

**DISEÑO E IMPLEMENTACIÓN DE UN APLICATIVO COMPUTACIONAL PARA
LA ADQUISICIÓN DE SEÑALES ELÉCTRICAS POR MEDIO DE UN FPGA**

CRISTIAN CAMILO ACEVEDO RINCÓN

JESSICA LORENA PLAZAS NIÑO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES**

BUCARAMANGA

2018

**DISEÑO E IMPLEMENTACIÓN DE UN APLICATIVO COMPUTACIONAL PARA
LA ADQUISICIÓN DE SEÑALES ELÉCTRICAS POR MEDIO DE UN FPGA**

CRISTIAN CAMILO ACEVEDO RINCÓN

JESSICA LORENA PLAZAS NIÑO

**Trabajo de Grado para optar al título de
Ingeniero Electrónico**

Director

Carlos Andrés Angulo Julio

Ingeniero Electrónico

Codirector

Ana Beatriz Ramírez Silva

Doctora en Ingeniería Eléctrica

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES**

BUCARAMANGA

2018

DEDICATORIAS

A mis padres José Acevedo y Patricia Rincon y hermanos Daniel Acevedo y Gina Acevedo, quienes siempre estuvieron apoyándome incondicionalmente, su ayuda fue muy útil en el desarrollo de este proyecto.

A mi compañera de proyecto y novia Jessica Plazas, ya que su apoyo, trabajo y dedicación es un pilar muy importante en mi vida.

A mi Director Carlos Angulo, quien siempre nos brindó su respaldo, tiempo y sus conocimientos para la solución de los diferentes problemas presentados antes, durante y después del producto final de este trabajo.

A la laboratorista Liliana por sus vitales conocimientos, apoyo y dispositivos para nuestro proyecto.

Cristian Camilo Acevedo Rincon

A Dios por darme paciencia y fortaleza para poder sobrellevar toda la carga académica y emocional que ésta carrera conlleva.

A mis padres Martha Niño y Luis Plazas por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como mi aprendizaje como persona, por su incondicional apoyo mantenido a través del tiempo.

A mis tíos Nancy Plazas, Germán Niño y abuelos Alfonso Niño, Soledad Moreno, Luis Plazas y Rosa Niño por el apoyo incondicional y económico, cada paso fue logrado con la ayuda de éstas personas tan magníficas.

A mi compañero de proyecto de grado, amigo y novio Cristian Acevedo por soportar mis cambios de humor repentinos por el estrés que éste proyecto nos dejaba cada día de traspaso, además de ser mi soporte, mi compañero de alegrías y tristezas.

A nuestro Director Carlos Angulo por su dedicación, apoyo y severidad en cada fase de nuestro proyecto, ya que nos formó como personas íntegras y responsables.

A nuestra laboratorista Liliana por apoyarnos en todo el proceso de nuestro proyecto, dándonos ánimo y facilitando los implementos de trabajo para poder culminarlo.

Jessica Lorena Plazas Niño

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. OBJETIVOS	18
1.1. OBJETIVO GENERAL	18
1.2. OBJETIVOS ESPECIFICOS	18
2. DISEÑO DE FILTROS CON HDL CODER	19
2.1. HDL CODER	19
2.2. DISEÑO DE FILTROS	20
3. ESTRUCTURA DEL PROTOTIPO	27
3.1. DISEÑO E IMPLEMENTACIÓN EN FPGA	27
3.1.1. Control del convertidor analógico digital	29
3.1.1.1. Modos de operación del XADC	33
3.1.1.2. Cuantificación del XADC	33
3.1.1.3. Frecuencia de muestreo del XADC	34
3.1.1.4. Sincronización del XADC con los otros componentes	38
3.1.2. Protocolo de comunicación UART	38
3.1.2.1. Bloque de recepción (UART-RX):	39
3.1.2.2. Bloque de transmisión (UART_TX):	40
3.1.2.3. Bloque de control (FSM):	43
3.2. DISEÑO E IMPLEMENTACIÓN DEL APLICATIVO COMPUTACIONAL	46
4. RESULTADOS	51

4.1. RESULTADOS EXPERIMENTALES	53
4.2. ANÁLISIS DE RESULTADOS.....	66
5. CONCLUSIONES	68
BIBLIOGRAFÍA	71

LISTA DE FIGURAS

	Pág.
Figura 1. Interfaz de HDL Coder	19
Figura 2. Interfaz de Filter Design HDL Coder	20
Figura 3. Interfaz de filter designer	21
Figura 4 Estructura filtro digital	24
Figura 5. Filtro pasa bajas 1 para $F_s = 1$ MHz	24
Figura 6. pasa bajas 2 para $F_s = 1$ MHz	25
Figura 7. Filtro rechaza banda 1 para $F_s = 1$ MHz	25
Figura 8. Filtro rechaza banda 2 para $F_s = 1$ MHz	26
Figura 9. Esquema del procedimiento para adquirir las señales	27
Figura 10. Sistema de adquisición implementado.	29
Figura 11 XADC Wizard	30
Figura 12. Diagrama del XADC.	31
Figura 13. Función de transferencia unipolar	34
Figura 14. Diagrama de sincronización para adquisición y conversión de una muestra	36
Figura 15. Diagrama de sincronización para adquisición y conversión	36
Figura 16. Estructura de datos transmisión / Recepción	39
Figura 17. Diagrama de muestreo de bits	39
Figura 18. Bloque de Recepción (UART_RX)	40
Figura 19. Bloque de transmisión (UART_TX)	41
Figura 20. Máquina de estado de UART transmisión (FSM_INTERNA)	42
Figura 21. Control de comunicación (FSM)	44
Figura 22. Máquina de estados (FSM_CONTROL_INT)	45
Figura 23. Configuración ADC y Filtros.	47
Figura 24. Buscar archivo IMPACT	48

Figura 25. Archivo incorrecto	48
Figura 26. Configuración ADC y Filtros prueba	49
Figura 27. Ventana grafica señal eléctrica y densidad espectral de potencia	50
Figura 28. Aplicativo para generar señales de audio.	51
Figura 29. Circuito de adecuación de la señal de audio	52
Figura 30. Señal sin filtrar (2590 [Hz] Y 14030 [Hz])	53
Figura 31. Espectro de la señal de la Figura 30	54
Figura 32. Señal adquirida por el osciloscopio	54
Figura 33. Espectro de la señal adquirida por el osciloscopio	55
Figura 34. Señal filtrada con filtro pasa bajas 1	55
Figura 35. Espectro de la señal filtrada con filtro pasa bajas 1	56
Figura 36. Señal sin filtrar (2112 [Hz], 10090 [Hz] y 14200 [Hz])	56
Figura 37. Espectro de la señal de la Figura 36	57
Figura 38. Señal adquirida por el osciloscopio	57
Figura 39. Espectro de la señal adquirida por el osciloscopio	58
Figura 40. Señal filtrada con filtro pasa bajas 2	58
Figura 41. Espectro de la señal filtrada con filtro pasa bajas 2	59
Figura 42. Señal sin filtrar (5637 [Hz], 9664 [Hz] y 13860 [Hz])	59
Figura 43. Espectro de la señal de la Figura 42	60
Figura 44. Señal adquirida por el osciloscopio	60
Figura 45. Espectro de la señal adquirida por el osciloscopio	61
Figura 46. Señal filtrada con filtro rechaza banda 1	61
Figura 47. Espectro de la señal filtrada con filtro rechaza banda	62
Figura 48. Señal sin filtrar (471 [Hz], 8229 [Hz] y 19330 [Hz])	62
Figura 49. Espectro de la señal de la Figura 48	63
Figura 50. Señal adquirida por el osciloscopio	63
Figura 51. Espectro de la señal adquirida por el osciloscopio	64
Figura 52. Señal filtrada con filtro rechaza banda 2	64
Figura 53. Espectro de la señal filtrada con filtro rechaza banda 2	65

LISTA DE TABLAS

Pág.

Tabla 1 Características de la Artix 7	22
Tabla 2. Descripción de los Registros de control XADC.....	31
Tabla 3. Direcciones de los canales del XADC.....	32
Tabla 4. Estados vs salidas (FSM_CONTROL_INT)	45
Tabla 5. Resumen resultados para Fs= 1MHz.....	65
Tabla 6. Resumen resultados para Fs= 500kHz.....	66
Tabla 7. Resumen resultados para Fs= 154kHz.....	66

LISTA DE ANEXOS

(Ver anexos adjuntos en el CD y pueden visualizarlos en la Base de Datos de la Biblioteca UIS)

Anexo A. Graficas de los filtros con frecuencia de muestreo 500kHz y 154kHz.

Anexo B. Código en MATLAB 2017 del aplicativo computacional.

Anexo C. Manual de Usuario del aplicativo computacional.

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UN APLICATIVO COMPUTACIONAL PARA LA ADQUISICIÓN DE SEÑALES ELÉCTRICAS POR MEDIO DE UN FPGA. •

AUTORES: CRISTIAN CAMILO ACEVEDO RINCON

JESSICA LORENA PLAZAS NIÑO. **

PALABRAS CLAVES: FPGA, FILTROS DIGITALES, ADC, APLICATIVO, UART.

DESCRIPCIÓN:

El grupo de investigación GOTS se encuentra realizando estudios que involucran la adquisición de señales eléctricas con el sistema de desarrollo *ADC12D1800RFRB* fabricado por Texas Instruments. Los datos adquiridos con esta tarjeta pueden ser enviados a un computador para ser analizados por el software *Wavevision5*. Sin embargo, en este software la modificación de parámetros de adquisición tales como número de muestras y frecuencia de muestreo está limitada.

En el presente proyecto se realizará un prototipo a escala basado en FPGA que permita la modificación de algunos parámetros de adquisición con el fin de evaluar la viabilidad de desarrollar las pruebas deseadas por grupo GOTS. Se realizará un prototipo para la adquisición de señales eléctricas por medio de un convertidor analógico-digital controlado por el FPGA Artix 7 de la tarjeta Nexys 4 DDR, el cual provee una conversión de señales analógicas a digitales de hasta de 1 MSPS, donde la señal proveniente de uno de los cuatro canales de adquisición del ADC podrá ser visualizada en un aplicativo computacional elaborado en MATLAB, además de permitir el filtrado de la señal adquirida por medio de filtros predeterminados en el aplicativo. Para la validación de este prototipo se utilizarán señales de audio creadas por sinusoidales puras y combinaciones de estas suministradas por un aplicativo móvil a través del puerto de audio, las cuales deben estar comprendidas entre el rango de frecuencias de 10 Hz a 20 kHz.

El prototipo constará de los siguientes cuatro componentes: (1) Un módulo de adquisición de datos, (2) un módulo de comunicación FPGA-Computador, (3) un módulo de filtrado diseñado con ayuda de la herramienta *HDL Coder* de Matlab y (4) un aplicativo computacional que permita modificar los parámetros de adquisición y de visualizar los datos adquiridos.

• Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías eléctrica, electrónica y de telecomunicaciones. Director: Carlos Andrés Angulo Julio, Ingeniero electrónico

ABSTRACT

TITLE: DESIGN AND IMPLEMENTATION OF A COMPUTATIONAL APLICATIVE FOR THE ACQUISITION OF ELECTRICAL SIGNAL THROUGH AN FPGA.*

AUTHORS: CRISTIAN CAMILO ACEVEDO RINCON
JESSICA LORENA PLAZAS NIÑO.**

KEYWORDS FPGA, DIGITAL FILTERS, ADC, APLICATIVE, UART.

DESCRIPTION:

The GOTS research group is carrying out studies that involve the acquisition of electrical signals with system development ADC12D1800RFRB manufactured by Texas Instruments. The data acquired with this card can be sent to a computer to be analyzed by the Wavevision5 software. However, in this software modification of acquisition parameters such as number of samples and sampling frequency is limited.

In this Project will realized a prototype to scale based on FPGA that will allows the modification of some parameters of acquisition in order to evaluate the viability of development of the desired test for the group GOTS. A prototype will be made for the acquisition of electrical signals by means of an analog-digital converter controlled by the FPGA Artix 7 of the Nexys 4 DDR card, which provides a conversion from analog to digital signals of up to of 1 MSPS, where the signal coming from one of the four channels of acquisition of the ADC can be visualized in a computational applicative developed in MATLAB, as well as allowing the filtering of the signal acquired by default in the application filters. For the validation of this prototype, audio signals will be used, pure sin waves and combination of these, supplied by a mobile applicative through an audio port, which must be between the frequency range of 10 Hz to 20 KHz.

The prototype will consist of the following four components: (1) A data acquisition module (2) a communication module FPGA computer, (3) a filtering module designed with help of the HDL Coder of MATLAB tool and (4) a computational application that allows to modify parameters of acquisition and display the acquired data.

* Bachelor Degree Project.

** Faculty of Physics Mechanics Engineering. Electrical, Electronics Engineering and Telecommunications School. Director: Carlos Andrés Angulo Julio, Electronic Engineer

INTRODUCCIÓN

La adquisición de las señales eléctricas ha facilitado el análisis de información con fines científicos, esto ha permitido tomar muestras del mundo real y generar datos con ayuda de dispositivos electrónicos diseñados para este tipo de situaciones. Se debe tener en cuenta qué tipo de señal se desea digitalizar para elegir los componentes adecuados para el funcionamiento del sistema de adquisición. En ocasiones, los componentes no cuentan con las características deseadas para manipular la información obtenida. En esta situación se encontraron miembros del grupo de Óptica y Tratamiento de Señales Ópticas (GOTS) de la Universidad Industrial de Santander, quienes actualmente se encuentran realizando estudios que involucran la adquisición de señales eléctricas con el sistema de desarrollo *ADC12D1800RFRB* de Texas Instruments¹. Este dispositivo permite convertir señales eléctricas analógicas a digitales con frecuencias de muestreo de hasta 3.6 Giga Muestras por segundo. Los datos adquiridos con esta tarjeta pueden ser enviados a una computadora por medio del software *Wavevision5*². Sin embargo, en este software la modificación de algunos parámetros para la adquisición de la señal eléctrica, tales como número de muestras y frecuencia de muestreo, es restringida.

Nosotros proponemos desarrollar un prototipo a escala basado en FPGA que posibilite modificar parámetros de adquisición con el fin de evaluar la viabilidad de realizar las pruebas deseadas por el grupo GOTS. Este prototipo permitirá la adquisición de señales eléctricas comprendidas entre 10 Hz y 20 KHz a través de la tarjeta de desarrollo Nexys 4 DDR³, la cual posee un ADC cuya máxima

¹ TEXAS INSTRUMENTS, 12-Bit, Dual 1.8 GSPS or Single 3.6 GSPS A/D Converter Reference Board, 2017.

² TEXAS INSTRUMENTS, WaveVision 5 Data Capture Board Version 5.1, 2017.

³ DIGILENT, *Nexys4 DDR FPGA Board*, 2017.

frecuencia de muestreo es 1 MHz, 12 bits de resolución y cuatro canales de adquisición. Las señales adquiridas podrán ser visualizadas por medio de un aplicativo computacional diseñado por nosotros. Adicionalmente, el aplicativo permitirá el filtrado de las señales adquiridas a través de filtros digitales predeterminados implementados con ayuda de la herramienta HDL coder de MATLAB⁴.

⁴ MATHWORKS, MATLAB & Simulink HDL Coder, 2017

1 OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar e implementar una herramienta computacional que permita visualizar en un computador los datos adquiridos por un convertidor analógico-digital controlado por el FPGA Artix 7 de la tarjeta Nexys 4 DDR.

1.2 OBJETIVOS ESPECIFICOS

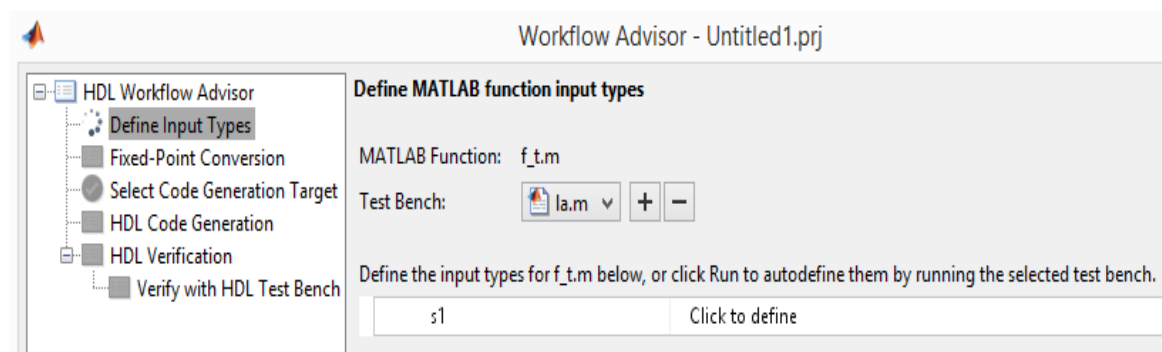
- Desarrollar en FPGA un módulo de control de un convertidor analógico digital para la adquisición de señales.
- Implementar un sistema de comunicación entre el FPGA y un computador para el envío de los datos adquiridos por el ADC.
- Diseñar un módulo de filtrado digital en FPGA con la ayuda de la herramienta *HDL coder* de Matlab.
- Elaborar un aplicativo computacional con interfaz gráfica que permita configurar algunos parámetros de adquisición y visualizar los datos adquiridos o filtrados.

2 DISEÑO DE FILTROS CON HDL CODER

2.1 HDL CODER

Es una herramienta de Matlab que, a partir de funciones y bloques de simulink, genera circuitos digitales descritos en VHDL y Verilog que pueden ser implementados en FPGAs o prototipos ASIC. Esta herramienta también permite estimar el uso de los recursos que se emplearían de acuerdo al FPGA seleccionado y genera bancos de pruebas para simular el comportamiento del circuito obtenido⁵. Adicionalmente, HDL coder (*Figura 1*) cuenta con la herramienta Filter Design HDL Coder (*Figura 2*) que permite la implementación de filtros digitales⁶.

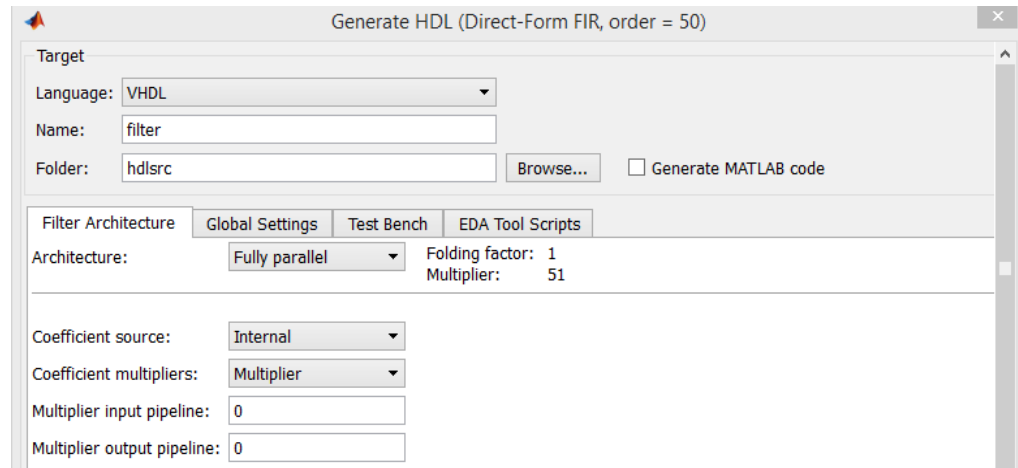
Figura 1. Interfaz de HDL Coder



⁵ MATHWORKS, MATLAB & Simulink HDL Coder, 2017

⁶ MATHWORKS, Filter Design HDL Coder, 2017

Figura 2. Interfaz de Filter Design HDL Coder

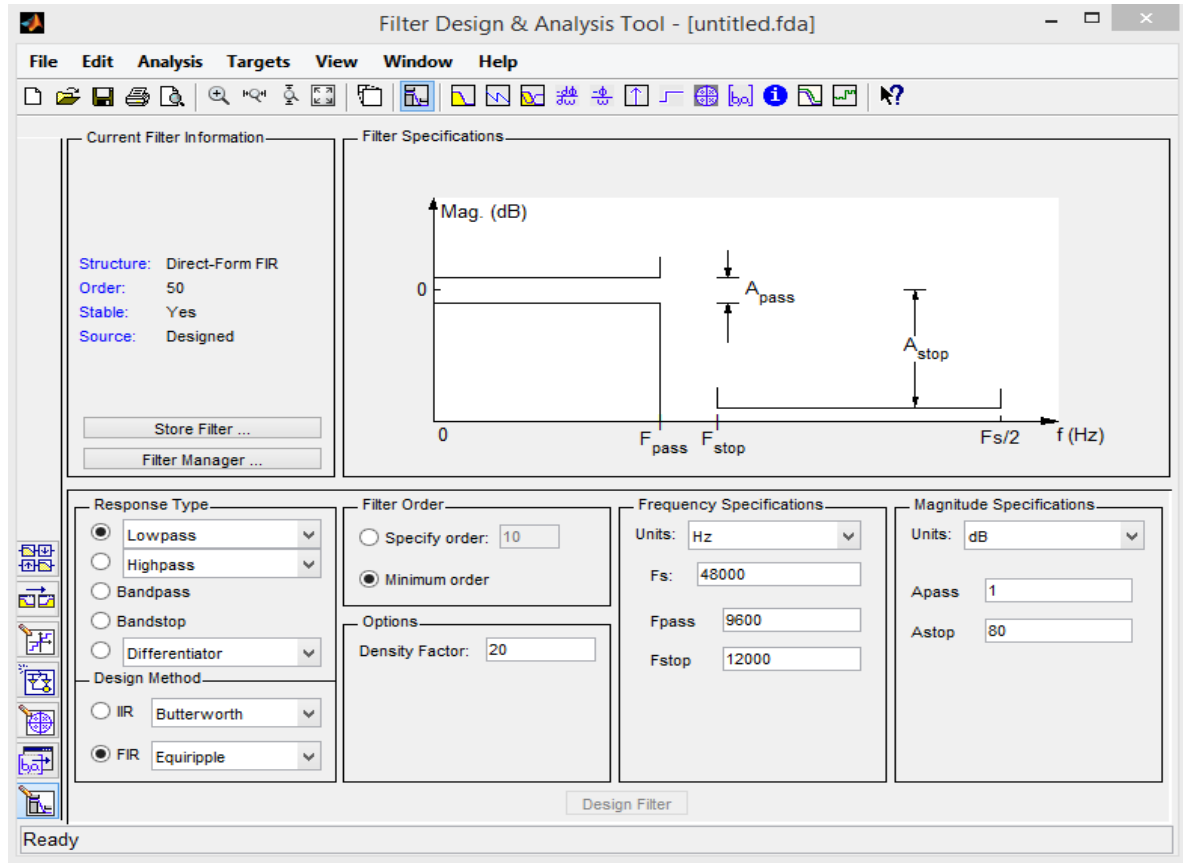


2.2 DISEÑO DE FILTROS

Los filtros implementados fueron diseñados con ayuda de la herramienta Filter Designer de Matlab (Mathworks, s.f.). Para el correcto uso de la herramienta se deben tener en cuenta, entre otros, los siguientes parámetros (ver *Figura 3*):

- Tipo de filtro
- Orden del filtro
- Frecuencia de muestreo
- Frecuencias de corte

Figura 3. Interfaz de filter designer



Una vez diseñado el filtro, se deben especificar los parámetros de cuantificación para su implementación en FPGA teniendo en cuenta las prestaciones de esta última, lo cual hará que la respuesta del filtro sea diferente a la deseada. Para la implementación del prototipo se escogió el sistema de desarrollo Nexys 4 DDR, el cual posee un puerto UART para la comunicación con un computador y un FPGA Artix-7 (XC7A100T) que cuenta con un convertidor analógico digital de 12 bits y 4860 kbits en bloques RAM para almacenar resultados (ver *Tabla 1*)⁷.

⁷ XILINX, 7 Series FPGAs Data Sheet, 2017

Tabla 1 Características de la Artix 7

<i>FPGA</i>	<i>XC7A100T</i>
<i>Celdas logicas</i>	<i>101440</i>
<i>Slices</i>	<i>15850</i>
<i>CLB Flip-Flops</i>	<i>126800</i>
<i>Maximo Distribuido (Kb)</i>	<i>1188</i>
<i>Bloque RAM/FIFO w/ECC (36 Kb each)</i>	<i>135</i>
<i>Ram de Bloque Total (Kb)</i>	<i>4860</i>
<i>Señal Mixta Analogica (AMS) / XADC</i>	<i>1</i>

Fuente: Adaptado de XILINX, *7 Series FPGAs Data Sheet*, 2017.

Dado que la transmisión de datos por el puerto UART se realiza en tramas de 8 bits, se decidió que la salida de los filtros podría tener 16 bits con lo cual podremos almacenar hasta 278480 muestras, si se desea aumentar la calidad del filtro se debe aumentar el número de bits que representa cada coeficiente y los bits de salida, además, disminuirá la cantidad de muestras a adquirir y aumentará la cantidad de tramas a enviar. Sin embargo, para pruebas de este prototipo se decidió utilizar 65536 muestras que pueden ser enviadas al computador en dos tramas de 8 bits cada muestra.

En este proyecto se decidió usar dos tipos de filtros FIR Equiripple, Pasa bajas y Rechaza banda ya que estos filtros pueden diseñarse para que tengan una fase lineal y estabilidad en su respuesta en frecuencia. Cada filtro fue diseñado con 3 frecuencias de muestreo diferentes: 154kHz, 500kHz y 1000 kHz. El resultado final son 4 filtros por frecuencia de muestreo, dos filtros pasa-bajas y dos rechaza-banda, para un total de 12 filtros.

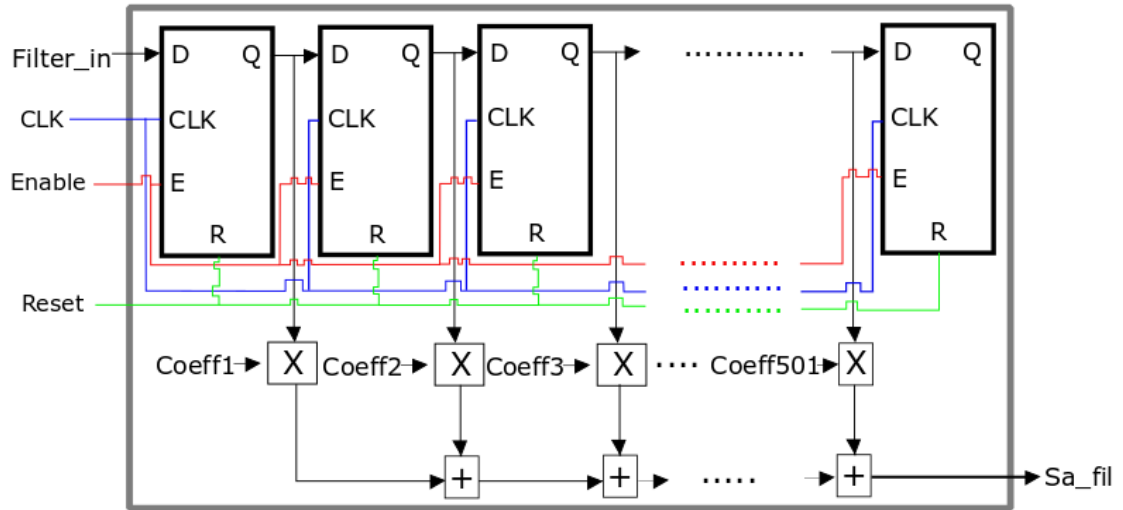
A continuación, se presentan las características empleadas en el diseño de los filtros:

- Orden del filtro: 500
- Aritmética del filtro: Punto fijo.
- Bits de entrada: 9 bits.
- Bits de salida: 16 Bits
- Bits para representación de cada constante: 9 bits
- Cantidad de bits para multiplicaciones: Se debe tener en cuenta que la entrada del filtro se multiplicará con cada constante, por lo tanto, este valor es de 18 Bits.

Cantidad de bits para acumulación de sumas: se debe tener en cuenta la cantidad de bits de las multiplicaciones (18) y el número de constantes del filtro (501), este valor es 519 (ver *Figura 4*) para conservar el número máximo de muestras y no aumentar el tiempo de transmisión. Sin embargo, disminuir la resolución implica que no se podrán filtrar señales con voltajes del orden de los micro voltios (μV).

Los Bits de entrada corresponden a los datos arrojados por el XADC, en este caso la resolución es disminuida, de 12 bits a 8 bits, esto se hace con el fin de garantizar una salida del filtro de 16 bits para no aumentar el tiempo de transmisión y no disminuir el número de muestras. Sin embargo, disminuir la resolución implica que no se podrán filtrar señales con voltajes del orden de los micro voltios (μV).

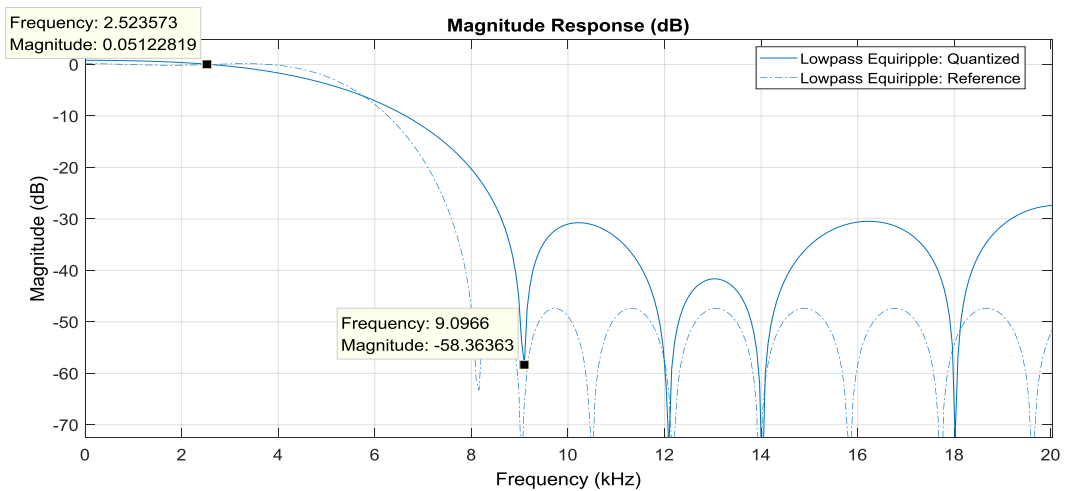
Figura 4 Estructura filtro digital



En las figuras 5 a 8 se muestra la respuesta en frecuencia de los filtros implementados con frecuencia de muestreo 1 MHz. En ellas se puede observar el efecto de la cuantificación en la respuesta del filtro:

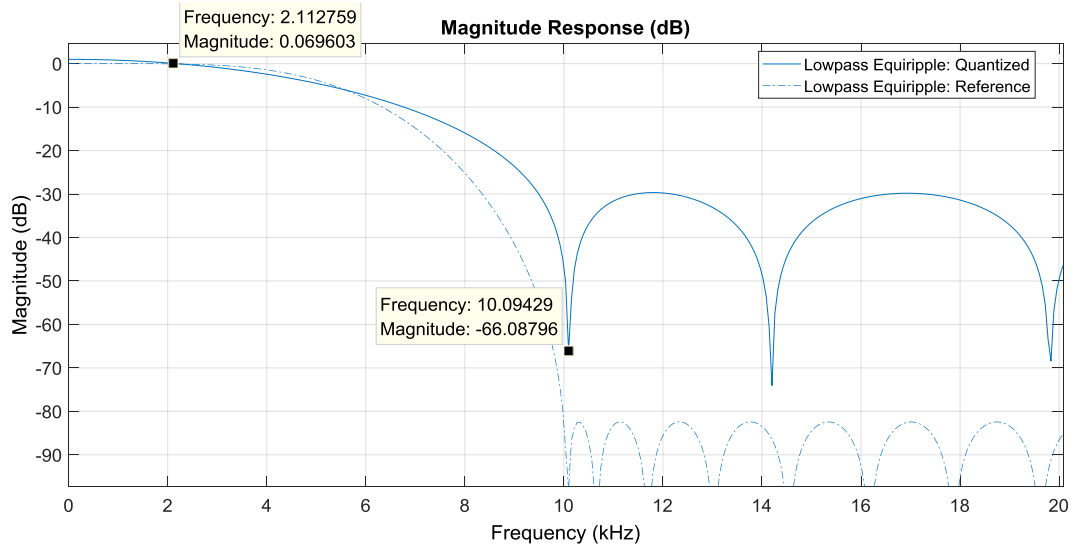
- Características: $F_{\text{pass}}=4000$ y $F_{\text{stop}}=8000$ Hz.

Figura 5. Filtro pasa bajas 1 para $F_s=1$ MHz



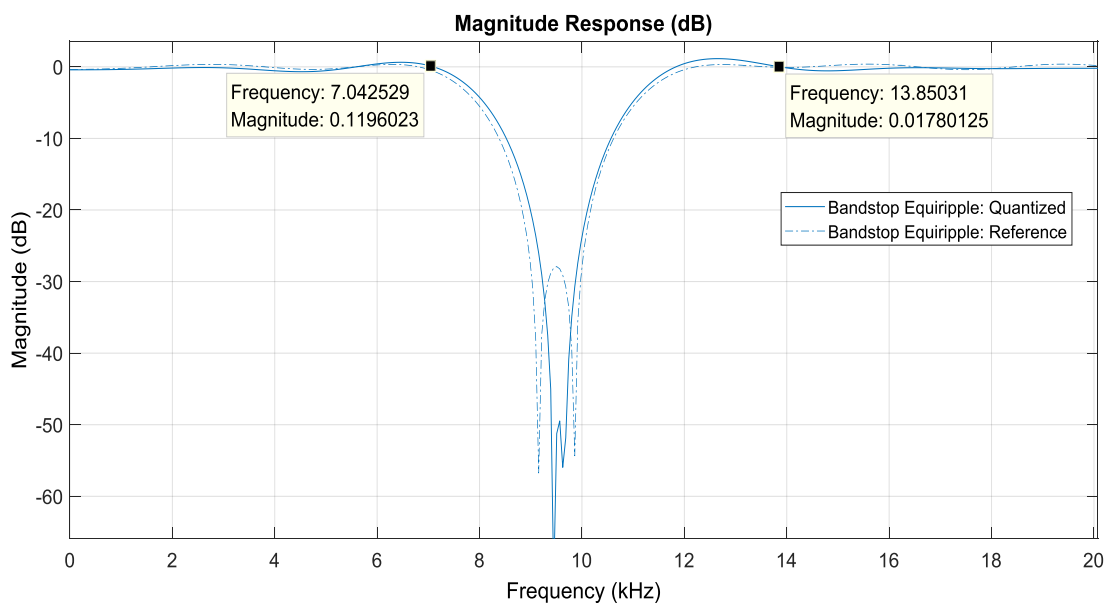
- Características: $F_{\text{pass}}=1000$ y $F_{\text{stop}}=10000$ Hz.

Figura 6. pasa bajas 2 para $F_s=1$ MHz



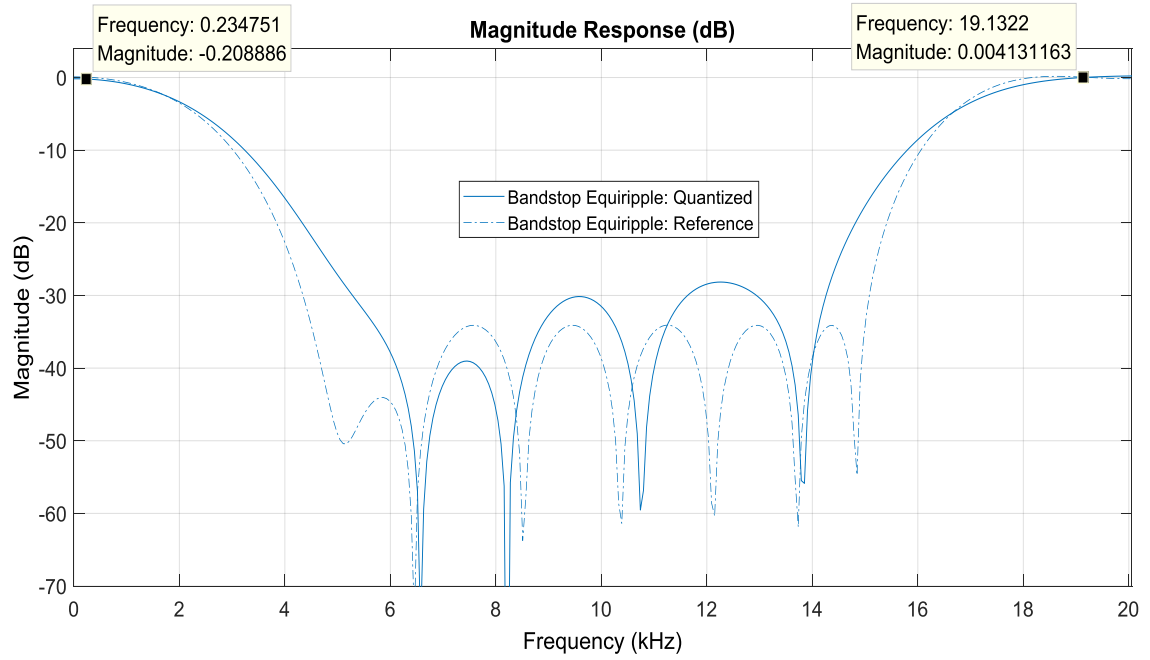
- Características: $F_{\text{pass1}}=7000$, $F_{\text{stop1}}=9000$, $F_{\text{stop2}}=10000$ y $F_{\text{pass2}}=12000$ Hz.

Figura 7. Filtro rechaza banda 1 para $F_s=1$ MHz



- Características: $F_{\text{pass1}}=500$, $F_{\text{stop1}}=4500$, $F_{\text{stop2}}=15000$ y $F_{\text{pass2}}=18000$ Hz.

Figura 8. Filtro rechaza banda 2 para $F_s = 1$ MHz



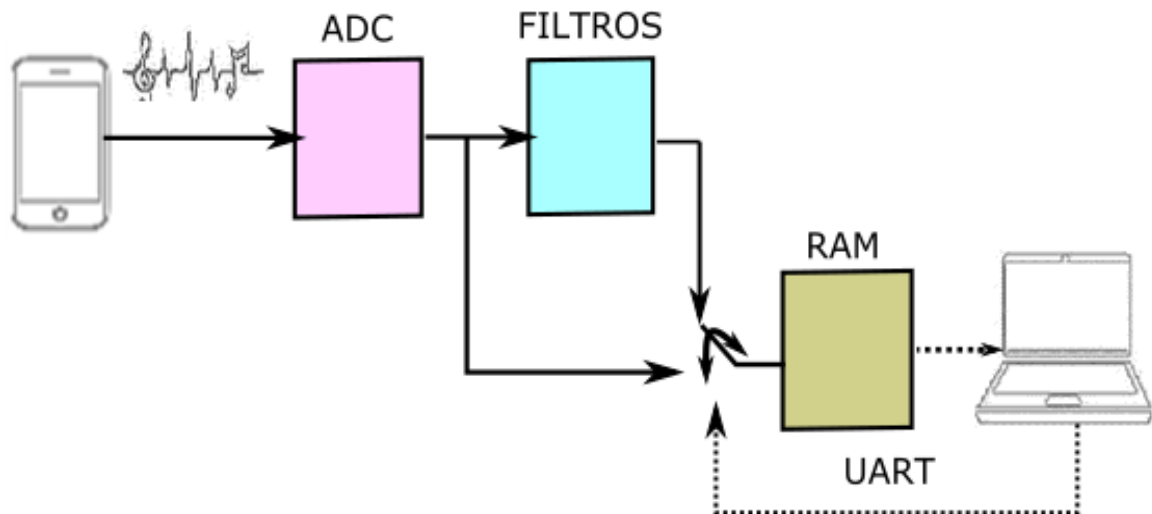
3 ESTRUCTURA DEL PROTOTIPO

Atendiendo los requerimientos planteados por el grupo GOTS, se propuso la implementación de un sistema de adquisición basado en FPGA y un aplicativo elaborado en MATLAB para visualizar en computador los datos adquiridos.

3.1 DISEÑO E IMPLEMENTACIÓN EN FPGA

El ADC recibe la señal de voltaje unipolar obtenida por uno de los cuatro canales y realiza la conversión a digital usando 12 bits de resolución. Los datos de salida del ADC son enviados al módulo de filtrado o hacia la RAM para luego ser transmitidos al aplicativo computacional por medio del módulo de transmisión (ver *Figura 9*).

Figura 9. Esquema del procedimiento para adquirir las señales



La *Figura 10* muestra el diseño implementado en FPGA. Este diseño está conformado por seis bloques principales:

- Bloque de Adquisición de datos (Mod_ADC): convierte las señales analógicas a digitales a través del canal del ADC elegido por el usuario.
- Bloque de filtros (FILTROS): filtra la señal digital adquirida por el bloque de adquisición de acuerdo a la opción elegida por el usuario.
- Bloque de almacenamiento (RAM): guarda las muestras de la señal adquirida o la señal filtrada.
- Bloque de transmisión UART (Uart_Tx): envía las muestras guardadas al aplicativo computacional.
- Bloque de recepción UART (Uart_Rx): recibe los comandos de configuración elegidos por el usuario desde el aplicativo computacional.
- Bloque de control (FSM): controla el flujo de datos al interior del sistema.

Figura 11 XADC Wizard

LogiCORE **XADC Wizard** xilinx.com:ip:xadc_wiz:2.1

Component Name

Sim File Name

Startup Channel Selection

- Simultaneous Sampling
- Independent ADC
- Single Channel
- Channel Sequencer

Timing Mode

- Continuous Mode
- Event Mode

DRP Timing options

- Enable DCLK

DCLK Frequency (MHz): Range: 8..250

ADC conversion rate (kSPS): Range: 154..1000

- Increase acquisition time

Clock divider value = 4

ADC Clock Frequency(MHz) = 25.00

Actual Conversion Rate(kSPS) = 961.54

[Datasheet](#) [< Back](#) Page 1 of 5 [Next >](#) [Generate](#) [Cancel](#) [Help](#)

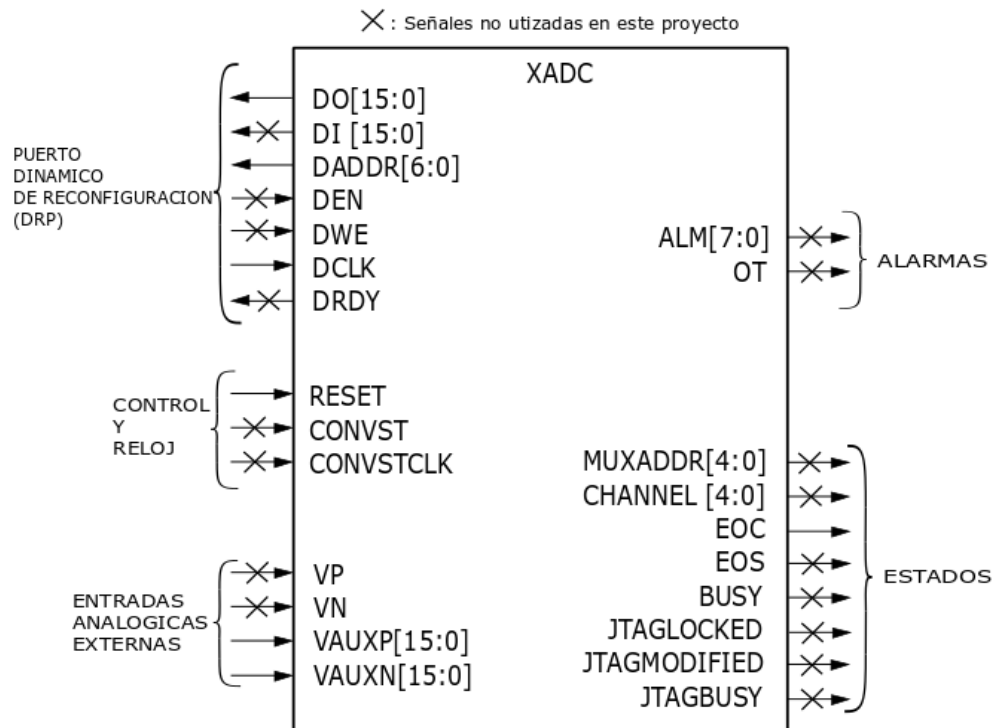
Fuente: Xilinx, *LogiCore IP XADC Wizard v2.1,2017.*

Tabla 2. Descripción de los Registros de control XADC.

NOMBRE	DIRECCIÓN	DESCRIPCIÓN
Registro 0	40h	Selecciona canal de entrada del ADC
Registro 1	41h	Habilita la calibración, el modo de operación del ADC
Registro 2	42h	Cambia el reloj del ADC y por ende la frecuencia de muestreo
Registros de prueba	43h a 4Fh	Estos registros se utilizan para la prueba de fábrica y deben dejarse en la inicialización predeterminada
Registros secuencia	48h a 4Fh	Programan la función del secuenciador de canales
Registros de alarma	50h a 5Ch	Umbral para la función de alarma del XADC, la alarma de temperatura se enciende con valores por encima de 85 °C y con apagado automático con 125 °C

Fuente: Adaptada de XILINX, XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER, 2017.

Figura 12. Diagrama del XADC.



Fuente: Adaptada de XILINX, XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER, 2017.

Para acceder a los puertos de entradas del XADC, se debe usar el conector PMOD JXADC de la placa de desarrollo Nexys 4 DDR. Hay cuatro canales disponibles, cada canal con un pin positivo y uno negativo. Debido a que en este proyecto las señales de entrada son señales unipolares, el pin negativo de cada canal va conectado a tierra, mientras que la positiva va conectada directamente a la señal que se desea muestrear. Para seleccionar el canal de entrada se debe tener en cuenta que cada uno de los 4 canales cuenta con una dirección específica, la cual debe ser configurada en el registro_0. En la *Tabla 3* se muestran las direcciones correspondientes a cada uno de los canales de adquisición empleados.

El XADC posee 16 entradas analógicas diferenciales auxiliares (VAUXP [15:0], VAUXN [15:0]) de las cuales 4 pertenecen a los canales del ADC (ver *Figura 12*). Éstas están predeterminadas por la notación del archivo UCF de la Nexys 4 DDR, en el cual se indica que las entradas del ADC son los Auxiliares 3, 10, 2 y 11 (ver *Tabla 3*). Los otros auxiliares son usados para otros componentes como el sensor de temperatura, el acelerómetro, entre otros.

Tabla 3. Direcciones de los canales del XADC

Canal	Nombre	PIN CONEXIÓN (PMOD)	Dirección(Hexadecimal)
1 (+)	AD3P	pin 1	"0013"
1 (-)	AD3N	pin 7	"0013"
2 (+)	AD10P	pin 2	"001a"
2 (-)	AD10N	pin 8	"001a"
3 (+)	AD2P	pin 3	"0012"
3 (-)	AD2N	pin 9	"0012"
4 (+)	AD11P	pin 4	"001b"
4 (-)	AD11N	pin 10	"001b"

3.1.1.1 Modos de operación del XADC

EL XADC posee 4 modos de operación⁹: muestreo simultáneo, independiente, canal único y secuenciador. La frecuencia de muestreo del convertidor se divide en cada uno de los canales habilitados cuando se utiliza en los modos de operación de muestreo continuo, independiente y secuenciador. Lo anterior lleva a que la frecuencia de muestreo por canal sea menor, limitando las señales que se podrían digitalizar por el ADC.

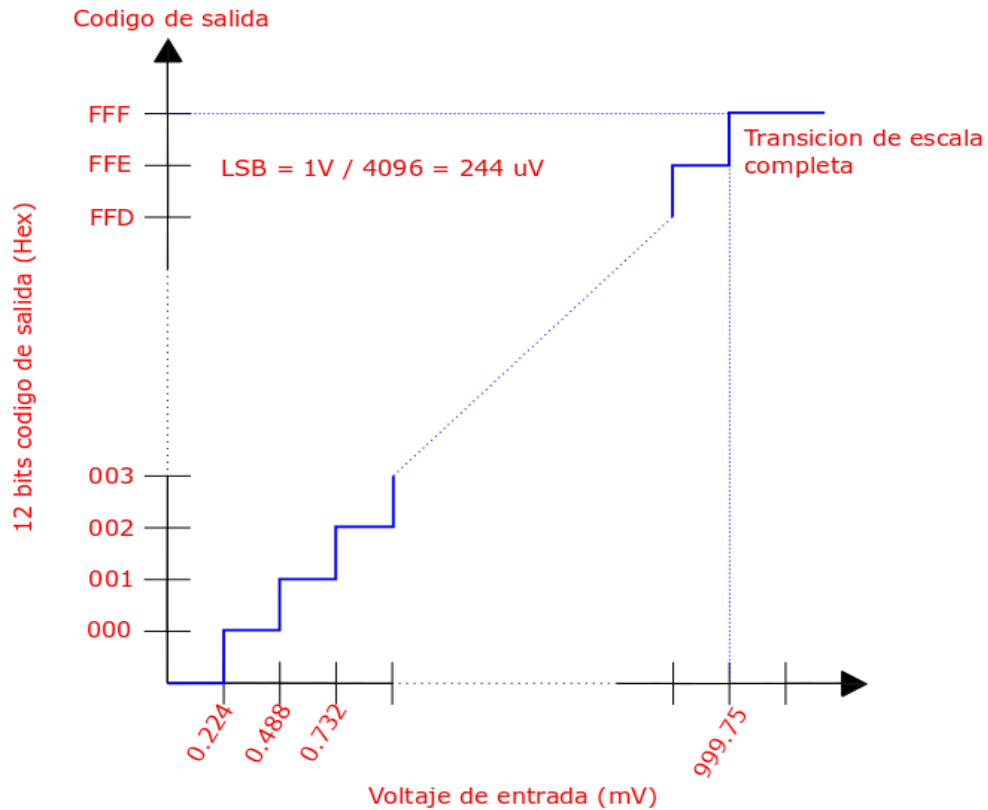
Por lo tanto, el modo de operación utilizado fue el canal único, ya que en este actúa un canal a la vez mientras los otros canales están desconectados. Pero cabe aclarar que el usuario puede elegir por medio del aplicativo cuál de los 4 canales desea usar para la adquisición.

3.1.1.2 Cuantificación del XADC

En la *Figura 13* se muestra la función de transferencia unipolar de 12 bits para el ADC. El rango de voltaje en las entradas analógicas del ADC es de 0[V] a 1[V]. Las transiciones de código diseñadas se producen en valores sucesivos enteros de LSB, cuyo valor está dado por la ecuación 1:

⁹ XILINX, *XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER*, 2017.

Figura 13. Función de transferencia *unipolar*



Fuente: Adaptada de XILINX, XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER, 2017.

$$LSB = \frac{1[V]}{2^{12}} = 244 [\mu V] \quad (1)$$

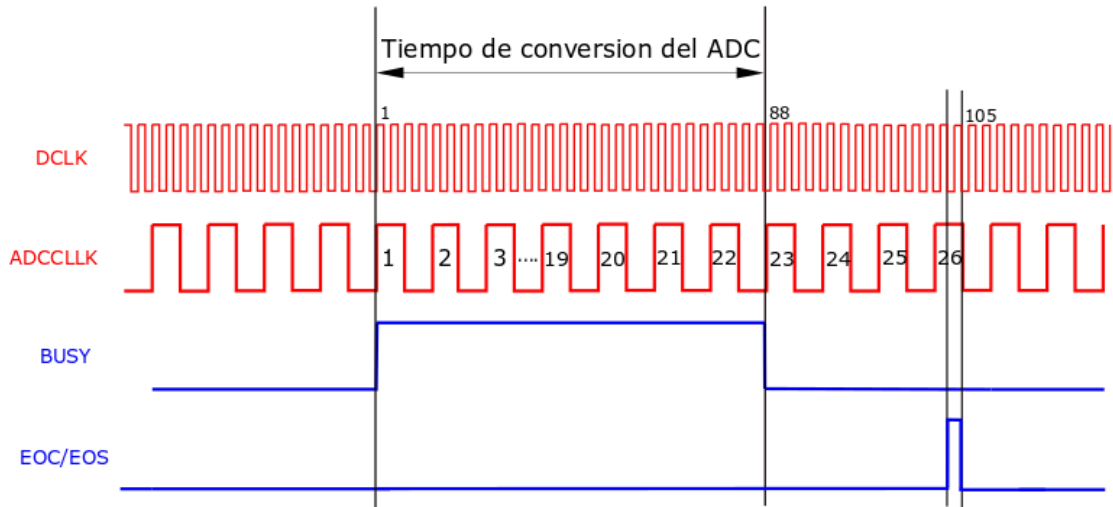
3.1.1.3 Frecuencia de muestreo del XADC

El XADC puede operar en dos modos de sincronización: modo continuo y por eventos. Utilizamos el modo continuo ya que en este modo la frecuencia de muestreo depende de $ADCCLK$ y esta señal se deriva del $DCLK$, mientras que en modo por eventos el usuario debe proporcionar una señal de inicio de conversión que define la frecuencia de muestreo. El periodo de muestreo se compone de dos

etapas: tiempo de conversión y tiempo de adquisición. Durante el tiempo de adquisición, el ADC comienza a adquirir la próxima muestra que se digitalizará en la fase de conversión.

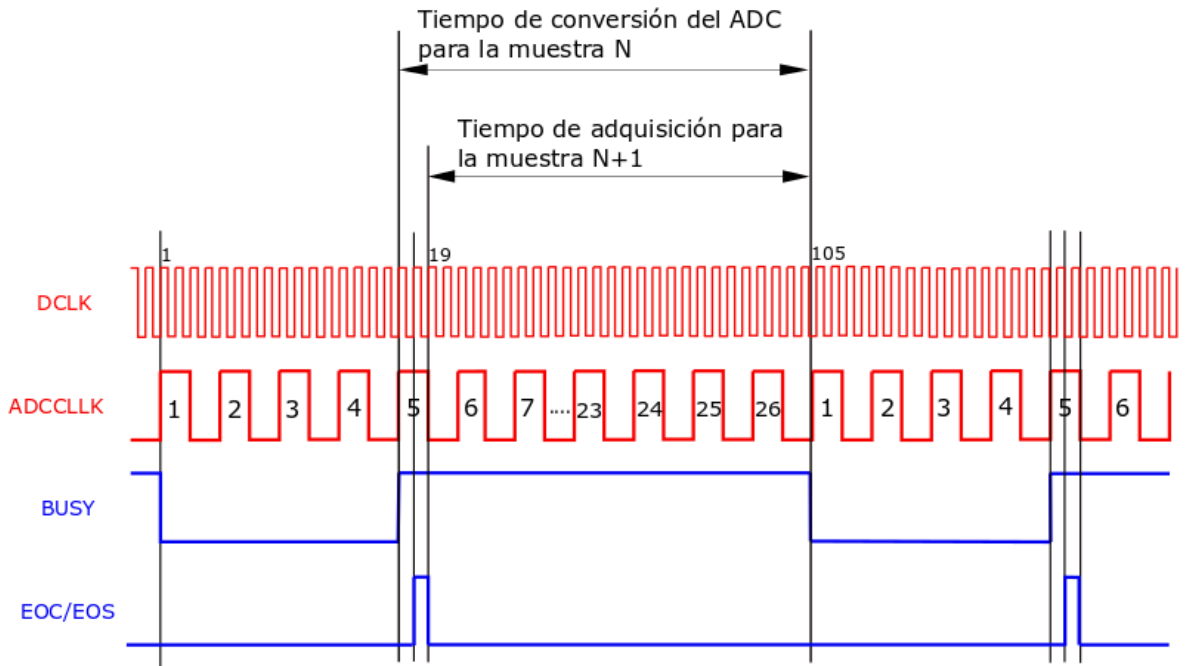
La tasa de conversión máxima específica para el ADC es 1MSPS o un tiempo de conversión de $1\mu\text{s}$, además, en modo de muestreo continuo se requieren 26 ciclos del *ADCCLK* para adquirir una señal analógica y realizar una conversión. Esto implica una frecuencia máxima del *ADCCLK* de 26 MHz, el rango de frecuencia de *ADCCLK* según el datasheet es de 4 a 26 MHz. La señal *DCLK* corresponde al reloj de 100 MHz perteneciente a la Nexys 4, *DCLK* puede ser hasta de 250 MHz, el *ADCCLK* es un reloj derivado de *DCLK* que cambiara dependiendo la frecuencia de muestreo deseada por el usuario. La fase de conversión empieza al final del ciclo 4 de *ADCCLK*, este tiempo corresponde a tiempo de establecimiento de los componentes de *XADC*. La señal *BUSY* se pone en alto en el siguiente flanco ascendente de *DCLK* para indicar que el ADC está realizando una conversión. La fase de conversión dura 22 ciclos del *ADCCLK*, 16 ciclos del *DCLK* después de que *BUSY* pasa a bajo, *EOC* pasa a un alto un ciclo del *DCLK* e indica cuando el resultado de conversión ha terminado, la señal *EOS* indica el final de una secuencia y tiene el mismo tiempo que *EOC*.

Figura 14. Diagrama de sincronización para adquisición y conversión de una muestra



Fuente: Adaptada de XILINX, XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER, 2017.

Figura 15. Diagrama de sincronización para adquisición y conversión



Fuente: Adaptada de XILINX, XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER, 2017.

En la *Figura 14* se ilustra el proceso de conversión para solo un dato mientras que en la *Figura 15* se muestra el proceso para adquirir varios datos.

Para establecer la frecuencia de muestreo primero se debe tener en cuenta la relación entre el reloj del sistema y el reloj del ADC. Esta relación se define de acuerdo a la ecuación 2:

$$Div = round\left(\frac{DCLK}{ADCCLK}\right) \quad (2)$$

La frecuencia de muestreo del XADC está regida por la ecuación 3:

$$XADC_{fs} = \frac{DCLK}{C * Div}, \quad (3)$$

donde $DCLK$ es $100MHz$. C es 26 (correspondiente al número de ciclos de reloj necesarios para la conversión) y Div es un valor entre 4 y 25 (de acuerdo la relación de relojes de la ecuación 2).

Las frecuencias de muestreo utilizadas en este proyecto se determinaron así:

$$XADC_{fs1} = \frac{100MHz}{26 * 4} \cong 961.54 kHz \quad (4)$$

$$XADC_{fs2} = \frac{100MHz}{26 * 8} \cong 480.77 \text{ kHz} \quad (5)$$

$$XADC_{fs3} = \frac{100MHz}{26 * 25} \cong 153.85 \text{ kHz} \quad (6)$$

3.1.1.4 Sincronización del XADC con los otros componentes

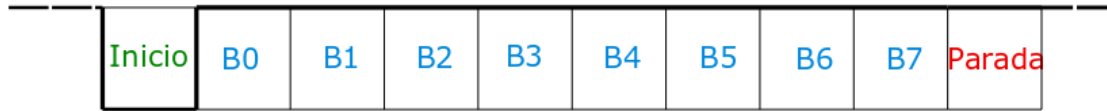
Dado que el convertidor trabaja con una señal de reloj más lenta que la del resto de componentes, la señal que indica la finalización de conversión se puede mantener activa por más de un ciclo de reloj de la Nexys 4 DDR (100 MHz). Lo anterior provocaría que la muestra dada por el ADC ingrese varias veces al filtro o a la memoria. Para evitar esto se hace uso de un circuito detector de flanco, el cual a su salida da una señal en alto solamente un ciclo de reloj de la Nexys 4 DDR, generando así una sincronización entre el ADC, y los demás componentes.

3.1.2 Protocolo de comunicación UART

El protocolo de comunicación empleado en este proyecto fue modificado del DEMO GPIO que proporciona DIGILENT¹⁰. Este diseño fue implementado en tres bloques: Bloque de recepción (Uart_Rx), bloque de transmisión (Uart_Tx) y bloque de control (FSM). La *Figura 16* muestra cómo deben enviarse los datos bidireccionalmente entre la FPGA y el computador.

¹⁰ DIGILENT, *Nexys 4 DDR GPIO Demo*, 2017.

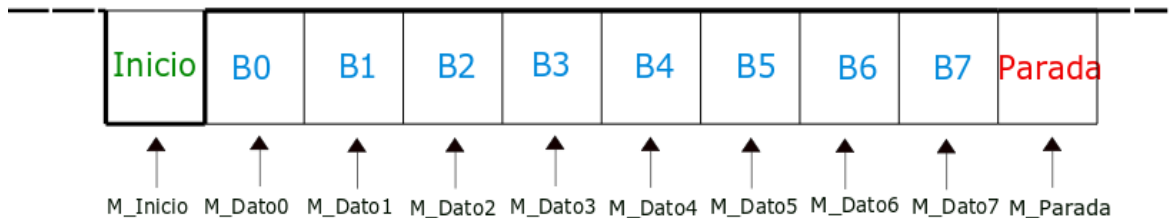
Figura 16. Estructura de datos transmisión / Recepción



3.1.2.1 Bloque de recepción (UART-RX):

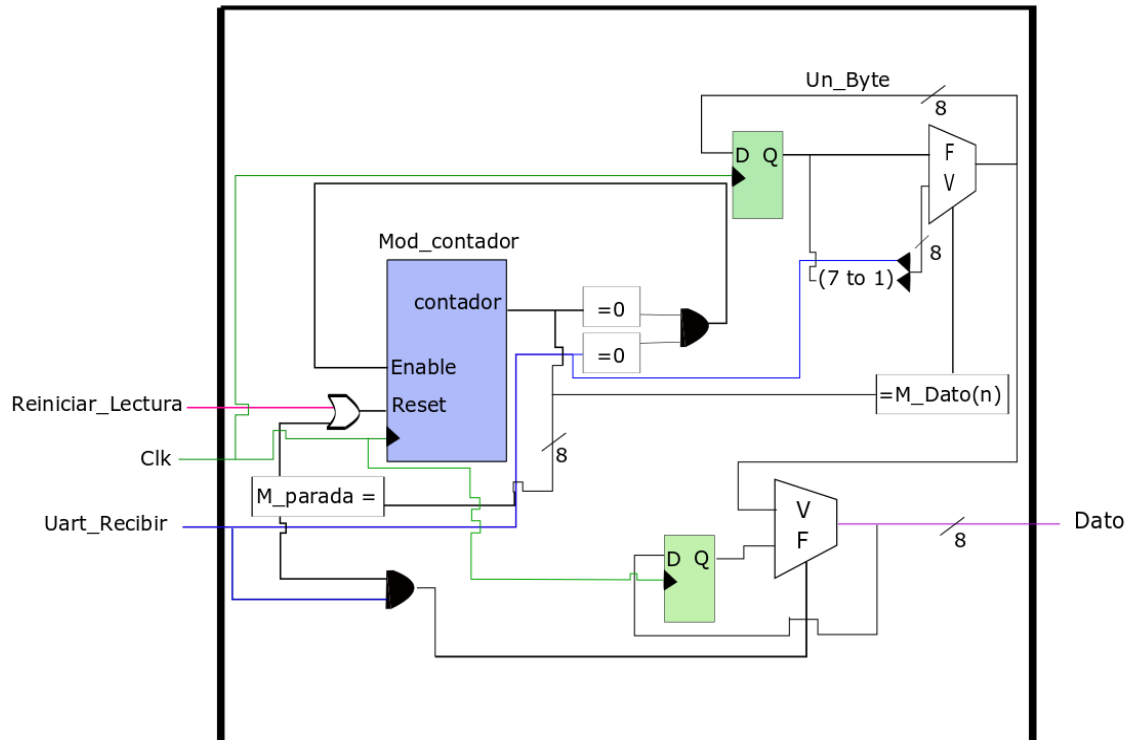
La recepción de los datos que ingresan al sistema se realiza por medio de muestreo de los diez bits que componen la trama. El muestreo se realiza tal cómo se muestra en la *Figura 17*, puesto que si se tomara entre los bordes habría error de lectura del dato.

Figura 17. Diagrama de muestreo de bits



La Figura 18 muestra el esquema del circuito implementado para la recepción de datos. El proceso de recepción inicia cuando ocurre un cambio de un 1 lógico a un 0 lógico en la señal de entrada $UART_{RECIBIR}$. Esto activa un contador que determina cuándo se debe capturar el valor (de acuerdo al muestreo establecido en la *Figura 17*) y almacenarlo en el registro de desplazamiento. Una vez guardados los 8 bits de información, se espera al bit de parada para enviar el dato a la señal de salida del circuito.

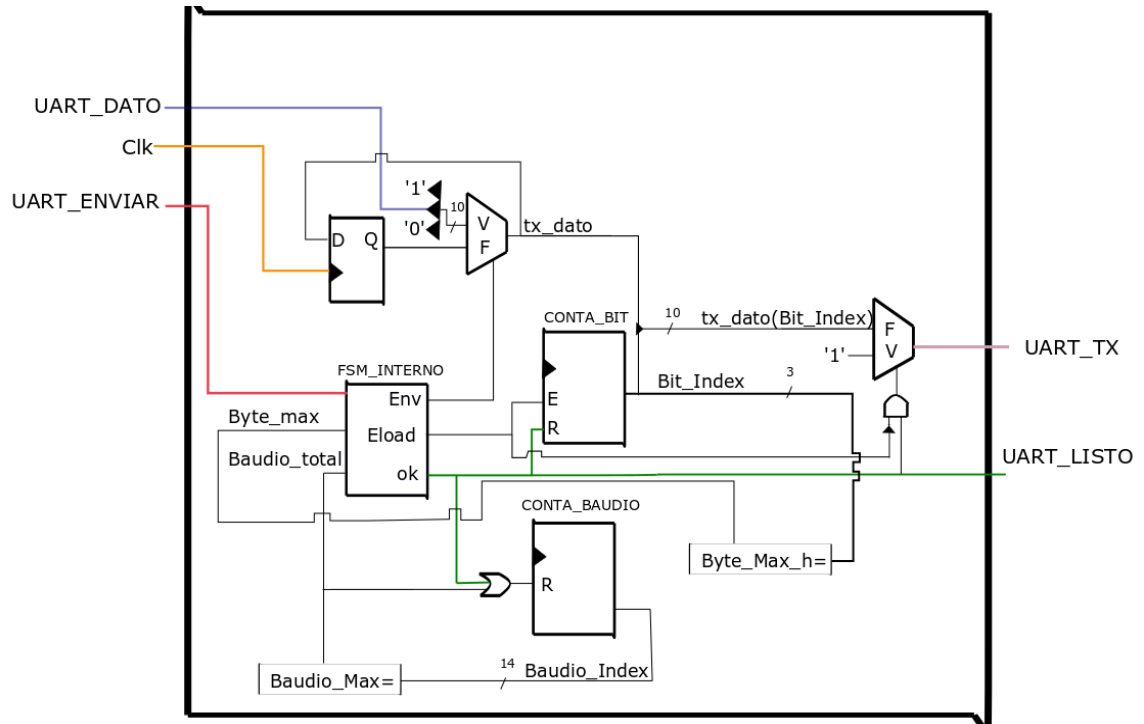
Figura 18. Bloque de Recepción (UART_RX)



3.1.2.2 Bloque de transmisión (UART_TX):

El protocolo de transmisión UART TX convierte los datos entrada paralelos (UART_DATO) a serie para ser enviados al computador. Para esto utiliza un registro de desplazamiento que se activa acorde con la tasa de baudios establecida y el conteo de bits a enviar. La máquina de estados (FSM INTERNA) se encarga de sincronizar el sistema para el envío de los datos teniendo en cuenta las banderas, tal como se muestran en la *Figura 19* y *Figura 20*.

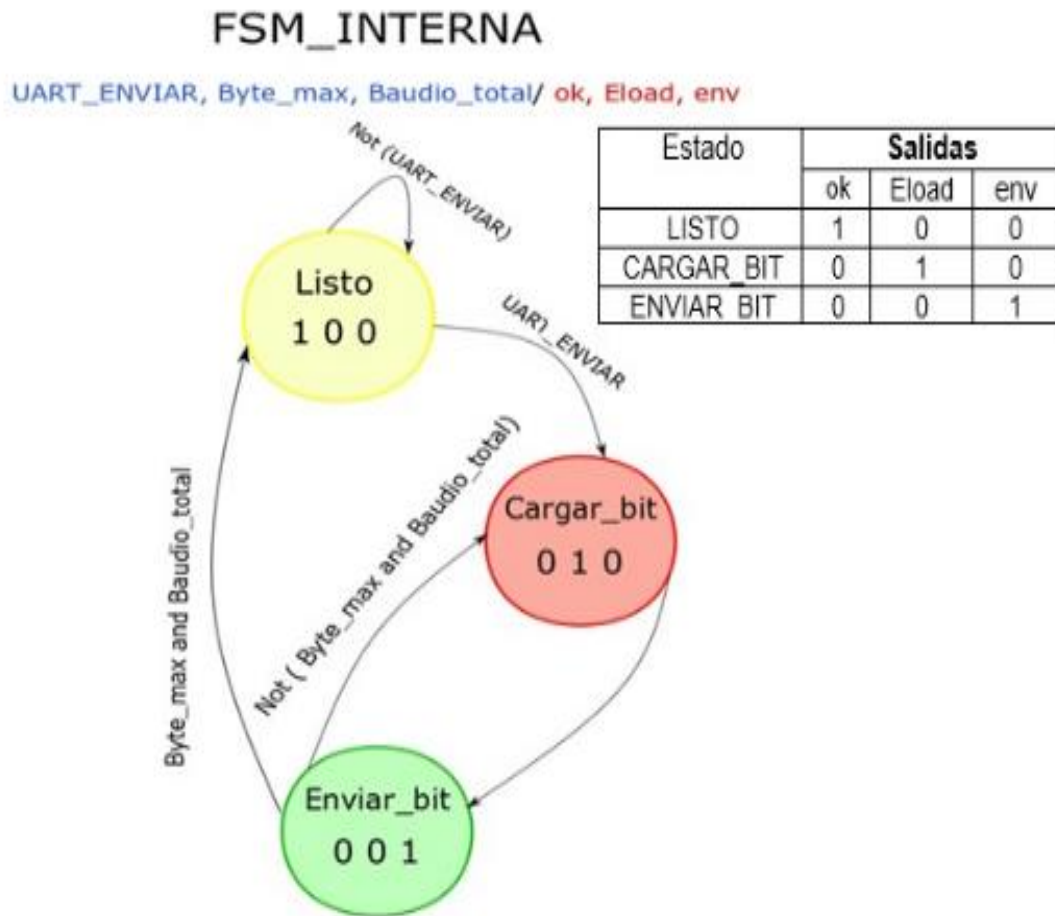
Figura 19. Bloque de transmisión (UART_TX)



Para transmitir los datos almacenados en el sistema se tiene en cuenta la señal *UART_ENVIAR* ya que ésta da la orden de comenzar el proceso de envío. De acuerdo a la tasa de baudios establecida, la velocidad de transmisión se calcula de acuerdo a la ecuación 7, sin muestrear los datos en comparación con el bloque *UART_RX*:

$$Baudio_total = round\left(\frac{Frecuencia_{sistema}}{Baudios}\right) - 1 \quad (7)$$

Figura 20. Máquina de estado de UART transmisión (FSM_INTERNA)



Teniendo en cuenta que la $Frecuencia_{sistema}$ que es de 100MHz y $Baudios$ 9600, la velocidad de transmisión es de 10416.

Cuando el valor de la señal $Baudio_Index$ llegue a ser igual a la señal $Baudio_total$ y el valor de la señal Bit_Index igual a $Byte_max_h$ se habrá enviado un dato de 10 bits a través de la señal $UART_TX$, además de avisar al sistema que el dato se envió por medio de la señal $UART_LISTO$.

3.1.2.3 Bloque de control (FSM):

El bloque de control (FSM) es el encargado de sincronizar el sistema entre el bloque de recepción y el bloque de transmisión. Para el envío de información se tiene en cuenta la longitud del dato de salida, como este dato es de 16 bits se distribuye en dos tramas, ya que el protocolo UART sólo permite enviar máximo 8 bits por trama.

El tiempo de transmisión/Recepción de una trama de datos se define de acuerdo a la ecuación 8:

$$Tiempo_{trc} = \frac{\#Bits_datos}{Baudios} \quad (8)$$

En este caso $\#Bits_datos$ corresponden a 10, ya que el primer bit es el bit de inicio, los siguientes 8 bits son los bits de información y el último bit corresponde al bit de parada, y los Baudios establecidos son 9600. Para calcular el tiempo de transmisión de los datos almacenados en la memoria del FPGA al aplicativo computacional, se utiliza la ecuación 9:

$$Tiempo_{transmisión} = Tiempo_{trc} * C_{datos} * Fraccion, \quad (9)$$

donde C_{datos} es la cantidad de datos almacenados en memoria (esto es, 65356 muestras) y $Fraccion$ es la cantidad de tramas a enviar (en nuestro caso se necesitan dos tramas para enviar los 16 bits de cada muestra). Por tanto, el tiempo de transmisión de los datos almacenados en la memoria del FPGA al aplicativo es de 136,53 Segundos.

La *Figura 21* muestra el esquema del circuito de control. Cuando las señales $UART_{LISTO}$ y $LEER_{BAN}$ están activadas (1 lógico) y la señal $FILAS$ está desactivada (0 lógico), es decir que el $CONTADOR_{FILA}$ no ha llegado a 2 (este número corresponde al número de filas que tiene la $MATRIZ_{[2 \times 8]}$ del dato de entrada X), ocurre que la máquina de estado (ver *Figura 22*) pasa de INICIO a ENVIAR_DATO, acá se activa las señales $UART_ENVIAR$ para dar aviso que el dato está listo para enviarse y la señal $FILA_E$ activa el registro para desplazar la primera trama de datos que corresponde a los primeros 8 bits más significativos, además, de activar el contador de filas.

Figura 21. Control de comunicación (FSM)

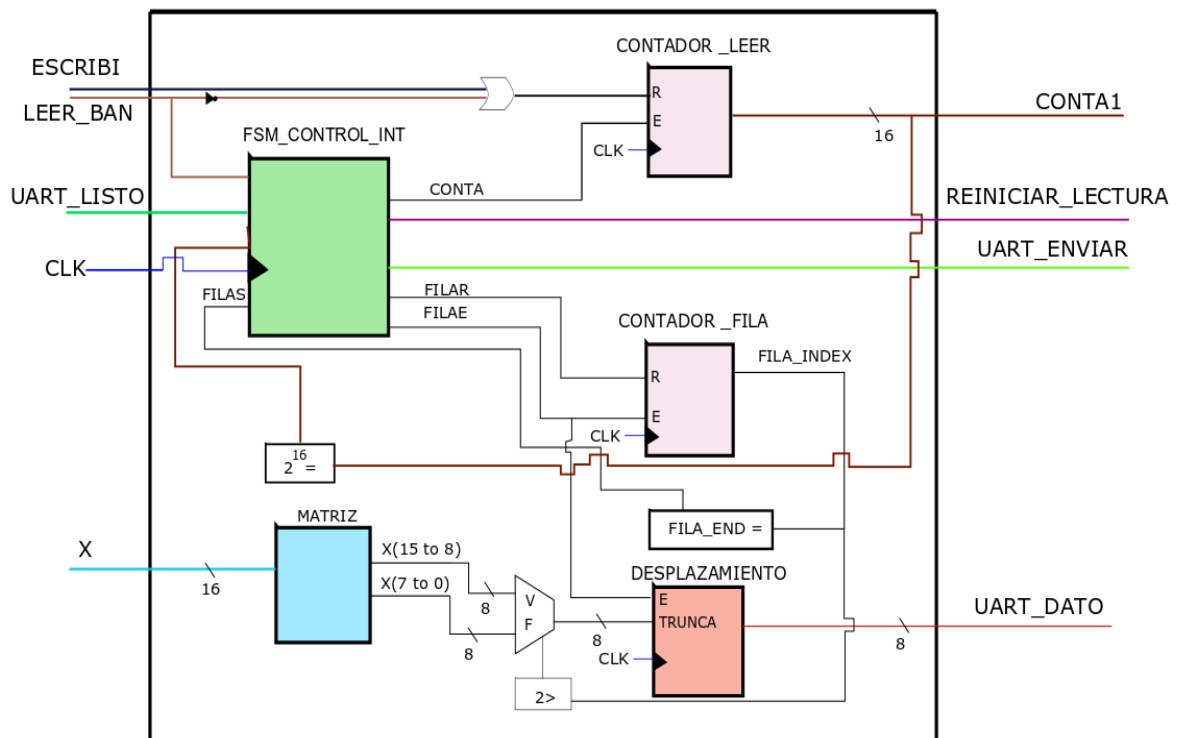


Figura 22. Máquina de estados (FSM_CONTROL_INT)

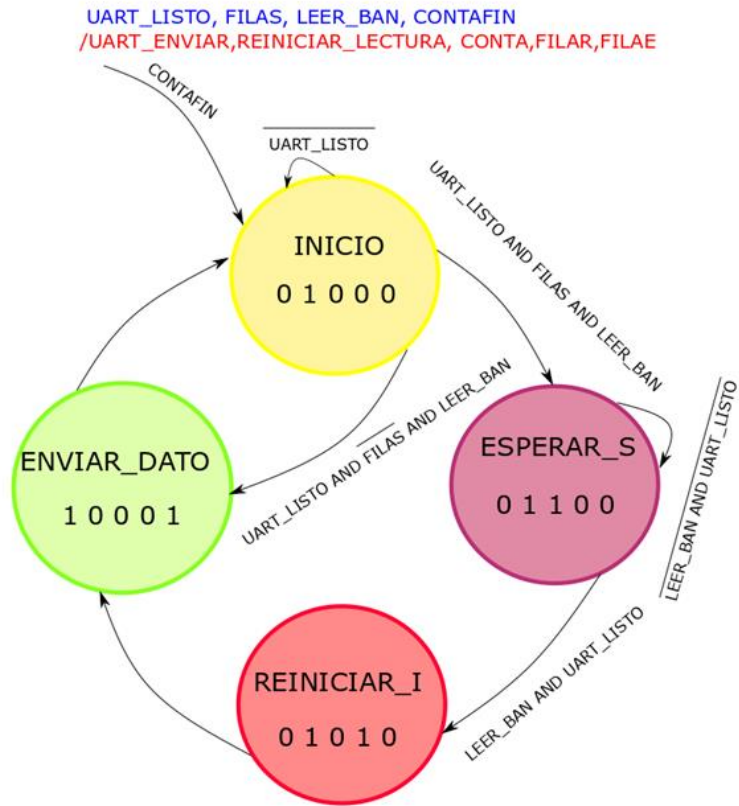


Tabla 4. Estados vs salidas (FSM_CONTROL_INT)

Estado	Salidas				
	UART_ENVIAR	REINICIAR_LECTURA	CONTA	FILAR	FILAE
INICIO	0	1	0	0	0
ESPERAR_S	0	1	1	0	0
REINICIAR_I	0	1	0	1	0
ENVIAR_DATO	1	0	0	0	1

Una vez enviado el primer dato la máquina de estados vuelve a su estado de *INICIO*. Cuando las señales *UART_LISTO*, *LEER_{BAN}* y *FILAS* están activas, la maquina pasa al estado *ESPERAR_SEÑAL* donde se activan las señales

REINICIAR_LECTURA y *CONTA*. Se quedará en ese estado si las señales *LEER_{BAN}* y *UART_LISTO* están desactivadas.

Cuando las señales *LEER_{BAN}* y *UART_LISTO* están activadas la máquina de estados pasa al estado *REINICIAR_I* allí se activa la señal *FILAR* corresponde a inicializar el contador de filas.

Luego de inicializar el contador de filas, pasa un ciclo de reloj después al estado *ENVIAR_DATO* para enviar la segunda trama de datos que corresponde a los 8 bits menos significativos.

El proceso se repite cíclicamente hasta que la señal *CONTA1* sea igual a (2^{16}) ya que este es el valor máximo de almacenamiento que tiene la RAM.

3.2 DISEÑO E IMPLEMENTACIÓN DEL APLICATIVO COMPUTACIONAL

El aplicativo se elaboró mediante la GUI de MATLAB. En él se usa el comando *COM* para usar el puerto serial, el comando *FFT* para calcular la transformada rápida de Fourier y los comandos *Freqz* y *mag2dB* para determinar la densidad espectral de potencia.

Cuenta con dos ventanas que proporcionan al usuario interactuar con la tarjeta. El aplicativo tiene unos requisitos básicos que están estipulados en el manual del usuario (carpeta de anexos).

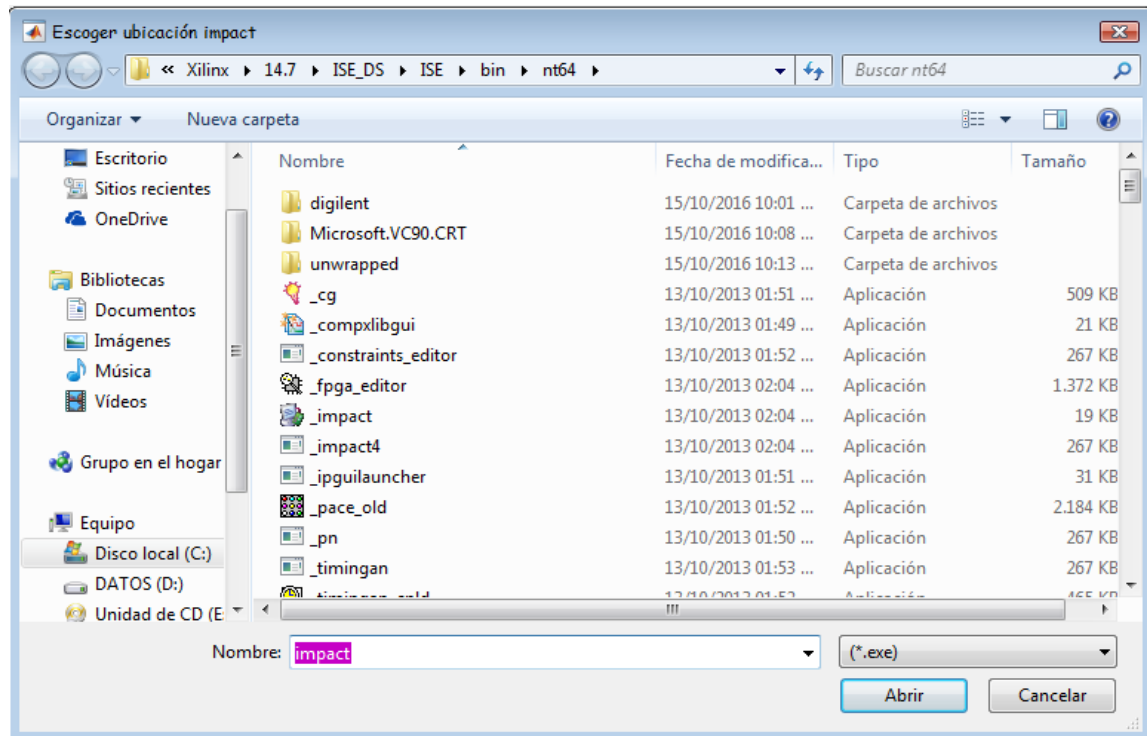
La primera ventana es llamada “Configuración ADC y Filtros” (ver Figura 23). En primera instancia el aplicativo pide al usuario validar la instalación de IMPACT de

XILINX ya que uno de los requisitos para utilizar el aplicativo es tenerlo instalado. Al dar click en el botón validar, emerge una ventana para buscar el archivo IMPACT (ver Figura 24).

Figura 23. Configuración ADC y Filtros.

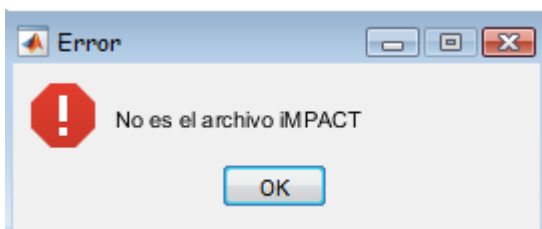


Figura 24. Buscar archivo IMPACT



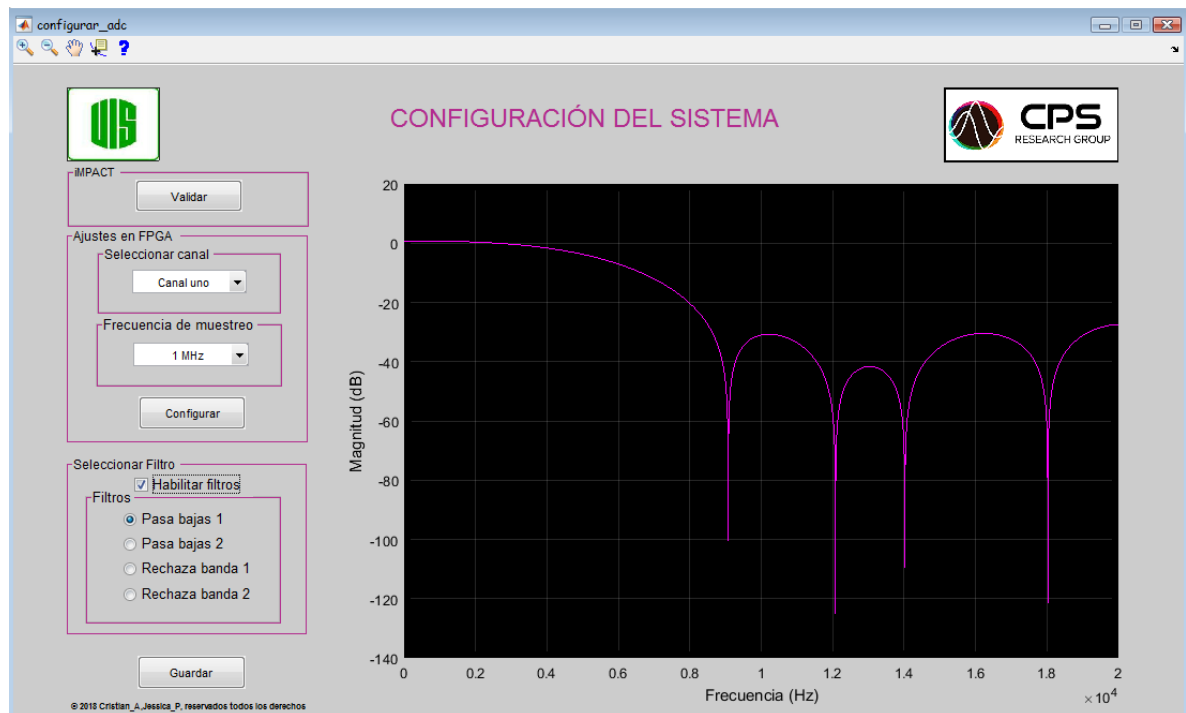
Si la dirección es correcta se habilitarán las opciones de los parámetros de adquisición (canal y frecuencia de muestreo), si no mostrará un error (ver *Figura 25*).

Figura 25. Archivo incorrecto



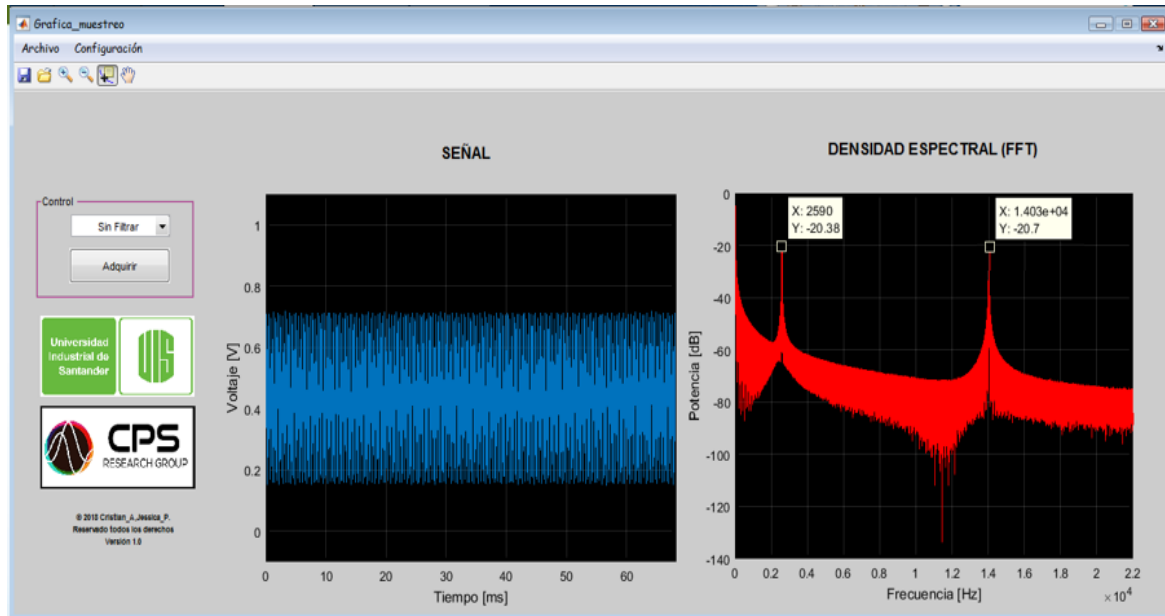
La sección “Ajustes en FPGA” permite seleccionar uno de los cuatro canales y configurar la frecuencia de muestreo, mientras que la sección “Seleccionar Filtro” permite escoger el tipo de filtro. Al elegir uno de los filtros, se podrá visualizar su respuesta en frecuencia en una gráfica tal como se muestra en la *Figura 26* .

Figura 26. Configuración ADC y Filtros prueba



Luego de configurar los parámetros de adquisición y la elección de filtrado, se abre la ventana de adquisición (ver *Figura 27*). Esta cuenta con una lista desplegable que da la opción de adquirir la señal sin filtrar o la señal filtrada. Además de visualizar la señal adquirida, se visualiza su densidad espectral de potencia a través de la transformada rápida de Fourier (FFT). Esta ventana cuenta con la opción de guardar en un archivo .xls las muestras recibidas y también de volverlas a cargar desde archivo (el aplicativo sólo puede abrir archivos en Excel previamente guardados por él).

Figura 27. Ventana grafica señal eléctrica y densidad espectral de potencia



4 RESULTADOS

Para realizar las pruebas en el prototipo se empleó el aplicativo Movil de Android *Multi-tone generator* para generar las señales de audio¹¹. Este aplicativo permite generar varias señales de audio permitiendo la suma entre ellas y modificar su amplitud, tal como se muestra en la *Figura 28*.

Figura 28 Aplicativo para generar señales de audio.

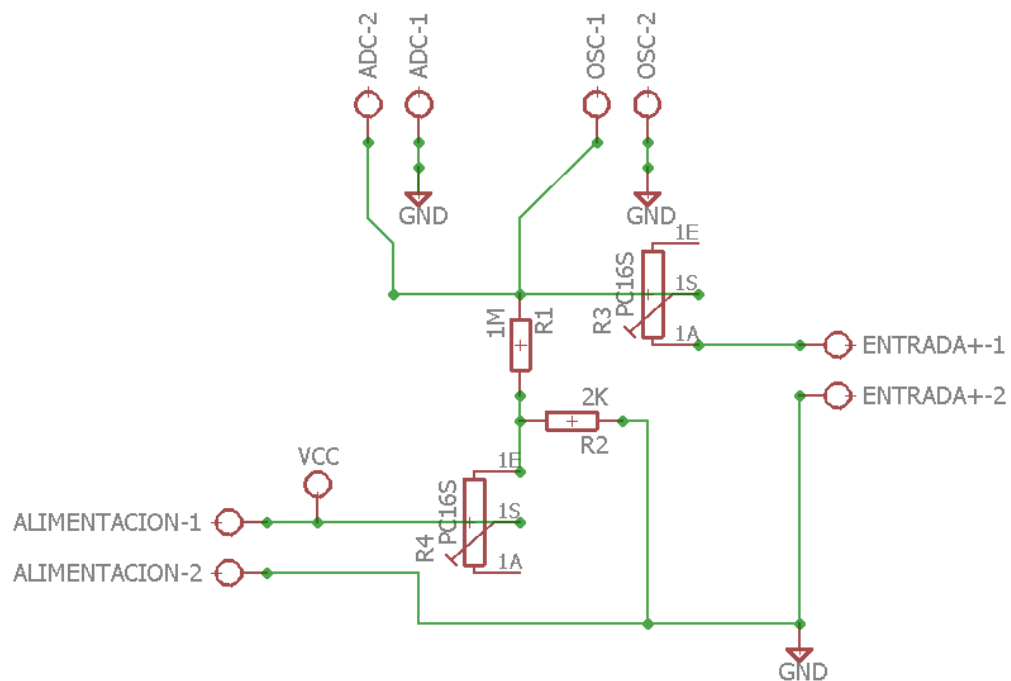


¹¹ MORAIS,J,Multi-tone generator,2017.

Dependiendo del dispositivo empleado será necesario adecuar la señal resultante para evitar que la tarjeta sufra daños ya que el sistema de adquisición está configurado para trabajar con señales de magnitudes de 0 a 1 [V].

La etapa de adecuación está diseñada para elevar el nivel de OFFSET y modificar la amplitud de la señal de audio generada por un dispositivo electrónico. En la *Figura 29* se muestra el diseño de la etapa de adecuación, para cada canal se tiene un circuito idéntico.

Figura 29. Circuito de adecuación de la señal de audio



El circuito dispone de un puerto de entrada para la señal de audio, un puerto de para visualizar la señal en el osciloscopio, un puerto de salida para enviar la señal de audio modificada a la tarjeta y un puerto para energizar el circuito de adecuación.

El usuario puede modificar la amplitud y el nivel de *offset* de la señal de audio a través de los potenciómetros incorporados en la placa de adecuación.

4.1 RESULTADOS EXPERIMENTALES

A continuación, se mostrarán los resultados de las señales de audio adquiridas con frecuencia de muestreo 1MHz, sin filtrar y filtradas y su respectiva gráfica en frecuencia (FFT), además de su comparación con la señal visualizada en osciloscopio.

- Filtro pasa-bajas 1 (PB1):

Figura 30. Señal sin filtrar (2590 [Hz] Y 14030 [Hz])

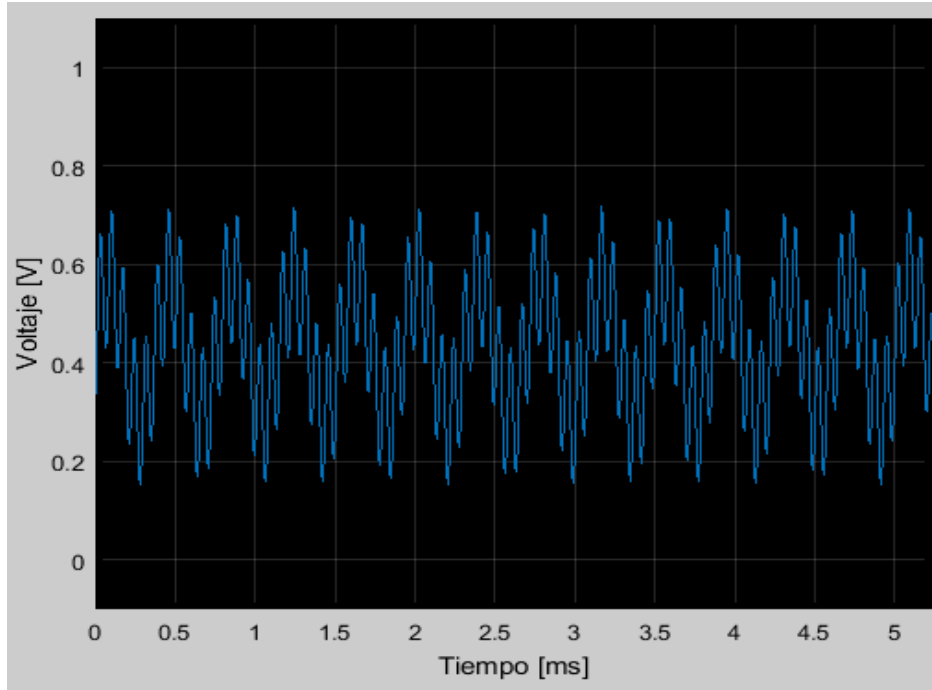


Figura 31. Espectro de la señal de la Figura 30

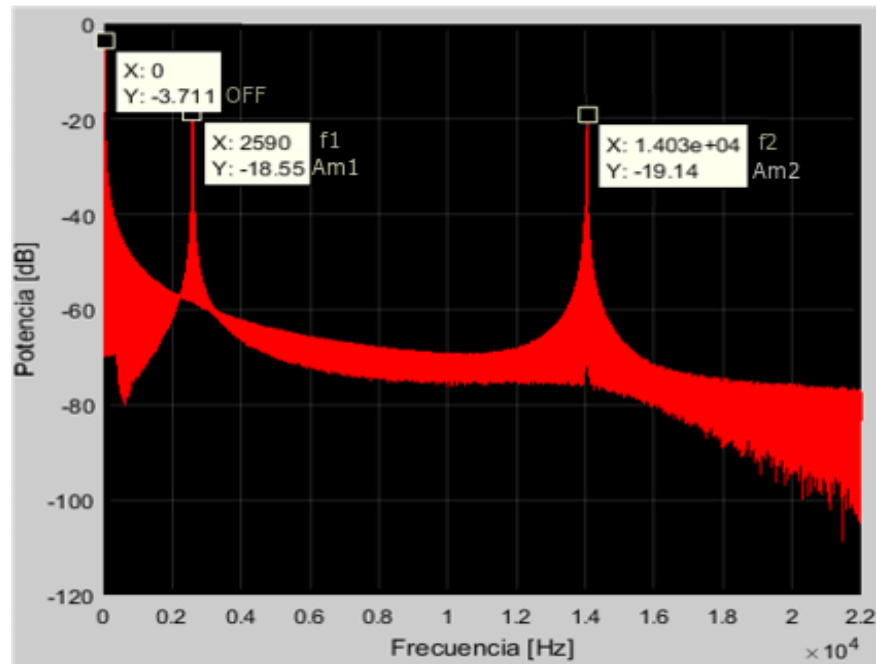


Figura 32. Señal adquirida por el osciloscopio

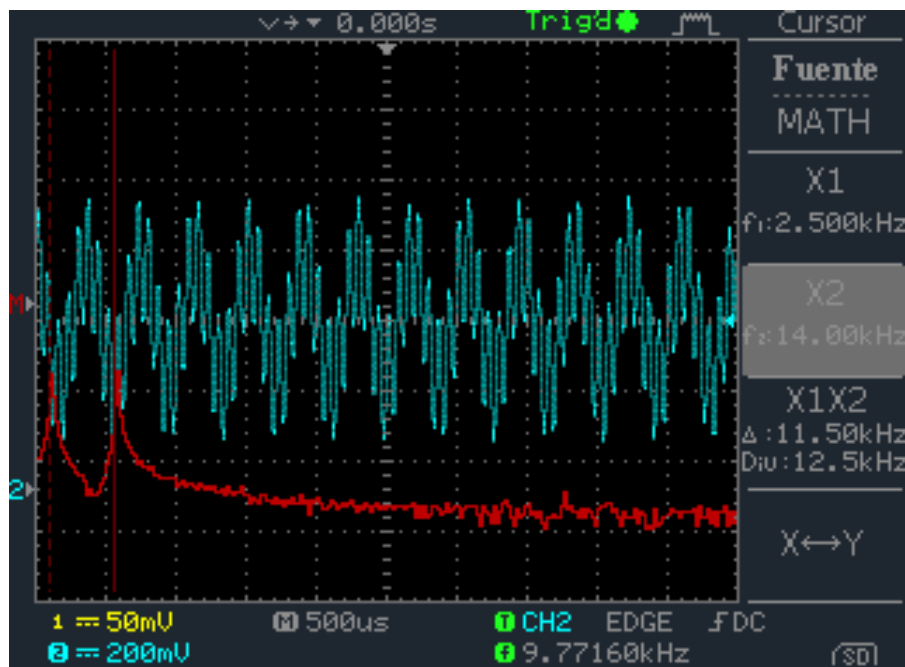


Figura 33. Espectro de la señal adquirida por el osciloscopio

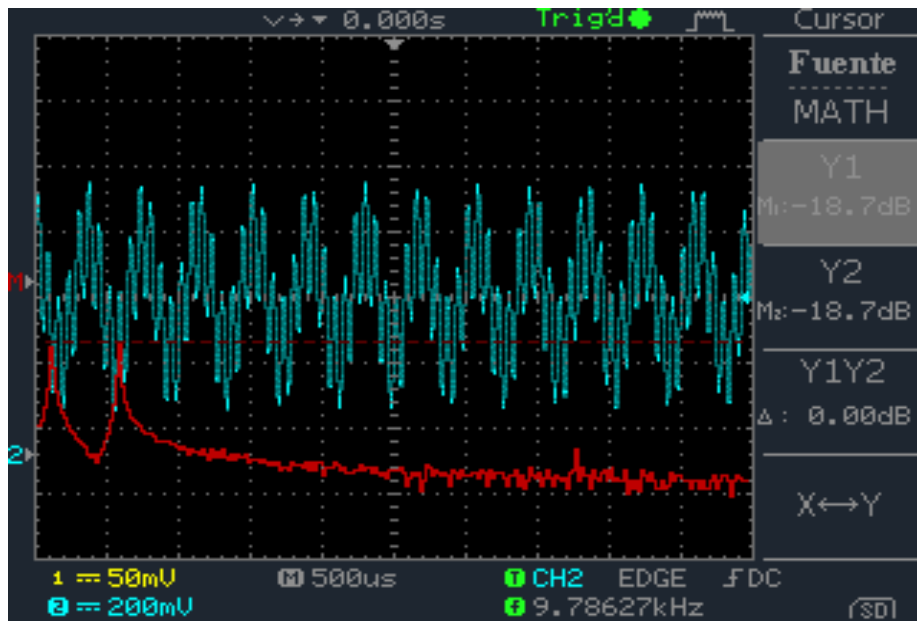


Figura 34. Señal filtrada con filtro pasa bajas 1

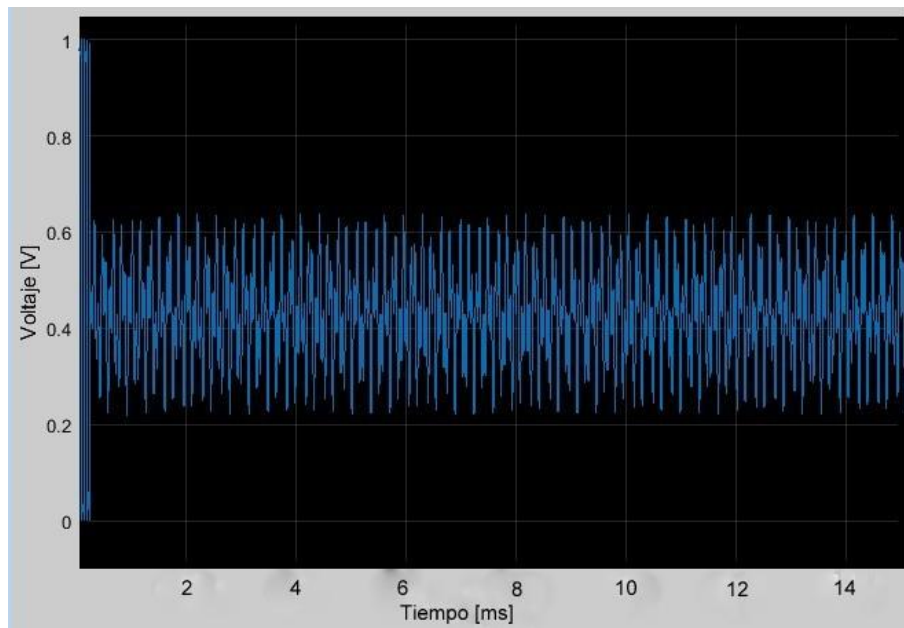
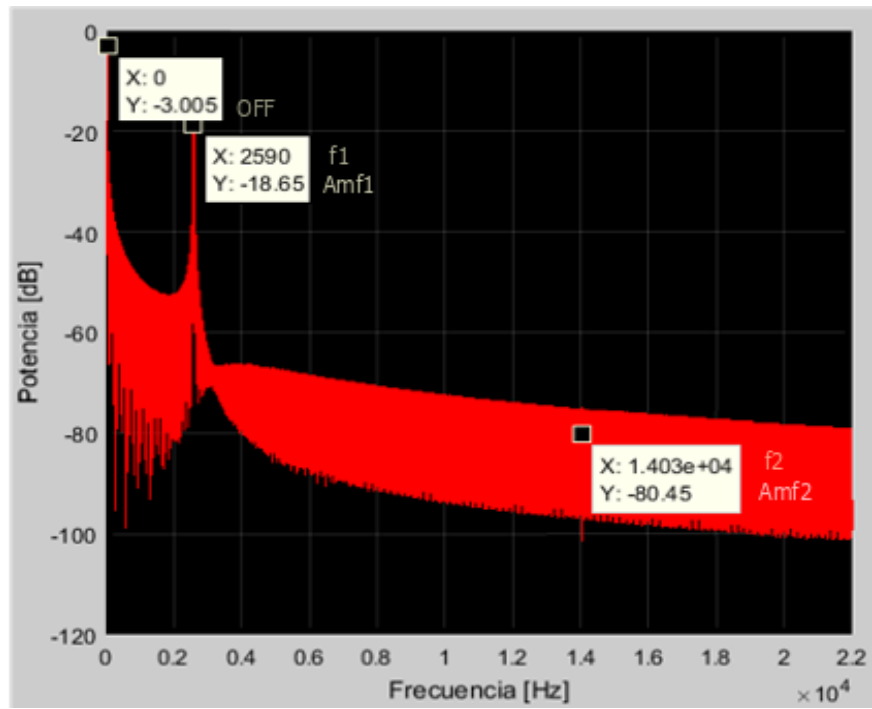


Figura 35. Espectro de la señal filtrada con filtro pasa bajas 1



- Filtro pasa-bajas 2 (PB2)

Figura 36. Señal sin filtrar (2112 [Hz], 10090 [Hz] y 14200 [Hz])

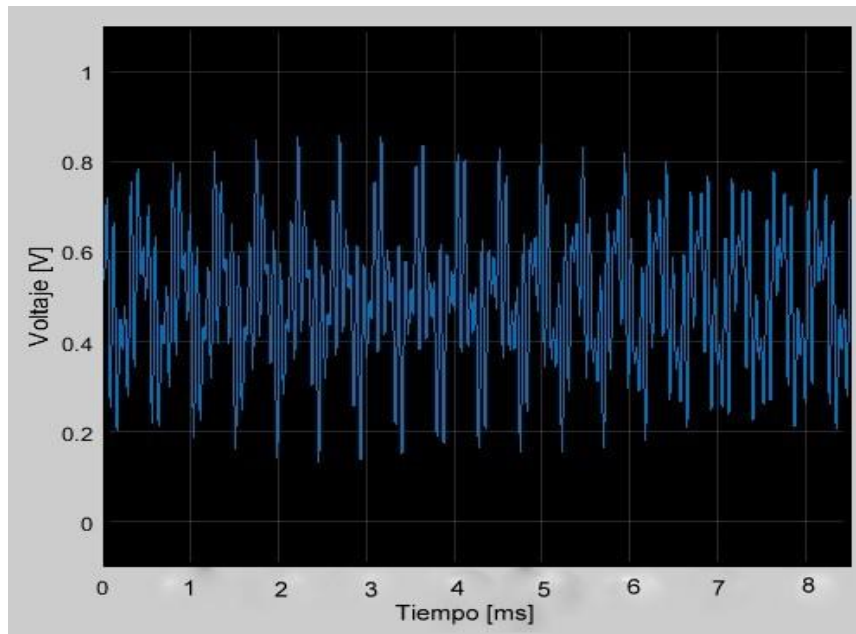


Figura 37. Espectro de la señal de la Figura 36

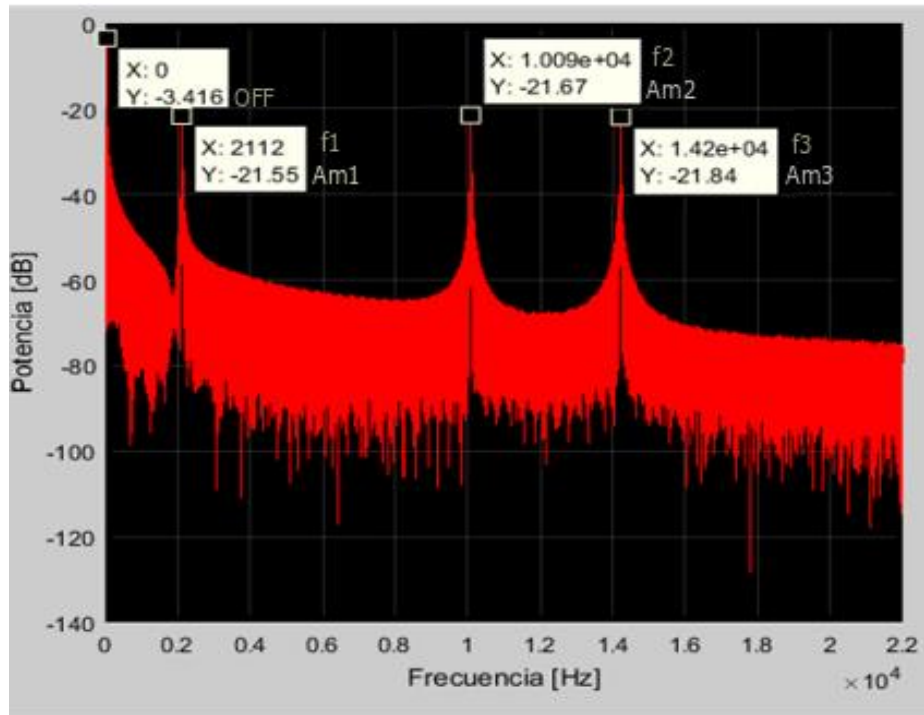


Figura 38. Señal adquirida por el osciloscopio

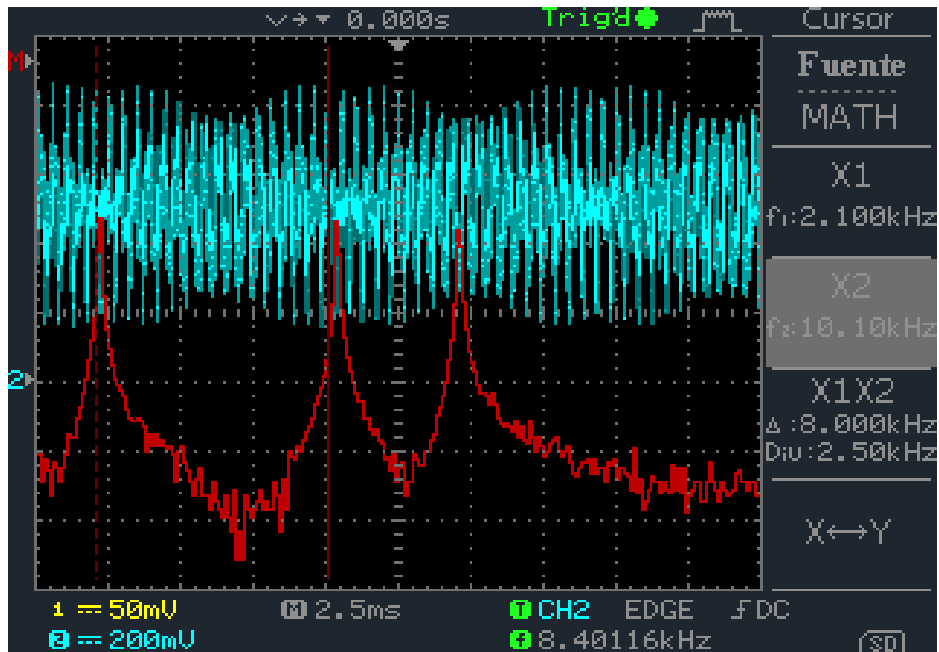


Figura 39. Espectro de la señal adquirida por el osciloscopio

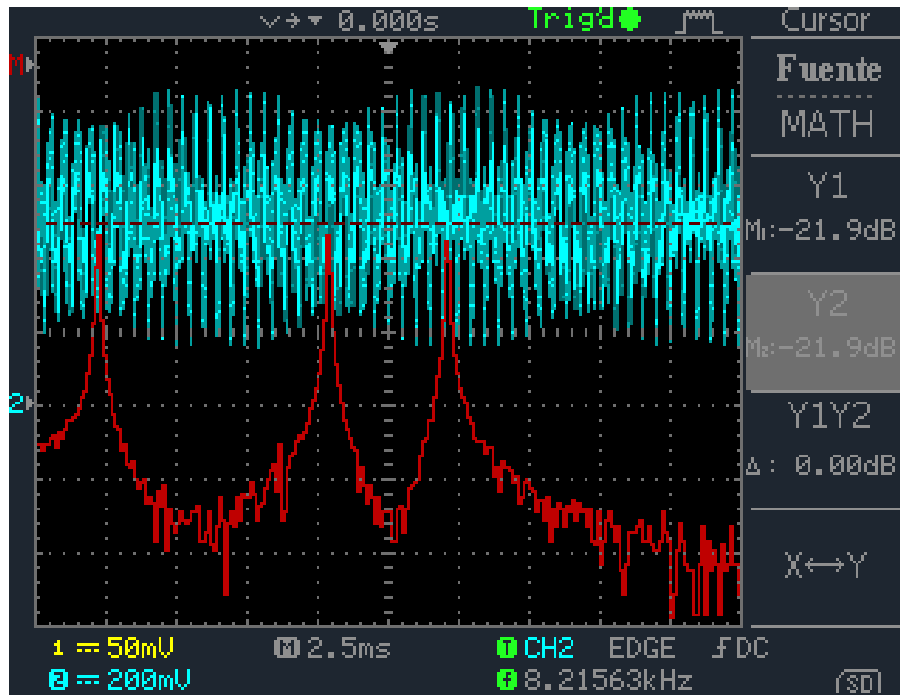


Figura 40. Señal filtrada con filtro pasa bajas 2

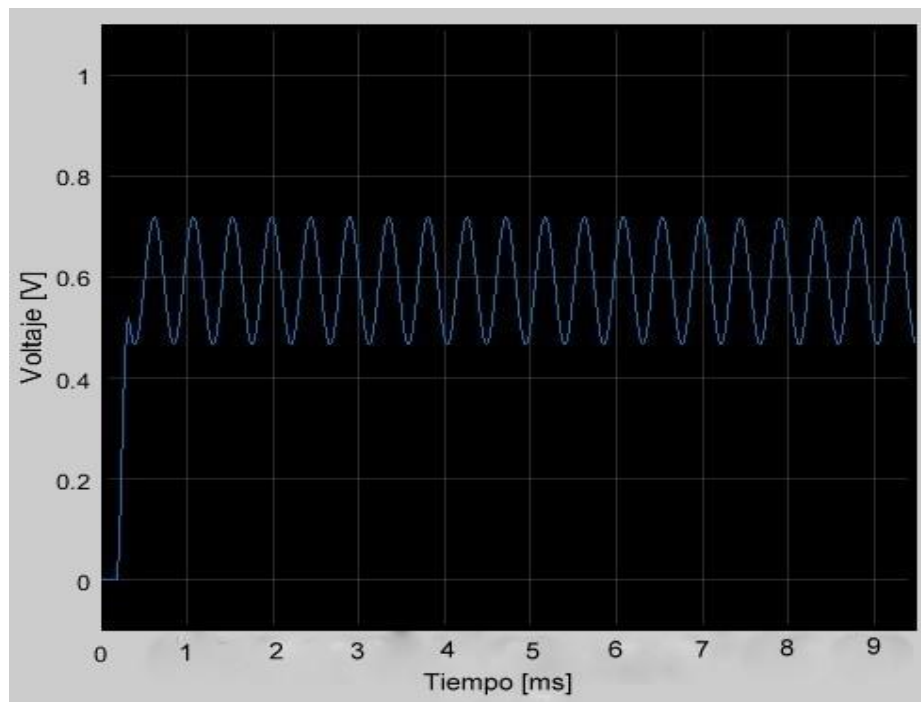
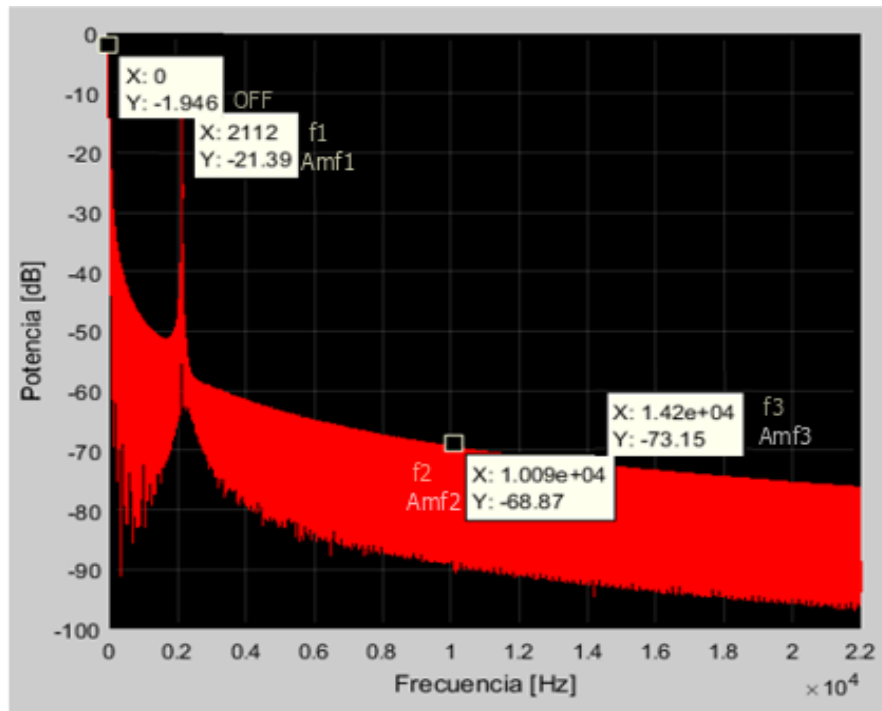


Figura 41. Espectro de la señal filtrada con filtro pasa bajas 2



- Filtro rechaza bandas 1 (RB1)

Figura 42. Señal sin filtrar (5637 [Hz], 9664 [Hz] y 13860 [Hz])

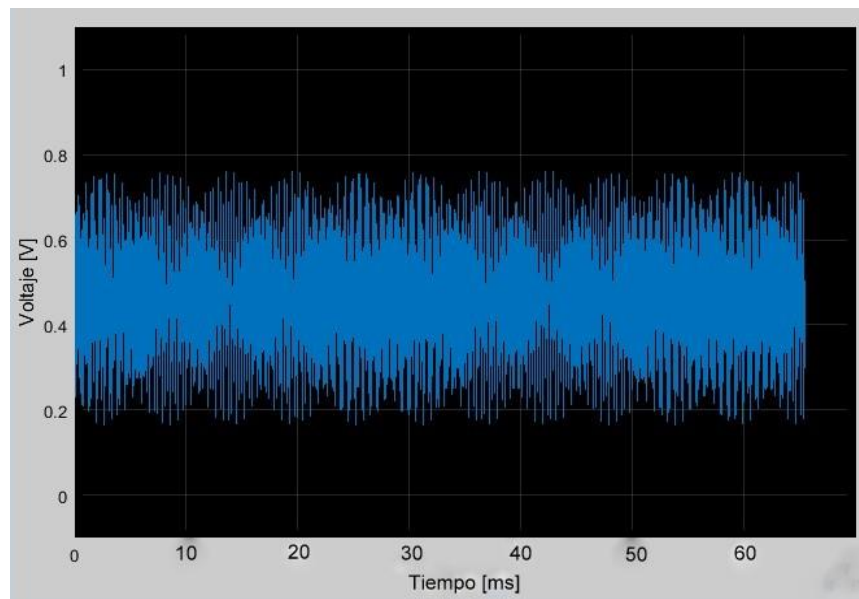


Figura 43. Espectro de la señal de la Figura 42

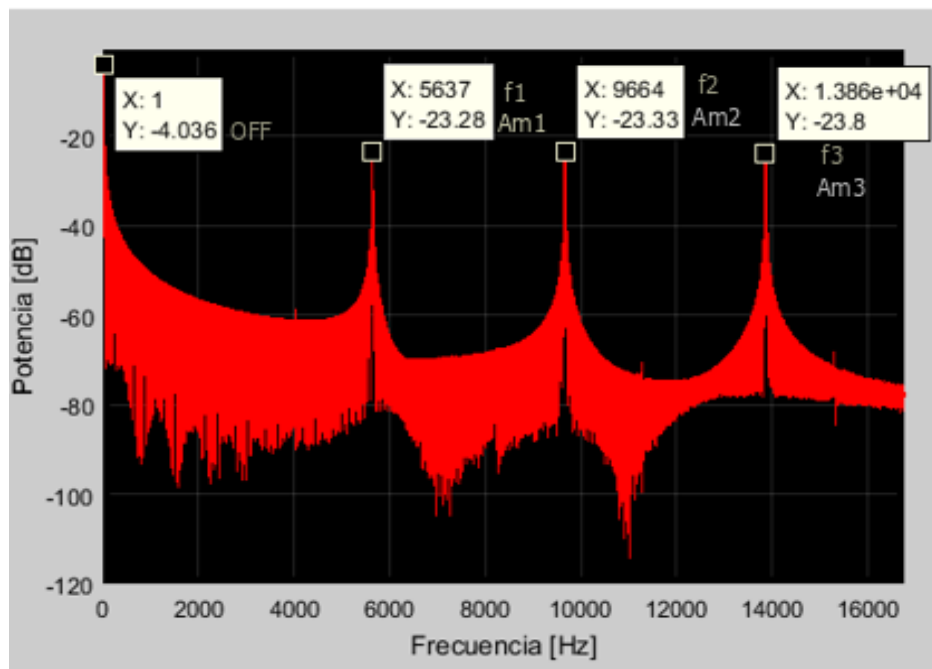


Figura 44. Señal adquirida por el osciloscopio



Figura 45. Espectro de la señal adquirida por el osciloscopio

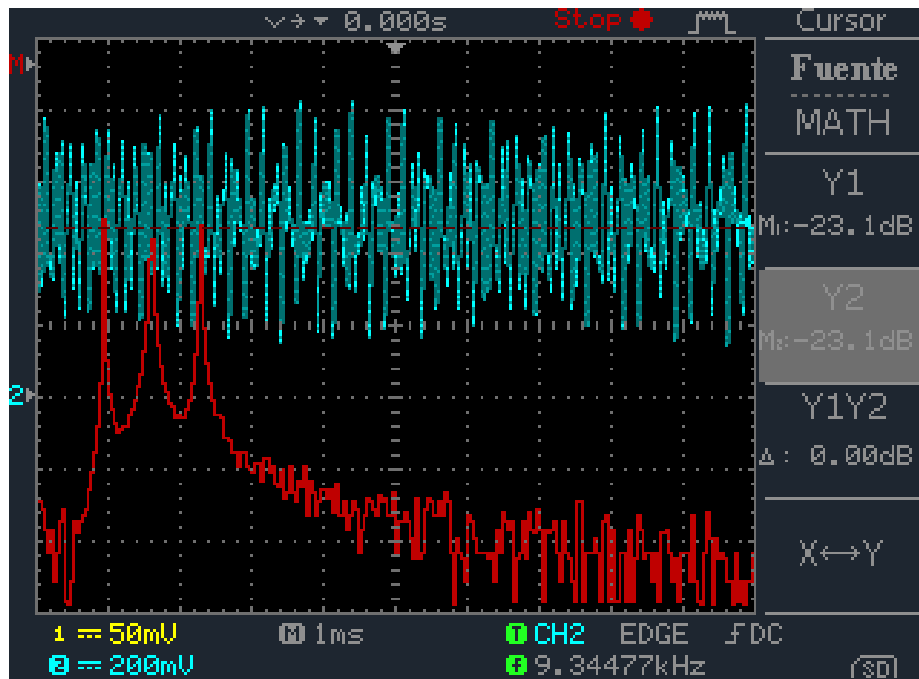


Figura 46. Señal filtrada con filtro rechaza banda 1

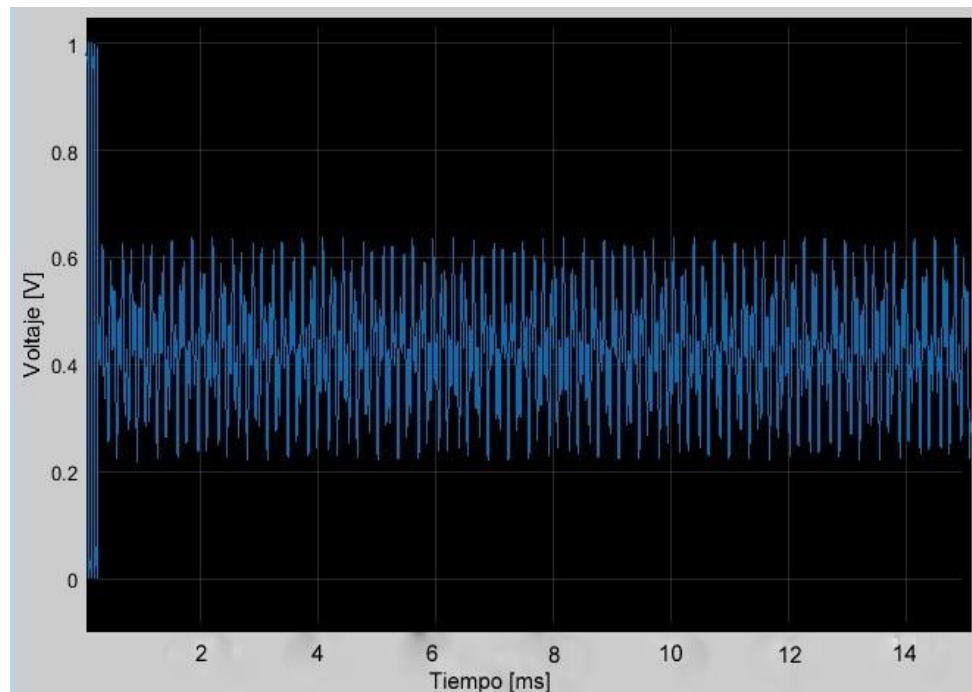
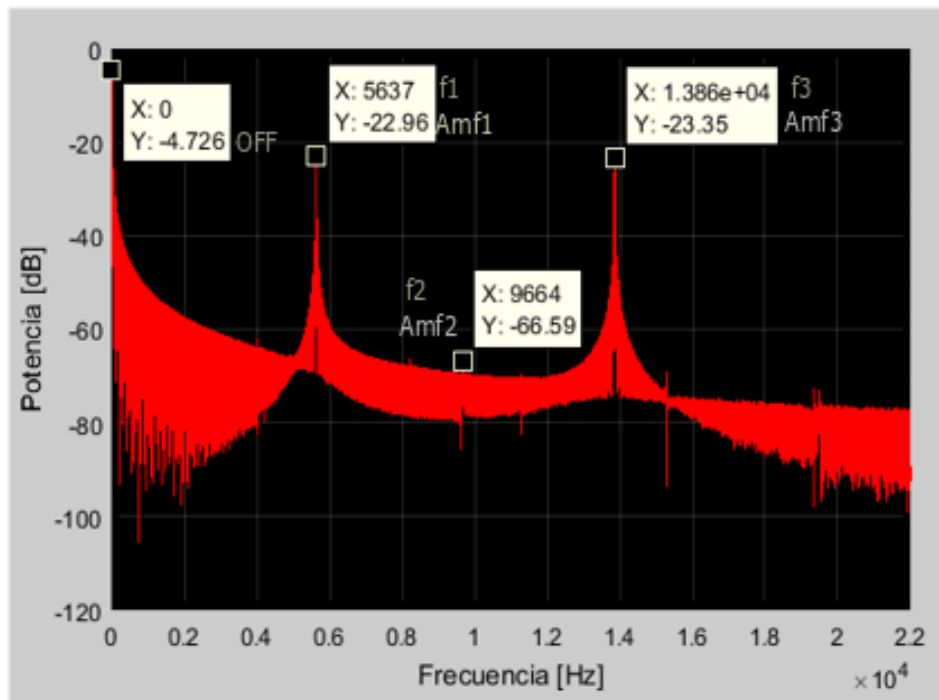


Figura 47. Espectro de la señal filtrada con filtro rechaza banda



- Filtro rechaza bandas 2 (RB2)

Figura 48. Señal sin filtrar (471 [Hz], 8229 [Hz] y 19330 [Hz])

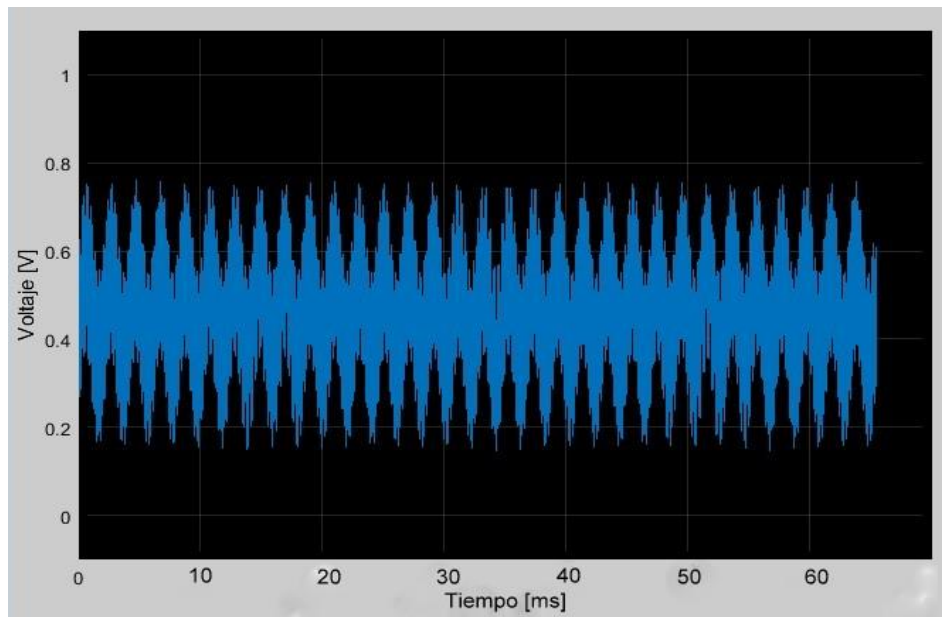


Figura 49. Espectro de la señal de la Figura 48

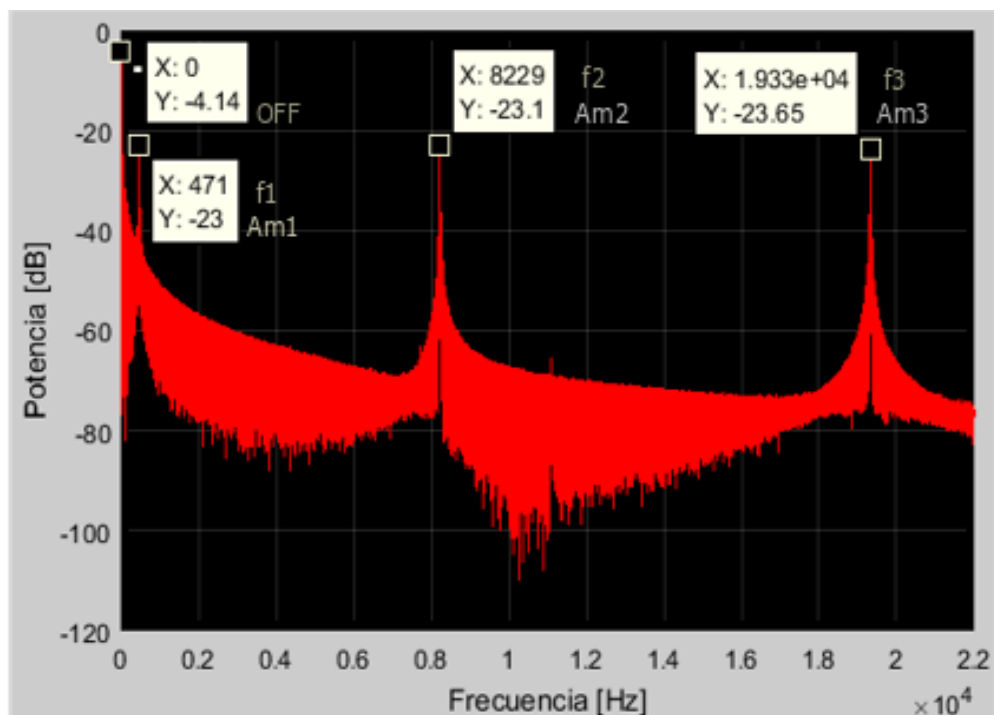


Figura 50. Señal adquirida por el osciloscopio

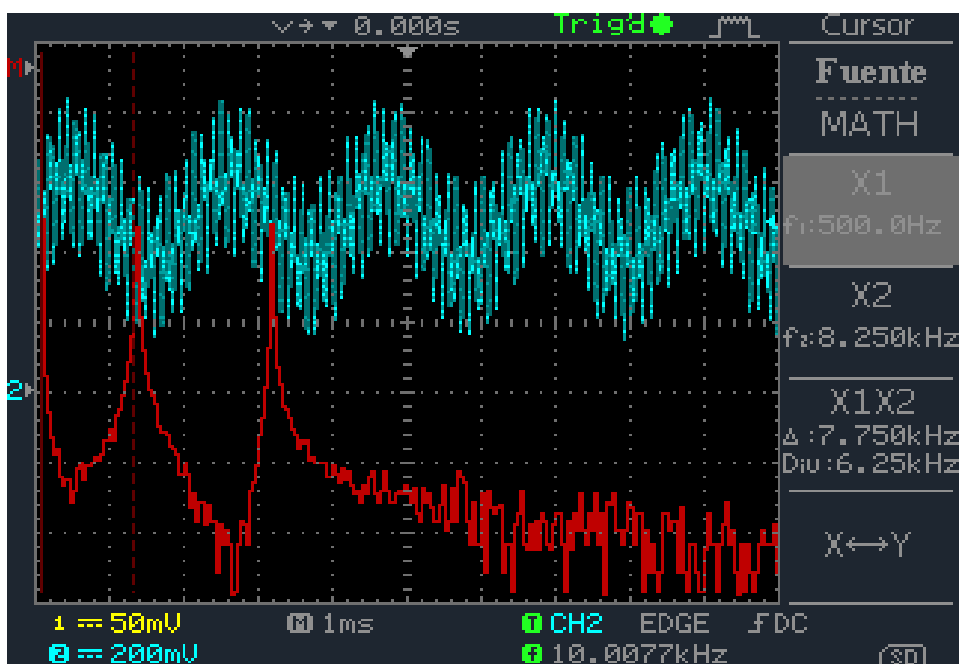


Figura 51. Espectro de la señal adquirida por el osciloscopio

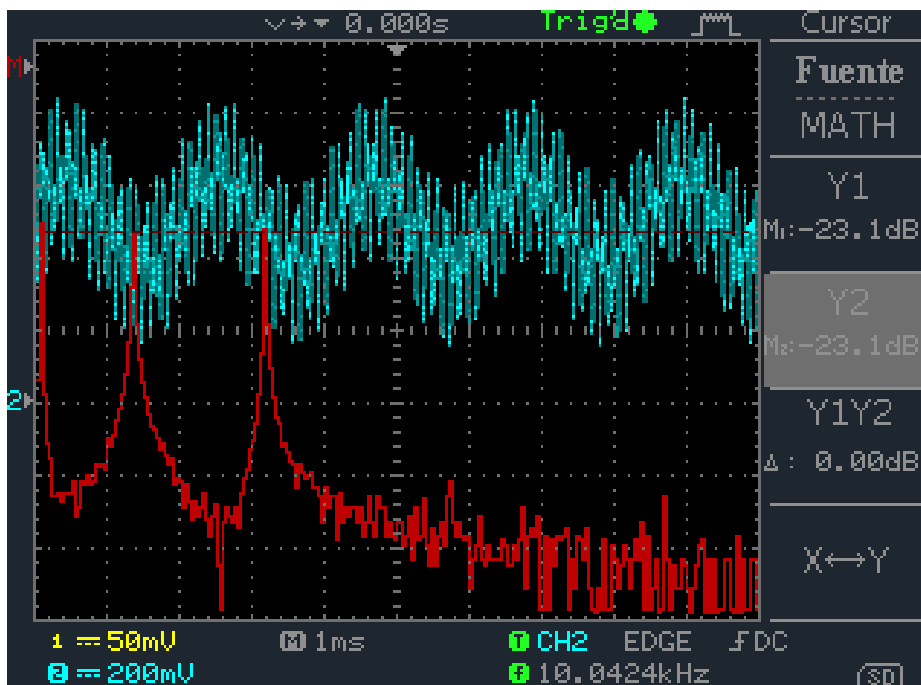


Figura 52. Señal filtrada con filtro rechaza banda 2

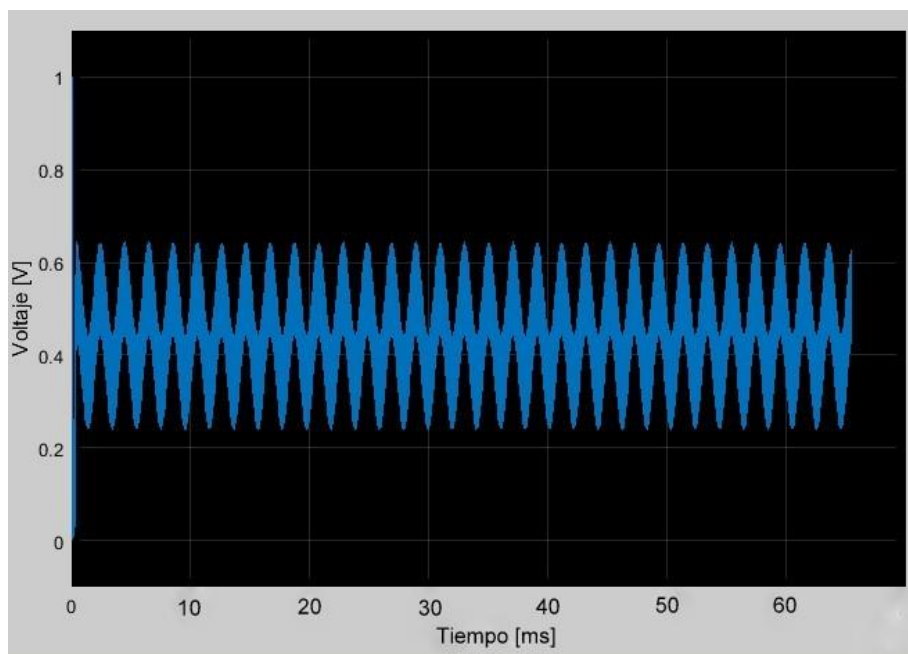
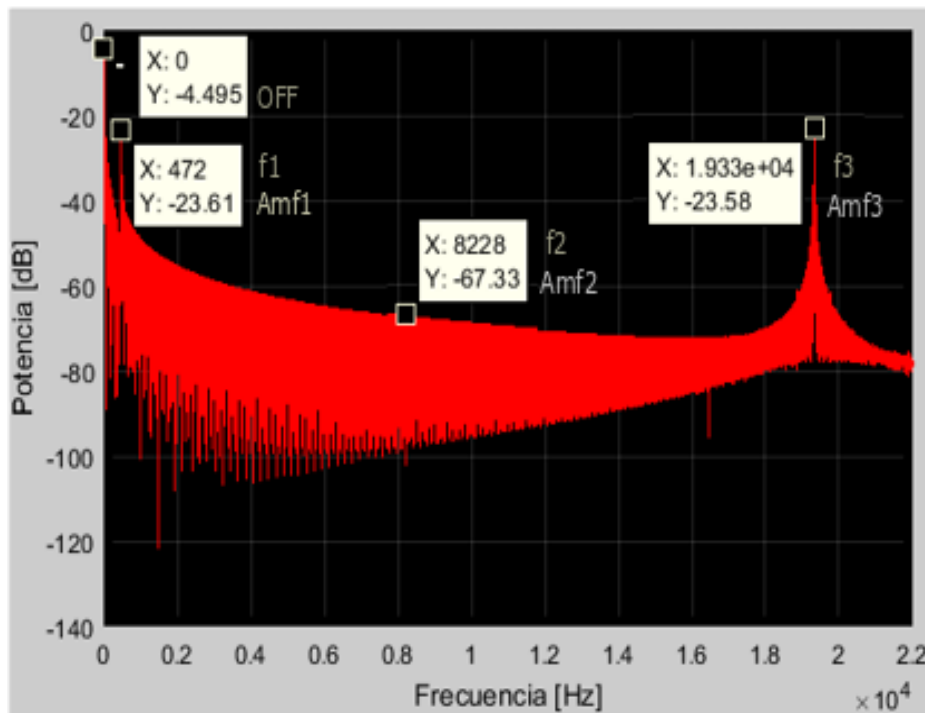


Figura 53. Espectro de la señal filtrada con filtro rechaza banda 2



En la Tabla 5 se resumen los resultados los obtenidos en el aplicativo para cada filtro cuando la frecuencia de muestreo es 1MHz mientras que la Tabla 6 y la Tabla 7 muestran los resultados para las frecuencias de muestreo 500 kHz y 154kHz respectivamente.

Tabla 5. Resumen resultados para $F_s = 1\text{MHz}$

Filtro	Frecuencias [Hz]			Amplitud señal original [dB]				Amplitud señal filtrada [dB]			
	f1	f2	f3	OFF	Am1	Am2	Am3	OFF	Amf1	Amf2	Amf3
PB1	2590	14030	---	-3.71	-18.5	-19.1	---	-3.48	-18.6	-80.4	---
PB2	2112	10090	14200	-3.4	-21.5	-21.6	-21.8	-1.9	-21.3	-68.8	-73.1
RB1	5637	9664	13860	-4.03	-23.2	-23.3	-23.8	-4.72	-22.9	-66.5	-23.3
RB2	471	8229	19330	-4.14	-23.0	-23.1	-23.6	-4.49	-23.6	-67.3	-23.5

PB1, PB2: Filtros pasa bandas 1 y 2 respectivamente.

RB1, RB2: Filtros rechaza bandas 1 y 2 respectivamente.

f1, f2, f3: Frecuencia en Hz de las señales generadas con *Multi-tone generator*.

Am1, Am2, Am3: Amplitud en dB de la respuesta en frecuencia de la señal sin filtrar.

Amf1, Amf2, Amf3: Amplitud en dB de la respuesta en frecuencia de la señal filtrada.

OFF: valor de la componente en frecuencia 0 (OFFSET).

Tabla 6. Resumen resultados para Fs= 500kHz

Filtro	Frecuencias [Hz]			Amplitud señal original [dB]				Amplitud señal filtrada [dB]			
	f1	f2	f3	OFF	Am1	Am2	Am3	OFF	Amf1	Amf2	Amf3
PB1	1308	14070	---	-4.05	-19.5	-19.8	---	-4.63	-19.6	-78.8	---
PB2	7231	18520	---	-3.89	-19.5	-19.9	---	-4.20	-19.8	-79.5	---
RB1	3856	9612	16360	-2.75	-21.0	-21.2	-21.5	-3.10	-21.1	-79.1	-21.4
RB2	12090	14130	19460	-3.48	-20.2	-20.5	-20.5	-4.34	-20.5	-77.6	-20.7

Tabla 7. Resumen resultados para Fs= 154kHz

Filtro	Frecuencias [Hz]			Amplitud señal original [dB]				Amplitud señal filtrada [dB]			
	f1	f2	f3	OFF	Am1	Am2	Am3	OFF	Amf1	Amf2	Amf3
PB1	1772	8240	---	-2.96	-16.5	-16.9	---	-3.47	-16.6	-81.9	---
PB2	4555	10820	---	-3.82	-18.6	-19.1	---	-3.89	-18.76	-86.55	---
RB1	1520	8491	11020	-3.50	-20.1	-20.5	-20.7	-3.63	-20.2	-81.9	-20.9
RB2	555	2686	6960	-3.69	-20.1	-20.1	-20.3	-4.16	-20.1	-73.5	-20.4

4.2 ANÁLISIS DE RESULTADOS

- Los resultados obtenidos validan que las señales de audio adquiridas por el aplicativo son similares a las mostradas por el osciloscopio en cuanto a frecuencia y potencia [dB].

- Los valores de frecuencia y amplitud de las señales adquiridas por el FPGA y los adquiridos por el osciloscopio en algunas ocasiones no coinciden. Esto se debe ya sea por la resolución del osciloscopio o por distorsión de la señal. La distorsión de la señal ocurre debido a la influencia de los elementos electrónicos que modifican la señal a adquirir.
- Al hacer un acercamiento a las gráficas de las señales en el modo de filtrado se visualiza que en los primeros 500 datos hay una diferencia grande de amplitud y forma en comparación con los siguientes datos. Esto ocurre ya que inicialmente, mientras ingresan más muestras, el filtro está procesando la señal de entrada con valores establecidos en cero.
- Se encontró que la frecuencia de las señales a adquirir debe ser al menos 10 veces menor que la frecuencia de muestreo para que puedan ser observadas adecuadamente.
 - Para $F_s = 961.54 \text{ kHz} \rightarrow F_{max} < 96.15 \text{ kHz}$.
 - Para $F_s = 480.77 \text{ kHz} \rightarrow F_{max} < 48.07 \text{ kHz}$.
 - Para $F_s = 153.85 \text{ kHz} \rightarrow F_{max} < 15.85 \text{ kHz}$.

5 CONCLUSIONES

En este proyecto se implementó un sistema de adquisición de señales en FPGA y un aplicativo computacional, con características tales como: 4 canales independientes para toma de datos, filtros digitales predeterminados, visualización de la densidad espectral de potencia de la señal adquirida, frecuencia de muestreo variable, entre otros. Teniendo en cuenta los resultados obtenidos, podemos decir que es viable construir un prototipo a escala que permita modificar los parámetros de adquisición por medio de un FPGA. Basado en este prototipo, se podrá diseñar e implementar un sistema fiable en un dispositivo con mayores prestaciones.

La frecuencia de muestreo por canal cambia con cada modo de operación del XADC, por ello el modo canal único presenta una frecuencia mayor por canal en comparación con los otros modos de operación, ya que los otros modos actúan de forma secuencial en los 4 canales, provocando que la frecuencia de muestreo por canal del XADC se divida en el número de canales.

De acuerdo a la cantidad de bits con que se representan los datos, el puerto de comunicación, medio de transmisión y el software implementado se generan retardos en el envío de la información, provocando la no visualización de los datos en tiempo real. El sistema de comunicación presenta un retraso de aproximadamente dos minutos. Se sugiere que para trabajos futuros se aumente la velocidad de transmisión y recepción de acuerdo a la velocidad máxima permitida por anteriores aspectos. Se sugiere implementar un bit de paridad en el protocolo de comunicación, ya que en este proyecto no se utilizó, esto avisará al sistema que hay una posible pérdida de información.

Los datos generados por el circuito de *Filter Design HDL CODER* para los filtros digitales implementados en FPGA, no son exactamente iguales a los datos de filtrado generados en Matlab. Esto se debe a que la herramienta *HDL Coder* busca utilizar menos recursos del FPGA modificando algunas operaciones.

La cantidad de bits con la que se representan los coeficientes del filtro cambia la respuesta en frecuencia de éste. Por lo tanto, si se desea mejorar dicha respuesta, es necesario aumentar dicha cantidad de bits, pero esto implicaría que el filtro necesite mayores recursos en hardware.

Los filtros diseñados en este proyecto son predeterminados, sin embargo, para trabajos futuros, se puede implementar una plantilla donde el usuario pueda ingresar los coeficientes para que éste modifique las características del filtro de acuerdo a su necesidad.

La sincronización de todos los componentes del sistema es vital para el funcionamiento esperado del mismo, ya que varios de estos elementos actúan a frecuencias de reloj distintas y algunos tiene un tiempo de respuesta más largos que otros, lo que generaría que los datos esperados no fueran enviados al aplicativo computacional.

En este sistema no se puede visualizar la señal filtrada y la señal sin filtrar al mismo tiempo, ya que sólo se implementó una memoria para almacenar una de las señales, es por esto que se recomienda para futuros trabajos que se implemente dos memorias para almacenar cada señal por aparte, teniendo en cuenta las señales de sincronización para que sean guardadas en un tiempo determinado.

Las señales de audio se pueden visualizar correctamente en este aplicativo, sin embargo, se recomienda que la frecuencia de muestreo sea 10 veces mayor que la frecuencia de la señal para poderla visualizar correctamente.

BIBLIOGRAFÍA

CHI-JUI C. ET al," FPGA implementation of Digital Filters", Proceedings of the international conference on signal processing applications and technology, 1993.

DANIEL, RUBEN Y FELIPE. Osciloscopio Digital Mediante FPGA [en línea]. [Citado el 17 Noviembre de 2017], Disponible en internet: <http://espacio.uned.es/fez/eserv/taee:congreso-2004-1106/SD106.pdf>

DIGILENT. Nexys4 DDR FPGA Board Reference Manual [en línea]. [Citado el 20 Noviembre de 2017], Disponible en internet: https://reference.digilentinc.com/_media/nexys4-ddr:nexys4ddr_rm.pdf

EVANS J.," Efficient FIR Filter Architectures Suitable for FPGA Implementation ", IEEE transactions. On circuits and systems, July 1994.

GONZÁLEZ G. et al, "Diseño e implementación de un filtro FIR con coeficientes programables y fase lineal en FPGA ", trabajo de grado, Universidad Distrital Francisco José de Caldas, Santafé de Bogotá, junio de 1998

LORENA, BÁRBARO. Osciloscopio Digital básico sobre FPGA [en línea]. [Citado el 20 de diciembre de 2017], Disponible en internet: https://www.researchgate.net/publication/274371152_Osciloscopio_Digital_basico_sobre_FPGA

MATHWORKS. (s.f.). *Filter Designer*. (Matlab) [En línea]. [Citado el 21 de enero de 2018], Disponible en internet: <https://la.mathworks.com/help/signal/ug/getting-started-with-filter-designer.html?requestedDomain=true>

MELGAREJO M. et al, " Filtro FIR adaptativo sobre celdas lógicas programables FPGA ", memorias del Primer congreso internacional de ingenierías eléctrica y electrónica, Universidad Industrial de Santander, Bucaramanga, marzo de 2000.

PROAKIS J. et al, "Tratamiento digital de señales", tercera edición, Madrid, PRENTICE HALL, 1998, pág. 509-520.

TEXAS INSTRUMENTS. 12-Bit, Dual 1.8 GSPS or Single 3.6 GSPS A/D Converter Reference Board [en línea]. [Citado el 18 de Noviembre de 2017] Disponible en internet: <http://www.ti.com/tool/ADC12D1800RFRB>

TEXAS INSTRUMENTS. WaveVision 5 Data Capture Board Version 5.1 [en línea]. [Citado el 18 de Noviembre de 2017] Disponible en internet <http://www.ti.com/tool/WAVEVSN-BRD-5.1>

XILINX. 7 Series FPGAs Data Sheet: Overview [en línea]. [Citado el 2 de Diciembre de 2017], Disponible en internet: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

XILINX. LogiCore IP XADC Wizard v2.1 [en línea]. [Citado el 2 de Noviembre de 2017], Disponible en internet: https://www.xilinx.com/support/documentation/ip_documentation/xadc_wiz/v2_1/ug772_xadc_wiz.pdf

XILINX. XADC Dual 12-Bit 1MSPS ANALOG-to- DIGITAL CONVERTER [en línea]. [Citado el 16 de diciembre de 2017], Disponible en internet: https://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf