



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO
DE INGENIERÍA
ELECTRÓNICA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Ingeniería Electrónica

Diseño de un circuito electrónico de muy bajo consumo y gran capacidad de almacenamiento adaptado a las necesidades marcadas por la Marine Strategy Framework Directive de la Unión Europea

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Sistemas Electrónicos

AUTOR/A: Lara Martínez, Guillermo Fernán

Tutor/a: Ramos Peinado, Germán

Cotutor/a: Martínez Pérez, Jorge Daniel

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO
DE INGENIERÍA
ELECTRÓNICA

DISEÑO DE UN CIRCUITO ELECTRÓNICO DE MUY BAJO CONSUMO Y GRAN CAPACIDAD DE ALMACENAMIENTO ADAPTADO A LAS NECESIDADES MARCADAS POR LA MARINE STRATEGY FRAMEWORK DIRECTIVE DE LA UNIÓN EUROPA

Autor: Guillermo Lara Martínez

Tutor: Germán Ramos Peinado

Cotutor: Jorge Daniel Martínez Pérez

Trabajo Fin de Máster presentado en el Departamento de Ingeniería Electrónica de la Universitat Politècnica de València para la obtención del Título de Máster Universitario en Ingeniería de Sistemas Electrónicos

Curso 2021-22

Valencia, Septiembre de 2022

Resumen

En las últimas décadas el medio marino ha sufrido un cambio drástico en diversos campos: el incremento de transporte de mercancías a través de cargueros o la instalación de molinos de viento o windfarms para la obtención de energía eólica son algunas de las problemáticas a la que se está exponiendo sin el control debido. Todas estas prácticas tienen un reflejo en el medio marino, concretamente el incremento de ruido submarino, ya que el agua es un medio óptimo para la propagación de la onda acústica. De hecho, las diversas especies que pueblan los mares y océanos están adaptadas al medio marino y por tanto son expertas en la comunicación acústica aprovechando sus propiedades. Diversos organismos se están encargando de establecer en qué parámetros y con qué evolución esta flora y fauna se está viendo afectada por el incremento del ruido antropogénico, entre ellos el TGNoise, la comisión de expertos europea encargada de definir el buen estado ambiental de los mares y océanos europeos.

Por el contrario, la tecnología aplicada al medio marino también ha evolucionado en estos últimos años, permitiendo el uso de diversos dispositivos para conocer y monitorizar lo que ocurre. Entre los más interesantes se encuentran los dispositivos de monitorización acústica pasiva (PAM), encargados de captar y almacenar el ruido submarino desde el fondo del mar.

Este trabajo fin de master ha intentado mejorar uno de los dispositivos PAM que existen en el mercado desarrollando un diseño electrónico de muy bajo consumo que permita escuchar y almacenar el ruido submarino con una tasa de muestreo elevada durante varios meses de duración, con el objetivo de ayudar a conocer mejor lo que sucede en los mares y océanos.

Resum

En les últimes dècades el medi marí ha patit un canvi dràstic en diversos camps: l'increment de transport de mercaderies a través de vaixells de càrrega o la instal·lació de molins de vent o windfarms per a l'obtenció d'energia eòlica són algunes de les problemàtiques a la qual s'està exposant sense el control degut. Totes aquestes pràctiques tenen un reflex en el medi marí, concretament l'increment de soroll submarí, ja que l'aigua és un mitjà òptim per a la propagació de l'ona acústica. De fet, les diverses espècies que poblen les mars i oceans estan adaptades al medi marí i per tant són expertes en la comunicació acústica aprofitant les seues propietats. Diversos organismes s'estan encarregant d'establir en quins paràmetres i amb quina evolució aquesta flora i fauna s'està veient afectada per l'increment del soroll antropogènic, entre ells el TGNoise, la comissió d'experts europea encarregada de definir el bon estat ambiental de les mars i oceans europeus.

Per contra, la tecnologia aplicada al medi marí també ha evolucionat en aquests últims anys, permetent l'ús de diversos dispositius per a conèixer i monitorar el que ocorre. Entre els més interessants es troben els dispositius de monitoratge acústic passiu (PAM), encarregats de captar i emmagatzemar el soroll submarí des del fons de la mar.

Aquest treball fi de màster ha intentat millorar un dels dispositius PAM que existeixen en el mercat desenvolupant un disseny electrònic de molt baix consum que permeti escoltar i emmagatzemar el soroll submarí amb una taxa de mostreig elevada durant diversos mesos de duració, amb l'objectiu d'ajudar a conèixer millor el que succeeix en les mars i oceans.

Abstract

In recent decades, the marine environment has undergone a drastic change in various fields: the increase in the transport of goods through freighters or the installation of windfarms to obtain wind energy are some of the problems that is exposing without a proper control. All these practices have a reflection on the marine environment, specifically the increase in underwater noise, since water is an optimal medium for the propagation of the acoustic wave. In fact, the various species that inhabit the seas and oceans are adapted to the marine environment and are therefore experts in acoustic communication, taking advantage of its properties. Various organizations are in charge of establishing in what parameters and with what evolution this flora and fauna is being affected by the increase in anthropogenic noise, among them, the TGNoise, the European commission of experts in charge of defining the Good Environmental Status of the seas and European oceans.

On the contrary, the technology applied to the marine environment has also evolved in recent years, allowing the use of various devices to know and monitor what is happening. Among the most interesting are passive acoustic monitoring (PAM) devices, responsible for capturing and storing underwater noise from the bottom of the sea.

This master's thesis has tried to improve one of the PAM devices that exist on the market by developing a very low-power electronic design that allows listening to and storing underwater noise with a high sampling rate for several months, with the aim of help to better understand what happens in the seas and oceans.

Gracias al Instituto Español de Oceanografía, en concreto al grupo de Ruido Submarino, por dar cabida a este trabajo y elegir el diseño al completo del dispositivo que aquí se cuenta como referencia para realizar el encargo del Ministerio para la Transición Ecológica y el Reto Demográfico a cuenta de la Directiva 11 (C2) de Ruido Submarino. A su vez, gracias a todos los componentes de la Unidad Mixta IEO (CSIC) - UPV por conseguir que un prototipo desarrollado íntegramente en la UPV pueda transferirse de manera satisfactoria a una empresa pública con el interés que ello supone.

Índice general

I Memoria

1. Introducción	1
2. Entorno y especificaciones del diseño	5
2.1. Hidrófono	9
2.2. Niveles de presión y consideraciones respecto a la conversión Analógico-digital .	10
3. Diseño hardware	13
3.1. Conversor analógico digital	13
3.2. Procesador y otros componentes	14
3.3. Diseño PCB en ALTIUM	16
3.3.1. Esquemáticos	16
3.3.2. Stack-Up	21
3.3.3. Rutado	22
4. Programación firmware	25
4.1. Código generado	25
4.2. Sistema de escritura en memoria	27
5. Resultados	31
5.1. Placa de evaluación eZDSP5535	31
5.2. Diseño auxiliar	32
5.3. Alimentación	34
5.4. Resultados	35
6. Conclusiones y líneas futuras	39

II Anexos

A. Listados componentes diseño auxiliar	47
B. Código firmware	49

Índice de figuras

1.1. Ilustración adaptada de las Curvas de Wenz.	2
2.1. Representación de las fuentes de ruido existentes en el mar/océano.	5
2.2. Fases principales del fondeo de un dispositivo PAM autónomo.	6
2.3. (a) Representación gráfica de un tren de fondeo autónomo y (b) dispositivo SAMARUC finalizado con la boya anillada de flotación.	7
2.4. Imagen del hidrófono de Cetacean Research. Modelo C57XRS.	9
2.5. Representación del rango de presiones acústicas (en rojo) captado por el sistema (hidrófono + ADC) en función de la configuración de la ganancia y los bits de adquisición.	10
2.6. Diagrama de la cadena de entrada de audio. Pa: Pascales, V: Voltios, Cd: Cuentas digitales.	11
3.1. Cadena de entrada en el ADC, con las distintas fases por las que pasa la señal analógica.	14
3.2. Visualización y conceptualización global del diseño electrónico a realizar.	15
3.3. Esquemático del regulador LT3995.	16
3.4. Captura del diseño jerarquico global.	17
3.5. Captura del esquemático al códec.	18
3.6. Captura del esquemático referente al emulador XDS100v2.	18
3.7. Captura del esquemático referente al regulador de amplio rango de tensión.	19
3.8. Captura del esquemático referente a la conexión de la μ SD.	19
3.9. Captura del esquemático referente a los reguladores utilizados en el diseño, así como al medidor de voltaje y corriente.	20
3.10. Captura del STACKUP realizado.	21
3.11. Detalle de las vías utilizadas.	22
3.12. Captura de las impedancias entre las capas.	22
3.13. Footprint del rutado en una fase intermedia.	23
3.14. Captura 3D del posicionamiento de los componentes en la PCB.	24
4.1. Flujo global del diseño firmware desarrollado.	25
4.2. Representación del modo de funcionamiento Ping Pong utilizando 2 Direct Memory Access.	26
4.3. Detalle en forma de tabla del sistema de escritura en memoria.	28
5.1. (a) Imagen de dicha placa de evaluación EZDSP5535 y (b) diagrama con sus principales componentes.	32
5.2. Esquemático del diseño auxiliar.	33

5.3. (a) Rutado de la PCB del diseño auxiliar y (b) Detalle del diseño auxiliar fabricado y montado	33
5.4. Curva simulada de la descarga de la alimentación de la electrónica.	34
5.5. Imágenes de (a) bloque de baterías ensamblado que alimenta a la electrónica del dispositivo y (b) distribución de las baterías en el bloque diseñado.	35
5.6. Detalle del espectrograma de uno de los archivos grabado así como su forma temporal.	37
5.7. Imágenes del SAMARUC (a) en el momento de su visualización una vez acaba de subir desde el fondo marino y (b) una vez recuperado del fondo del mar despues de 3 meses	37

Índice de tablas

2.1. Elementos vs función	7
2.2. Duración del fondeo según el modo de grabación a 16 bits, frecuencia de muestreo: 192 kHz y 1 canal	8
2.3. Número de archivos de 5 minutos que se pueden grabar según el modo de grabación de: 16 bits, frecuencia de muestreo: 192 kHz y 1 hidrófono mono	8
2.4. Especificaciones del hidrófono	10
3.1. Resumen de las especificaciones globales del diseño	13
5.1. Características de la Placa de evaluación comercial EZDSP5535	31
5.2. Funcionalidades aportadas por el diseño auxiliar	32
5.3. Funcionalidades cumplidas por el diseño auxiliar vs diseño propuesto	34
5.4. Resumen de los resultados de los fondeos	36

Parte I

Memoria

Capítulo 1

Introducción

La evaluación del buen estado medioambiental “Good Environmental Status”(GES) en los mares y océanos es una cuestión que tiene un papel cada vez más importante en relación con la biodiversidad y el control de los ruidos antropogénicos creados por el hombre y sus infraestructuras. De hecho, el documento Monitoring Guidance for Underwater Noise in European Seas: Monitoring Guidance Specifications, dentro de la European Union Marine Strategy Framework Directive (MSFD) [1], especifica como acción fundamental monitorizar y posteriormente analizar los datos acústicos submarinos para alcanzar adecuadamente los objetivos de obligado cumplimiento (relacionados con el nivel de presión acústico submarino) por parte de todos los países miembros de la Unión Europea que poseen mares y océanos colindantes.

Existen múltiples fuentes causantes de ruido antropogénico en la biota marina, y la monitorización acústica pasiva, en inglés “passive acoustic monitoring (PAM)” permite que dichos sonidos se adquieran y estudien de manera continua. Para ello, dichos dispositivos deben diseñarse y utilizarse correctamente, siguiendo las directrices marcadas por el Technical Group on Underwater Noise (TGNoise).

Para mostrar como ha evolucionado la tecnología relacionada con la monitorización acústica pasiva es importante contextualizar el habitat submarino de los mares y océanos. Existen diversos estudios sobre organismos y especies acústicamente sensibles [2, 3]. Dichos estudios apuntan el papel fundamental del sonido para especies y ecosistemas en procesos tan diversos como las interacciones depredador-presa [4] o el comportamiento coordinado [5]. El entorno acústico de estos fenómenos no es un mundo silencioso: la evolución de la monitorización de la acústica submarina ha tenido lugar en el contexto de un fondo natural variable [6, 3] al que los organismos adaptan su comportamiento acústico [7]. Este ruido de fondo es generado por procesos climáticos (viento, lluvia, truenos), eventos sísmicos y sonido biótico (fauna y flora) [8, 9].

La Figura 1.1 corresponde a las curvas de Wenz [10], un referente en el ámbito del sonido submarino. En ella se pueden observar en qué frecuencias (eje x, logarítmico) con qué nivel de presión acústica (eje y) se sitúan los diferentes sonidos que existen en el medio marino. Destacar que los niveles de presión acústica en el agua se dan en dB re 1 μ Pa, no como en el aire donde la unidad de referencia para su cálculo son 20 dB re 20 μ Pa). Concretamente, se muestra que los terremotos y las explosiones se pueden encontrar en el rango frecuencial de entre [1 - 100 Hz] o que los sonidos biológicos pueden llegar a cubrir un rango frecuencial mayor, entre algo menos de 10 Hz y pudiendo llegar a los 100 kHz. Además, en las curvas azules numeradas se refleja el

ruido submarino provocado por la cantidad de viento que existe en la superficie, es decir, a mayor número de Beaufort, el ruido submarino provocado es mayor en todo el rango frecuencial (hasta los 50 kHz aproximadamente). Un aspecto importante que también muestran la Figura 1.1 son las curvas con sombreado rojo, en ellas se puede ver cómo el tráfico marino tiene su reflejo en el rango frecuencial entre los [5 - 100 Hz]. Es por ello que la MSFD [1] interpreta que en su Directiva 11 (concretamente en su apartado D11.C2, acorde al ruido submarino), que la monitorización del ruido submarino debe realizarse en las bandas de frecuencia de tercio de octava de 63 Hz y 125 Hz con el objetivo de ver como influye el tráfico marino en el hábitat submarino [11].

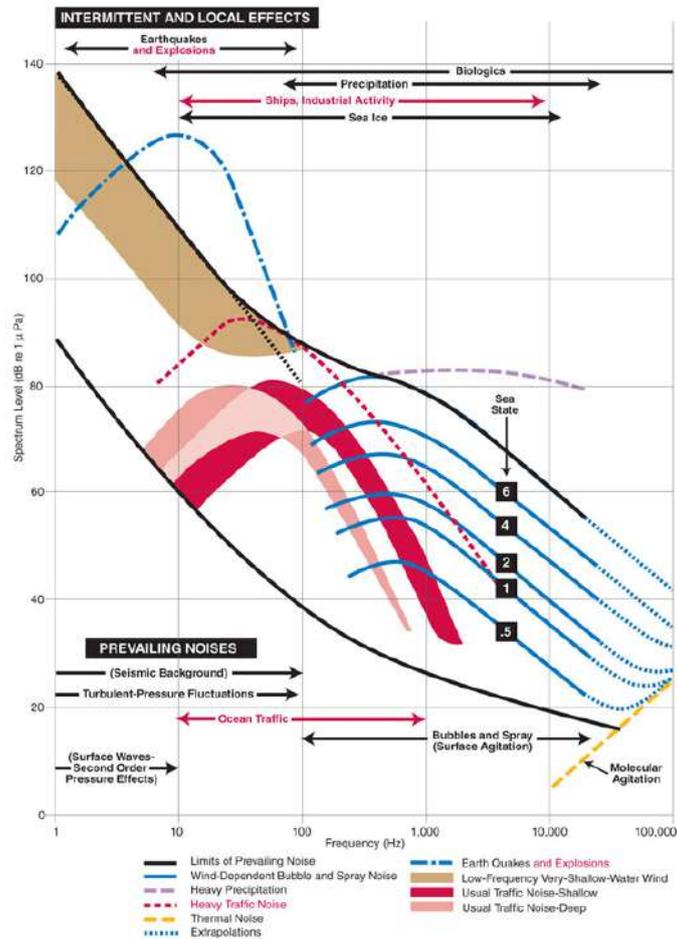


Figura 1.1: Ilustración adaptada de las Curvas de Wenz.

Los hábitats acústicos se han visto cada vez más alterados por el ruido antropogénico. En tierra, estas fuentes incluyen el transporte por carretera, ferroviario y aéreo, y la actividad industrial [12], mientras que las operaciones submarinas, marítimas de construcción en alta mar, exploración de petróleo y gas, y los sónares activos de los diferentes contribuyen al paisaje sonoro [8]. En ambos dominios, este ruido puede enmascarar señales acústicas [3, 13] o provocar respuestas conductuales [14, 15], con el potencial de causar estrés fisiológico crónico [16] y efectos más amplios sobre las poblaciones y las comunidades [17]. La conciencia de estos impactos humanos ha traído una renovada urgencia al estudio de los hábitats acústicos y sus influencias en los procesos de los ecosistemas [16]. La rápida expansión de este campo ha sido impulsada por importantes avances

en la tecnología PAM y ha llevado a una creciente demanda de experiencia en acústica submarina. Durante las últimas dos décadas, el desarrollo de unidades de grabación autónomas ha revolucionado la bioacústica tanto en el aire [18, 19] como bajo el agua [20]. La bioacústica marina ha sido impulsada particularmente por la innovación tecnológica, con nuevas perspectivas ofrecidas por etiquetas de grabación no invasivas [21, 22], planeadores autónomos [23, 24], plataformas a la deriva [25] y observatorios oceánicos cableados a escala regional [26].

La tecnología ha dado forma al desarrollo de la bioacústica. Los avances en la instrumentación, la capacidad de almacenamiento de datos y el análisis de datos con nuevos desarrollos como el Deep Learning, han abierto nuevas vías de investigación, al tiempo que permite evolucionar el conocimiento previamente establecidas. Para ayudar a contextualizar las limitaciones históricas en la monitorización de datos acústicos en el habitat submarino, se resumen a continuación las capacidades y limitaciones de los principales tipos de estructuras para la colocación de dispositivos PAM. Entre ellas, las plataformas fijas, aquellas diseñadas para implementarse en una localización durante varios días, o también las plataformas móviles, aquellas que graban mientras están en movimiento, como los gliders, los cuales se mueven siguiendo las corrientes marinas.

La reciente expansión en el estudio de los hábitats acústicos se ha visto facilitada por el desarrollo de dispositivos acústicos autónomos: instrumentos digitales autónomos que se pueden fijar a estructuras terrestres o amarrados al fondo marino para registrar el paisaje sonoro durante varios meses [18, 20]. Estos son generalmente más rentables y más fáciles de implementar que los sistemas cableados y se pueden colocar en trenes de fondeo previamente instalados. Cada unidad consta de una electrónica encargada de la adquisición de datos digitales y almacenamiento dentro de una carcasa impermeable o resistente a la intemperie. El transductor acústico (hidrófono en este caso) puede montarse en el dispositivo o conectarse mediante un cable. La longevidad de la implementación está limitada por el consumo de energía, la potencia del bloque de baterías que va alojado dentro de la misma carcasa, y la capacidad de almacenamiento de datos, los cuales continúan mejorando a medida que la tecnología evoluciona [20]. Complementar el bloque de baterías con una fuente externa (panel solar o batería adicional) es una posibilidades para aumentar la duración de la grabación. La monitorización del hábitat puede continuar indefinidamente si los dispositivos se revisan regularmente, reponiendo las baterías y recogiendo los datos almacenados. A su vez, se están desarrollando sistemas para la recuperación remota de datos a través de redes inalámbricas de comunicación. Los sistemas montados en el lecho marino se pueden recuperar utilizando dispositivos de liberación acústica, aunque el mal funcionamiento de las emisiones acústicas y la pesca de arrastre por parte de los barcos pesqueros son peligros comunes en el campo. El robo y el vandalismo también son preocupaciones potenciales, así como los daños causados por la vida silvestre y la contaminación biológica.

Los sistemas cableados se han utilizado durante mucho tiempo para monitorizar el sonido submarino y consisten en hidrófonos montados en el fondo marino conectados a estaciones costeras que proporcionan energía y datos adquisición. En los últimos años, los sistemas montados por cable han experimentado un resurgimiento con la expansión de los observatorios oceánicos cableados, por ejemplo, en Europa, en infraestructuras con múltiples hidrófonos como ORCA o ARCA dentro del proyecto KM3NeT [27].

La capacidad de monitorización acústica en tiempo real y el mantenimiento menos frecuente requerido (en comparación con las unidades autónomas) son las principales ventajas de los sistemas cableados, aunque los costes asociados de despliegue, mantenimiento y gestión de estos dispositivos a largo plazo y los grandes volúmenes de datos que generan son demasiado altos.

En este trabajo fin de master se persigue el diseño electrónico y la programación de su firmware de una PCB para su incorporación en el sistema de monitorización acústica pasiva SAMARUC [28], un dispositivo PAM autónomo que se instala en el fondo marino y permite monitorizar durante varios meses el paisaje sonoro.

El trabajo se estructura de la siguiente manera: en el Capítulo 2 se describen las características fundamentales del diseño priorizando la elección de los componentes electrónicos y periféricos necesarios. En el Capítulo 3 se aborda el diseño del circuito utilizando el software de diseño ALTIUM y en el Capítulo 4 se explica la programación firmware realizada en CODE COMPOSER, el software proporcionado por Texas Instruments para la programación firmware del procesador y los periféricos escogidos. Para finalizar, en el Capítulo 5 se mostrarán los resultados obtenidos en entornos de prueba reales, utilizando la placa de evaluación comercial ezdspC5535 de Texas Instruments, acabando con el Capítulo 6, donde se describirán las conclusiones y líneas futuras a incluir en próximos diseños.

Capítulo 2

Entorno y especificaciones del diseño

El sonido es, por definición, cualquier fenómeno que involucre la propagación de ondas mecánicas (sean audibles o no al ser humano), a través de un medio (fluido o sólido) que esté generando el movimiento vibratorio de un cuerpo. Los mares y océanos no están exentos de él. El viento y las olas, los terremotos submarinos, el rompimiento de hielos y glaciares, las erupciones de volcanes subterráneos generan ruido en el mar (ver Figura 2.1).

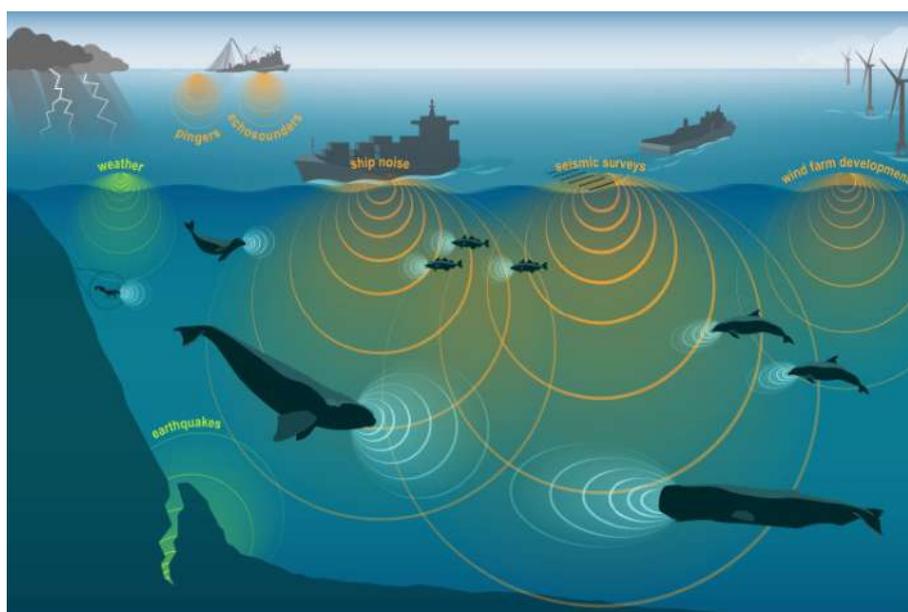


Figura 2.1: Representación de las fuentes de ruido existentes en el mar/océano.

La existencia de los sonidos de los procesos físicos usuales del océano, ha llevado a que los animales hayan desarrollado sistemas auditivos en presencia de estas fuentes de ruido, en algunos casos aprovechando las señales que proporcionan.

Durante los últimos años, el transporte marítimo, la navegación recreativa y las exploraciones del océano han aumentado en las costas y el mar abierto. El sonido generado por estas actividades puede viajar largas distancias bajo el agua, y aumentar los niveles de ruido en el océano; dicho ruido puede enmascarar la comunicación de algunas especies [29], cambiando su comportamiento.

Con el objetivo de extraer información sobre los sonidos de diferentes especies de animales y ruidos antropogénicos provocados por el hombre, los dispositivos de monitorización acústica pasiva (PAM) autónomos son fondeados en lugares donde resulta relevante conocer dicha información. Varios meses después, el dispositivo se recupera y los datos se extraen de la memoria del dispositivo como archivos wav en discos duros. Finalmente, estos discos duros son conectados de manera local a ordenadores con la potencia adecuada para poder analizar los archivos grabados a frecuencias de muestreo más altas comparadas con las del audio habitual. Más concretamente, en la Figura 2.2 se resume el fondeo y la recogida de un dispositivo PAM.

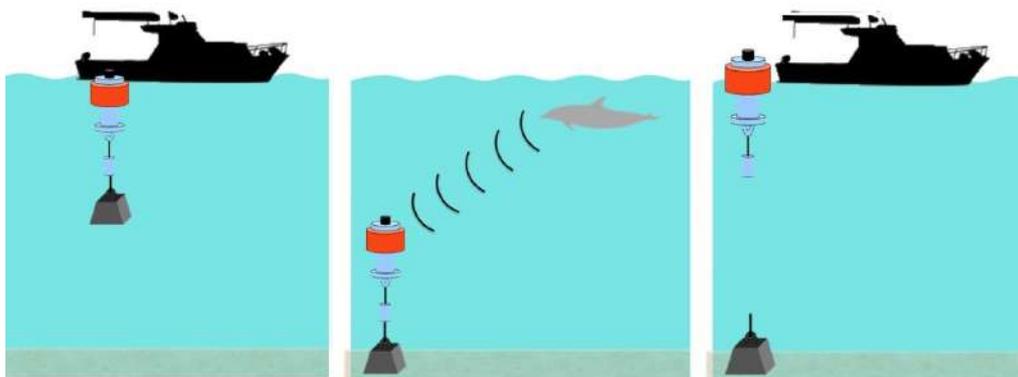
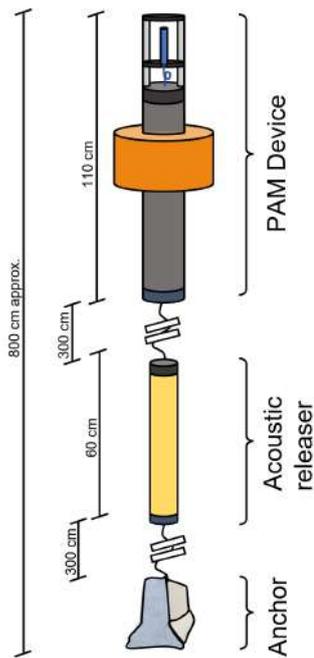


Figura 2.2: Fases principales del fondeo de un dispositivo PAM autónomo.

Ya se ha comentado previamente como los fondeos autónomos ganan en importancia tanto la cantidad de datos sean capaces de adquirir. Es cierto que dado que el intercambio de datos inalámbrico a través del medio marino sigue siendo un reto tecnológico a solventar, y para ello se siguen desarrollando módems acústicos con cada vez más ancho de banda. Este tipo de soluciones no son nada económicas y menos cuando es necesario escalar la instalación de más de un módem, por ejemplo, a la hora de cubrir las cinco Demarcaciones Marinas (DM) [30] de las que consta nuestro país. Debido a esto, los dispositivos PAM toman una importancia relevante con un coste tecnológico asumible, almacenando los datos acústicos sin pérdidas durante varios meses, para que, una vez recuperado el dispositivo, dichos datos sean descargados y analizados.

A la hora de realizar fondeos autónomos es necesario la realización de un tren de fondeo cómodo de manejar, contando como elemento clave el liberador acústico, el dispositivo que se encarga de la recuperación de los dispositivos del tren de fondeo que se encuentran en el lecho marino. Según el comando que se ejecute, se obtiene información de la profundidad a la cual está el dispositivo, el nivel de las baterías que le quedan al propio liberador, o bien se realizan las acciones correspondientes a la liberación del tren de fondeo del muerto a través de el giro del eslabón que los une.

En la Figura 2.3a se puede ver el esquema de un tren de fondeo para un dispositivos PAM autónomo (el SAMARUC en nuestro caso, ver Figura 2.3b), consistente en un elemento pesado, habitualmente llamado muerto, hecho de hormigón o de sacos de yuta, un liberador acústico y el dispositivo de grabación autónoma con una boya de flotación anillada, son lanzados al fondo del mar. Durante varios meses, el dispositivo de grabación escucha y almacena el ruido submarino con el ciclo de trabajo seleccionado hasta llenar su memoria. Una vez se decide volver al lugar de fondeo para proceder a la recogida del dispositivo, desde el barco se realizará la comunicación con el liberador acústico desde su unidad de superficie (terminal de comandos). Desde ella se



(a)



(b)

Figura 2.3: (a) Representación gráfica de un tren de fondeo autónomo y (b) dispositivo SAMARUC finalizado con la boya anillada de flotación.

dará la orden al liberador de que el eslabón que le une al muerto se desprenda y, gracias al collar de flotación (en naranja), tanto el liberador acústico como el dispositivo PAM, volverán a la superficie del mar y serán recogidos.

Elementos externos del SAMARUC	Detalle
Carcasa	Tubo cilíndrico de aluminio 6082 anodizado
Boyas anilladas de flotación	Fácil manejo, evitando posibles sombras acústicas
Jaula protectora	Protección del hidrófono

Tabla 2.1: Elementos vs función

Las boyas anilladas al cuerpo del dispositivo PAM dotan de la flotabilidad necesario al tren de fondeo y evitan sombras acústicas. El dispositivo consta de una jaula para sujetar mecánicamente el hidrófono en posición vertical, aislarlo de vibraciones y protegerlo de golpes. La Tabla 2.1 resume la relación entre los elementos externos del dispositivo y la función que desarrollan. El dispositivo donde se aloja tanto el bloque de baterías como la electrónica es un cilindro de aluminio 6082 anodizado en NITUFF con las siguientes dimensiones: 120 mm. de diámetro exterior, 100 mm. de diámetro interior y 465 mm. de largo. En él deberá ir alojada la electrónica y también las baterías que servirán como alimentación, tanto para dicha electrónica como para los 2 hidrófonos que podrán ser instalados en el dispositivo.

En la Tabla 2.2 se puede ver un resumen de los días que puede estar grabando el dispositivo según la memoria o hidrófonos que se vayan a utilizar en un fondeo. Es obvio que cuanto más memoria tenga el dispositivo en forma de tarjetas *muSD*, más días podrán ser cubiertos. No obstante,

	Contínuo	5 ON / 5 OFF	5 ON / 10 OFF
1 Hidrófono - 1 Tbyte	1 mes	2 meses	3 meses
1 Hidrófono - 2 Tbytes	2 meses	4 meses	6 meses
2 Hidrófonos - 1 Tbyte	15 días	1 mes	45 días
2 Hidrófonos - 2 Tbytes	1 mes	2 meses	3 meses

Tabla 2.2: Duración del fondeo según el modo de grabación a 16 bits, frecuencia de muestreo: 192 kHz y 1 canal

cabe destacar que la única manera de incrementar los días en los cuales el dispositivo está grabando es realizar ciclos de encendido y apagado. Por ejemplo, para poder grabar durante 6 meses, deberemos configurar el firmware con ciclos de 5 minutos ON y 10 minutos OFF (ver Fila 4 de la Tabla 2.2), grabando únicamente con un hidrófono (mono) y utilizando una memoria de 2 Tbytes.

Número de archivos grabados	5 ON / 1 OFF	5 ON / 5 OFF	5 ON / 10 OFF
Al día	240	144	96
Al mes (30 días)	7200	4320	2880
En 2 meses	14400	8640	8640
En 4 meses	28800	17280	11520
En 6 meses	43200	25920	17280

Tabla 2.3: Número de archivos de 5 minutos que se pueden grabar según el modo de grabación de: 16 bits, frecuencia de muestreo: 192 kHz y 1 hidrófono mono

Con esta configuración se almacenarán aproximadamente unos 17374 archivos de 5 minutos a una frecuencia de muestreo de 192 kHz y con 16 bits de adquisición (ver Tabla 2.3). Cabe destacar que en 2 Tbytes de memoria no cabrían todas estas configuraciones, solo las destacadas en negrita (relacionado con la Tabla 2.2).

La cantidad de tiempo que el sistema debe grabar va ligado a los requisitos de consumo de los componentes que se van a seleccionar. En el caso de tener que realizar un fondeo de 6 meses de duración (ciclo 5/10), el sistema estaría activo 1440 horas y en reposo 2880 horas. Conforme a ello, si suponemos que tenemos disponibles 600 Wh para alimentar la electrónica durante ese tiempo, sabiendo que la mayoría de componentes trabajan a 3.3 V y suponiendo que en modo reposo el consumo del sistema es 1/6 comparándolo cuando está grabando ($x_{ON} = 6 * x_{reposo}$), tenemos la siguiente ecuación:

$$600[Wh] = 3,3[V] * x_{ON}[A] * 1440[h] + 3,3[V] * x_{reposo}[A] * 2880[h]; \quad (2.1)$$

$$600[Wh] = 3,3[V] * x_{ON}[A] * 1440[h] + 3,3[V] * \frac{x_{ON}}{6}[A] * 2880[h]; \quad (2.2)$$

$$x_{ON}[A] = \frac{600}{3,3 * 1920} = 94,69mA \quad (2.3)$$

Es decir, el diseño completo deberá consumir algo menos de 95 mA en activo y menos de 15 mA en reposo. Por tanto, se seleccionarán los distintos componentes del sistema teniendo en cuenta este requisito.

2.1. Hidrófono

La elección de un hidrófono de calidad es parte fundamental a la hora de extraer los datos con la mejor sensibilidad posible, adaptado al ruido ambiente de cada una de las localizaciones donde se van a realizar los fondeos.

El dispositivo PAM SAMARUC permite la elección de cualquier marca de hidrófono. Cabe destacar que dependiendo de la sensibilidad que requiera la aplicación para la cual el dispositivo sea utilizado, la elección del hidrófono será una de las cuestiones más importantes, así la elección de un conversor analógico digital (ADC) capaz de aportar una ganancia previa a la conversión, es decir, en el dominio analógico.

Tal y como marca la MSFD [1], para la implantación de las Directivas de Ruido D11 es necesario que las grabaciones submarinas sean adquiridas con al menos 16 bits de adquisición y una frecuencia de muestreo de al menos 96 kHz, y que el equipo de adquisición (hidrófono) cuente con una sensibilidad entre $[-165$ y -185 dB re $1 \text{ V}/\mu\text{Pa}$]. Estos datos proporcionan medidas absolutas de los niveles sonoros bióticos, abióticos y antropogénicos, los cuales son necesarios para establecer comparaciones significativas de hábitats a lo largo del tiempo y en diferentes lugares.

Si por ejemplo, el hidrófono elegido tiene una sensibilidad de -190 dB re $1 \text{ V}/\mu\text{Pa}$, será necesario complementarlo con una ganancia analógica interna por parte del ADC seleccionado. El diseño y elección de componentes principales de la placa electrónica será descrita en el Capítulo 3.



Figura 2.4: Imagen del hidrófono de Cetacean Research. Modelo C57XRS.

Los hidrófonos seleccionados son de la marca Cetacean Research, en concreto su modelo C57XRS [31] (ver Figura 2.4), con una sensibilidad cercana a los -187 dB re $1 \text{ V}/\mu\text{Pa}$ con un preamplificador incorporado de 20 dB, al cual habrá que alimentar con una tensión entre $[5 - 32 \text{ V}]$ (parte del sistema de baterías irá destinado a alimentar esta preamplificación). La sensibilidad efectiva del hidrófono será de -167 dB re $1 \text{ V}/\mu\text{Pa}$. Estos hidrófonos tienen las características resumidas en la Tabla 2.4.

Rango de Frecuencia útil (-3 dB) (kHz)	0.008 hasta 250
Sensibilidad del transductor [dB re 1V/ μ Pa]	-200
Ganancia del preamplificador [dB]	33
Sensibilidad efectiva [dB re 1V/ μ Pa]	-167
Ruido propio en SPL Equiv. a 1kHz [dB re 1 μ Pa/ \sqrt{Hz}]	61
Alimentación requerida [V]	5 - 32
Máxima profundidad [m]	920
RMS Overload Acoustic Pressure [dB re 1 μ Pa]	188
Rango de temperatura[°C]	-40 to 60
Impedancia de salida [Ω]	10
Conector	Subconn MCBH3MSS

Tabla 2.4: Especificaciones del hidrófono

2.2. Niveles de presión y consideraciones respecto a la conversión Analógico-digital

Para entender como puede influir un Conversor Analógico Digital (ADC) a la hora de adquirir la señal que le proporcione un sensor, en este caso, un hidrófono. La elección de los bits de adquisición y de la ganancia analógica y/o digital que pueda incorporar el ADC previo y/o post a la adquisición del dato es fundamental.

En la Figura 2.5 se muestra como cambia el rango de valores de presiones acústicas que podemos obtener según los bits de adquisición del ADC y la ganancia analógica de la cadena de adquisición del dispositivo, es decir, el hidrófono y el ADC.

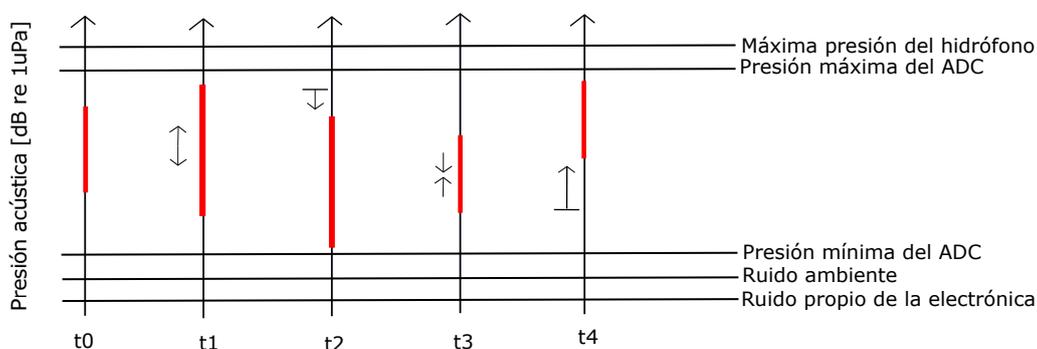


Figura 2.5: Representación del rango de presiones acústicas (en rojo) captado por el sistema (hidrófono + ADC) en función de la configuración de la ganancia y los bits de adquisición.

Los diferentes instantes temporales se marcan como t_0 , t_1 , t_2 , t_3 y t_4 . El eje 'y' representa el nivel de presión acústica y en rojo podemos ver el rango dinámico que el dispositivo puede captar según los diferentes escenarios. Partimos del rango de valores mostrado en el instante t_0 (pongamos por ejemplo que adquirimos con 16 bits de adquisición efectivos y una ganancia 0 dB). Si en t_1 aumentamos los bits de adquisición (por ejemplo a 24), el rango dinámico aumentará por la parte inferior. En t_2 el rango dinámico se mantiene como t_1 pero los valores de presión que somos capaces de captar están desplazados hacia valores inferiores, esto se consigue subiendo la ganancia del dispositivo. A cambio, el sistema saturará antes en el ADC. Si volvemos a reducir el

rango dinámico del sistema adquiriendo otra vez con 16 bits, veremos el efecto en t3. Por último, si quitamos la ganancia y la configuramos a 0 dB, el rango dinámico se desplazará hacia valores más altos.

Cabe destacar que el rango dinámico efectivo no es el teórico (calculado a partir de los bits de adquisición), sino que debemos fijarnos en los valores que nos da el fabricante. En la mayoría de ocasiones, los bits de adquisición efectivos no son 32, 24 o 16 bits, sino algunos menos. Esto es debido a que los bits menos significativos, suelen tener variaciones al estar cerca del ruido propio de la electrónica. Es por ello que el rango dinámico siempre será menor que el teórico.

Una vez la presión acústica llega al hidrófono, tendremos que fijarnos en el rango de la señal eléctrica a su salida conociendo su sensibilidad. Se deben tener en cuenta varios escenarios:

1. La presión que llega al hidrófono no debe superar el overload de su membrana, de esta manera se evitará que la señal eléctrica a su salida esté saturada. En este escenario, aunque atenúemos dicha señal en una etapa previa al ADC, seguirá estando saturada al entrar en el ADC y llegando, ya digital, al procesador.
2. El rango de valores de adquisición ADC debe ser menor que el de la señal analógica a la salida del hidrófono. En este caso, en la conversión del ADC, la señal digital quedará saturada. La solución será atenuar la señal previamente a la entrada del ADC.

Haciendo números con los datos del hidrófono seleccionado, siguiendo el diagrama de la Figura 2.6, si la presión máxima sin saturar que soporta es de 188 dB re $1\mu\text{Pa}$ y su sensibilidad es de -167 dB re $1\text{V}/\mu\text{Pa}$, tendremos que la amplitud máxima de la señal a la salida del hidrófono sin saturar es de 21 dB re 1V. Cabe decir que se está suponiendo que el valor de sensibilidad es válido en todo el ancho de banda frecuencial del hidrófono (la sensibilidad del hidrófono no es un valor constante con la frecuencia, de hecho, los fabricantes suelen dar la curva de la sensibilidad a lo largo de la frecuencia).

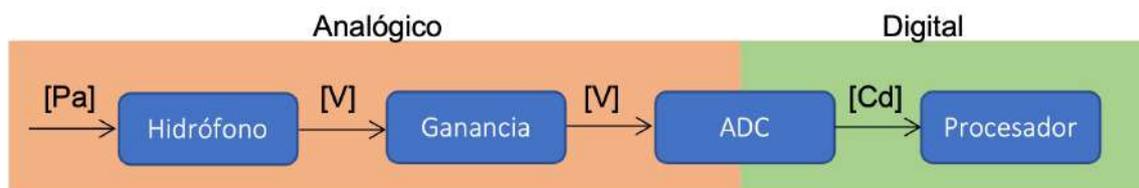


Figura 2.6: Diagrama de la cadena de entrada de audio. Pa: Pascales, V: Voltios, Cd: Cuentas digitales.

Ahora, desde la otra perspectiva, a la entrada del ADC, (dado que todavía no hemos hablado del ADC seleccionado, haremos los cálculos tomando el valor máximo de entrada por adelantado según la configuración que veremos en el Capítulo 4) tenemos un valor máximo de $0,375 * \sqrt{2} = 0,53[V]$ o lo que es lo mismo, -5.51 dB re 1V. Este valor, teniendo en cuenta la sensibilidad del hidrófono correspondería a una presión acústica de 161.49 dB re $1\mu\text{Pa}$ a su entrada. En otras palabras, el ADC, sin amplificación y atenuación analógica de la señal a su entrada, es capaz de adquirir señales correspondientes a una presión acústica de 161.49 dB re $1\mu\text{Pa}$ sin saturarlas. En acústica submarina es un valor bastante elevado (ver las curvas de Wenz en la Figura 1.1 del Capítulo 1).

Sin embargo podremos ajustar la presión máxima a adquirir, tal y como hemos visto en la Figura 2.5. Si tenemos valores de presión acústica entre [161 - 188 dB re $1\mu\text{Pa}$], podremos atenuar la señal

a la entrada del ADC para evitar la saturación en la adquisición. Cuando el fondeo esté situado en un lugar con muchas contribuciones ruidosas (suelen ser sonidos impulsivos antropogénicos cercanos a puerto) será lo que debemos hacer. Si por el contrario, tenemos unos valores de presión acústica entre [70 - 100 dB re $1\mu\text{Pa}$] podremos amplificar la señal analógica a la entrada del ADC para aprovechar mejor su rango de valores de adquisición.

Capítulo 3

Diseño hardware

En este capítulo se describe cómo se ha realizado el diseño electrónico para el dispositivo de monitorización acústica pasiva SAMARUC en el programa ALTIUM. El diseño hará hincapié en la grabación y almacenado del audio sin pérdidas durante largos periodos de tiempo. Es por ello que el requisito principal del diseño será la utilización de componentes electrónicos de muy bajo consumo.

La Tabla 3.1 resume las capacidades que el diseño global debe incluir. Entre ellas, una amplificación analógica programable para poder adaptarse a las diferentes zonas de fondeo y su ruido ambiente asociado, así como permitir la alimentación del diseño electrónico con diversas configuraciones de bloques de baterías de diferente voltaje y amperaje.

Sensibilidad del hidrófono preamplificado	-167 dB re 1V/ μ Pa
Rango de tensión de alimentación	4.3 - 60 V
Ganancia analógica programable	0 - 47.5 dB
Capacidad de almacenamiento	2 Tbytes
Frecuencia de muestreo	192 kHz
Ancho de banda -3 dB del dispositivo (ADC + hidrófono)	6 Hz - 96 kHz
Autonomía	6 meses

Tabla 3.1: Resumen de las especificaciones globales del diseño

3.1. Conversor analógico digital

Existen diferentes alternativas en el mercado en cuanto a ADCs o códecs de muy bajo consumo, en nuestro caso ha sido elegido el códec AIC3204 de Texas Instruments [32] debido a las siguientes razones:

- La mayoría de las placas de evaluación de bajo consumo de Texas Instruments lo llevan en su diseño, por tanto es fácil probar sus prestaciones.
- Ultra bajo consumo.

- Posibilidad de programar por software una ganancia analógica a la señal antes de entrar al ADC.
- Hasta 192 kHz de frecuencia de muestreo.

Es el componente encargado de adquirir la señal eléctrica proveniente del hidrófono y acondicionarla de manera adecuada para aprovechar de la mejor manera la sensibilidad del sensor (hidrófono). Además posee amplificación analógica (previa al ADC) y digital, como se puede observar en la Figura 3.1, donde se resumen en forma de bloques la entrada analógica del códec AIC3204.

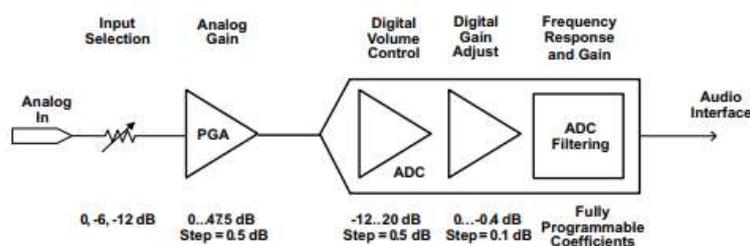


Figura 3.1: Cadena de entrada en el ADC, con las distintas fases por las que pasa la señal analógica.

El códec tiene una impedancia de entrada seleccionable (tres opciones: 10 kOhm, 20 kOhm y 40 kOhm), las cuales atenúan la señal analógica de entrada 0 dB, 6 dB o 12 dB respectivamente. Es posible aplicar una ganancia analógica a los 2 canales de la entrada estereoscópica seleccionada, de hasta 47.5 dB en pasos de 0.5 dB, también configurable por software. Esto permite que el ruido de cuantificación no sea amplificado. Además también se puede controlar la ganancia digital después de la conversión de la señal, existiendo además diferentes filtros configurables también por software y filtros anti-aliasing.

Aunque en nuestro diseño no vamos a tener que seleccionar entre las 3 entradas estéreo, el ADC puede elegir entre cualquiera de ellas para que sea la que entre en él. Además, pueden ser configuradas como entradas diferenciales. Una vez elegida una de las entradas, solo sus dos canales pueden ser amplificados por la etapa de ganancia analógica, como bien se ha comentado anteriormente. En nuestro caso, dado que trabajamos con señales mono, se podrán adquirir señales de dos hidrófonos.

3.2. Procesador y otros componentes

Se empleará un procesador de señal digital (DSP) de uso general y ultra bajo consumo de Texas Instruments (en concordancia con el códec seleccionado), el TMS320C5535 [33]. En la selección del procesador se tuvo en cuenta conseguir un compromiso entre precio, bajo consumo de energía, flexibilidad, fiabilidad y software disponible. Además, el TMS320C5535 tiene disponible las siguientes características:

- 4 Direct Memory Access Controllers
- Posibilidad de implementar FFTs

- Real Time Clock interno
- Un watchdog timer
- 2 buses Emmc/SD
- Buses I2C para control de periféricos
- I2S (InterICSound bus)
- 16 Puertos GPIO

La Figura 3.2 muestra el funcionamiento global en un diagrama de bloques. Se puede ver cómo los datos digitales del ADC llegan a través del bus de datos I2S al búfer PINGPONG (ver Sección 4.1). Con el objetivo de aislar los picos de corriente pasajeros del procesador producidos al grabar en las μ SD, las masas del circuito analógico y digital serán aisladas correctamente.

El diseño electrónico se activa mediante un interruptor de relé magnético para que no se desperdicie energía mientras el sistema se transporta al lugar de fondeo. Una vez el imán enroscado a la tapa superior de la carcasa sea desenroscado, el interruptor magnético queda en cortocircuito y el sistema inicializará el programa cargado en la EPROM, comenzando a almacenar datos.

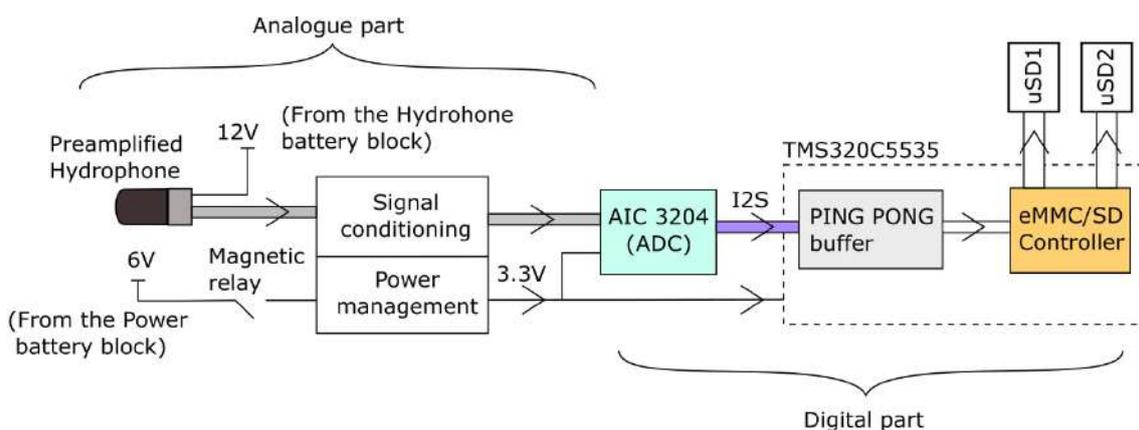


Figura 3.2: Visualización y conceptualización global del diseño electrónico a realizar.

En la Figura 3.2 también es posible observar el bloque llamado “Power Management”, el cual hace referencia a los componentes electrónicos necesarios para dotar al ADC y al procesador de los 3.3 Voltios requeridos, en concreto el regulador LT3995 de Analog Devices [34], el cual es capaz de reducir el voltaje desde un rango de tensión de entre [60 V y 4.3 V] a 3.3 V (ver montaje fabricante en la Figura 3.3). El regulador puede proporcionar un voltaje constante de 3.3 V siempre y cuando el voltaje del paquete de baterías sea superior a 4.3 V. Más concretamente es un regulador de conmutación reductor monolítico de frecuencia ajustable que acepta un amplio rango de voltaje de entrada de hasta 60 V. Consume solo 2,7 μ A de corriente de suministro mientras se regula sin carga y mantiene una alta eficiencia con corrientes de salida bajas mientras mantiene la ondulación de salida por debajo de 15 mV en una aplicación típica. El LT3995 puede suministrar hasta 3 A de corriente de carga y tiene un límite de corriente para limitar la disipación de energía durante un cortocircuito.

Por otro lado, el bloque de acondicionamiento de señal aísla la señal de audio analógica capturada por el hidrófono (incluidos sus componentes amplificados) de las señales digitales y la fuente

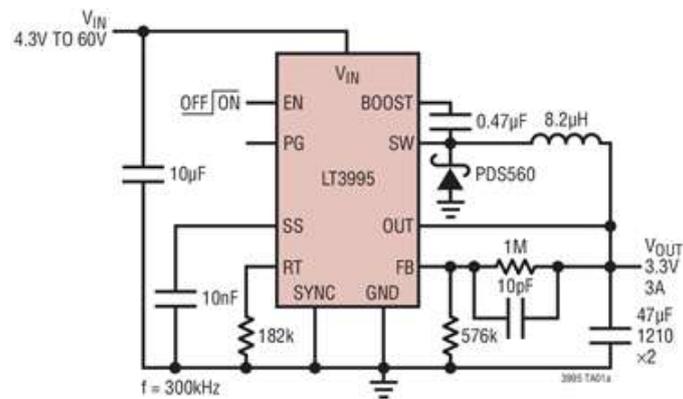


Figura 3.3: Esquemático del regulador LT3995.

de alimentación de los periféricos y el procesador al agregar dos masas independientes, reduciendo las interferencias de la electrónica del procesador en la señal de audio captada por el hidrófono. En resumen, las principales características del diseño que se ha realizado son las siguientes:

- 2 canales de entrada mono, uno para cada hidrófono.
- 16 bits de adquisición por canal.
- RTC asociado a una alimentación independiente.
- Alojamiento de 2 tarjetas μ SD.

3.3. Diseño PCB en ALTIUM

Una vez descritos los componentes más importantes se procede a la realización del diseño en el programa ALTIUM, el cual nos permite, tanto realizar los esquemáticos, como el Stack-Up y el rutado de los componentes teniendo en cuenta sus particularidades y recomendaciones del fabricante.

3.3.1. Esquemáticos

Se ha elegido un diseño jerárquico que dote de mayor claridad y orden al diseño. Dicho diseño consta de diversas partes tal y como se puede ver en la Figura 3.4. Se puede observar como el circuito tiene 2 conectores molex de 3 pines a los cuales se conectarán los 2 hidrófonos de la marca seleccionada. Estos 3 pines corresponden a los 3 cables por los que se conecta cada hidrófono al circuito, correspondientes a:

1. Señal de audio mono (output). La señal de audio tiene colocados condensadores de bloqueo de continua en serie tal y como marca el fabricante.
2. Masa (input/output). A esta masa va conectada parte del ADC seleccionado. Se obtiene del negativo del bloque de alimentación que proporciona la energía al hidrófono.

3. Alimentación entre [4.3 - 60 V]. Va conectado al positivo del bloque de alimentación del hidrófono.

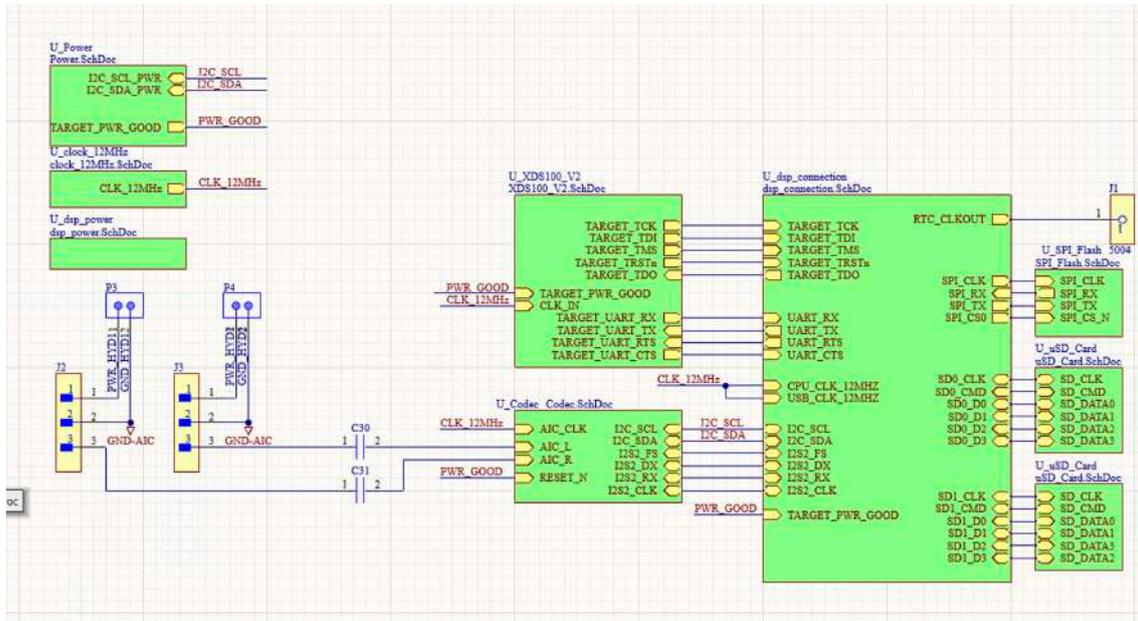


Figura 3.4: Captura del diseño jerarquico global.

A continuación se enumeran y comentan los diferentes esquemáticos que componen el diseño global:

- Procesador TMS320C5535. Es el componente principal del diseño y se ha dividido en dos partes, una de ellas conforme a los pines relacionados con su correcta alimentación (llamado dsp_power) y la otra (dsp_connection) relacionada con la utilización de sus prestaciones. Entre ellas se incluye la utilización de la UART (con el cual se establece parte de la conexión con el emulador XDS100v2 para poder programar dispositivo) el bus I2C, el bus I2S (a través del cual se establece la transferencia de los datos acústicos), el SPI (por el cual se accede a la memoria flash necesaria para grabar el .bin que se ejecutará de manera automática una vez se encienda el dispositivo y los 2 buses EMMC/SD con los cuales se realizará el envío de datos a 2 tarjetas SD.
- ADC AIC3204 (Codec). ver Figura 3.5
- El emulador XDS100v2 (llamado igualmente en el diseño) (ver Figura 3.6), gracias al cual se puede programar los registros del procesador a través del Code Composer Studio, el software que proporciona Texas Instruments para la programación firmware de sus componentes (en el Capítulo 4 se comentará este aspecto). En este esquemático también se ha dividido el componente en dos partes para facilitar un diseño más claro.
- La memoria flash (SPI_flash) a partir de la cual se puede cargar el programa realizado al inicializar la tarjeta.

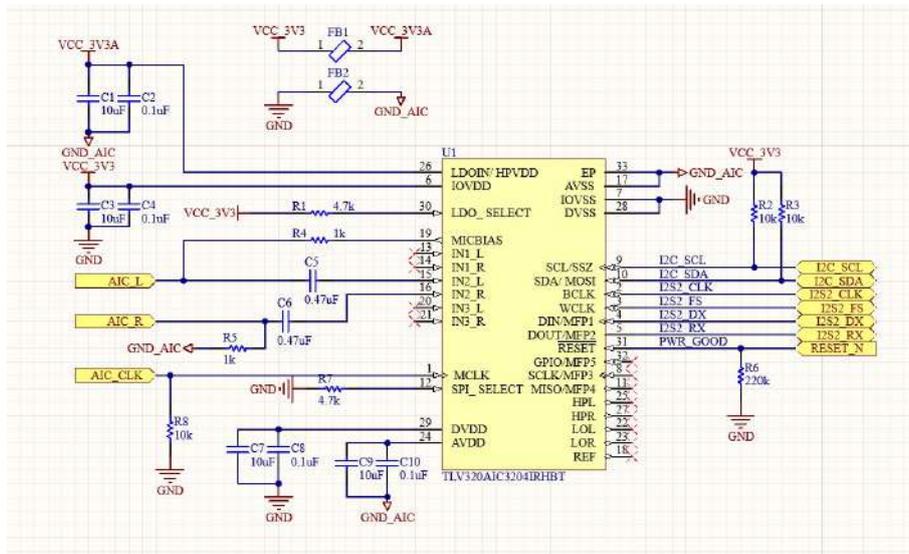


Figura 3.5: Captura del esquemático al código.

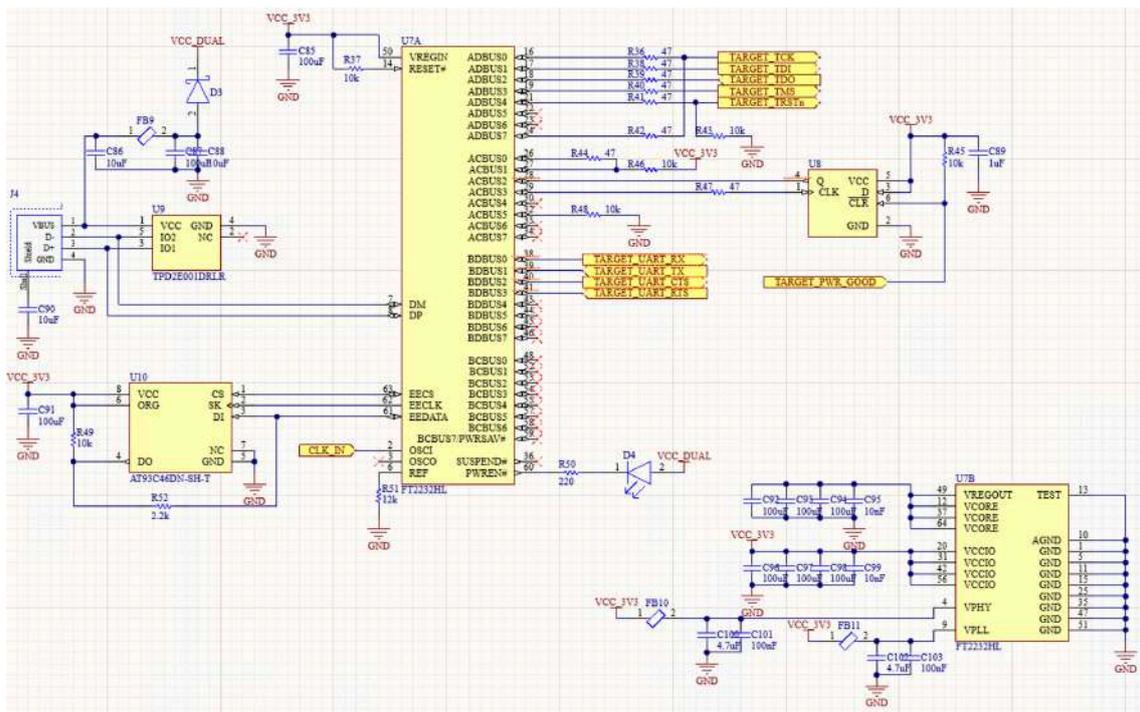


Figura 3.6: Captura del esquemático referente al emulador XDS100v2.

- El bloque de alimentación (Power). En él se incluye todo lo relacionado con los reguladores que se utilizan para proporcionar la tensión adecuada a los demás componentes. Además, incluye los pines de conexión de alimentación al sistema, así como un medidor de corriente y voltaje (conectado a través del bus I2C) justo antes de los reguladores para monitorizar la descarga de las baterías. En el se incluye el esquemática de la Figura 3.7, correspondiente al regulador LT3995.

- Existen tres GNDs diferentes: GND_AIC, GND y AGND. Respectivamente, GND_AIC es la masa a la cual se conecta tanto el negativo de las alimentaciones de los dos hidrófonos, así como la referencia de ambas señales mono de audio y parte del circuito del ADC (el AIC3204 es un códec). Por otro lado, GND es masa digital de la mayoría del circuito y AGND tiene que ver con la masa analógica de la parte dos del bloque de alimentación del procesador (dsp_power). Es importante realizar esta consideración ya que en el caso de no aislar GND_AIC de GND, al compartir el procesador y el ADC el mismo regulador, la corriente que demanda el procesador es mayor en el momento de la escritura de los datos, causando interferencias en el audio adquirido. Este es uno de los motivos por el cual la alimentación de los hidrófonos es también independiente de la alimentación del circuito electrónico.
- El diseño tiene dos osciladores: el ASDM1-12.000MHZ-LC-T, que controla el reloj global del sistema, y otro conectado al Real-time Clock (RTC) del procesador, el SSPT7F-12.5PF20-R, el cual es el encargado de despertar al sistema o ejecutar las posibles alarmas a programar, además, podrá ser alimentado independientemente a los componentes electrónicos del sistema. Ambos son dispositivos de bajo consumo de energía y tienen buena estabilidad a la temperatura. El oscilador del RTC tiene una estabilidad de frecuencia de ± 20 ppm, lo que da una deriva de $\pm 10,5$ min/año.
- Reset del sistema. En la Figura 3.9 se puede observar como se ha utilizado la salida “Good” del regulador TPS76601 (3.3 V a 1.3 V) como reset de todos los componentes del sistema.

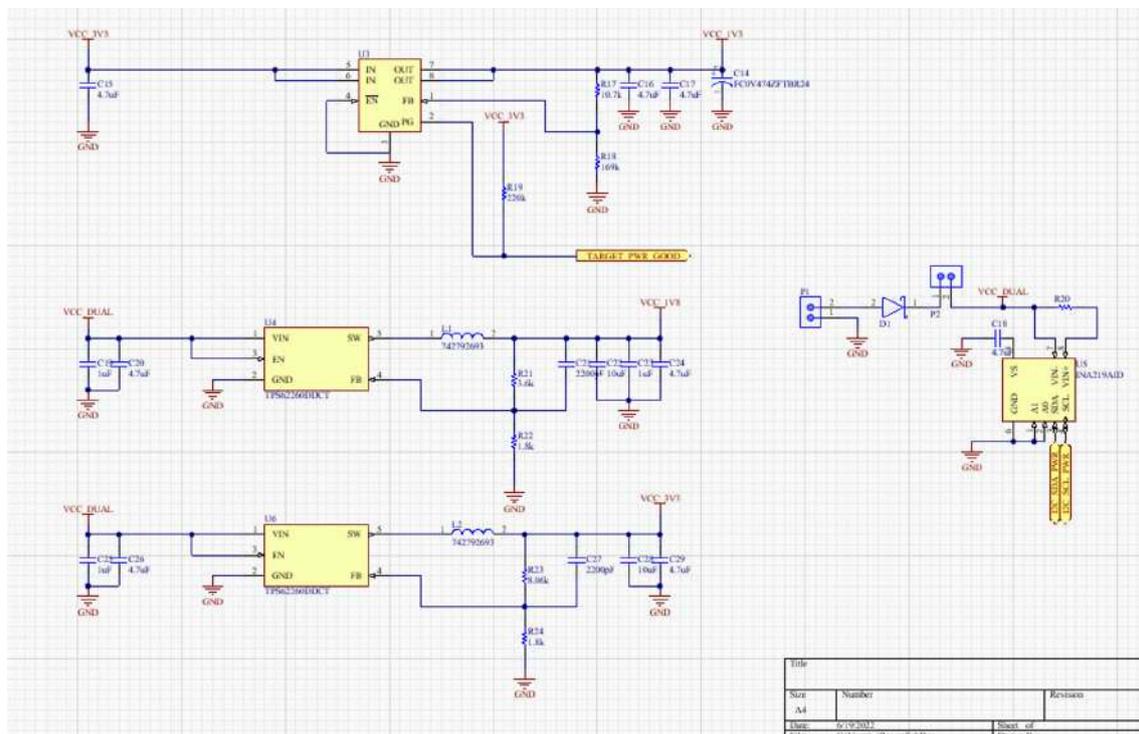


Figura 3.9: Captura del esquemático referente a los reguladores utilizados en el diseño, así como al medidor de voltaje y corriente.

- Monitorización de la corriente y el voltaje de la alimentación del circuito electrónico. En el esquemático “power”(ver Figura 3.9), además de establecer los diferentes voltajes necesarios

para la correcta alimentación de los componentes electrónicos elegidos a través de varios reguladores de tensión (en concreto a 3.3 V, 1.8 V y 1.3 V), también se incluye un medidor de la tensión y la corriente de las baterías que alimentan el circuito electrónico con el cual se monitorizará su descarga con el dispositivo en funcionamiento. Para ello, el conector de dichas baterías está presente también en el esquemático.

- Alimentación del RTC del procesador. En el mismo esquemático “power” se puede observar como a la salida del regulador TPS76601 que baja el voltaje a 1.3 V para la alimentación del RTC interno del procesador se ha colocado un supercondensador que permita mantener el RTC alimentado en caso de que se produzca algún corte de tensión en la alimentación del dispositivo.
- Alimentación y emulador a través del USB. En el esquemático XDS100v2 (ver Figura 3.6). Gracias al USB hembra se podrá debuggear el sistema y proceder a la programación firmware del diseño electrónico.

3.3.2. Stack-Up

Para el diseño del Stack-UP se ha elegido un diseño de 6 capas, tal y como se puede ver en la Figura 3.10. Concretamente, de arriba a abajo: TOP (S1), GND1, PWR, S2, PWR, BOTTOM (S3).

Top Overlay		Overlay				
Top Solder	SM-001	...	Solder Mask		0.4mil	4
Top (S1)	CF-004	...	Signal	1oz	1.4mil	Above
Dielectric 1	PP-006	...	Prepreg		8mil	4.1
GND1	CF-004	...	Signal	1oz	1.378mil	Above
Dielectric 2	Core-035	...	Core		15mil	4.7
PWR	CF-004	...	Signal	1oz	1.378mil	Below
Dielectric 3	PP-006	...	Prepreg		4mil	4.1
Dielectric 4	PP-006	...	Prepreg		4mil	4.1
S2	CF-004	...	Signal	1oz	1.378mil	Above
Dielectric 5	Core-035	...	Core		15mil	4.7
GND2	CF-004	...	Signal	1oz	1.378mil	Below
Dielectric 6	PP-006	...	Prepreg		8mil	4.1
Bottom (S3)	CF-004	...	Signal	1oz	1.4mil	Below
Bottom Solder	SM-001	...	Solder Mask		0.4mil	4
Bottom Overlay			Overlay			

Figura 3.10: Captura del STACKUP realizado.

La razón por la cual se ha elegido un diseño de 6 capas es debido a que el footprint del procesador TMS320C5535 es bastante complejo al tratarse de un encapsulado BGA, y se hace necesario contar con una capa de señal extra para poder acceder de una manera más sencilla a sus pines. En futuros diseños se buscarán procesadores de la misma familia que el C5535 (familia de bajo consumo C5000 de Texas Instruments) para hacer un rutado más sencillo utilizando 4 capas.

Cabe destacar que el diseño no consta de ningún par diferencial y que la velocidad del reloj que demanda una aplicación con una tasa de muestreo permite no fijar ninguna regla extra en el



Figura 3.11: Detalle de las vías utilizadas.

diseño del circuito a la hora de rutar. Por otra parte, en la Figura 3.11 se muestran las vías planteadas en el diseño para proceder al rutado. A su vez, se comprueba que las impedancias generadas son las adecuadas, es decir, 50 Ohmios (ver detalle en la Figura 3.12).

	Top Ref	Bottom Ref	Width (W1)	Impe...	Devia...	Delay...
<input checked="" type="checkbox"/>		2 - GND1	14.677mil	50	0.01%	150.44...
<input type="checkbox"/>					0%	
<input checked="" type="checkbox"/>	2 - GND1	4 - S2	7.853mil	50	0%	177.32...
<input checked="" type="checkbox"/>	3 - PWR	5 - GND2	7.853mil	50	0%	177.32...
<input type="checkbox"/>					0%	
<input checked="" type="checkbox"/>		5 - GND2	14.677mil	50	0.01%	150.44...

Figura 3.12: Captura de las impedancias entre las capas.

3.3.3. Rutado

Gracias al stackup de PCB de 6 capas, hay una capa completa dedicada a PWR. En general, esta es una buena práctica en una PCB de 6 capas, ya que libera área en las superficies para los componentes, y es más fácil obtener energía para estos componentes a través de vías. Poner PWR y SIG en las capas interiores da más desacoplamiento con un par de planos PWR/GND.

Un footprint BGA en procesadores es típico de un controlador de interfaz de alta velocidad el cual requiere mucha corriente a múltiples voltajes, por lo que muchas de las bolas estarán conectadas a la alimentación y a masa. Dedicar una sola capa a la alimentación permite hacer split en el

plano para que se puedan usar múltiples voltajes a alta corriente si es necesario. De esta manera, no se necesita superponer estos planos a diferentes voltajes, lo que evita un problema adicional de EMI.

En la Figura 3.13 se puede visualizar tanto la huella BGA del procesador elegido, así como un ejemplo del rutado que se ha llevado a cabo. Cabe decir que se incluye el enlace de GitHub donde se ha subido y puede ver el proyecto en Altium de manera completa.

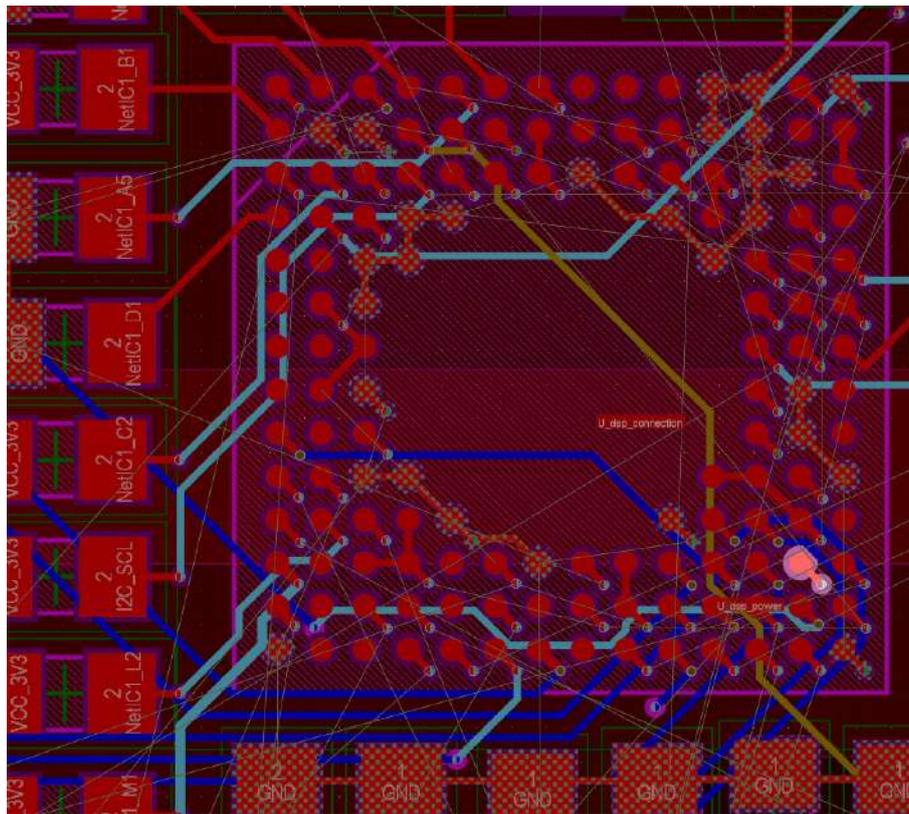


Figura 3.13: Footprint del rutado en una fase intermedia.

Finalmente, en la Figura 3.14 se puede ver cómo se posicionan los componentes en la PCB, en una fase casi definitiva.

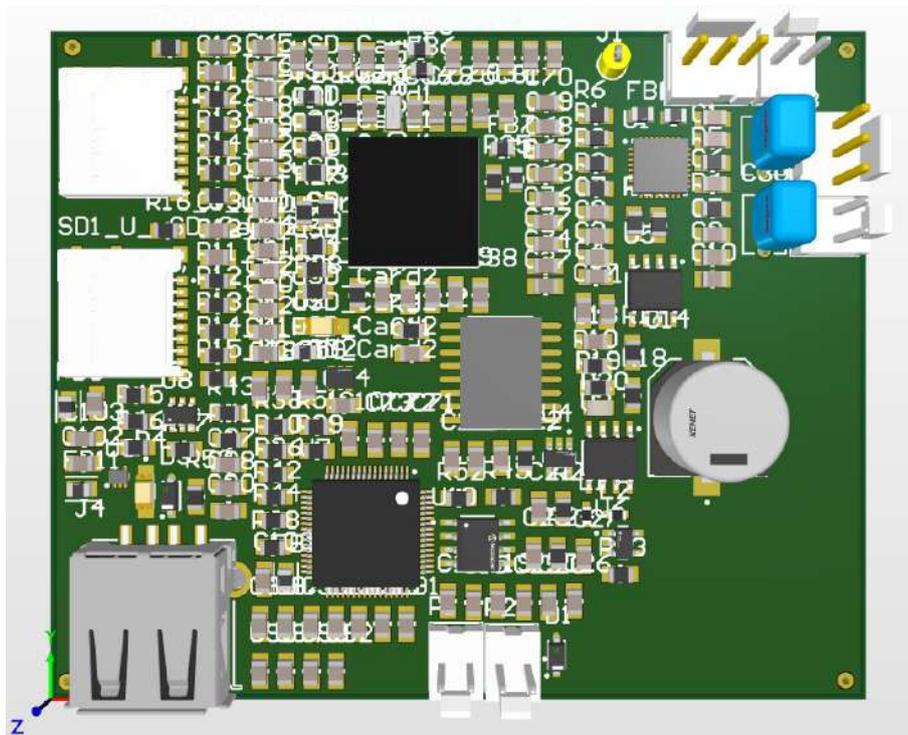


Figura 3.14: Captura 3D del posicionamiento de los componentes en la PCB.

Capítulo 4

Programación firmware

En este trabajo también se ha realizado la programación firmware, con el objetivo de probar el funcionamiento de los componentes descritos en el Capítulo 3.

4.1. Código generado

Una de las partes más importantes de la programación es la de configurar el códec seleccionado. En concreto se ha seleccionado una frecuencia de muestreo de 192 kHz y una impedancia de entrada de 10 kOhmios. Esta parte es relevante debido a que el hidrófono elegido tiene como salida de audio esa misma impedancia.

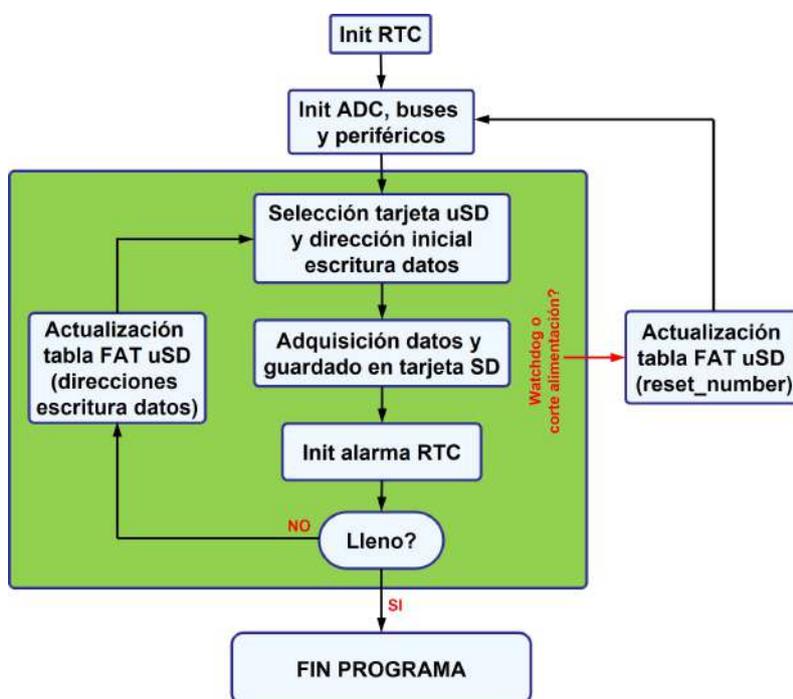


Figura 4.1: Flujo global del diseño firmware desarrollado.

La Figura 4.1 resume de manera global el firmware diseñado, el cual, básicamente permite la adquisición y grabación sin pérdidas del audio captado por el/los hidrófonos en las tarjetas μ SD. El primer paso consiste en el formateo de las dos tarjetas μ SD donde se configura la ganancia analógica de los 2 canales del ADC, el ciclo de minutos ON/OFF y la frecuencia de muestreo adecuadas para el fondeo a realizar. Seguidamente, el programa se inicializa leyendo estos parámetros comenzando a grabar los minutos ON configurados. Una vez pasados estos minutos, el sistema se pondrá en modo de bajo consumo los minutos OFF. Para despertarse, se ha programado una alarma en el Real Time Clock (RTC) que posee el procesador. La grabación de los datos acústicos en las tarjetas se realiza de forma secuencial, es decir, hasta que no se llene una de ellas no empezará la siguiente. Cabe destacar que se ha configurado un watchdog de forma que, si el programa se queda en alguna parte del código por mas tiempo del que se ha programado, éste hará que el programa vuelva a empezar.

La parte más interesante del código es la que permite capturar el audio mientras a su vez se guarda en las tarjetas μ SD. Gracias a la estructura PINGPONG, el código utiliza 2 Direct Memory Access (DMA) uno dedicado a la captura de datos desde el ADC y otro para el almacenamiento de los datos en la/s tarjetas μ SD a través del controlador Emmc/SD.

De una manera más concreta, la configuración de Ping-Pong permite utilizar un búffer (buffer PING_PONG), que se divide internamente en dos subbuffers (llamados subbuffer_PING y subbuffer_PONG), lo que permite comprobar el estado de cada uno de ellos, realizándolo todo en paralelo, gracias a la integración de los dos DMA con la memoria y los periféricos. La Figura 4.2 muestra el funcionamiento de la estructura Ping-Pong. Cuando se configura un DMA (DMA1, por ejemplo) para transferir datos del ADC a uno de los subbuffers, DMA2 enviará los datos del otro al bus Emmc/SD, alternando estas dos tareas mientras el ADC llena el subbuffer donde está trabajando.

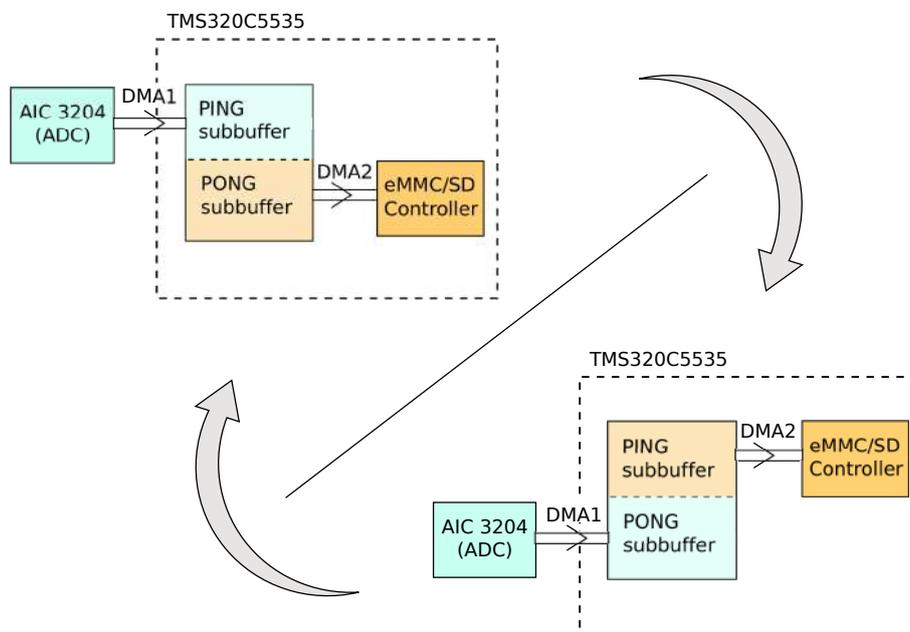


Figura 4.2: Representación del modo de funcionamiento Ping Pong utilizando 2 Direct Memory Access.

Conforme cada ciclo de grabación se completa, el código genera y va completando un archivo

txt con la información de en qué fecha y hora ha sido grabada dicho ciclo, el número de ciclos que se llevan grabados, así como información referente a los periféricos conectados por i2C, es decir, la tensión a la entrada del diseño propuesto al final de cada ciclo. En el Anexo B se incluyen varias partes del código firmware programado.

4.2. Sistema de escritura en memoria

La necesidad de ampliar la memoria del sistema al grabar audio a 192 kHz plantea el reto de diseñar un nuevo sistema de escritura propio, capaz de direccionar toda la memoria disponible (2 Tbytes) sin necesidad de utilizar el sistema de almacenamiento exFAT o FAT32, las cuales ralentizan el funcionamiento del sistema conforme se tienen que direccionar en la memoria disponible una gran cantidad de archivos (el dispositivo prevé llegar a grabar más de 17000 archivos de 110 MBytes cada uno). Con el fin de evitar el manejo de archivos se ha diseñado un sistema de escritura directo en memoria de forma que los datos quedan grabados en “crudo”.

El proceso de formateo divide la memoria en tres partes de diferentes tamaños (ver Figura 4.3): el encabezado (en color blanco), la tabla de asignación de datos (en color amarillo y verde) y la sección de datos (en color rosa). El encabezado se utiliza para configurar el funcionamiento del dispositivo y almacena información sobre la frecuencia de muestreo, el ciclo de grabación (minutos ON/OFF) y la ganancia del ADC. Toda esta información se carga al encender el dispositivo. La tabla de asignación de datos almacena la dirección de memoria inicial y final donde están guardadas las muestras audio de cada ciclo de grabado, así como la cantidad de resets producidos. Se define reset a las veces el programa firmware diseñado ha comenzado desde cero, ya sea por cortes en la alimentación del dispositivo o porque el watchdog ha inicializado otra vez el programa.

Finalmente, la sección de datos almacena las muestras de la grabación en los diferentes bloques de la tarjeta μ SD de una manera totalmente ordenada. El sistema de escritura diseñado, al escribir directamente en memoria de una manera secuencial, mejora la velocidad y baja la latencia en el acceso de lectura y escritura. Una vez recuperado el fondeo, se extraerán las tarjetas μ SD y se procederá a la creación de los archivos en un disco duro externo leyendo en la sección de datos correspondiente de dichas tarjetas desde el bloque inicial al bloque final (indicados en la tabla de asignación de archivos).

En el sistema diseñado se ha incluido la condición de que únicamente se guarde la dirección del bloque inicial y final en la tabla de asignación si ha acabado el ciclo de grabado. De esta manera, si se produce un corte de tensión en la alimentación del dispositivo durante la adquisición de los datos, las muestras del ciclo que se ha interrumpido quedarán sin utilizar, ya que cuando el programa se recupere, el sistema comenzará a grabar un nuevo ciclo en la última dirección de memoria inicial guardada, es decir, donde había empezado el ciclo que se interrumpió. En la Figura 4.3 se puede observar un detalle del sistema de escritura en memoria diseñado.

	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8	byte 9	byte 10	byte 11	byte 12	byte 13	byte 14	byte 15	byte 16	byte 17	byte 18	byte 19	byte 20	byte 21	byte 22	byte ...	byte 512										
Block 0	fs																						Reset number ⁽¹⁾		Last written block ⁽²⁾		Fat Block entry		Fat block offset		uSD number ⁽⁴⁾		Header	
	Minutes on		Seconds off		Gain		Block end																											
Block 1	Sep: 13	Reset	File #: 1	Block start	Block start	Block start																												
Block 2	Sep: 13	Reset	File #: 33	Block start	Block start	Block start																												
...	Sep: 13	Reset	...	Block start	Block start	Block start																												
...	Sep: 13	Reset	...	Block start	Block start	Block start																												
Block 2047	Sep: 13	Reset	File #: 65473	Block start	Block start	Block start																												
Block 2048	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word									
Block 2049	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word									
...									
...									
...									
Block end	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word	Data Word									

Figura 4.3: Detalle en forma de tabla del sistema de escritura en memoria.

donde:

1. *Resetnumber*: Controla por si se ha producido algún reinicio por fallo temporal de la alimentación. Inicialmente está a 0 y nada mas empezar a escribir se coge su valor y se sustituye por *Resetnumber*+1.
2. Si *Resetnumber* \neq 0: Last written block es el lugar de “Data” donde tiene que seguir escribiendo.
3. *Resetnumber* \neq 0: Fat Block entry y Fat block offset es el lugar de la “Block-File-Time Allocation Table” donde tiene que seguir escribiendo.
4. *μ SDnumber*: Número de la tarjeta μ SD en dos dígitos (XX). Se emplea para añadir al “logXX.txt” y que no se sobrescriban al extraer datos de un fondeo que emplea más de una tarjeta μ SD en un mismo directorio. Se crean un log por cada tarjeta desde la que se vuelcan datos.
5. La codificación de Month, Day y Hour en 16 bits es como sigue: Month (6 bits), Day (5 bits), Hour (5 bits).

Capítulo 5

Resultados

En este capítulo se ha tratado de comprobar el funcionamiento de parte del diseño electrónico mostrado en el Capítulo 3 así como del firmware programado en el Capítulo 4. Para ello, y dado que varios componentes del diseño planteado no tenían la disponibilidad requerida, se ha utilizado un diseño auxiliar de PCB, el cual, unido a la placa de evaluación EZDSP5535, permitirá probar la mayoría de funcionalidades del diseño propuesto del Capítulo 3, así como el firmware de una manera global.

5.1. Placa de evaluación eZDSP5535

La placa de evaluación EZDSP5535 (ver Figura 5.1a) ha servido para realizar las pruebas necesarias del funcionamiento y programación tanto del procesador TM320C5535 como del códec AIC3204 utilizados en el diseño electrónico del Capítulo 3, además de la conexión entre el procesador y el códec, las diferentes tensiones a las cuales hay que alimentar el procesador, etc. En la Figura 5.1b se muestra un esquema básico de las principales características de la placa EZDSP5535, resumidas en la Tabla 5.1.

Características
El DSP TMS320C5535
Un Stereo Codec TLV320AIC3204 (stereo in, stereo out)
Un conector para tarjetas μ SD
Un interfaz USB 2.0 para la alimentación de la placa
8 Mbytes SPI flash. Pantalla OLED a través de I2C
5 LEDs controlados por el usuario
2 switches
3 INA219 para la medida de corriente y tensión a través de I2C
Un emulador JTAG (XDS100) a través de USB
Wireless interface
Expansion edge connector
Test points para la monitorización de la tensión
Compatibilidad con Code Composer Studio

Tabla 5.1: Características de la Placa de evaluación comercial EZDSP5535

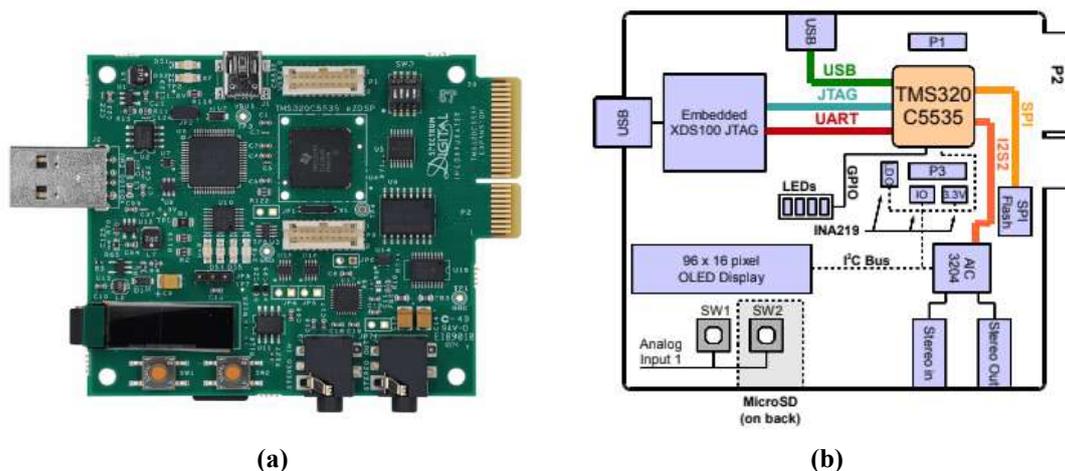


Figura 5.1: (a) Imagen de dicha placa de evaluación EZDSP5535 y (b) diagrama con sus principales componentes.

5.2. Diseño auxiliar

Este diseño auxiliar se ha utilizado para poder probar la mayor parte del código desarrollado en el Capítulo 4 y funcionalidades que se han incluido en el diseño electrónico propuesto en el Capítulo 3.

Una de sus funciones es recibir como inputs las salidas de 2 bloques de baterías, el encargado de la alimentación de la placa electrónica comercial eZDSP5535 y el correspondiente al único hidrófono conectado, además se conectan a ella los 3 cables procedentes de dicho hidrófono (ver en el esquemático de la Figura 5.2). A su vez, como output, a través de un conector hembra sale la alimentación a la placa de evaluación comercial. En el Anexo A se puede ver la lista de componentes comprados para la fabricación de la PCB.

Cabe destacar que se ha podido acceder al segundo bus EMMC/SD del procesado gracias a la conexión con los puertos 1 y 3 de la placa de evaluación. El diseño auxiliar incluye la conexión física donde se monta la segunda tarjeta μ SD. La Tabla 5.2 resume las funcionalidades aportadas por la placa auxiliar.

Funcionalidad	Resultado
Ampliación memoria	Utilización del 2do controlador EMMC/SD del procesador
Eliminación intererencias audio	Alimentación del hidrófono y electrónica independientes
Entrada audio analógica	Diseño acorde a especificaciones del fabricante del hidrófono

Tabla 5.2: Funcionalidades aportadas por el diseño auxiliar

Se puede observar como este diseño independiza la alimentación del hidrófono y la alimentación de la placa de evaluación. En el Capítulo 3 se describe este problema resolviéndose mediante la utilización de distintas alimentaciones y GND, ya que se comprobó que si se compartían, la corriente que exige el DSP cada vez que accede a la μ SD para guardar los datos generaba crosstalk en la señal de audio grabada por el hidrófono.

En la Figuras 5.3a y 5.3b se muestran respectivamente el rutado del diseño auxiliar y el aspecto

final del diseño auxiliar ya montado encima de la placa comercial. Además, en esta última figura se pueden ver las conexiones los cables del hidrófono y con el relé magnético.

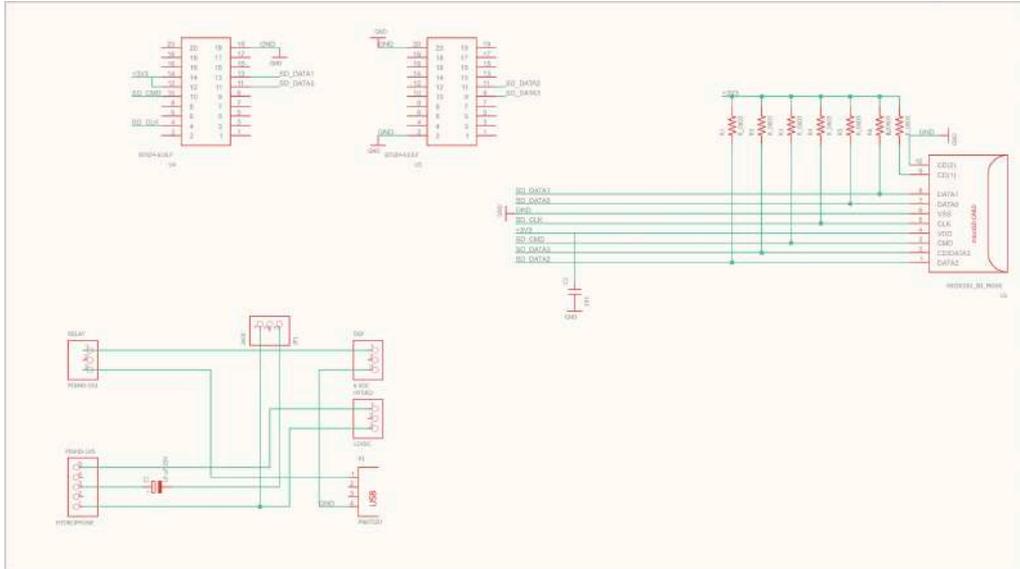


Figura 5.2: Esquemático del diseño auxiliar.

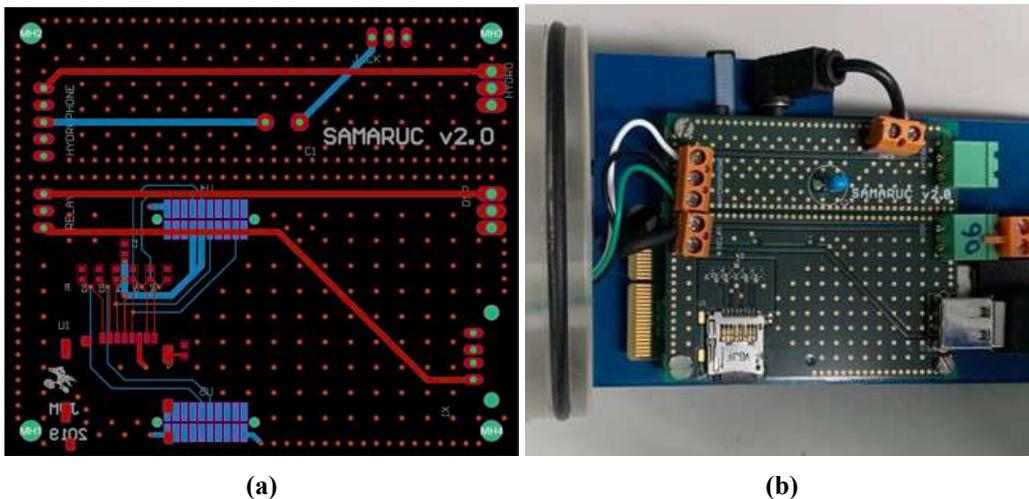


Figura 5.3: (a) Rutado de la PCB del diseño auxiliar y (b) Detalle del diseño auxiliar fabricado y montado

La conexión entre la placa de evaluación comercial y la placa del diseño auxiliar se hace por los puertos 1 y 3 de la primera. En el Anexo A se lista los detalles de los componentes del diseño de esta placa auxiliar.

La Tabla 5.3 muestra las principales diferencias entre las funcionalidades del conjunto placa de evaluación EZDSP5535 + PCB auxiliar y el diseño propuesto en el Capítulo 3. Tal y como se puede ver en ella, las funcionalidades que no han sido testeadas son la adquisición de varias entradas analógicas, la alimentación independiente del RTC, la medición del voltaje y la corriente a la entrada del circuito antes del primer regulador y la utilización del regulador LT9335 de Analog Devices

que se encarga de permitir la conexión de bloques de baterías de hasta 60 V para la alimentación del diseño electrónico. Cabe recordar que el diseño propuesto en el Capítulo 3, permite la conexión de 2 hidrófonos y el diseño auxiliar solo uno.

Funcionalidad	EZDSP5535 y Auxiliar	Propuesto
Sistema de escritura en memoria μ SD	x	x
Utilización 2 buses Emmc/SD	x	x
Masas independientes	x	x
Activación dispositivo mediante relé magnético	x	x
Audio input ganancia analógica configurable	1	2
Alimentación independiente del RTC		x
Medición voltaje/corriente a la entrada del circuito		x
Mayor rango de voltage de alimentación		x

Tabla 5.3: Funcionalidades cumplidas por el diseño auxiliar vs diseño propuesto

5.3. Alimentación

Para la alimentación de la placa de evaluación EZDSP553 se ha tenido en cuenta que posee el regulador a su entrada, y por tanto se ha tenido que realizar un bloque de baterías que no supere dicho valor, además de Figura 5.4

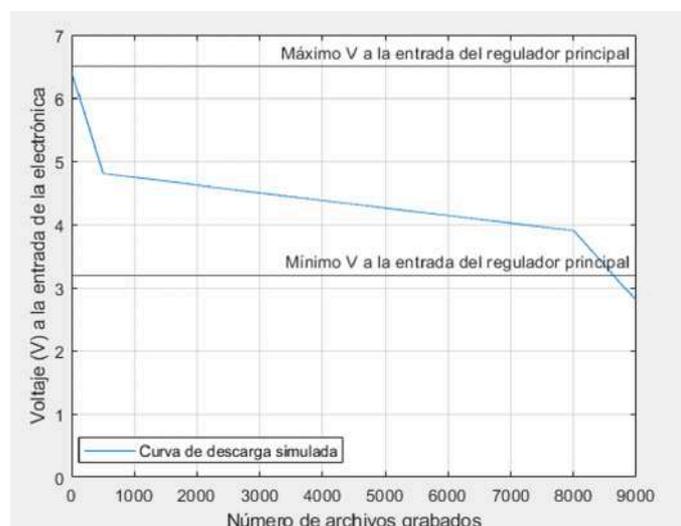


Figura 5.4: Curva simulada de la descarga de la alimentación de la electrónica.

La tensión de alimentación del hidrófono y del DSP son bastante distantes, en concreto, el hidrófono, por recomendación del fabricante, debe estar alimentado a 12 V aproximadamente y la placa de evaluación entre 2.8 V y 6.5 V que marca su regulador de entrada.

Los bloques de baterías para pilas tipo D que serán los encargados de alimentar tanto los hidrófonos como a la electrónica, en este caso la placa electrónica auxiliar + la placa de evaluación EZDSP5535. En la Figura 5.5a se el resultado final del bloque de baterías terminado, emplazando

4 pisos de pilas en serie, cada uno de ellos con la distribución que se puede observar en la Figura 5.5b.

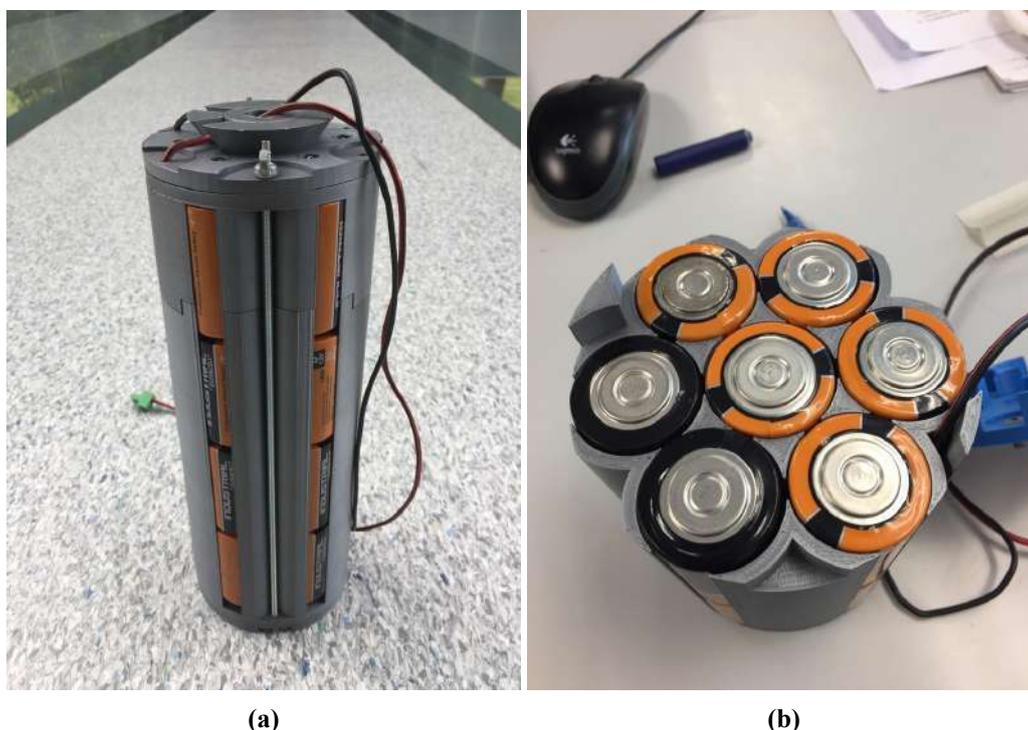


Figura 5.5: Imágenes de (a) bloque de baterías ensamblado que alimenta a la electrónica del dispositivo y (b) distribución de las baterías en el bloque diseñado.

En el caso llenar los 7 tubos utilizando 28 pilas tipo D de 19 Ah, se obtendrá una potencia de 638.4 [Wh] ($7 * 19[Ah] * 4 * 1,2[V]$), tomando como 1.2 V la salida nominal de las baterías tipo D que se van a utilizar.

Gracias a la colocación de 4 pilas tipo D en serie, en el bloque de baterías de la electrónica se obtendrá una tensión de 6.3 V al comienzo de la grabación (con las pilas nuevas), tensión que se sitúa dentro de la tensión máxima del regulador que se utilizará en el diseño electrónico del Capítulo 3. Además, las pilas tipo D, cuando están casi descargadas se quedan con una tensión de entre 0.7 y 0.8 V cada una de ellas (una tensión total de entre 2.8 y 3.2 V), justo cuando el regulador deja de funcionar. De esta manera aprovecharemos todo el ciclo de descarga de las pilas. En la Figura 5.4 se ha representado una curva de descarga ficticia para ejemplificar cómo se quieren aprovechar todos los amperios hora (Ah) de las baterías tipo D.

5.4. Resultados

Se realizaron 2 fondeos de mas de 3 meses de duración, donde el bloque de baterías que alimenta la electrónica fué llenado con 20 pilas tipo D de 13 [Ah] (5 de los 7 tubos) en ambos fondeos. La potencia disponible fue por tanto de $5 * 13[Ah] * 4 * 1,2[V] = 312Wh$. Uno de ellos fue configurado con un ciclo de 5 minutos ON, 5 minutos OFF y 1.5 TBytes de memoria. El segundo fue configurado con un ciclo de 5 minutos ON y 10 minutos OFF con 1 TByte de memoria (ver

resumen en Tabla 5.4). Ambos fondeos se realizaron a unos 80 metros de profundidad.

Fondeo (localización)	Ciclo	Num. Archivos	Mem. ocupada	Estado
DM Levantino Balear	5/5	8712	933 GBytes	Baterías sin carga
DM Estrecho, Mar Alborán	5/10	8238	895 GBytes	Baterías sin carga

Tabla 5.4: Resumen de los resultados de los fondeos

Se comprobó que el archivo log.txt generado por el firmware donde se puede ver el seguimiento de como ha ido funcionando y grabando el código implementado ha guardado correctamente la información. Aquí será donde se guardará en un futuro la tensión a la entrada de la electrónica y así poder ver la curva de descarga real de las baterías cuando el diseño propuesto en el Capítulo 3 sea fabricado.

Respecto al consumo que la placa electrónica comercial tiene, sabiendo los datos de la Tabla 5.4 y la potencia que tienen ambos bloques de baterías, podemos obtener cual ha sido los amperios consumidos cuando el código está en funcionamiento (x_{ON}) y cuando el sistema está en reposo (x_{reposo}) en los minutos OFF. Tal y como se ha comentado en el anteriormente, la configuración y pilas utilizadas en los bloques de baterías de la electrónica de ambos fondeos poseen una potencia a consumir de 312 Wh. Además sabiendo que cada fondeo tiene ciclos ON/OFF diferentes (DM Levantino Balear 5/5 y DM Estrecho y Mar de Alborán 5/10), obtenemos el sistema de ecuaciones 5.1:

$$312[Wh] = 3,3[V] * x_{ON}[A] * 686,5[h] + 3,3[V] * x_{reposo}[A] * 1373[h] \quad (5.1)$$

$$312[Wh] = 3,3[V] * x_{ON}[A] * 726[h] + 3,3[V] * x_{reposo} * 726[h] \quad (5.2)$$

Cabe destacar que las horas de funcionamiento de cada fondeo se han dividido en horas x_{ON} , y horas x_{reposo} . Resolviendo el sistema de ecuaciones, se obtiene que:

$$x_{reposo} = 8,23mA; x_{ON} = 122mA \quad (5.3)$$

Lo que comprueba que el modo de bajo consumo se activa correctamente (sólo se consumen 8.23 mA) por el firmware. Cabe decir que, aunque se esperaba un valor más bajo cuando el dispositivo está adquiriendo y almacenando los datos en las memorias μ SDs, el consumo obtenido es algo lógico para una placa de evaluación con tantos componentes alimentados con el reloj del sistema.

En la Figura 5.6 se puede ver un ejemplo de la forma temporal y el espectrograma de uno de los archivos de audio grabados. Se comprueba como se ha aprovechado de una manera correcta el rango dinámico del ADC y el ajuste de la ganancia analógica que éste permite.

En las Figuras 5.7a y 5.7b, se puede observar respectivamente la forma en la que el dispositivo se divisó desde el barco una vez alcanzó la superficie del mar y en las condiciones en las cuales se encontraban despues de más de 3 meses de fondeo.

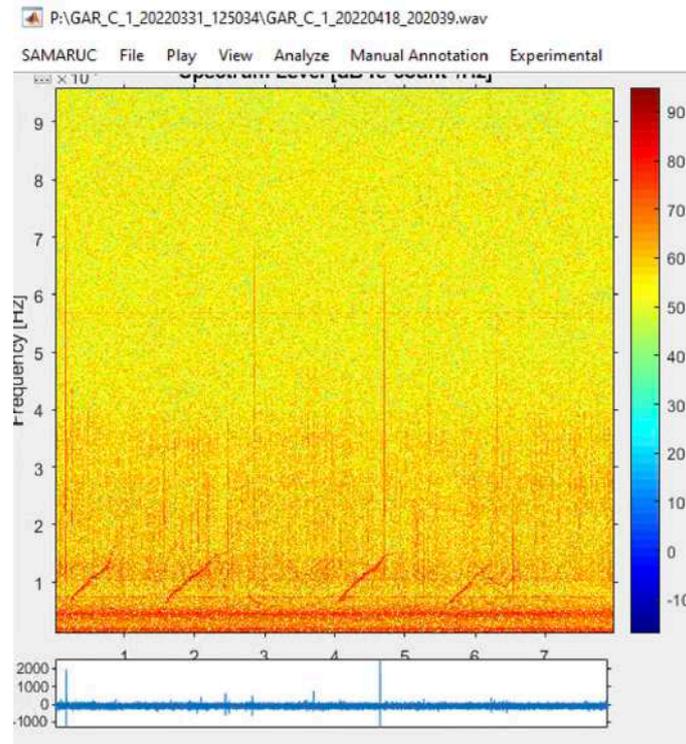


Figura 5.6: Detalle del espectrograma de uno de los archivos grabado así como su forma temporal.

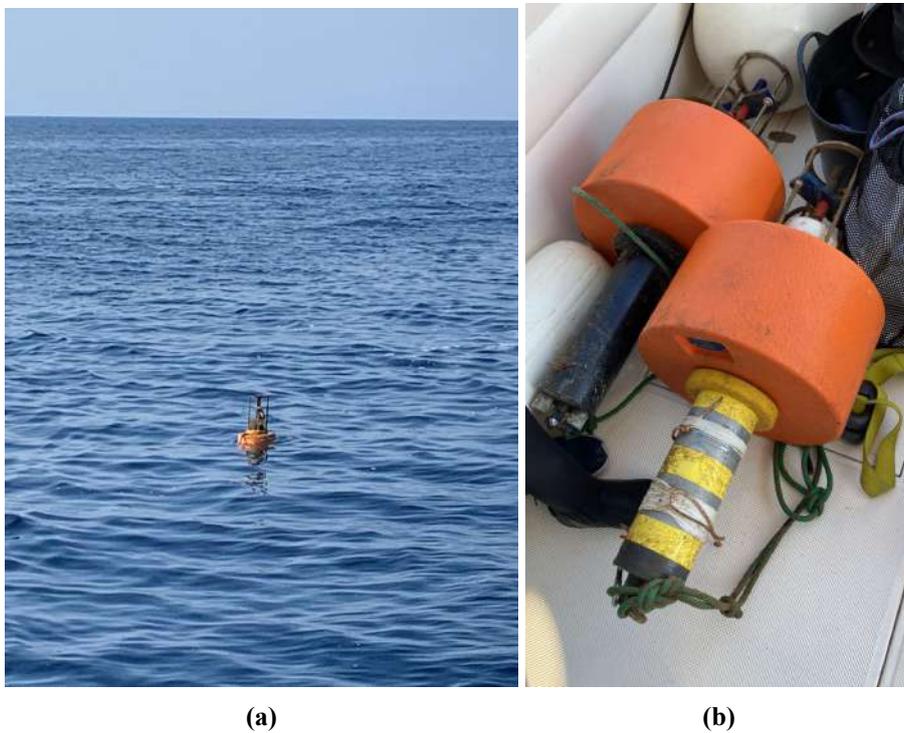


Figura 5.7: Imágenes del SAMARUC (a) en el momento de su visualización una vez acaba de subir desde el fondo marino y (b) una vez recuperado del fondo del mar después de 3 meses

Capítulo 6

Conclusiones y líneas futuras

Se ha diseñado la electrónica del dispositivo PAM SAMARUC teniendo en cuenta los aspectos más importantes en su etapa de conceptualización y realizando el proyecto en Altium, desde la elección de los componentes hasta su rutado y conexión teniendo en cuenta las pistas más críticas. Además se ha programado el firmware necesario para comprobar el funcionamiento correcto del diseño electrónico realizado. Esta comprobación se ha realizado por medio de la placa de evaluación comercial EZDSP5535, diseño en el cual se ha basado parte del Capítulo 3, con el objetivo de poder probar las funcionalidades requeridas. Entre todas ellas, la adecuada captación de los datos sin pérdidas a través de la estructura PingPong utilizando dos DMAs, así como la activación del modo de bajo consumo del procesador y la utilización del segundo bus de datos EMMC/SD. Para ello se ha realizado una placa electrónica de apoyo la cual se ha conectado placa de evaluación comercial por uno de sus slots.

Gracias a esta placa electrónica y al diseño auxiliar realizado se ha podido comprobar que, con una gran probabilidad, parte del diseño electrónico realizado en el Capítulo 3 es decir, tanto la calidad de audio obtenida (teniendo en cuenta la alimentación del hidrófono y las masas digitales y analógicas), así como el segundo controlador EMMC/SD del procesador va a funcionar de manera correcta. Además, gracias a la adquisición del módulo de evaluación EZDSP5535, la programación firmware explicada en el Capítulo 4 ha sido testeada satisfactoriamente. De hecho, tal y como se detalla en el Capítulo 5, este diseño auxiliar ha sido probado en cuatro fondeos diferentes, todos ellos con buenos resultados, poniendo a prueba sus carencias, a falta de la realización del diseño completo explicado en el Capítulo 3.

Los próximos pasos serán los siguientes: primeramente se procederá a la fabricación al testeo del diseño electrónico una vez sea recibido. Gracias a las pruebas reales realizadas con el diseño auxiliar, se ha comprobado que la funcionalidad de dotar al RTC interno de una alimentación independiente, de forma indirecta con un condensador con mucha capacidad de forma que evite posibles cortes de tensión. Además, se probarán los 2 canales de audio (recordemos que los hidrófonos son mono) con 2 hidrófonos diferentes. En caso del firmware, se probará el modo de cero consumo cuando el dispositivo está en el ciclo de descanso. Además se plantea incluir medidores de la temperatura interna del dispositivo en el diseño hardware con el objetivo de comprobar que el rendimiento de la electrónica no disminuye debido al sobrecalentamiento de los componentes al estar en un entorno cerrado y sin ventilación.

Por último, cabe destacar que una parte de este trabajo fin de master ha sido publicada en la

revista Sensors [28], concretamente la centrada en Capítulo 4.

Bibliografía

- [1] M. Van Der Graaf S. Ainslie M. Andersson M. André M. Borsani J. Brensing K. Castellote M. Cronin D. et al Dekeling, R. Tasker. Monitoring guidance for underwater noise in european seas: Monitoring guidance specifications, part ii: Monitoring guidance specifications. *JRC Scientific and Policy Report EUR 26555*, pages 1–49, 2014.
- [2] Sofie Van Parijs, Christopher Clark, Renata Sousa-Lima, Susan Parks, Shannon Rankin, Denise Risch, and Ilse Van Opzeeland. Management and research applications of real-time and archival acoustic sensors over varying temporal and spatial scales. *Marine Ecology Progress Series*, 395:37 – 53, 12 2009.
- [3] Henrik Brumm and Hans Slabbekoorn. Acoustic communication in noise. *Advances in the Study of Behavior*, 35:151–209, 12 2005.
- [4] Luke Remage-Healey, Douglas Nowacek, and Andrew Bass. Dolphin foraging sounds suppress calling and elevate stress hormone levels in a prey species, the gulf toadfish. *The Journal of experimental biology*, 209:4444–51, 12 2006.
- [5] Campbell A. F. Boinski, S. Use of trill vocalizations to coordinate troop movement among white-faced capuchins: A second field test. *Behaviour*, 132:875–901, 1995.
- [6] Richard Wiley and Douglas Richards. Physical constraints on acoustic communication in the atmosphere: Implications for the evolution of animal vocalizations. *Behavioral Ecology and Sociobiology*, 3:69–94, 03 1978.
- [7] Eugene S. Morton. Ecological sources of selection on avian sounds. *The American Naturalist*, 109(965):17–34, 1975.
- [8] JA Hildebrand. Anthropogenic and natural sources of ambient noise in the ocean. *Marine Ecology-progress Series - MAR ECOL-PROGR SER*, 395:5–20, 12 2009.
- [9] Gage S.H. Pijanowski B.C., Farina A. What is soundscape ecology? an introduction and overview of an emerging new science. *Landscape Ecol.*, 26:1213–1232, 2011.
- [10] G. M. Wenz. Acoustic ambient noise in the ocean: Spectra and sources. *The Journal of the Acoustical Society of America*, (34(12)):1936–1956, 1962.
- [11] Xinguo Zhao Chao Peng and Guangxu LiuInt. Noise in the sea and its impacts on marine organisms. *J.Environ. Res. Public Health* 12, (12):12304–12323, 2015.
- [12] Fristrup KM. Barber JR, Crooks KR. The costs of chronic noise exposure for terrestrial organisms. *Trends Ecol Evol.*, 25(3):180–9, 2010.

- [13] Christopher Clark, William Ellison, Brandon Southall, L. Hatch, Sofie Van Parijs, Adam Frankel, and Dimitri Ponirakis. Acoustic masking in marine ecosystems: Intuitions, analysis, and implication. *Marine Ecology Progress Series*, 395:201–222, 12 2009.
- [14] Jennifer C Sun and Peter M. Narins. Anthropogenic sounds differentially affect amphibian call rate. *Biological Conservation*, 121:419–427, 2005.
- [15] Npwacek P. Thorne L.H. Johnston D.W. Tyack P.L. Douglas, G. Responses of cetaceans to anthropogenic noise. *Mammal review*, 37(2):81–115, 2007.
- [16] Jesse R Barber Clinton D Francis. A framework for understanding noise impacts on wildlife: an urgent conservation priority. 11(6):305–313, 2013.
- [17] Cruz A. Francis CD, Ortega CP. Noise pollution changes avian communities and species interactions. *Curr Biol.*, 19(16):1415–9, 2009.
- [18] Daniel Mennill, Stéphanie Doucet, Kara Anne Ward, Dugan Maynard, Brian Otis, and John Burt. A novel digital telemetry system for tracking wild animals: A field test for studying mate choice in a lekking tropical bird. *Methods in Ecology and Evolution*, 3, 08 2012.
- [19] Ben D. Bell Paul D. Teal Andrew Digby, Michael Towsey. A practical comparison of manual and autonomous methods for acoustic monitoring. *Methods in ecology and evolution*, 4(7):675–683, 2013.
- [20] Renata Sousa-Lima. A review and inventory of fixed autonomous recorders for passive acoustic monitoring of marine mammals. *Aquatic Mammals*, 39:23–53, 03 2013.
- [21] William C. Burgess, Peter Lloyd Tyack, Burney J. Le Boeuf, and Daniel P. Costa. A programmable acoustic recording tag and first results from free-ranging northern elephant seals. *Deep-sea Research Part Ii-topical Studies in Oceanography*, 45:1327–1351, 1998.
- [22] Mark Johnson and Peter Tyack. A digital acoutic recording tag for measuring the response of wild marine mammals to sound. *Oceanic Engineering, IEEE Journal of*, 28:3 – 12, 02 2003.
- [23] Daniel Rudnick, Russ Davis, Charles Eriksen, David Fratantoni, and Mary Jane Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38:73–84, 06 2004.
- [24] Mark F. Baumgartner, David M. Fratantoni, Thomas P. Hurst, Moira W. Brown, Tim V. N. Cole, Sofie M. Van Parijs, and Mark Johnson. Real-time reporting of baleen whale passive acoustic detections from ocean gliders. *The Journal of the Acoustical Society of America*, 134(3):1814–1823, 2013.
- [25] Ben Wilson, Steven Benjamins, and Jim Elliott. Using drifting passive echolocation loggers to study harbour porpoises in tidal-stream habitats. *Endangered Species Research*, 22:125–143, dec 2013.
- [26] Paolo Favali, Laura Beranzoli, and Angelo De Santis. *Seafloor observatories: A new vision of the earth from the Abyss*. 01 2015.
- [27] High-energy neutrino searches in the mediterranean sea: probing the universe with antares and km3net/arca. *XXVII International conference on neutrino physics and astrophysics*, 2016.

- [28] G. Lara, R. Miralles, M. Bou-Cabo, J.A. Esteban, and V. Espinosa. New insights into the design and application of a passive acoustic monitoring system for the assessment of the good environmental status in spanish marine waters. *Sensors*, 20(18), 2020.
- [29] Manuel Bou-Cao, Guillermo Lara, Paula Gutiérrez-Muñoz, Camilo Saavedra, Ramón Miralles, and Víctor Espinosa. A risk-based model using communication distance reduction for the assessment of underwater continuous noise: An application to the bottlenose dolphin (*tursiops truncatus*) inhabiting the spanish north atlantic marine demarcation. *Journal of Marine Science and Engineering*, 10(5), 2022.
- [30] MITECO. Demarcaciones marinas españolas. <https://www.miteco.gob.es/es/cartografia-y-sig/ide/descargas/costas-medio-marino/demarcaciones-marinas.aspx>, (último acceso: 01.09.2022).
- [31] Cetacean Research. <https://www.cetaceanresearch.com/hydrophones/c57-hydrophone/index.html>, (último acceso: 01.09.2022).
- [32] Texas Instruments. <https://www.ti.com/product/tlv320aic3204>, (último acceso: 01.09.2022).
- [33] Texas Instruments. <https://www.ti.com/product/tms320c5535>, (último acceso: 01.09.2022).
- [34] Analog Devices. <https://www.analog.com/en/products/lt3995.html>, (último acceso: 01.09.2022).

Parte II

Anexos

Apéndice A

Listados componentes diseño auxiliar

- Zocalo microSD. Farnell: 206406. Marca Molex. (1 Unidades). Precio: 1.94€.
- Conector hembra para PCB Amphenol FCI Recto 20 pines 2 filas paso 1.27 mm. Montaje Superficial
- Resistencias de montaje en superficie de 10 K (7 unidades)
- Condensador de montaje en superficie de 104. (3 unidades)
- Condensador bloqueo continua - Condensador cerámico multicapa MLCC, Murata, 10 μ F, \pm 20 %, 50V dc, Orificio Pasante, X7R dieléctrico (1 Unidad). Código RS 811 - 8373. Precio: 2.592€
- Conector NARANJA 5.08 placa para cable 90 grados 3 vias tornillo (1 unidad). Fabricante: Weidmuller. Terminal para PCB naranja 3 vias,5.08mm. Código RS408 - 7887 N° ref. fabric.PM 5.08/ 3/90 3.5 - 1760500000. Precio: 0.502€
- Conector NARANJA 5.08 placa para cable 90 grados 2 vias tornillo (2 Unidades). Fabricante: Weidmuller Código RS408 - 7871 N° ref. fabric.PM 5.08/2/90 3.5 - 1760490000. Precio: 0.762€
- Conector placa VERDE 5.08 - 90 grados soldar hembra 2 vias. (1 Unidad). Terminal block socket, 90 deg, 2 way (Código RS707 - 5335) – Precio: 0.372€
- Conector placa BLANCO 5.08 - 90 grados soldar hembra 2 vias (1 Unidad). Fabricante: Wieland En blanco Código RS181 - 4018. Precio: 0.523 €

Apéndice B

Código firmware

```
config_main{
/* Enable periph clocks in idle control register */
*((volatile ioport unsigned *) (0x0001)) = 0x000E;

/* Flush the pipeline */
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16");
asm("\tnop_16"); // Execute IDLE instr to apply changes

InitDate.year = 1;
InitDate.month = 1;
InitDate.day = 1;
InitTime.hours = 0;
InitTime.mins = 0;
InitTime.secs = 0;
InitTime.mSecs = 0;

rtcConfig.rtcyear = 8;
rtcConfig.rtcmonth = 8;
rtcConfig.rtcday = 8;
rtcConfig.rtchour = 8;
rtcConfig.rtcmin = 8;
rtcConfig.rtcsec = 8;
rtcConfig.rtcMSec = 8;

rtcConfig.rtcyeara = 8;
rtcConfig.rtcmontha = 8;
rtcConfig.rtcdaya = 8;
```

```

rtcConfig.rthour = 8;
rtcConfig.rtcmin = 8;
rtcConfig.rtcsec = 8;
rtcConfig.rtcminSec = 10;
rtcConfig.rtcintcr = 0x8000; //Enabled only RTC alarm interrupt

/* Set the RTC init structure */
AlarmTime.year = InitDate.year;
AlarmTime.month = InitDate.month;
AlarmTime.day = InitDate.day;
AlarmTime.hours = InitTime.hours;
AlarmTime.mins = InitTime.mins;
AlarmTime.secs = InitTime.secs+offset_on;
AlarmTime.mSecs = InitTime.mSecs;

/* Register the ISR function */
isrAddr.AlarmEvtAddr = rtc_alarmEvt;
status = RTC_setCallback(&rtcDispatchTable, &isrAddr);
IRQ_globalDisable();
IRQ_clearAll();
IRQ_disableAll();
IRQ_setVecs((Uint32)&VECSTART);
IRQ_clear(RTC_EVENT);
IRQ_plug(RTC_EVENT, &rtc_isr);
IRQ_enable(RTC_EVENT);
IRQ_globalEnable();
RTC_reset();

status = RTC_config(&rtcConfig);
status = RTC_getConfig(&rtcGetConfig);
EZDSP5535_waitusec(100); // Wait
status = RTC_setDate(&InitDate);
EZDSP5535_waitusec(100); // Wait
status = RTC_setTime(&InitTime);

EZDSP5535_waitusec(100); // Wait
status = RTC_setAlarm(&AlarmTime);
status = RTC_eventEnable(CSL_RTC_ALARM_INTERRUPT);

CSL_SYSCTRL_REGS->CGCR1 = 0x8173;
CSL_FINS(CSL_SYSCTRL_REGS->CGOCR, SYS_CGOOCR_OD, 0);
CSL_FINS(CSL_SYSCTRL_REGS->CGOCR, SYS_CGOOCR_OUTDIVEN, 0);
printf("\nWorking at 12MHz");

CSL_SYSCTRL_REGS->CCR2 = 0x0001;
}

```

```

CSL_Status configSdCard (CSL_MmcSdInstId  instId)
{

    Uint32          sectCount;
    CSL_Status      status;
    Uint16          actCard;
    Uint16          rca;
    Uint16          clockDiv;
    CSL_CardType    cardType;

    cardType = CSL_CARD_NONE;
    sectCount = 0;

    /* Initialize MMCSd CSL module */
    status = MMC_init();
    status = SYS_setEBSR(CSL_EBSR_FIELD_SP0MODE, CSL_EBSR_SP0MODE_0);
    status = SYS_setEBSR(CSL_EBSR_FIELD_SP1MODE, CSL_EBSR_SP1MODE_0);
    if(CSL_SOK != status)
    {
        printf("SYS_setEBSR failed\n");
        return (status);
    }

    /* Open MMCSd CSL module */

    mmcSdHandle = MMC_open(&pMmcSdContObj,
        instId, CSL_MMCSd_OPMODE_DMA, &status);
    if(mmcSdHandle == NULL)
    {
        printf("MMC_open_Failed\n");
        return (status);
    }

    /* Initialize Dma */
    status = DMA_init();
    if (status != CSL_SOK)
    {
        printf("DMA_init_Failed!\n");
        return (status);
    }

    /* Open Dma channel for MMCSd write */
    dmaWrHandle = DMA_open(CSL_DMA_CHAN0, &dmaWrChanObj, &status);
    if((dmaWrHandle == NULL) || (status != CSL_SOK))
    {
        printf("DMA_open for MMCSd Write Failed!\n");
        return (status);
    }

```

```

}

/* Open Dma channel for MMCSD read */
dmaRdHandle = DMA_open(CSL_DMA_CHAN1, &dmaRdChanObj, &status);
if((dmaRdHandle == NULL) || (status != CSL_SOK))
{
printf("DMA_open for MMCSD Read Failed!\n");
return(status);
}

/* Set the DMA handle for MMC read */
status = MMC_setDmaHandle(mmcscdHandle, dmaWrHandle, dmaRdHandle);

/* Reset the SD card */
status = MMC_sendGoIdle(mmcscdHandle);

/* Check for the card */
status = MMC_selectCard(mmcscdHandle, &mmcCardObj);
if((status == CSL_ESYS_BADHANDLE) ||
    (status == CSL_ESYS_INVPARAMS))
{
printf("MMC_selectCard Failed\n");
return(status);
}

/* Verify whether the SD card is detected or not */
if(mmcCardObj.cardType == CSL_SD_CARD)
{
//printf("SD Card detected\n");
cardType = CSL_SD_CARD;

/* Check if the card is high capacity card */
if(mmcscdHandle->cardObj->sdHcDetected == TRUE)
{
printf("\nSD card is High Capacity Card");
printf("\nMemory Access will use Block Addressing");

/* For the SDHC card Block addressing will be used.
Sector address will be same as sector number */
cardAddr = sectCount;
}
else
{
printf("\nSD card is Standard Capacity Card");
printf("\nMemory Access will use Byte Addressing");

/* For the SD card Byte addressing will be used.

```

```

Sector address will be product of sector number and sector size */
cardAddr = (sectCount)*(CSL_MMCSDBLOCK_LENGTH);
        }
    }
    else
    {
    if(mmcCardObj.cardType == CSL_CARD_NONE)
    {
    printf("\nNo Card detected");
    }
    else
    {
    printf("\nSD Card not detected");
    }
    printf("\nPlease Insert SD Card");
    return(CSL_ESYS_FAIL);
    }

    /* Set the init clock */
    status = MMC_sendOpCond(mmcHandle, 70);

    /* Send the card identification Data */
    status = SD_sendAllCID(mmcHandle, &sdCardIdObj);

    /* Set the Relative Card Address */
    status = SD_sendRca(mmcHandle, &mmcCardObj, &rca);

    /* Read the SD Card Specific Data */
    status = SD_getCardCsd(mmcHandle, &sdCardCsdObj);
    if(status != CSL_SOK)
    {
    printf("SD_getCardCsd Failed\n");
    return (status);
    }

    /* Get the clock divider value for the current CPU frequency */
    clockDiv = computeClkRate();

    /* Set the card type in internal data structures */
    status = MMC_setCardType(&mmcCardObj, cardType);

    /* Set the card pointer in internal data structures */
    status = MMC_setCardPtr(mmcHandle, &mmcCardObj);

    /* Get the number of cards */
    status = MMC_getNumberOfCards(mmcHandle, &actCard);

```

```

/* Set clock for read-write access */
status = MMC_sendOpCond(mmcHandle, clockDiv);

/* Set Endian mode for read and write operations */
status = MMC_setEndianMode(mmcHandle,
CSL_MMCSO_ENDIAN_LITTLE, CSL_MMCSO_ENDIAN_LITTLE);

/* Set block length for the memory card
* For high capacity cards setting the block length will have
* no effect */
status = MMC_setBlockLength(mmcHandle, CSL_MMCSO_BLOCK_LENGTH);

return (status);
}

CSL_DMA_Handle CSL_configDmaForI2s(CSL_DMAChanNum chanNum)
{
CSL_DMA_Handle dmaHandle;

ioport volatile CSL_SysRegs *sysRegs;

sysRegs = (CSL_SysRegs *)CSL_SYSCTRL_REGS;

/*Enable the corresponding DMA clock from PCGCR Registers*/
CSL_FINS(sysRegs->PCGCR1, SYS_PCGCR1_DMA0CG, CSL_SYS_PCGCR1_DMA0CG_ACTIVE);
CSL_FINS(sysRegs->PCGCR2, SYS_PCGCR2_DMA1CG, CSL_SYS_PCGCR2_DMA1CG_ACTIVE);

status = DMA_init();
dmaHandle = DMA_open(chanNum, &dmaObj2, &status);
status = DMA_config(dmaHandle, &dmaConfig2);
return(dmaHandle);
}

interrupt void dma_isr(void)
{

int ifrValue;

ifrValue = CSL_SYSCTRL_REGS->DMAIFR;
CSL_SYSCTRL_REGS->DMAIFR |= ifrValue;

if ((DMA_getLastTransferType (dmaLeftRxHandle1, &status)) == 1)
{
/* PONG */
for(index=0; index< CSL_I2S_DMA_BUF_LEN/2; index++)
{
i2sDmaReadLeftBuff2aux[index] = i2sDmaReadLeftBuff2[(index*2)+1];
}
}
}

```

```

}
status = MMC_write(mmcSDHandle, cardAddr,
CSL_I2S_DMA_BUF_LEN, i2sDmaReadLeftBuff2aux);
cardAddr = cardAddr+16; //16=(CSL_I2S_DMA_BUF_LEN/512)
}
else
{
/* PING */
for(index=0; index < CSL_I2S_DMA_BUF_LEN/2; index++)
{
i2sDmaReadLeftBuff1aux[index] = i2sDmaReadLeftBuff1[(index*2)+1];
}

status = MMC_write(mmcSDHandle, cardAddr,
CSL_I2S_DMA_BUF_LEN, i2sDmaReadLeftBuff1aux);
cardAddr = cardAddr+16; //16=(CSL_I2S_DMA_BUF_LEN/512)
}
if (cuenta == times_on) {
CSL_FINS(dmaLeftRxHandle1->dmaRegs->DMACH0TCR2, DMA_DMACH0TCR2_AUTORLD,0);
}
cuenta=cuenta+1;

}

Uint16 i2s_dma(void)
{

Uint16 result;
CSL_I2sHandle hI2s1;
I2S_Config hwConfig1;

cuenta=1;

EZDSP5535_I2C_init( );

/* Configure AIC3204 */
AIC3204_rset( 0, 0 );
AIC3204_rset( 1, 1 );
AIC3204_rset( 0, 1 );
AIC3204_rset( 1, 8 );
AIC3204_rset( 2, 1 );
AIC3204_rset( 0, 0 );

/* PLL and Clocks config and Power Up */
AIC3204_rset( 27, 0x0d );
AIC3204_rset( 4, 3 );

```

```

AIC3204_rset( 6, 8 );
AIC3204_rset( 7, 0x07 );
AIC3204_rset( 8, 0x80 );
AIC3204_rset( 30, 0x88 );
AIC3204_rset( 5, 0x91);
AIC3204_rset( 13, 0 );
AIC3204_rset( 14, 0x20 );
AIC3204_rset( 20, 0x20 );
AIC3204_rset( 11, 0x82 );
AIC3204_rset( 12, 0x88 );
AIC3204_rset( 18, 0x82 );
AIC3204_rset( 19, 0x88 );
AIC3204_rset( 0x3c, 8 );
AIC3204_rset( 0x3d, 8 );

/* DAC ROUTING and Power Up */
AIC3204_rset( 0, 1 );
AIC3204_rset( 0x0c, 8 );
AIC3204_rset( 0x0d, 8 );
AIC3204_rset( 0x0a, 0x40 );
AIC3204_rset( 0x03, 0 );
AIC3204_rset( 0x04, 0 );
AIC3204_rset( 0x47, 0x31);
AIC3204_rset( 0x7b, 1 );
AIC3204_rset( 0, 0 );
AIC3204_rset( 64, 0x0e );
AIC3204_rset( 63, 0xd4 );
AIC3204_rset( 0, 1 );
AIC3204_rset( 0x10, 0x1c );
AIC3204_rset( 0x11, 0x1c );
AIC3204_rset( 9, 0x30 );
AIC3204_rset( 0, 0 );

* ADC ROUTING and Power Up */
AIC3204_rset( 0, 1 );
AIC3204_rset( 0x33, 0x40);
AIC3204_rset( 0x34, 0x10 );
AIC3204_rset( 0x37, 0x10 );
AIC3204_rset( 0x36, 1 );
AIC3204_rset( 0x39, 1 );
AIC3204_rset( 0x3b, gain );
AIC3204_rset( 0x3c, gain );

AIC3204_rset( 0, 8 );
AIC3204_rset( 36, 0x00 );
AIC3204_rset( 37, 0x00 );
AIC3204_rset( 38, 0x00 );

```

```

AIC3204_rset( 40, 0x65 );
AIC3204_rset( 41, 0x6e );
AIC3204_rset( 42, 0xe3 );
AIC3204_rset( 44, 0x00 );
AIC3204_rset( 45, 0x00 );
AIC3204_rset( 46, 0x00 );
AIC3204_rset( 48, 0x00 );
AIC3204_rset( 49, 0x00 );
AIC3204_rset( 50, 0x00 );
AIC3204_rset( 52, 0x00 );
AIC3204_rset( 53, 0x00 );
AIC3204_rset( 54, 0x00 );

AIC3204_rset( 0, 0 );
AIC3204_rset( 81, 0xc0 );
AIC3204_rset( 82, 0x00 );
AIC3204_rset( 83, 0x28 );
EZDSP5535_waitusec(100);

WriteBuff[iter_control*8]= 0x0D00 + reset_iter;
WriteBuff[iter_control*8+1]= iteration;
WriteBuff[iter_control*8+2]=cardAddr >> 16;
WriteBuff[iter_control*8+3]= cardAddr & 65535;

EZDSP5535_LED_init( );

/** Open the device with instance 2 */
hI2s1 = I2S_open(I2S_INSTANCE2, DMA_POLLED, I2S_CHAN_STEREO);
if(NULL == hI2s1)
{
printf ("\nI2S□Module□Instance□not□opened");
return (1);
}

/** Set the value for the configure structure */
hwConfig1.dataType = I2S_STEREO_ENABLE;
hwConfig1.loopBackMode = I2S_LOOPBACK_DISABLE;
//hwConfig1.dataFormat = I2S_DATAFORMAT_LJUST;
hwConfig1.fsPol = I2S_FSPOL_LOW;
hwConfig1.clkPol = I2S_FALLING_EDGE;
hwConfig1.datadelay = I2S_DATADELAY_ONEBIT;
hwConfig1.datapack = I2S_DATAPACK_ENABLE;
hwConfig1.signext = I2S_SIGNEXT_DISABLE;
hwConfig1.wordLen = I2S_WORDLEN_32;
hwConfig1.i2sMode = I2S_SLAVE;
hwConfig1.FError = I2S_FSERROR_DISABLE;
hwConfig1.OuError = I2S_OUERROR_DISABLE;

```

```

/** Configure hardware registers */
result = I2S_setup(hI2s1, &hwConfig1);
if(result != CSL_SOK)
{
printf ("\nI2S Module not Configured");
return (2);
}

IRQ_globalDisable();
IRQ_clearAll();
IRQ_disableAll();
IRQ_setVecs((Uint32)&VECSTART);
IRQ_clear(DMA_EVENT);
IRQ_plug(DMA_EVENT, &dma_isr);
IRQ_enable(DMA_EVENT);
IRQ_globalEnable();

/** Configure DMA channel for I2S Read */
dmaConfig2.pingPongMode = CSL_DMA_PING_PONG_ENABLE;
dmaConfig2.autoMode     = CSL_DMA_AUTORELOAD_ENABLE;
dmaConfig2.burstLen     = CSL_DMA_TXBURST_1WORD;
dmaConfig2.trigger      = CSL_DMA_EVENT_TRIGGER;
dmaConfig2.dmaEvt       = CSL_DMA_EVT_I2S2_RX;
dmaConfig2.dmaInt       = CSL_DMA_INTERRUPT_ENABLE;
dmaConfig2.chanDir      = CSL_DMA_READ;
dmaConfig2.trfType      = CSL_DMA_TRANSFER_IO_MEMORY;
dmaConfig2.dataLen      = CSL_I2S_DMA_BUF_LEN*4;
dmaConfig2.srcAddr      = (Uint32)&hI2s1->hwRegs->I2SRXLT0;
dmaConfig2.destAddr     = (Uint32)i2sDmaReadLeftBuff1;

dmaLeftRxHandle1 = CSL_configDmaForI2s(CSL_DMA_CHAN4);
if(dmaLeftRxHandle1 == NULL)
{
printf("DMA Config for I2S Read Failed!\n!");
return (3);
}

CSL_FINS(mmcDsdHandle->mmcRegs->MMCCLK, MMCSD_MMCLK_CLKRT, 0x00);
CSL_FINS(mmcDsdHandle->mmcRegs->MMCCTL, MMCSD_MMCTL_PERMDR, 0);
CSL_FINS(mmcDsdHandle->mmcRegs->MMCCTL, MMCSD_MMCTL_WIDTH, 1);

status = RTC_getTime(&GetTime);
status = RTC_getDate(&GetDate);
secs=GetTime.secs;
mins=GetTime.mins;
hours=GetTime.hours;

```

```

day=GetDate . day;
month=GetDate . month;

dayandhour = (month << 10) + (hours & 0x001F) + (day << 5);

minuteandsecond = mins << 8;
minuteandsecond = minuteandsecond + (secs & 255);

WriteBuff[iter_control*8+6]= dayandhour;
WriteBuff[iter_control*8+7]= minuteandsecond;

mmcsdHandle->readEndianMode = CSL_MMCSD_ENDIAN_BIG;
mmcsdHandle->writeEndianMode = CSL_MMCSD_ENDIAN_BIG;
status = MMC_write(mmcsdHandle, cardAddr_control,
WRITE_BUFFER_MAX_SIZE, WriteBuff);

mmcsdHandle->writeEndianMode = CSL_MMCSD_ENDIAN_LITTLE;
mmcsdHandle->readEndianMode = CSL_MMCSD_ENDIAN_LITTLE;

I2S_transEnable(hI2s1, TRUE);
status = DMA_start(dmaLeftRxHandle1);
if(status != CSL_SOK)
{
printf("I2S□Dma□Read□Failed!!\n");
return (7);
}

while(cuenta <= times_on)
{
if (cuenta>times_on)
{
I2S_transEnable(hI2s1, FALSE);
result = I2S_reset(hI2s1);
result = I2S_close(hI2s1);
IRQ_clearAll();
IRQ_disableAll();
IRQ_globalDisable();

WriteBuff[iter_control*8+4]= cardAddr-16 >> 16;
WriteBuff[iter_control*8+5]= cardAddr-16 & 65535;

iter_control=iter_control+1;
mmcsdHandle->readEndianMode = CSL_MMCSD_ENDIAN_BIG;
mmcsdHandle->writeEndianMode = CSL_MMCSD_ENDIAN_BIG;
status = MMC_write(mmcsdHandle, cardAddr_control,
WRITE_BUFFER_MAX_SIZE, WriteBuff);

```

```

if (iter_control==32){
iter_control=0;
cardAddr_control=cardAddr_control+1;
for(index = 0; index < WRITE_BUFFER_MAX_SIZE/2; index++)
{
WriteBuff[index] = 0x0;
}
}

// guardo la info en el bloque 0 cuando acaba el archivo
ReadBuff[5] = reset_iter;
ReadBuff[6] = cardAddr >> 16;
ReadBuff[7] = cardAddr & 65535;
ReadBuff[8] = cardAddr_control;
ReadBuff[9] = iter_control;
ReadBuff[10] = num_sd_actual+1;

status = MMC_write(mmcSDHandle, 0, WRITE_BUFFER_MAX_SIZE, ReadBuff);
return (CSL_TEST_PASSED);
}
}

I2S_transEnable(hI2s1, FALSE);

/** Reset the registers */
result = I2S_reset(hI2s1);
result = I2S_close(hI2s1);

IRQ_clearAll();
IRQ_disableAll();
IRQ_globalDisable();

WriteBuff[iter_control*8+4]= cardAddr-16 >> 16;
WriteBuff[iter_control*8+5]= cardAddr-16 & 65535;

iter_control=iter_control+1;
mmcSDHandle->readEndianMode = CSL_MMCSO_ENDIAN_BIG;
mmcSDHandle->writeEndianMode = CSL_MMCSO_ENDIAN_BIG;
status = MMC_write(mmcSDHandle, cardAddr_control,
WRITE_BUFFER_MAX_SIZE, WriteBuff);

if (iter_control==32){
iter_control=0;
cardAddr_control=cardAddr_control+1;
for(index = 0; index < WRITE_BUFFER_MAX_SIZE/2; index++)
{
WriteBuff[index] = 0x0;
}
}

```

```
}  
  
}  
// guardo la info cuando acaba el archivo  
ReadBuff[5] = reset_iter;  
ReadBuff[6] = cardAddr >> 16;  
ReadBuff[7] = cardAddr & 65535;  
ReadBuff[8] = cardAddr_control;  
ReadBuff[9] = iter_control;  
ReadBuff[10] = num_sd_actual+1;  
  
status = MMC_write(mmcHandle, 0, WRITE_BUFFER_MAX_SIZE, ReadBuff);  
  
return (CSL_TEST_PASSED);  
}
```