

# Procesamiento de señal visualizado sobre un espectrograma

## Signal processing displayed on a spectrogram

Martin E. Paz<sup>#1</sup>, Guillermo Friedrich<sup>#2</sup>, Christian L. Galasso<sup>\*3</sup>

<sup>#</sup> *Cátedra de Técnicas Digitales III, Departamento de Ingeniería Electrónica  
 Universidad Tecnológica Nacional - Facultad Regional Bahía Blanca.  
 Bahía Blanca - Argentina*

<sup>1</sup> pazmartin35@gmail.com

<sup>2</sup> gfried@frbb.utn.edu.ar

<sup>\*</sup> *Cátedra de Teoría de Circuitos II, Universidad Tecnológica Nacional  
 Bahía Blanca - Argentina*

<sup>3</sup> clgalasso@frbb.utn.edu.ar

Recibido: 29/09/19; Aceptado: 01/03/20

**Abstract**— This paper describes the implementation of a digital spectrogram on an embedded system, with the aim of allowing the visualization of the effects of applying digital filters and Fast Fourier Transform over a signal. The FreeRTOS operating system was used to facilitate the management of processing resources.

**Keywords:** embedded system; spectrogram; FFT; operating system.

**Resumen**— En el presente trabajo se describe la implementación de un espectrograma digital sobre un sistema embebido, con el objetivo de permitir la visualización de los efectos de aplicar filtros digitales y Transformada Rápida de Fourier sobre una señal. Se utilizó el sistema operativo FreeRTOS para facilitar la administración de los recursos de procesamiento.

**Palabras clave:** sistema embebido; espectrograma; FFT; sistema operativo.

### I. INTRODUCCIÓN

Este proyecto se desarrolló en el marco de la cátedra Técnicas Digitales III perteneciente a la Universidad Tecnológica Nacional (sede Bahía Blanca, Argentina), como un proyecto final de materia, con el objetivo de favorecer el aprendizaje del funcionamiento de un analizador de espectro basado en transformada rápida de fourier (FFT), filtros digitales de tipo respuesta finita al impulso (FIR) y de tipo respuesta infinita al impulso (IIR), y sistemas operativos de tiempo real. Dado que a lo largo de la carrera se trabaja intensamente sobre sistemas embebidos, se decidió implementar el sistema sobre un microcontrolador Cortex M4, realizando la adquisición, procesamiento y análisis de las señales por medio de periféricos dedicados a este fin. Asimismo, se utilizó como base para el desarrollo del software al sistema operativo FreeRTOS.

El espectrograma es una herramienta básica de representación que se utiliza para el análisis de una señal eléctrica, de comunicación, o audiovisual, en la que se

visualiza la energía del contenido frecuencial de la señal, según va variando a lo largo del tiempo. La representación se realiza en tres dimensiones: temporal, frecuencial y amplitud, y el instrumento que genera el espectrograma es llamado espectrómetro.

El espectrograma consiste en tomar una determinada cantidad de muestras por medio de una ventana temporal, de un tamaño concreto según sea el tipo de análisis que se requiera, para luego realizar el cálculo del contenido espectral de las muestras obtenidas por medio de la FFT [1], representando los resultados en una gráfica de tres dimensiones. Posteriormente, la ventana temporal se desplaza a lo largo del tiempo sobre la señal a analizar y se adquiere una nueva cantidad de muestras, repitiendo el ciclo de cálculo y graficando los resultados. La suma de las representaciones de la FFT obtenidas aportan información en el dominio de la frecuencia, de la variación de la energía y la frecuencia de la señal en función del tiempo.

La FFT es un algoritmo eficiente que permite reducir el costo computacional de la transformada discreta de fourier (DFT), mediante la que se realiza el análisis en el dominio frecuencial de una señal.

Un filtro eléctrico o filtro electrónico es un elemento que discrimina una determinada frecuencia o gama de frecuencias de una señal que pasa a través de él, modificando los parámetros de amplitud y fase, y pueden ser activos o pasivos, analógicos o digitales. Un filtro digital es un tipo de filtro que trabaja sobre señales discretas y cuantizadas, implementado con tecnología digital como un circuito digital ad-hoc o mediante software.

El equipo da la posibilidad de elegir usar filtros Pasa Bajo o Pasa Alto de tipos IIR o FIR [2], a fin de comparar sus efectos y visualizar los resultados en un espectrograma.

### II. IMPLEMENTACIÓN

#### A. Hardware

En la Fig. 1 se muestra el diagrama de conexionado del espectrómetro, integrado por el circuito normalizador, el

circuito controlador y la pantalla LCD TFT (Liquid Crystal Display Thin Film Transistor).

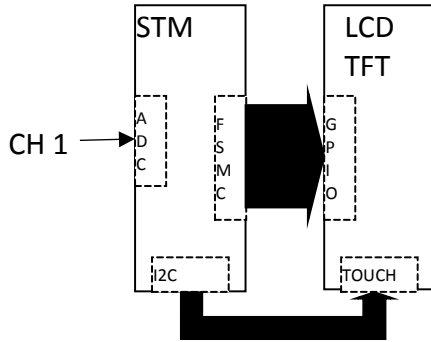


Fig. 1. Diagrama de conexión del espectrómetro.

El microcontrolador utilizado es el STM32F407 de la firma STMicroelectronics [3], el cual puede trabajar a una velocidad máxima de 168 MHz; posee convertidor analógico digital (ADC), acceso directo a memoria (DMA), controlador de memoria estática flexible (FSMC), temporizadores programables, hardware de procesamiento digital de señal (DSP), y unidad de punto flotante (FPU), permitiendo mayor velocidad de cálculo. El microcontrolador se encuentra montado en un kit de desarrollo STM32F4DISCOVERY [4], permitiendo su uso para múltiples aplicaciones.

El ADC interno del microcontrolador trabaja con tensión de 3 V máxima y una resolución de 12 bits, es decir que el valor mínimo que puede leer es de 732  $\mu$ V. Para señales menores a 1 mVpp, estas pueden ser amplificadas mediante un circuito externo.

El temporizador utiliza como frecuencia de referencia 84 MHz, siempre que el oscilador interno del microcontrolador este configurado para trabajar a 168 MHz. Se configura utilizando dos variables de 16 bits denominadas período y prescaler, las cuales se modifican en función de las necesidades del usuario, permitiendo cambiar la frecuencia con la que se muestrea la señal. Teniendo en cuenta el tiempo requerido por el ADC para convertir la señal de analógica a digital, la frecuencia máxima de muestreo es de 1 MHz.

El DMA es un periférico que permite almacenar el valor digital de la señal entregado por el ADC, en un arreglo utilizando el temporizador como disparador y desplazando dicho arreglo automáticamente para almacenar un nuevo dato en la siguiente posición. Posee una configuración de doble-buffer que alterna entre un arreglo y otro al momento en que uno de estos se llene, evitando la pérdida de datos mientras se analiza uno de los arreglos.

El FSMC es un periférico que permite controlar la pantalla LCD como si fuera una dirección de memoria interna del microcontrolador, otorgando mayor velocidad de escritura y lectura. La comunicación es a 16 bits de datos, configurando cada pixel en formato 565 (5 rojo, 6 verde y 5 azul) [5].

El módulo de pantalla LCD TFT es de 7 pulgadas con una resolución de 800 x 480 y una comunicación paralela de 16 bits. El circuito controlador de la pantalla es el SSD1963 [6], el cual soporta hasta 864 x 480 x 24 bits. El módulo incluye una interface táctil de tipo capacitiva (Fig. 2) con protocolo de comunicación I2C, y un zócalo para una memoria de almacenamiento SD (Fig. 3), con protocolo de comunicación SPI con el microcontrolador.

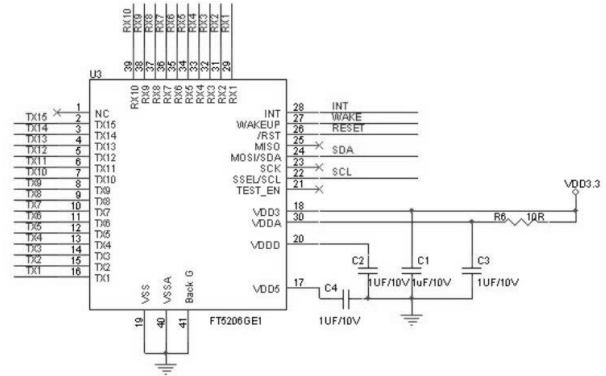


Fig. 2. Circuito eléctrico del panel Táctil.

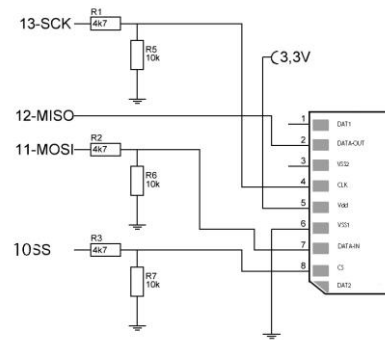


Fig. 3. Circuito eléctrico del zócalo para la memoria SD.

B. Software

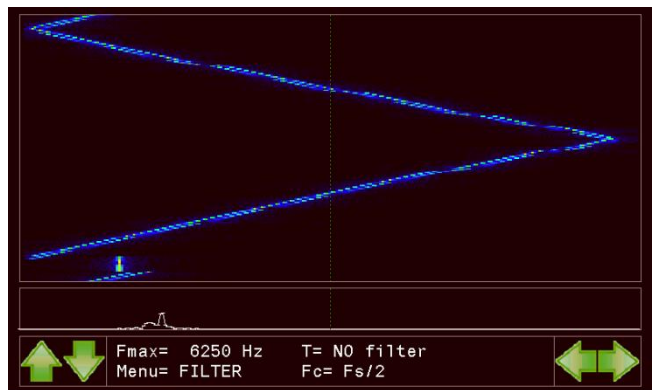


Fig. 4. Captura de pantalla del espectrograma.

La Fig. 4 muestra una captura de pantalla de la interfaz gráfica de usuario (GUI), la cual se separa en tres ventanas: ventana inferior – donde se encuentra el menú de usuario junto con los botones de selección, ventana central – donde se grafica la FFT de la señal de entrada (analizador de espectro), y ventana superior – donde se registra el histograma de la FFT, que es el objetivo principal del Espectrómetro.

La librería utilizada para el algoritmo de FFT [7] pertenece a la firma STMicroelectronics. Este algoritmo se encarga de convertir la señal muestreada en tiempo a frecuencia permitiendo visualizar su componente espectral. Una variable importante al configurar este algoritmo es la cantidad de datos que se va a utilizar, ya que de ésta depende la resolución del espectro. A mayor cantidad de datos, mejor resolución, pero requiere más tiempo de cálculo y mayor memoria del microcontrolador.

```
arm_rfft_init_f32(&S, &S_CFFT, FFT_LENGTH,
ifftFlag, doBitReverse);
```

arm\_rfft\_init\_f32 es la función utilizada para inicializar la configuración del algoritmo. Los parámetros &S y &S\_CFFT son estructuras globales que utiliza la función para almacenar los valores de configuración, FFT\_LENGTH determina la longitud de la FFT, los valores admitidos son 128, 512 o 2048. ifftFlag y doBitReverse son banderas binarias que determinan si se realiza una transformación directa (FFT) o inversa (IFFT), y si la salida está en orden normal o en orden de bit invertido.

```
arm_rfft_f32(&S, FFT_DATA_IN, FFT_CMPLX_DATA);
```

arm\_rfft\_f32 es la función encargada de realizar la FFT. El parámetro &S es la estructura global utilizada en la función antes mencionada. FFT\_DATA\_IN es el arreglo de tipo float32\_t y de longitud FFT\_LENGTH que contiene los datos de la señal muestreada. FFT\_CMPLX\_DATA es el arreglo de tipo float32\_t y de longitud FFT\_LENGTH\*2 que contiene los datos resultantes de la transformada. El arreglo resultante es el doble de tamaño del arreglo de entrada porque los datos están en formato rectangular (real e imaginario).

```
arm_cmplx_mag_f32(FFT_CMPLX_DATA, FFT_MAG_DATA,
FFT_LENGTH);
```

arm\_cmplx\_mag\_f32 es la función que convierte los datos del arreglo FFT\_CMPLX\_DATA de rectangular a polar (magnitud y fase), y almacena únicamente los valores de magnitud en FFT\_MAG\_DATA, que es un arreglo de tipo float32\_t y de longitud FFT\_LENGTH. Debido a la naturaleza del algoritmo para realizar la transformada, los datos en el arreglo FFT\_MAG\_DATA se encuentran reflejados, siendo útiles FFT\_LENGTH / 2.

El algoritmo se configuró para trabajar con FFT\_LENGTH igual a 512 datos, y frecuencia de muestreo configurable.

R es la resolución de la FFT (Eq. 1 y Eq. 2), FS es la frecuencia de muestreo y FN es la frecuencia de Nyquist. FN es equivalente a la mitad de la frecuencia de muestreo y aparece como FMAX en la Fig. 4. Puede ser modificada por el usuario dentro de una gama de frecuencias antes mencionadas. Como los datos útiles son la mitad de FFT\_LENGTH, la resolución se calcula de la siguiente manera:

$$R = \frac{F_S}{FFT\_LENGTH} = \frac{F_S}{512} \quad (1)$$

$$R = \frac{F_S/2}{FFT\_LENGTH/2} = \frac{F_N}{256} \quad (2)$$

La resolución de la ventana superior es de 768 x 330 pixeles configurada de forma tal que 768 pixeles de ancho corresponden a los 256 datos útiles, utilizando 3 pixel por dato para una mejor apreciación, y 330 pixeles de alto que corresponden al eje temporal del espectrograma. El inicio de la gráfica es en la parte inferior de la ventana, avanzando hacia la parte superior repitiendo el proceso y sobre escribiendo los datos anteriores. El desplazamiento del eje

temporal, para graficar una nueva trama de datos, se realiza con un período de 50 ms dando como resultado un historial de 16,5 segundos del comportamiento de la señal de entrada al instrumento.

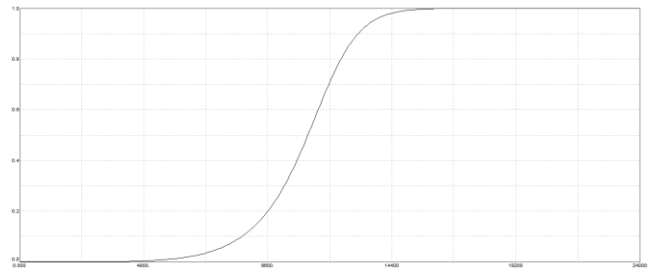


Fig. 5. Respuesta en frecuencia filtro IIR.

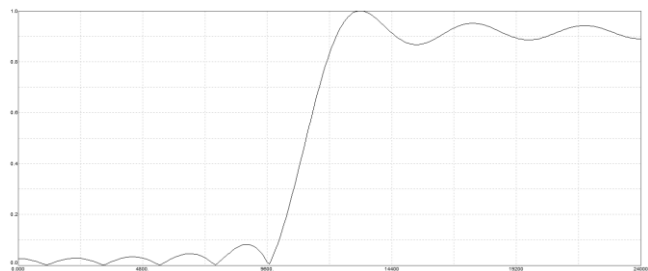


Fig. 6. Respuesta en frecuencia filtro FIR.

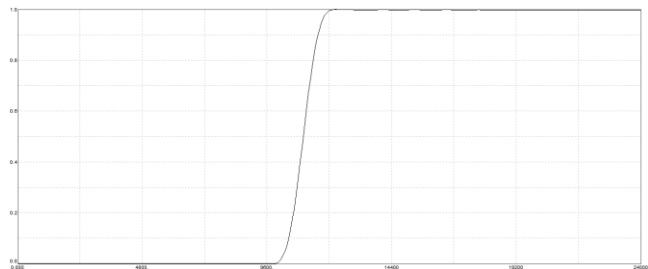


Fig. 7. Respuesta en frecuencia filtro FIR.

La Fig. 5 y la Fig. 6 corresponden a la representación de la respuesta en frecuencia de dos filtros digitales tipo IIR y FIR. Ambos filtros poseen idénticas configuraciones básicas, tales como:

- Filtro Pasa Alto
- Frecuencia de muestreo: 48000 Hz
- Frecuencia de banda de paso: 12000 Hz
- Frecuencia de banda de stop: 10000 Hz
- Ripple banda de paso: -3 dB
- Ripple banda de stop: -10 dB

Los filtros antes mencionados son dos de los diversos filtros que pueden ser seleccionados por el usuario a través del menú de selección.

El filtro digital IIR con los requerimientos supra mencionados se lo diseñó a partir de una configuración de tipo Butterworth, resultando en un filtro de orden 5. En el caso del FIR se realizó un diseño con un enventanado Rectangular obteniéndose un filtro de orden 23.

Como el filtro FIR presentó una notable oscilación antes y después de la zona de corte, se procedió a cambiar la ventana por una de Hamming resultando en un filtro de orden 79 (Fig. 7).

FreeRTOS [8] es el sistema operativo encargado de administrar los recursos del microcontrolador. Es de tiempo

real (kernel) para dispositivos embebidos que ha sido portado a 35 plataformas de microcontrolador y está distribuido bajo la Licencia de código abierto del MIT. Está configurado para crear 3 tareas con distintos niveles de prioridad (0, 1 y 2) dependiendo de sus funciones, las cuales son:

- Menú de selección – nivel de prioridad 0.
- Analizador de espectro – nivel de prioridad 1.
- Espectrograma – nivel de prioridad 2.

Se decidió darle mayor prioridad al menú de selección, ya que un retardo de respuesta al seleccionar una opción, desemboca en un mal funcionamiento del equipo.

### III. ENSAYOS REALIZADOS

Se realizaron ensayos con diferentes tipos de señales, como senoidal de tono puro con frecuencia fija y frecuencia variable, y amplitud modulada con modulación al 30% y 90%. El instrumento utilizado para generar señal es un Keysight 33500B, configurando la señal senoidal con 1V de pico para todas las señales utilizadas.

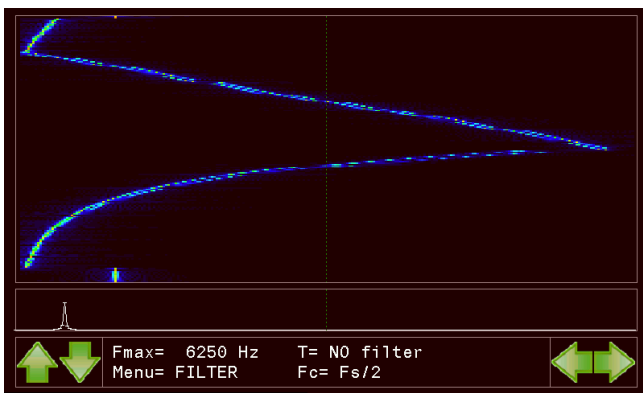


Fig. 8 – Captura de pantalla del espectrograma.

En la Fig. 8 se visualiza una señal senoidal que incrementa su frecuencia de forma logarítmica desde 100 Hz hasta 6 KHz, y que decrementa su frecuencia de forma lineal en un lapso de 13 segundos aproximadamente.

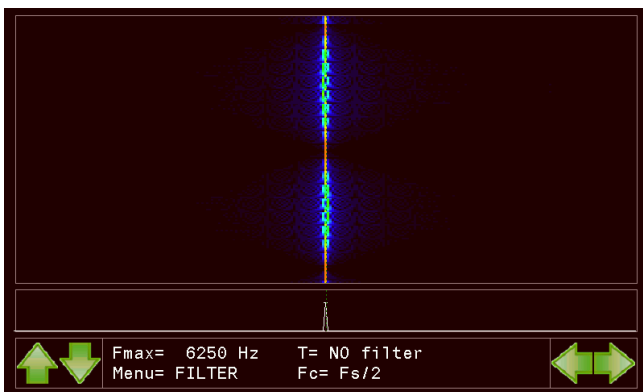


Fig. 9 – Captura de pantalla del espectrograma.

En la Fig. 9 se visualiza una señal senoidal con incremento y decremento de su frecuencia de forma lineal desde 3120 Hz hasta 3130 Hz.

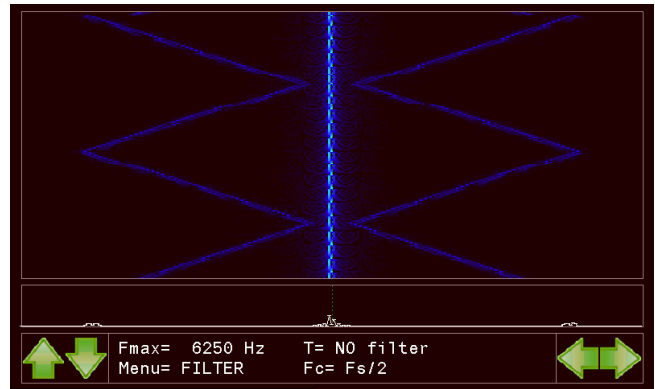


Fig. 10 – Captura de pantalla del espectrograma

En la Fig. 10 se visualiza una señal senoidal modulada en amplitud con frecuencia de portadora de 3120 Hz, y frecuencia moduladora con incremento y decremento de su frecuencia de forma lineal desde 200 Hz hasta 2500Hz, con modulación al 30%.

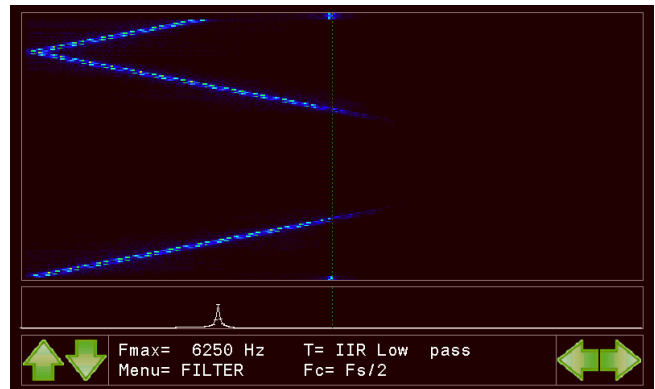


Fig. 11 – Captura de pantalla del espectrograma

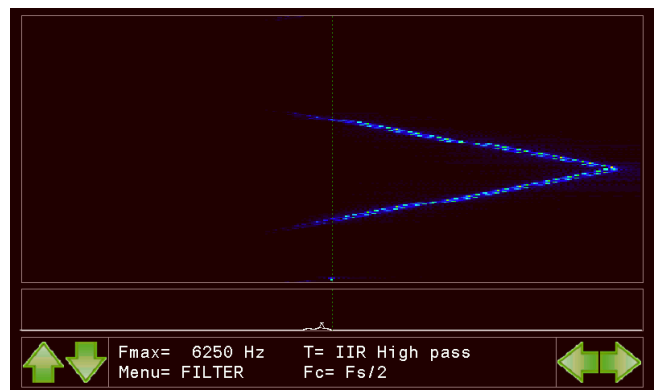


Fig. 12 – Captura de pantalla del espectrograma

En la Fig. 11 y la Fig. 12 se visualiza una señal senoidal que atraviesa dos filtros digitales pasa bajo y pasa alto tipo IIR, aplicados a la mitad de la frecuencia de muestreo. La señal de entrada es senoidal con incremento y decremento lineal de su frecuencia desde 100 Hz hasta 6 KHz. La Fig. 4 muestra la misma señal de entrada sin algún tipo de filtro aplicado en ella.

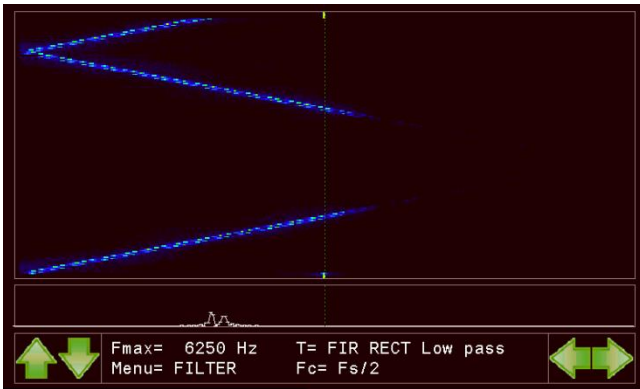


Fig. 13 – Captura de pantalla del espectrograma

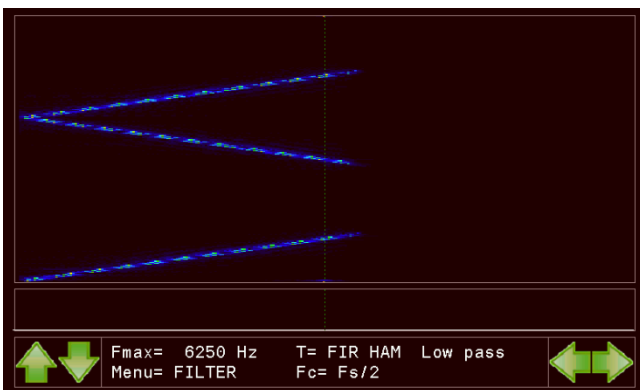


Fig. 14 – Captura de pantalla del espectrograma

En la Fig. 13 se visualiza una señal senoidal que atraviesa un filtro digital pasa bajo de tipo FIR con enventanado rectangular, y en la Fig. 14, la misma señal atraviesa un filtro digital tipo FIR con enventanado Hamming, ambos aplicados a la mitad de la frecuencia de muestreo. La señal de entrada es la misma que se aplicó en los casos de la Fig. 11 y 12.

#### IV. CONCLUSIÓN

Se logró una correcta implementación de un espectrómetro con limitaciones asociadas al kit de desarrollo que se utilizó para la misma. Se comprobó su adecuado funcionamiento mediante una serie de pruebas, entre ellas las mostradas. Como se puede apreciar, los ensayos apuntaron al fin didáctico de fijar conceptos vistos a lo largo de la asignatura como son sistemas operativos embebidos, filtros digitales y FFT. No se ahondó en cuantificar todos los límites del desarrollo obtenido por ser caso de estudio de otra cátedra, si bien se tiene en cuenta como trabajo futuro. Entre los parámetros que se cuantificaron se puede mencionar que la máxima frecuencia que admite el espectrómetro es de 500 kHz, dada la máxima velocidad de muestreo del ADC interno del microcontrolador (1 MHz).

Se pudo apreciar gráficamente y de una forma más amigable con el usuario el efecto de la aplicación de filtros

digitales sobre una señal de entrada que realizaba un barrido en frecuencia.

Se observó también limitaciones en el algoritmo que calcula la FFT, por ejemplo, en los casos en que la frecuencia de la señal de entrada fue igual a un múltiplo de la resolución de la FFT, se pudo apreciar un único pico en el analizador de espectro y una línea fina y continua en el espectrograma. Cuando no coincidieron las frecuencias, se produjo un derramamiento o leakage de la señal, resultando en picos extendidos a las señales vecinas con amplitud menor a la real. Este fenómeno se puede apreciar en la Fig. 9, donde la frecuencia de la señal oscila entre 3120 Hz y 3130 Hz. En el espectrograma se aprecian tramos del historial donde se visualiza una línea fina y continua, período donde la señal está a una frecuencia de 3125 Hz (múltiplo de la resolución), y tramos del historial donde la línea es gruesa y discontinua, período donde los valores se extiende a los datos vecinos del arreglo, que en este caso son los extremos de la oscilación.

Una opción para atenuar el fenómeno de derramamiento es aumentar el valor de FFT\_LENGTH, permitiendo una mejor discriminación de la señal en el analizador de espectro, a costa de aumentar el tiempo de cálculo para el procesamiento de la señal, y en un aumento de la memoria requerida. También se puede modificar la frecuencia de muestreo, manteniendo FFT\_LENGTH constante, y visualizando una parte del arreglo en la pantalla.

Otra opción es la aplicación de una función de ventana, siendo las más comunes las de Hamming, Hanning o Blackman, lo que implica multiplicar los valores de cada bloque de muestras de la señal de entrada por el coeficiente correspondiente de la función de ventana, lo que reduce el derramamiento espectral, para luego calcular la FFT.

#### REFERENCIAS

- [1] Samir S. Soliman, “Señales y Sistemas continuos y discretos”, segunda edición, Madrid 1999, ISBN: 84-8322-154-3.
- [2] Steven W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, Segunda edición, California 1999, ISBN 0-9660176-6-8.
- [3] ST Microelectronics. Reference Manual RM0090. STM32F407xx, STM32F415xx and STM32F417xx. Disponible en: [http://www.st.com/web/en/resource/technical/document/reference\\_manual/DM00031020.pdf](http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf)
- [4] STM32F407 DISCOVERY. Disponible en: [http://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html](http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html)
- [5] Jennifer N. Robbins, “Web design in a nutshell”, tercera edición, EEUU 2006, ISBN: 978-0-596-00987-8.
- [6] SSD1963 1215KB Embedded Display SRAM LCD Display Controller - Disponible en: <https://cdn.displaytech-us.com/sites/default/files/driver-ic-data-sheet/Solomon-Systech-SSD1963.pdf>
- [7] DSP library. Disponible en: [http://www.st.com/content/ccc/resource/technical/document/user\\_manual/b1/01/3b/ab/75/ef/4d/91/CD00144074.pdf/files/CD00144074.pdf/jcr:content/translations/en.CD00144074.pdf](http://www.st.com/content/ccc/resource/technical/document/user_manual/b1/01/3b/ab/75/ef/4d/91/CD00144074.pdf/files/CD00144074.pdf/jcr:content/translations/en.CD00144074.pdf)
- [8] FreeRTOS open source licensing. 22 de diciembre de 2017. – Disponible en <https://www.freertos.org/a00114.html>