000205
5	354992	355060	1,000192
6	354942	355005	1,000177
7	355008	355063	1,000155
8	355720	355774	1,000152
9	355634	355707	1,000205
10	355670	355734	1,000180

Sonar			
Núm.	Tiempo global	Tiempo contados	Relación
1	542455	542509	1,000100
2	543040	543098	1,000107
3	542370	542425	1,000101
4	542660	542715	1,000101
5	542470	542526	1,000103
6	542306	542359	1,000098
7	542488	542545	1,000105
8	542368	542425	1,000105
9	542386	542443	1,000105
10	542424	542478	1,000100

	Promedio relación
Sonar	1,00010249
Baliza 1	1,00020137
Baliza 2	1,00018797

Promedio balizas	1,00019467
------------------	------------

Tabla 11. Resultados de la sincronización manual de relojes

Finalmente, la sincronización de los relojes se realiza cuando el sonar calcula el tiempo de vuelo. En el caso del reloj del sonar, la sincronización se realiza dividiendo el valor del contador de tiempo de vuelo por el promedio de la relación calculada para el sonar. En el caso del reloj de la baliza, la sincronización se realiza dividiendo el valor de la marca de tiempo obtenida del mensaje de datos de la baliza por el promedio de la relación calculada para las balizas. Aún con esta sincronización manual, hay que tener presente que se produce un pequeño error del orden de unidades de microsegundos.

### 5.1.2. Ajuste de la ranura TDMA

El tamaño de la ranura TDMA es un parámetro crítico a la hora de configurar la aplicación para un escenario en particular dado que tiene un impacto directo sobre el consumo de energía. Sin embargo, el consumo de energía no va a ser un parámetro a tener en cuenta durante los experimentos ya que actualmente la aplicación no puede ofrecer una solución óptima de bajo consumo. Por este motivo, el ajuste de la ranura TDMA se va a realizar para facilitar los experimentos con la aplicación. Por lo tanto, en vez de ajustar la ranura minimizando su tamaño lo máximo posible para un escenario en particular, se va a ajustar la ranura dejando holgura suficiente para que se puedan realizar varios experimentos en distintos escenarios sin que sea necesario cambiar la implementación de la aplicación.

El tamaño de la ranura TDMA viene dado por la Ecuación 1 descrita en el punto 4.6.2. Por lo tanto es necesario obtener el tiempo de envío del mensaje de datos de la baliza y su tiempo de vuelo. Además hay que tomar una decisión sobre el tiempo de procesamiento de la baliza ya que depende tanto de la ocupación del canal como de la comunicación serie.

Como el tiempo de vuelo del mensaje depende de la velocidad del sonido y de la distancia de las balizas, no hay más remedio que obtenerlo teóricamente. Los experimentos que se pretenden realizar a corto plazo van a realizarse tanto

en piscina como en mar. Como la velocidad del sonido es más lenta en agua dulce que en agua salada se va a utilizar la velocidad del sonido en el agua dulce, 1495 m/s [44], para que la ranura TDMA sirva para ambos escenarios. Además, en los experimentos se pretende ir ampliando la distancia de las balizas hasta llegar al máximo alcance del módem, unos 100 m. Por lo tanto, el tiempo de vuelo del mensaje es de:

$$T_{vuelo} = \frac{100 \text{ m}}{1495 \text{ m/s}} = 0,0668896 \text{ s} \approx 67 \text{ ms}$$

El tiempo de envío del mensaje de datos de la baliza ya ha sido obtenido en un experimento anterior. Su valor es de 200 milisegundos si se redondean los decimales, siendo además idéntico al valor teórico. Sin embargo, el módem envía dos bytes adicionales de sincronización después del envío de cada mensaje. Por lo tanto hay que sumar el tiempo teórico de envío de estos dos últimos bytes.

$$T_{módem\_2bytes} = 1 \text{ kbps} * 16 \text{ bits} = 0,016 \text{ s} = 16 \text{ ms}$$

$$T_{transmisión} = 200 \text{ ms} + 16 \text{ ms} = 216 \text{ ms}$$

En cuanto al tiempo de procesamiento de la baliza, se espera que en una situación normal la deriva entre ranuras pueda alargar el envío del mensaje de una baliza y acabar dentro de la ranura de otra. Por lo tanto, es posible que en una ranura el canal esté ocupado y la baliza no pueda enviar su mensaje durante unos pocos milisegundos. Además, hay una posibilidad de que el módem pueda perder el mensaje de datos serie enviado por la baliza. Como consecuencia, la baliza necesitará un tiempo para permitirle realizar reenvíos de mensajes. Por este motivo se ha decidido que el tiempo de procesamiento máximo de la baliza sea de ¼ del tiempo de envío de un mensaje de datos, es decir, de 50 milisegundos.

Con estos tiempos y aplicando la Ecuación 1, el tamaño de la ranura TDMA es de:

$$T_{ranura} = 50 \text{ ms} + 216 \text{ ms} + 67 \text{ ms} = 333 \text{ ms}$$

Por último, el sonar necesita esperar las respuestas de las balizas teniendo en cuenta las ranuras TDMA. Como en la actualidad sólo se dispone de dos balizas para realizar los experimentos, el tiempo de espera de respuestas del sonar es de 2 ranuras TDMA más una holgura para evitar que el último mensaje se pierda por la posible deriva de las ranuras. Además hay que tener en cuenta que el sonar empieza a contar el tiempo desde que envía el mensaje de activación. Por lo tanto, al tiempo anterior hay que sumarle el tiempo de vuelo del mensaje de activación.

$$T_{espera\_respuestas} = 67 \text{ ms} + 333 \text{ ms} * 2 + 23 \text{ ms} = 750 \text{ ms}$$

### 5.1.3. Campaña de obtención de tiempos de vuelo

Con todos los tiempos caracterizados y la ranura TDMA ajustada, la aplicación está lista para dar soporte a la localización de los equipos subacuáticos. Sin embargo, es necesario evaluar la bondad de la implementación realizada. Para ello se ha llevado a cabo una campaña de experimentos en el laboratorio utilizando el acuario y el montaje explicado al principio de este apartado.

La campaña consiste en obtener los tiempos de vuelo para 2 balizas modificando la distancia que las separa del sonar. En la campaña se ha llevado a cabo un total de 4 experimentos en los cuales el sonar envía 1500 mensajes de activación. En el primer experimento las balizas están separadas del sonar exactamente 2 metros, en el segundo 1.5 metros, en el tercero 1 metro y en el cuarto 0.5 metros. De esta forma se pretende obtener el error cometido por la aplicación al obtener los tiempos de vuelo y comprobar si existe alguna correlación con la distancia.

Durante los experimentos, tanto las balizas como el sonar se han conectado a un PC para obtener sus mensajes de depuración mediante la aplicación de demostración de la API IUMOTE. En el caso del sonar, además, la aplicación de demostración se ha usado para simular el equipo subacuático, configurando el sonar para enviar los mensajes de activación y obteniendo así los mensajes de datos provenientes de las balizas.

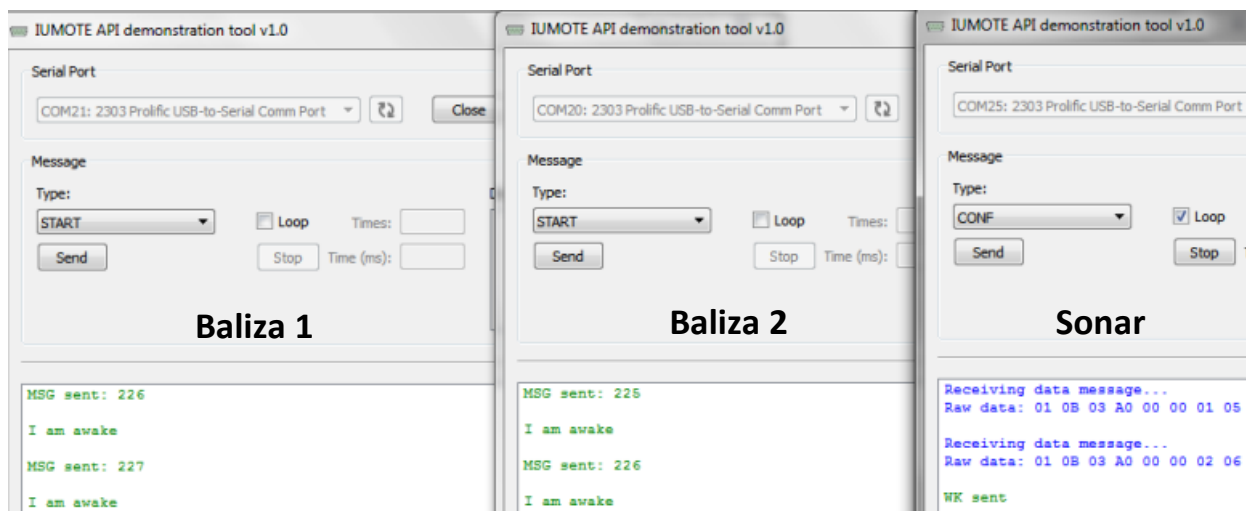


Figura 22. Campaña de prueba

Para automatizar el análisis de los registros almacenados por la aplicación de demostración se ha realizado un programa en python. Este programa obtiene como parámetros de entrada la carpeta donde se encuentran los registros de la aplicación y la velocidad del sonido. Como salida, el programa genera un fichero para cada una de las balizas que contiene los tiempos de vuelo obtenidos y la distancia calculada.

Al igual que en el ajuste de la ranura TDMA, se ha decidido utilizar una estimación de la velocidad del sonido de 1495 m/s en el cálculo de las distancias. Esta velocidad del sonido es la misma que la obtenida en la experimento en piscina realizado en [44].

A continuación se muestran y analizan los resultados obtenidos en cada uno de los experimentos.

- **Experimento 1: 2 metros de distancia**

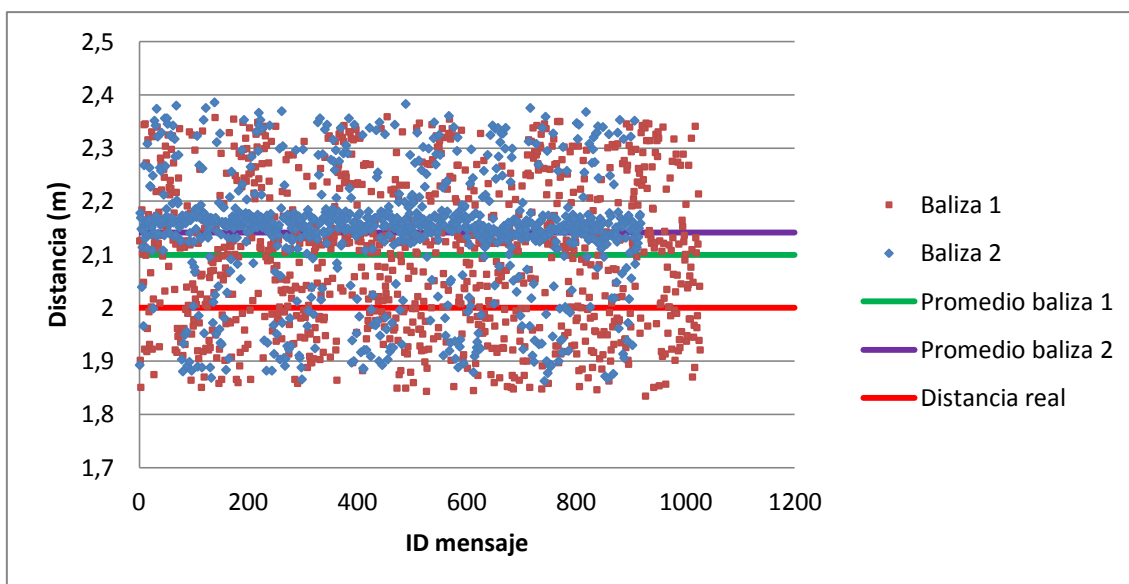


Figura 23. Resultados del experimento 1 de la campaña de pruebas.

	Baliza 1 (m)	Baliza 2 (m)
Promedio	2,099	2,141
Desviación típica	0,138	0,111
Error medio	0,099(4,73%)	0,141(6,60%)
Error máximo	0,359(17,11%)	0,386(18,03%)

Tabla 12. Resultados de medida de distancias del experimento 1

En ambos casos el error medio no es superior al 7%. Sin embargo, la baliza 1 consigue un error menor que la baliza 2. Además, puede observarse que la dispersión de los resultados obtenidos es bastante elevada, llegando a obtenerse un error máximo de un 17,11% para la baliza 1 y un 18,03% para la baliza 2.



- **Experimento 2: 1.5 metros de distancia**

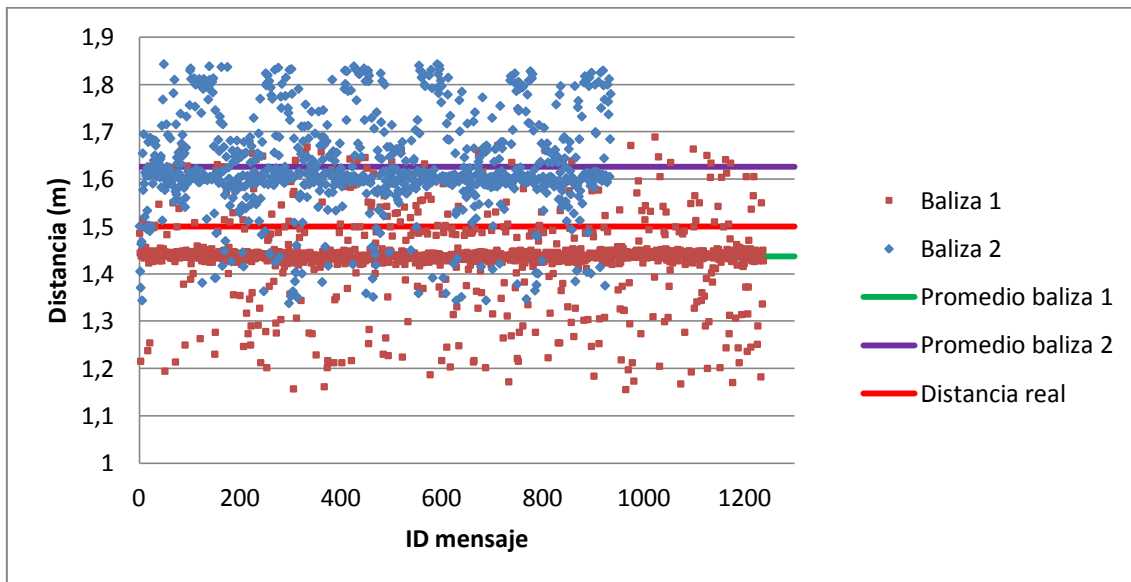


Figura 24. Resultados del experimento 2 de la campaña de pruebas.

	Baliza 1 (m)	Baliza 2 (m)
Promedio	1,437	1,626
Desviación típica	0,078	0,102
Error medio	0,063(4,37%)	0,126(7,75%)
Error máximo	0,344(23,96%)	0,343(21,11%)

Tabla 13. Resultados de medida de distancias del experimento 2

En ambos casos el error medio no es superior al 8%. Sin embargo, la baliza 1 consigue un error menor que la baliza 2. Además, puede observarse que la dispersión de los resultados obtenidos es bastante elevada, llegando a obtenerse un error máximo de un 23,96% para la baliza 1 y un 21,11% para la baliza 2.

Comparado con los resultados del experimento anterior, el error de la baliza 2 ha aumentado en aproximadamente un 1% y, en ambos casos, la dispersión sigue siendo elevada y el error máximo ha aumentado.

- **Experimento 3: 1 metros de distancia**

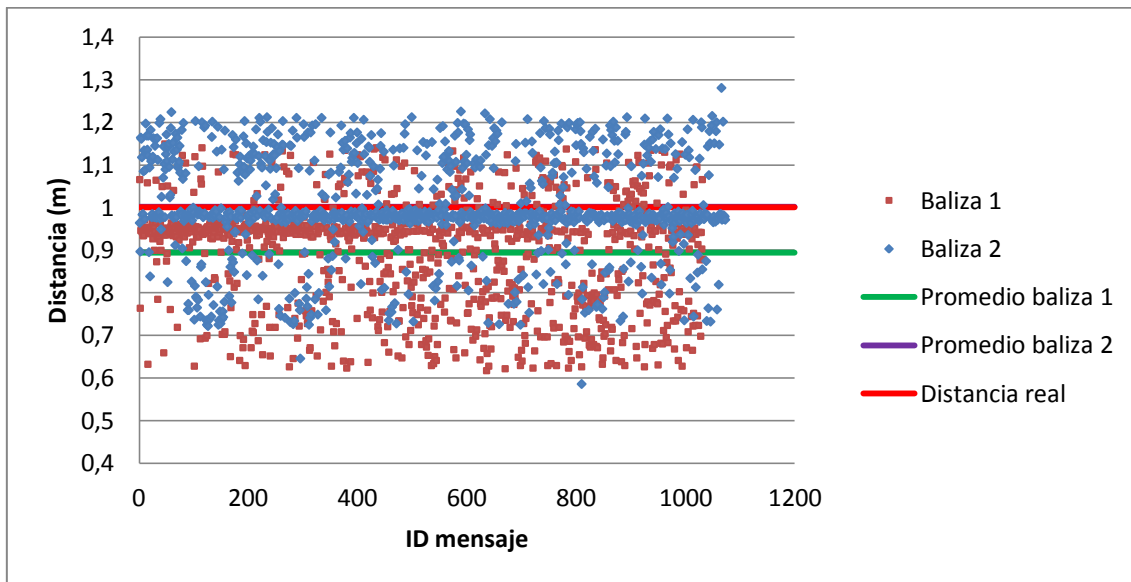


Figura 25. Resultados del experimento 3 de la campaña de pruebas.

	Baliza 1 (m)	Baliza 2 (m)
Promedio	0,894	1,002
Desviación típica	0,128	0,117
Error medio	0,106(11,89%)	0,002(0,17%)
Error máximo	0,383(42,81%)	0,414(41,32%)

Tabla 14. Resultados de medida de distancias del experimento 3

En este caso el error medio de la baliza 1 es de aproximadamente un 12%, mayor que en los experimentos anteriores. Por otra parte, el error medio de la baliza 2 está muy cercano al 0%. Esto significa que la tendencia vista en los experimentos anteriores ha cambiado, siendo ahora la baliza 2 la que consigue un error medio mucho menor que el de la baliza 1. Sin embargo, puede observarse que la dispersión de los resultados obtenidos es muy elevada, llegando a obtenerse un error máximo de más de un 40% para ambos casos.

Comparado con los resultados de los experimentos anteriores, la dispersión sigue siendo elevada y el error máximo ha aumentado hasta casi doblar el valor obtenido en el experimento anterior.

- **Experimento 4: 0.5 metros de distancia**

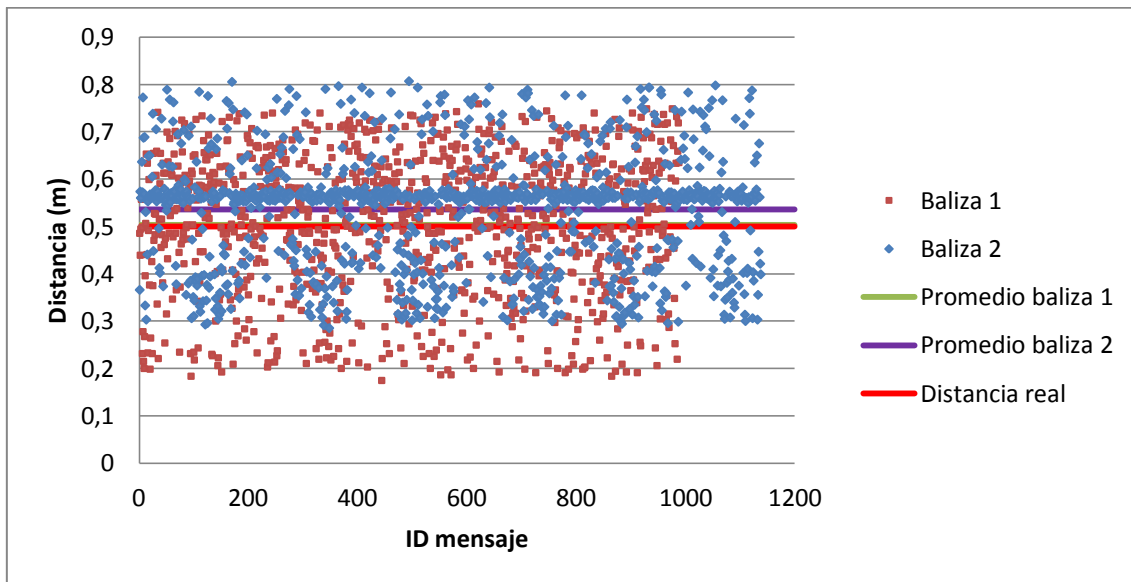


Figura 26. Resultados del experimento 4 de la campaña de pruebas.

	Baliza 1 (m)	Baliza 2 (m)
Promedio	0,504	0,536
Desviación típica	0,149	0,116
Error medio	0,004(0,72%)	0,036(6,74%)
Error máximo	0,325(64,55%)	0,307(57,32%)

Tabla 15. Resultados de medida de distancias del experimento 4

En este caso, la tendencia del error medio vista en el experimento anterior se vuelve a invertir y la baliza 1 vuelve a ser la que consigue un menor error medio. Sin embargo, puede observarse que la dispersión de los resultados obtenidos es muy elevada, llegando a obtenerse un error máximo mayor de un 55% en ambos casos. En el caso de la baliza 1 casi roza el 65%.

Comparado con los resultados de los experimentos anteriores, la dispersión sigue siendo elevada y el error máximo ha aumentado todavía más.

- **Análisis de los resultados**

En todos los experimentos la distancia media obtenida está cerca de la distancia real, siendo su error siempre menor de un 12%. Además, no parece que el error medio sea dependiente de la distancia. Para la baliza 2, exceptuando el tercer experimento, su error medio siempre ronda el 7% siendo prácticamente el mismo para una distancia de 0.5 m y de 2 m. Para la baliza 1, el error medio ronda el 5% en los primeros experimentos siendo mayor para el segundo experimento, justo al contrario que para la baliza 2.

Por otro lado, en todos los experimentos la variabilidad del error de la distancia calculada es bastante grande, aumentando el error máximo conforme disminuye la distancia. Sin embargo, si en vez de tener en cuenta el porcentaje se tiene en cuenta su valor absoluto, el error máximo se mantiene en torno a los 350 centímetros variando como máximo entre 300 y 400 centímetros. Este hecho puede significar que algún parámetro no se está teniendo en cuenta a la hora de obtener el tiempo de vuelo.

Para intentar identificar dicho parámetro, se ha llevado a cabo un último experimento en el cual las balizas están separadas una distancia de 2 m del sonar. En este experimento se ha modificado el firmware del módem para que el pin TX de la comunicación serie pueda ser utilizado para medir el tiempo de vuelo de los mensajes inalámbricos con un osciloscopio. Para ello, el pin TX se establece a nivel bajo al comienzo de la recepción y a nivel alto en el momento en el que se recibe un mensaje. En la transmisión, el pin TX se establece a nivel alto antes de enviar un mensaje y a nivel bajo justo después de enviar el último bit.

El experimento consiste en conectar el osciloscopio al pin TX del módem del sonar y de una de las balizas. A continuación, se envía un mensaje de activación mediante un mensaje CONF haciendo uso de la aplicación de demostración de la API IUMOTE para obtener la respuesta de la baliza y, así, medir el tiempo con el osciloscopio. El experimento se repite para cada baliza.

Los resultados de este experimento se muestran a continuación.

- Envío del mensaje de activación

Sonar a Baliza 1		Sonar a Baliza 2	
Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	1528	1	1545
2	1527	2	1550
3	1541	3	1548
4	1548	4	1539
5	1528	5	1537
6	1537	6	1535
7	1539	7	1537
8	1504	8	1537
9	1528	9	1541
10	1524	10	1541

Tabla 16. Tiempo de vuelo del mensaje de activación inalámbrico

- Envío del mensaje de datos

Baliza 1 a Sonar		Baliza 2 a Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	1537	1	1286
2	1427	2	1297
3	1305	3	1565
4	1325	4	1273
5	1191	5	1238
6	1515	6	1561
7	1433	7	1550
8	1372	8	1459
9	1437	9	1325
10	1288	10	1509

Tabla 17. Tiempo de vuelo del mensaje de datos inalámbrico

- Tiempo de vuelo promedio tal y como lo obtendría la aplicación

Baliza 1 - Sonar		Baliza 2 - Sonar	
Núm.	Tiempo(us)	Núm.	Tiempo(us)
1	1532,5	1	1415,5
2	1477	2	1423,5
3	1423	3	1556,5
4	1436,5	4	1406
5	1359,5	5	1387,5
6	1526	6	1548
7	1486	7	1543,5
8	1438	8	1498
9	1482,5	9	1433
10	1406	10	1525

Tabla 18. Tiempo de vuelo promedio de ambos mensajes

Finalmente, se analizan los resultados obtenidos después de calcular la distancia utilizando una velocidad del sonido de 1495 m/s.

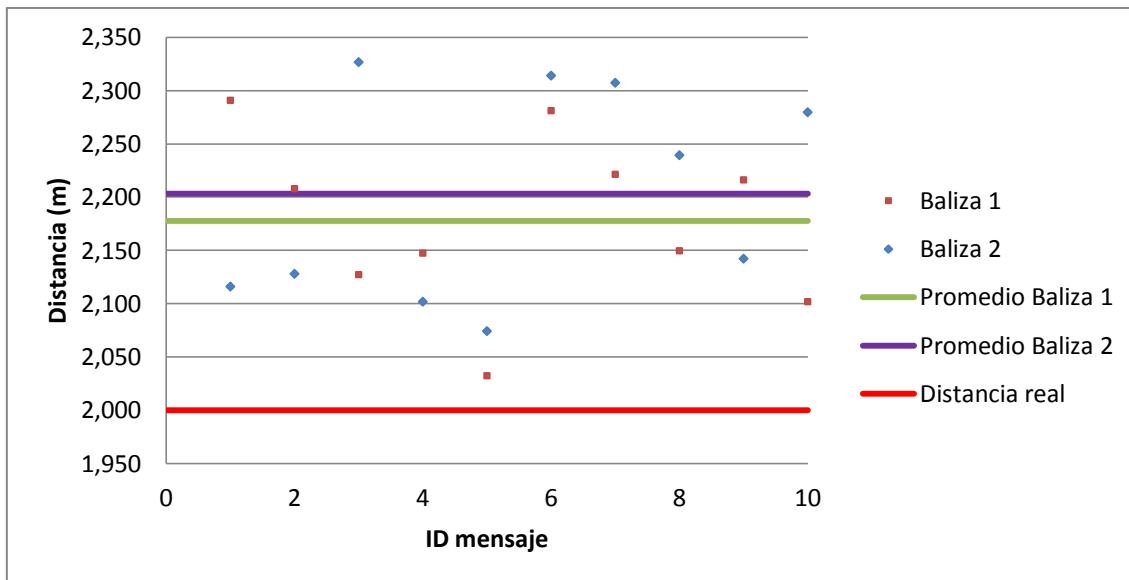


Figura 27. Resultados del experimento de obtención de distancias con osciloscopio

	Baliza 1 (m)	Baliza 2 (m)
Promedio	2,178	2,203
Desviación típica	0,077	0,095
Error medio	0,178(8%)	0,203(9%)
Error máximo	0,291(13%)	0,327(15%)

Tabla 19. Resultados de medida de distancias con osciloscopio

Como puede observarse, la variabilidad del error en la obtención del tiempo de vuelo sigue siendo grande. Por lo tanto es lógico pensar que no se trata de un parámetro temporal sino de la forma en la que se obtiene el tiempo de vuelo. Un análisis más profundo del firmware del módem ha identificado que la posible causa de esta variabilidad es el algoritmo utilizado para la sincronización de bit en las comunicaciones inalámbricas. Este algoritmo utiliza una ventana del 65% del tiempo de bit para intentar recibir un byte de sincronización, reiniciando el proceso si no lo ha obtenido. Por lo tanto, esta ventana tiene una movilidad del 35% sobre el tiempo de bit. Como consecuencia es posible que el mensaje se reciba un poco antes o un poco después del instante en el que debería recibirse. Sin embargo, esto no explica por qué el mensaje de activación tiene un comportamiento más estable que el mensaje de datos. Por lo tanto, en un futuro hace falta seguir investigando y haciendo pruebas del código para poder encontrar la causa exacta.

## 6. Conclusiones y trabajo futuro

Para dar soporte a la navegación de un vehículo autónomo subacuático mediante una red inalámbrica de sensores subacuáticos es necesario permitir que el AUV pueda obtener su localización mediante la UWSN siempre que sea necesario. En este trabajo se ha realizado una búsqueda bibliográfica de técnicas de localización y posicionamiento para conseguir una base de

conocimiento apropiada y aportar así una solución factible. Esta búsqueda bibliográfica ha dado lugar a una primera introducción al estado del arte en localización subacuática que será utilizado y ampliado posteriormente en una tesis doctoral en localización y posicionamiento subacuático.

Gracias a la búsqueda bibliográfica se ha obtenido una visión general de las técnicas de localización tradicionales y de las técnicas de localización utilizadas en redes inalámbricas de sensores subacuáticos.

Las técnicas de localización en redes inalámbricas de sensores subacuáticos permiten añadir información de localización en las aplicaciones de monitorización del entorno subacuático. Estas técnicas pueden obtener la localización de manera centralizada o distribuida. El principal inconveniente de las técnicas centralizadas es su incapacidad de dar soporte a las aplicaciones de monitorización en tiempo real. Por otro lado, el principal inconveniente de las técnicas distribuidas es el aumento de la complejidad y el consumo de energía en los nodos sensor ya que necesitan estimar su propia localización. En estas técnicas, la localización se calcula a partir de la distancia estimada a, al menos, tres puntos de referencia. Esta distancia puede calcularse haciendo uso de la telemetría o de otros métodos. Los principales inconvenientes de utilizar telemetría son obtener una buena estimación de la velocidad del sonido y la necesidad de tener en cuenta la sincronización de los dispositivos. Sin embargo, una vez se superan estos inconvenientes, la precisión obtenida en la localización es mayor que la obtenida por otros métodos.

Por otra parte, las técnicas de localización tradicionales LBL y USBL han vuelto a tener una gran importancia en los últimos años gracias a su capacidad para reducir el error de localización de los instrumentos de navegación inerciales de los vehículos subacuáticos. Estas técnicas hacen uso de la telemetría para obtener la distancia y calcular la localización. Por lo tanto, se necesita tener en cuenta la sincronización y obtener una estimación adecuada de la velocidad del sonido para reducir el error de localización.

Teniendo en cuenta las conclusiones obtenidas a partir de la búsqueda bibliográfica, se ha implementado una aplicación de obtención de tiempos de vuelo para dar soporte a la navegación de un AUV. Esta aplicación permite que el AUV pueda calcular su posición después de estimar mediante telemetría sus distancias a un conjunto de balizas en posiciones fijas. La aplicación se ha desarrollado utilizando un sistema de comunicación basado en el módem ITACA-S1000. Haciendo uso de la API de comunicaciones IWCA desarrollada con anterioridad a este trabajo, la aplicación es capaz de enviar y recibir mensajes subacuáticos para obtener los tiempos de vuelo y entregárselos al AUV. La comunicación entre el AUV y la aplicación se realiza mediante la API de comunicaciones IUMOTE desarrollada específicamente para esta aplicación en concreto. Finalmente, se ha realizado una campaña de experimentos en un acuario de 2,4 metros para evaluar la bondad de la aplicación desarrollada.

Los experimentos han demostrado que la implementación actual de la aplicación de obtención de tiempos de vuelo es una primera solución aceptable con un error menor del 12% en distancias cortas. Sin embargo, en un futuro es

necesario reducir dicho error y comprobar su funcionamiento en distancias largas.

El actual sistema de comunicación basado en el módem ITACA-S1000 tiene varios inconvenientes que impiden que se pueda reducir el error en la obtención del tiempo de vuelo mediante software. La variabilidad introducida por la recepción del módem tiene un gran impacto en distancias cortas, lo que impide que un equipo subacuático pueda fiarse de los tiempos de vuelo obtenidos con un número reducido de mensajes para calcular su localización. Por otro lado, la incapacidad del módem para realizar cualquier tarea durante la recepción tiene como consecuencia que la obtención de los tiempos tenga que trasladarse al puente del sonar. Como consecuencia, el proceso de obtención del tiempo de vuelo no puede tener en cuenta todo el tiempo involucrado y se necesita caracterizar manualmente los tiempos ignorados. Por lo tanto, este sistema no es adecuado en cuanto el número de balizas aumenta un poco.

A medio plazo es necesario modificar levemente el hardware del módem ITACA-S1000 para permitir que comparta reloj con el puente y que, además, pueda avisar de la recepción de un mensaje inalámbrico por medio de una interrupción. Esto permitiría modificar la implementación de la aplicación para que el proceso de obtención de tiempos de vuelo tenga en cuenta mucha más cantidad del tiempo involucrado. Incluso es posible que se evite caracterizar los tiempos manualmente. Además, también permitiría utilizar el modo de bajo consumo de forma más sencilla y eficiente. Por último, también sería interesante modificar el algoritmo de recepción del módem para evitar la variabilidad en la obtención de los tiempos de vuelo. De esta forma se podría utilizar el sistema actual en distancias cortas sin necesidad de enviar un gran número de mensajes.

A corto plazo, el objetivo más inmediato es probar la aplicación en distancias mucho mayores. Así, se podrá valorar si el sistema actual de comunicaciones permite dar soporte a la localización de un equipo subacuático mediante una malla de balizas con un error aceptable.



## 7. Referencias

- [1] A. Sánchez, S. Blanc, P. Yuste, A. Perles y J. J. Serrano, «An Ultra-Low Power and Flexible Acoustic Modem Design to Develop Energy-Efficient Underwater Sensor Networks,» *Sensors 2012*, nº 12, pp. 6837-6856, 2012.
- [2] R. Hirotsu, T. Ura, J. Kojima, H. Sugimatsu, R. Bahl y M. Yanagisawa, «Classification of sperm whale clicks and triangulation for real-time localization with SBL arrays,» *OCEANS 2008*, pp. 1-7, 15-18 Sept. 2008.
- [3] M. Morgado, P. Oliveira, C. Silvestre y J. Vasconcelos, «USBL/INS Tightly-Coupled Integration Technique for Underwater Vehicles,» *Information Fusion, 2006 9th International Conference on*, pp. 1-8, 10-13 July 2006.
- [4] Y. Chen, D. Zheng, P. Miller y J. Farrell, «Underwater inertial navigation with long base line transceivers: A Near-Real-Time approach,» *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 5042 - 5047, 10-13 Dec. 2013.
- [5] P. Pettinato, N. Wirström, J. Eriksson y a. T. Voigt, «Multi-channel two-way time of flight sensor network ranging,» *Proceedings of the 9th European conference on Wireless Sensor Networks (EWSN'12)*, pp. 163-178, 2012.
- [6] Y. Watanabe, H. Ochi y T. Shimura, «The SSBL Positioning for The AUV with Data Transmission,» *OCEANS 2007*, pp. 1-7, Sept. 29 2007-Oct. 4 2007.
- [7] Y. Watanabe, H. Ochi y T. Shimura, «A study of inverse SSBL acoustic positioning with data transmission for multiple AUV navigation,» *OCEANS, 2012 - Yeosu*, pp. 1-6, 21-24 May 2012.
- [8] Z. Yuyi, G. Zhenbang, W. Lei, Z. Ruiyong y L. Huanxin, «Study of Underwater Positioning Based on Short Baseline Sonar System,» *Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on*, vol. 2, pp. 343-346, 7-8 Nov. 2009.
- [9] S. Smith y D. Kronen, «Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation,» *OCEANS '97. MTS/IEEE Conference Proceedings*, vol. 1, pp. 714-720, 6-9 Oct 1997.
- [10] M. V. Jakuba, C. N. Roman, H. Singh, C. Murphy, C. Kunz, C. Willi, T. Sato y R. A. Sohn, «Long-baseline acoustic navigation for under-ice autonomous underwater vehicle operations,» *J. Field Robot*, vol. 25, pp. 11-12, November 2008.
- [11] X. Lurton y N. Millard, «The feasibility of a very-long baseline acoustic positioning system for AUVs,» *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*, vol. 3, pp. 403-408, 13-16 Sep 1994.

- [12] H. S. a. L. L. W. Ryan M. Eustice, «Synchronous-clock, one-way-travel-time acoustic navigation for underwater vehicles,» *J. Field Robot*, vol. 28, nº 1, pp. 121-136, January 2011.
- [13] H. Thomas, «GIB buoys: an interface between space and depths of the oceans,» *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, pp. 181-184, 20-21 Aug 1998.
- [14] R. W. Coutinho, A. Boukerche, L. F. Vieira y A. A. Loureiro, «GEDAR: Geographic and opportunistic routing protocol with Depth Adjustment for mobile underwater sensor networks,» *Communications (ICC), 2014 IEEE International Conference on*, pp. 251-256, 10-14 June 2014.
- [15] D. Mirza y C. Schurgers, «Motion-aware self-localization for underwater networks,» *Proceedings of the third ACM international workshop on Underwater Networks (WuWNeT '08)*, pp. 51-58, 2008.
- [16] D. Mirza y C. Schurgers, «Energy-Efficient Ranging for Post-Facto Self-Localization in Mobile Underwater Networks,» *Selected Areas in Communications, IEEE Journal on*, vol. 26, nº 9, pp. 1697 - 1707, December 2008.
- [17] V. Chandrasekhar y W. Seah, «An Area Localization Scheme for Underwater Sensor Networks,» *OCEANS 2006 - Asia Pacific*, pp. 1-8, 16-19 May 2007.
- [18] Q. Yao, S.-K. Tan, Y. Ge, B.-S. Yeo y Q. Yin, «An area localization scheme for large wireless sensor networks,» *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 5, pp. 2835 - 2839, 30 May-1 June 2005.
- [19] Y. Zhou, J. He, K. Chen, J. Chen y A. Liang, «An Area Localization Scheme for Large Scale Underwater Wireless Sensor Networks,» *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, vol. 1, pp. 543-547, 6-8 Jan. 2009.
- [20] M. Erol, F. Vieira y M. Gerla, «AUV-Aided Localization for Underwater Sensor Networks,» *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pp. 44-54, 1-3 Aug. 2007.
- [21] Y. Yu, G. Wang, Z. Li y C. Li, «Alternating Combination Trilateration for Unknown Nodes of Sensor Networks,» *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pp. 1747-1751, May 30 2007-June 1 2007.
- [22] X. Cheng, H. Shu y Q. Liang, «A Range-Difference Based Self-Positioning Scheme for Underwater Acoustic Sensor Networks,» *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pp. 38-43, 1-3 Aug. 2007.
- [23] X. Cheng, A. Thaeler, G. Xue y D. Chen, «TPS: a time-based positioning scheme for outdoor wireless sensor networks,» *INFOCOM 2004. Twenty-third Annual Joint Conference of the*

*IEEE Computer and Communications Societies*, vol. 4, pp. 2685-2696 , 7-11 March 2004.

- [24] H. Tan, A. Gabor, Z. A. Eu y W.-G. Seah, «A Wide Coverage Positioning System (WPS) for Underwater Localization,» *Communications (ICC), 2010 IEEE International Conference on*, pp. 1-5 , 23-27 May 2010.
- [25] Z. Zhou, J.-H. Cui y S. Zhou, «Localization for large-scale underwater sensor networks,» *Proceedings of the 6th international IFIP-TC6 conference on Ad Hoc and sensor networks, wireless networks, next generation internet (NETWORKING'07)*, pp. 108-119, 2007.
- [26] H. Luo, Y. Zhao, Z. Guo, S. Liu, P. Chen y L. Ni, «UDB: Using Directional Beacons for Localization in Underwater Sensor Networks,» *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, pp. 551-558, 8-10 Dec. 2008.
- [27] S. Lee y K. Kim, «Localization with a Mobile Beacon in Underwater Sensor Networks,» *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, pp. 316-319, 11-13 Dec. 2010.
- [28] V. A. Del Grosso, «New equation for the speed of sound in natural waters (with comparisons to other equations),» *The Journal of the Acoustical Society of America*, nº 56, pp. 1084-1091, 1974.
- [29] H. Medwin, «Speed of sound in water: A simple equation for realistic parameters,» *The Journal of the Acoustical Society of America*, nº 58, pp. 1318-1319 , 1975.
- [30] J. Snyder, «Doppler Velocity Log (DVL) navigation for observation-class ROVs,» *OCEANS 2010* , pp. 1-9, 20-23 Sept. 2010.
- [31] SiliconLabs, «<http://www.silabs.com/products/mcu/lowpower/pages/c8051f92x-93x.aspx>,» Accedido el 29/08/2014.
- [32] A. Sanchez, S. Blanc, P. Yuste y J. J. Serrano, «A low cost and high efficient acoustic modem for underwater sensor networks,» *OCEANS, 2011 IEEE - Spain*, pp. 1-10, 6-9 June 2011.
- [33] A. Sánchez, S. Blanc, P. Yuste, I. Piqueras y J. J. Serrano, «Advanced Acoustic Wake-up System for,» *CISME*, vol. 2, nº 2, pp. 1-10, 2012.
- [34] AMS, «<http://www.ams.com/eng%29/Products/RF-Transmitters-Receivers/LF-Receivers/AS3933>,» Accedido el 29/08/2014.
- [35] ARM, «<http://www.arm.com/products/processors/cortex-m/>,» Accedido el 01/09/2014.
- [36] ST, «<http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1848/PF252419>,» Accedido el 01/09/2014.

- [37] Qt-Project, «<http://qt-project.org>,» Accedido el 29/08/2014.
- [38] S. Climent, A. Sanchez, J. V. Capella y J. J. Serrano, «Study of MAC Protocols for a Real Underwater Sensor Network Application,» *In The 2012 International Conference on Wireless Networks*, 2012.
- [39] M. Molins y M. Stojanovic, «Slotted FAMA: a MAC protocol for underwater acoustic networks,» *OCEANS 2006-Asia Pacific*, pp. 1-7, 16-19 May 2007.
- [40] A. Syed, W. Ye y J. Heidemann, «T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks,» *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 13-18 April 2008.
- [41] I. Rhee, A. Warriar, J. Min y L. Xu, «DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks,» *Mobile Computing, IEEE Transactions on*, vol. 8, nº 10, pp. 1384 - 1396, Oct. 2009.
- [42] R. Patro y B. Mohan, «Mobile agent based TDMA slot assignment algorithm for wireless sensor networks,» *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2, pp. 663 - 667 , 4-6 April 2005.
- [43] F. Shad, T. Todd, V. Kezys y J. Litva, «Dynamic slot allocation (DSA) in indoor SDMA/TDMA using a smart antenna basestation,» *Networking, IEEE/ACM Transactions on*, vol. 9, nº 1, pp. 69 - 81, Feb 2001.
- [44] J. L. a. L. R. a. C. K. W. Lopes, «Mid-Frequency Sonar Backscatter Measurements from a Rippled Bottom,» *AIP Conference Proceedings*, nº 728, pp. 141-148, 2004.
- [45] P. Karn, «MACA a new channel access method for packet radio,» *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, 1990.