UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:
INGENIERO ELECTRÓNICO

TEMA:
DISEÑO E IMPLEMENTACIÓN DE FILTROS DIGITALES FIR E IIR UTILIZANDO EL MICROCONTROLADOR XMEGA DE ATMEL PARA EL TRATAMIENTO DE SEÑALES DE AUDIO

AUTOR:
DIEGO ANDRÉS JARRÍN OCHOA

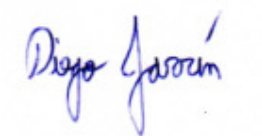
TUTOR:
LUIS GERMÁN OÑATE CADENA

Quito, junio del 2016

CESIÓN DE DERECHOS DE AUTOR

Yo Diego Andrés Jarrín Ochoa, con documento de identificación N° 1717547150, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: Diseño e implementación de filtros digitales FIR e IIR utilizando el microcontrolador XMEGA de Atmel para el tratamiento de señales de audio, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Diego Andrés Jarrín Ochoa
1717547150
Quito, junio del 2016

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, Diseño e implementación de filtros digitales FIR e IIR utilizando el microcontrolador XMEGA de Atmel para el tratamiento de señales de audio realizado por Diego Andrés Jarrín Ochoa, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, junio del 2016

Luis Germán Oñate Cadena

1712157401

Dedicatoria

Dedico mi trabajo de titulación a los jóvenes estudiantes de las carreras de Ingeniería, que no se rinden, que siguen adelante y no se dan por vencidos, que aprenden de sus errores y comparten sus éxitos y conocimientos con los demás.

A mi amada familia, en agradecimiento al tiempo, paciencia, amor y comprensión durante toda mi vida.

Agradecimiento

Agradezco a mis profesores y a mis compañeros de trabajo por su paciencia, confianza, ayuda, colaboración, conocimientos y consejos.

Al Ing. Luís Oñate por haberme enseñado el mundo del procesamiento digital de señales y por permitirme realizar este proyecto.

Al Ing. Luís Fernando Casanova por su gran dedicación por la docencia y por llevar adelante el laboratorio de Fabricación, espacio de trabajo donde realicé este y muchos más proyectos y donde tuve la oportunidad de conseguir mi primer empleo profesional.

Índice

Introducción.....1

Capítulo 1 3

Planteamiento del problema 3

1.1 Problema a resolver 3

1.2 Objetivos 3

1.2.1 Objetivo general 3

1.2.2 Objetivos específicos 3

1.3 Justificación.....4

1.4 Alcances 4

1.5 Metodología 5

1.5.1 Investigación preliminar 5

1.5.2 Desarrollo de hardware y software 5

1.5.3 Construcción final 5

1.5.4 Documentación.....6

Capítulo 2 7

Fundamentación teórica 7

2.1 Microcontrolador XMEGA de Atmel.....7

2.1.1 Tarjeta de desarrollo XMEGA-Ready de MikroElektronika 8

2.1.2 Convertidor análogo a digital 9

2.1.3 Convertidor digital a analógico 10

2.2 Procesamiento digital de señales 11

2.3 Filtros digitales 12

2.3.1 Filtros digitales FIR 13

2.3.2 Filtros digitales IIR 14

2.3.3 Respuesta en frecuencia 16
2.3.4 Escalamiento de los coeficientes de los filtros digitales 17
Capítulo 3 18
Diseño e implementación 18
3.1 Diseño de la tarjeta de filtros antialiasing, de reconstrucción y acondicionamiento de las señales a procesar 18
3.1.1 Filtro antialiasing y de reconstrucción 18
3.1.2 Acondicionamiento de la señal 18
3.2 Máxima frecuencia de adquisición y reconstrucción 19
3.3 Códigos en MATLAB y Mikro C para AVR para diseñar e implementar los filtros digitales FIR e IIR 19
3.3.1 Diseño de los filtros digitales en MATLAB 19
3.3.2 Códigos en Mikro C para AVR 21
Capítulo 4 25
Análisis de resultados 25
4.1 Implementación de la tarjeta de filtros antialiasing, reconstrucción y acondicionamiento de las señales 25
4.2 Determinación de la frecuencia máxima de adquisición y reconstrucción 26
4.2.1 Frecuencia de muestreo en los filtros digitales FIR e IIR y número de coeficientes 27
4.3 Resultados obtenidos con el filtro antialiasing y el filtro de reconstrucción 29
4.4 Respuesta en magnitud y fase de los filtros digitales 31
4.4.1 Filtro digital pasa-bajos FIR 31
4.4.2 Filtro digital pasa-altos FIR 33

4.4.3 Filtro digital pasa-banda FIR 35

4.4.4 Filtro digital rechaza-banda FIR.....37

4.4.5 Filtro digital pasa-bajos IIR 39

4.4.6 Filtro digital pasa-altos IIR 41

4.4.7 Filtro digital pasa-banda IIR 43

4.4.8 Filtro digital rechaza-banda IIR 45

Conclusiones 48

Recomendaciones 51

Referencias 52

Anexos 54

Lista de tablas

Tabla 1. Coeficientes de un filtro digital FIR por el método de la transformada de Fourier.....14
Tabla 2. Función de transferencia de filtros analógicos de segundo orden 15
Tabla 3. Coeficientes de los filtros digitales IIR de segundo orden.....16
Tabla 4. Respuesta en frecuencia de los filtros digitales 17
Tabla 5. Valores experimentales y teóricos para el filtro digital FIR pasa-bajos 32
Tabla 6. Valores experimentales y teóricos para el filtro digital FIR pasa-altos 34
Tabla 7. Valores experimentales y teóricos para el filtro digital FIR pasa-banda 36
Tabla 8. Valores experimentales y teóricos para el filtro digital FIR rechaza-banda.38
Tabla 9. Valores experimentales y teóricos para el filtro digital IIR pasa-bajos 40
Tabla 10. Valores experimentales y teóricos para el filtro digital IIR pasa-altos 42
Tabla 11. Valores experimentales y teóricos para el filtro digital IIR pasa-banda 44
Tabla 12. Valores experimentales y teóricos para el filtro digital IIR recha-zabanda .. 46

Lista de figuras

Figura 1. Microcontrolador XMEGA respecto a otros AVRs de 8 bits 7
Figura 2. Almacenamiento de un registro de 12 bits en 2 registros de 8 bits. 8
Figura 3. Fotografía de la tarjeta de desarrollo XMEGA-Ready. 8
Figura 4. Configuración del ADC en el AVR XMEGA.....9
Figura 5. Configuración del DAC en el AVR XMEGA.....10
Figura 6. Diagrama de bloques de un sistema DSP.....11
Figura 7. Filtro Sallen Key de ganancia unitaria.....12
Figura 8. Diagrama de bloques para los filtros digitales 13
Figura 9. Acondicionador de la señal de audio. 19
Figura 10. Diagrama de flujo del diseño de los filtros digitales en MATLAB.....20
Figura 11. Diagrama de flujo del programa utilizado para DSP con la XMEGA.....21
Figura 12. Diagrama de flujo para filtros digitales FIR.....22
Figura 13. Diagrama de flujo para filtros digitales IIR.....23
Figura 14. PCB de los filtros antialiasing y de reconstrucción. 25
Figura 15. Adquisición y reconstrucción de una señal analógica cuadrada de 100 kHz.....26
Figura 16. Adquisición y reconstrucción de una señal analógica sinusoidal de 50 kHz.....27
Figura 17. Máxima frecuencia de adquisición y reconstrucción para los filtros IIR..28
Figura 18. Máxima frecuencia de adquisición y reconstrucción para los filtros IIR..29
Figura 19. Ejemplo de aliasing mostrado en el osciloscopio. 30
Figura 20. Suavizado de una señal utilizando el filtro de reconstrucción.....30
Figura 21. Respuesta de frecuencia para filtro digital FIR pasa-bajos. 33
Figura 22. Respuesta de frecuencia para filtro digital FIR pasa-altos. 35
Figura 23. Respuesta de frecuencia para filtro digital FIR pasa-banda. 37
Figura 24. Respuesta de frecuencia para filtro digital FIR rechaza-banda. 39
Figura 25. Respuesta de frecuencia para filtro digital IIR pasa-bajos. 41
Figura 26. Respuesta de frecuencia para filtro digital IIR pasa-altos. 43
Figura 27. Respuesta de frecuencia para filtro digital IIR pasa-banda. 45
Figura 28. Respuesta de frecuencia para filtro digital IIR rechaza-banda. 47

Lista de anexos

Anexo 1. Esquemático y layout de la PCB para filtrado y acondicionamiento de las señales de audio 54
Anexo 2. Códigos de los programas en MATLAB y MikroC para AVR.....56
Anexo 3. Coeficientes calculados para los filtros digitales FIR e IIR utilizando MATLAB 64
Anexo 4. Documentación fotográfica del comportamiento de los filtros digitales FIR e IIR 65

Resumen

Las aplicaciones de los filtros digitales son muy importantes para la carrera de ingeniería electrónica, porque son utilizados para procesamiento digital de señales de audio, video e imágenes, telecomunicaciones digitales, análisis de señales biomédicas y en equipos de laboratorio para eliminar armónicos de la red eléctrica. El diseño de un filtro analógico requiere rediseñar su estructura si los parámetros del filtro cambian, el diseño de un filtro digital es más flexible al depender de un código de programa. El uso de microcontroladores para realizar procesamiento digital de señales se popularizó debido a su bajo costo y altas velocidades de procesamiento. El microcontrolador XMEGA perteneciente a la familia de microcontroladores AVR (Advanced Virtual Risc) de Atmel, se presenta como una alternativa económica para implementar los filtros digitales al incluir un conversor digital análogo (DAC) de 12 bits, lo que no ocurre con algunos dsPIC de Microchip. Los filtros digitales de respuesta finita al impulso (FIR) y los filtros digitales de respuesta infinita al impulso (IIR) del tipo rechaza-banda que fueron diseñados tienen los menores errores: en el caso del filtro digital FIR rechaza-banda se obtuvo un error relativo de 3.31% para la magnitud de la respuesta de frecuencia y un error relativo de 2.58% para la fase de la respuesta de frecuencia, en el caso del filtro digital IIR rechaza-banda se obtuvo un error relativo de 3.29 % para la magnitud de la respuesta de frecuencia y un error relativo de 8.22 % para la fase de la respuesta de frecuencia.

Abstract

The applications of digital filters are very important for the electronics engineering career, due to their use in digital signal processing of audio, video and image, digital telecommunications, biomedic signal analysis and harmonic elimination in laboratory equipment. If the design parameters changes for analogue filters, is required to redesign its structure, digital filters design shows more flexibility due to their dependence of programming code. The use of microcontrollers for digital signal processing has become popular for its low prices and high processing speeds. The XMEGA microcontroller belongs to Atmel AVR (Advanced Virtual Risc) microcontroller's family, this is presented as an economic alternative for implementing digital filters instead of dsPIC from Microchip. Finite impulse response (FIR) and infinite impulse response (IIR) digital filters of band-reject type, showed the lower errors for the magnitude an phase of the frequency response, with the band-reject FIR digital filter the relative error for the magnitude of the frequency response was 3.31% and the relative error for the phase of the frequency response was 2.58%, in the case of the band-reject IIR digital filter the relative error for the magnitude of the frequency response was 3.29% and the relative error for the phase of the frequency response was 8.22%.

Introducción

La aplicación de filtros digitales FIR e IIR dentro del procesamiento digital de señales es muy importante para los estudiantes de ingeniería electrónica en control y en telecomunicaciones debido a que los filtros digitales se incluyen en proyectos de procesamiento de audio y video, codificación y compresión de voz, procesamiento y mejora de imágenes, video vigilancia, en equipos médicos y de laboratorio como: electrocardiografía, control cardiaco, rayos X digitales, medición de señales físicas con sensores, eliminación de armónicos de la red eléctrica, comunicaciones celulares y telefonía. Los filtros digitales han adquirido gran importancia en las aplicaciones con señales de audio, debido a que se reduce el hardware con el que se trabaja, por ejemplo: en equipos de sonido como microcomponentes, la ecualización de sonido, resalte de bajos, aclaramiento de la voz, se consigue gracias al filtrado y tratamiento de señales de audio (Tan, 2007).

El procesamiento digital de señales a través de filtros digitales se realiza en microcontroladores especializados para esta tarea, ya que necesitan adquirir, procesar y reconstruir señales, sin embargo, su costo es elevado. Los microcontroladores de Microchip disponibles para procesamiento de señales, disponen de módulos conversores análogo a digital (ADC) para adquirir las señales, pero necesitan ser comunicados con un módulo DAC externo para reconstruir las señales (Clavijo, 2011).

El microcontrolador XMEGA de Atmel perteneciente a la familia de microcontroladores AVR de 8 bits de Atmel, aparece como una alternativa económica y robusta para realizar filtros digitales y procesamiento digital de señales, dispone de módulos ADC y DAC de 12 bits (Chicaiza & Reyes, 2013).

El uso de microcontroladores de la AVR de la casa ATMEL se ha popularizado en la última década gracias a su bajo costo y alta velocidad de procesamiento, las tarjetas de desarrollo comerciales más utilizadas por los estudiantes, ingenieros y hobbistas que utilizan AVR son las tarjetas Arduino (Atmel, 2016).

La implementación del proyecto se realiza en la tarjeta de desarrollo XMEGA-Ready disponible para los Laboratorios de Sistemas Microprocesados de la Universidad Politécnica Salesiana. Como apoyo al diseño de filtros digitales, los cálculos matemáticos de los coeficientes, su escalamiento a 8 bits y la simulación a través de la respuesta de frecuencia son realizados utilizando MATLAB.

El primer capítulo describe el problema a resolver con este proyecto, se plantea los objetivos, la justificación, alcances y la metodología a seguir para elaborar el proyecto. El segundo capítulo trata sobre la fundamentación teórica necesaria para desarrollar el proyecto, se describe el microcontrolador XMEGA, los conversores ADC y DAC, nociones de filtros digitales y su escalamiento para evitar desbordamiento en la memoria del microcontrolador. El tercer capítulo abarca el diseño e implementación del proyecto, documentando el método empleado para determinar la máxima frecuencia de adquisición y reconstrucción del microcontrolador XMEGA, los cálculos para el diseño de del filtro antialiasing utilizado en el proyecto y la lógica en el diseño de software para los filtros digitales a través de MATLAB y el compilador Mikro C para AVR. El cuarto capítulo describe los análisis y resultados de los experimentos llevados a cabo durante el proyecto, la máxima frecuencia de adquisición y reconstrucción obtenida y la respuesta de frecuencia y fase de cada uno de los filtros digitales FIR y IIR implementados en el microcontrolador XMEGA, comparando valores experimentales con valores teóricos y obteniendo el error al implementarlos. El último capítulo corresponde a las conclusiones y recomendaciones.

En los anexos se encuentra los diagramas de los circuitos del filtro antialiasing, el filtro de reconstrucción y el circuito acondicionador de señales, los códigos en MATLAB para calcular los coeficientes y la respuesta de frecuencia de los filtros digitales y los códigos en Mikro C para AVR para la implementación de los filtros digitales FIR e IIR en el microcontrolador XMEGA. Finalmente se anexa la documentación fotográfica sobre el comportamiento de cada filtro implementado.

Capítulo 1
Planteamiento del problema

Este primer capítulo analiza el problema a resolver y propone los objetivos, alcances y metodologías a desarrollarse en el proyecto.

1.1 Problema a resolver

Los filtros digitales son los sistemas de procesamiento digital de señales más comunes, con ellos se adquiere, procesa y reconstruye señales analógicas. Los microcontroladores son capaces de realizar procesamiento a altas velocidades y son utilizados para procesamiento digital de señales, sin embargo, la mayoría de microcontroladores disponibles en el mercado local no disponen de los módulos conversores analógicos necesarios para adquirir y reconstruir señales, obligando al usuario a comprar y comunicar módulos conversores externos con el microcontrolador.

Existe el microcontrolador XMEGA de Atmel, el cual posee módulos ADC y DAC de 12 bits incorporados, lo cual permite reducir el hardware a utilizar en los proyectos que requieran aplicaciones de procesamiento digital de señales a través de filtros digitales, entonces surge la pregunta:

¿Cómo se puede diseñar e implementar filtros digitales FIR e IIR en un microcontrolador XMEGA de Atmel para el tratamiento de señales de audio?

1.2 Objetivos

A continuación, se presentan el objetivo general y los objetivos específicos para la realización de este proyecto:

1.2.1 Objetivo general

- Diseñar e implementar filtros digitales FIR e IIR en el microcontrolador XMEGA de ATMEL para el tratamiento de señales de audio.

1.2.2 Objetivos específicos

- Determinar la frecuencia máxima de adquisición y reconstrucción de señales en el microcontrolador XMEGA de ATMEL.
- Diseñar y simular los filtros FIR e IIR utilizando MATLAB.
- Implementar los filtros digitales FIR e IIR utilizando el microcontrolador XMEGA para el tratamiento de señales de audio.

- Comparar la respuesta de frecuencia y fase de los filtros digitales implementados en el microcontrolador con respecto a la simulación.

1.3 Justificación

Existe un módulo de entrenamiento basado en el microcontrolador XMEGA de ATMEL desarrollado en el proyecto de titulación: Diseño y construcción de un módulo de entrenamiento basado en el microcontrolador XMEGA de ATMEL para los laboratorios de sistemas microprocesadores de la Universidad Politécnica Salesiana, por parte de la Ing. Nina Gabriela Chicaiza Enriquez y el Ing. Francisco Javier Reyes Almeida, pero no se han implementado filtros digitales FIR e IIR aún (Chicaiza & Reyes, 2013); sin embargo al disponer de ADC y DAC de 12 bits incorporados, el microcontrolador XMEGA de ATMEL muestra potencial para adquirir y reconstruir señales, por lo que sería de mucha utilidad aprovechar estas características de hardware para diseñar e implementar filtros digitales para el tratamiento de señales de audio.

Con el desarrollo de este proyecto se dispondrá de una herramienta de trabajo para estudiar y analizar los diferentes filtros digitales, además los estudiantes tanto de control como de telecomunicaciones podrán usar esta información para proyectos futuros para aplicaciones con filtros multifrecuencia y filtros de Kalman que sirven para análisis, síntesis de señales estocásticas.

1.4 Alcances

Este proyecto de filtros digitales en el microcontrolador XMEGA realiza lo siguiente:

- Diseño e implementación de una placa de circuito impreso (PCB) para un filtro antialiasing a la entrada del ADC y un filtro de reconstrucción a la salida del DAC, utilizando filtros Sallen Key de segundo orden con ganancia unitaria y acondicionamiento de la señal a un voltaje seguro para el microcontrolador XMEGA.
- Programación del microcontrolador XMEGA en lenguaje C con el compilador MikroC para AVR de MikroElektronika.
- Diseño de filtros digitales pasa-bajos, pasa-altos, pasa-banda y rechaza-banda FIR e IIR, filtros digitales FIR por el método de la transformada de Fourier y filtros digitales IIR por el método de la transformada bilineal.

- Cálculo de los coeficientes de los filtros digitales utilizando MATLAB.
- Escalamiento de los coeficientes de los filtros digitales FIR para microcontroladores de 8 bits.
- Diseño de filtros digitales para una frecuencia máxima de muestreo de 16 kHz en los filtros digitales FIR y una frecuencia máxima de muestreo de 8 kHz en los filtros digitales IIR.
- Cálculo de la respuesta de frecuencia en magnitud y fase utilizando MATLAB para simular el comportamiento de los filtros digitales.

El proyecto de filtros digitales con el microcontrolador XMEGA no realiza:

- Filtros digitales con frecuencias de muestreo mayores a 16 KHz.
- Filtros digitales FIR por métodos distintos al de la transformada de Fourier.
- Filtros digitales IIR por métodos diferentes a la transformada bilineal.

1.5 Metodología

Se detalla el procedimiento planteado para la realización del proyecto, empezando por la investigación preliminar, seguida por desarrollo de hardware y software para la construcción final y documentación respectiva.

1.5.1 Investigación preliminar

Se investigará el diseño en forma teórica de los filtros digitales FIR e IIR y su respuesta de frecuencia para magnitud y fase utilizando MATLAB y la realización de los filtros digitales FIR e IIR considerando el escalamiento de coeficientes para evitar desbordamiento en el microcontrolador XMEGA de 8 bits.

1.5.2 Desarrollo de hardware y software

Se diseñará e implementará una PCB (Printed circuit board: Placa de Circuito Impreso) con filtros Sallen Key de segundo orden para antialiasing, reconstrucción y acondicionamiento de la señal a un voltaje seguro entre 0 V y 1 V para el microcontrolador XMEGA.

1.5.3 Construcción final

Se realizarán pruebas correspondientes para comprobar el funcionamiento de cada filtro digital, utilizando el osciloscopio como instrumento de medición y el generador de señales de los laboratorios de la carrera de electrónica para producir la señal analógica de audio que será procesada. Se realizarán las mediciones de amplitud y

desfase de las señales de audio utilizadas para obtener experimentalmente la respuesta de frecuencia en magnitud y frecuencia.

1.5.4 **Documentación**

Como documentación al presente proyecto de investigación se escribirá el texto de la monografía respectiva.

Capítulo 2
Fundamentación teórica

Este capítulo describe las principales características del microcontrolador XMEGA, sus módulos ADC y DAC, breves nociones sobre procesamiento digital de señales y el diseño de filtros digitales.

2.1 Microcontrolador XMEGA de Atmel

XMEGA es un microcontrolador de 8 bits con procesador de punto fijo, de la familia de microcontroladores AVR de Atmel, tiene bajo consumo de energía y utiliza una fuente de alimentación entre 1.6 V a 3.6 V, la CPU (Unidad Central de Procesamiento) trabaja con un reloj interno que puede llegar hasta 32 MHz y cuenta con módulos ADC y DAC de 12 bits; se puede leer la hoja de datos para más información en http://www.atmel.com/Images/doc7925.pdf (Atmel, 2008, pág. 2).

XMEGA como el más avanzado AVR de 8 bits de Atmel

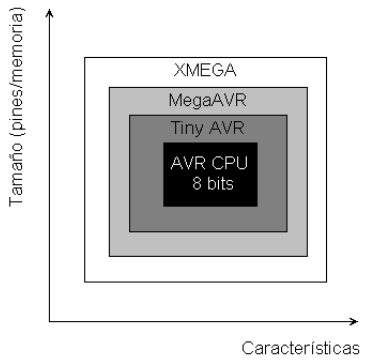


Figura 1. Microcontrolador XMEGA respecto a otros AVRs de 8 bits

Fuente: (Atmel, 2008, pág. 2)

La Figura 1 muestra una comparación entre los AVR de 8 bits de Atmel más populares, TinyAVR y MegaAVR siendo el más robusto el AVR XMEGA en tamaño, memoria y características.

La programación del microcontrolador XMEGA se realiza en lenguajes de alto nivel como C o Basic a través de un software compilador, accediendo a los registros del microcontrolador para configurar y activar el funcionamiento de sus módulos periféricos; por lo tanto, para trabajar con datos que ocupen más de 8 bits como el ADC y el DAC de 12 bits en el microcontrolador AVR XMEGA se divide los valores en dos registros de 8 bits (Atmel, 2008, pág. 5).

Representación de un registro de en el AVR XMEGA

Figura 2. Almacenamiento de un registro de 12 bits en 2 registros de 8 bits.

Fuente: (Atmel, 2008, pág. 5)

La Figura 2 muestra el uso de dos registros de 8 bits para almacenar un valor de 12 bits, un registro con los bits más significativos (MSB) denominado registro alto y un registro con los bits menos significativos (LSB) denominado registro bajo (Atmel, 2008, pág. 5).

2.1.1 Tarjeta de desarrollo XMEGA-Ready de MikroElektronika

El proyecto técnico se realizó en base al microcontrolador AVR ATxmega128A1, se utilizó la tarjeta de desarrollo XMEGA-Ready de MikroElektronika, disponible en los Laboratorios de Sistemas Microprocesados de la Universidad Politécnica Salesiana, como se puede ver en la Figura 3.

PCB con el microcontrolador AVR ATxmega128A1.

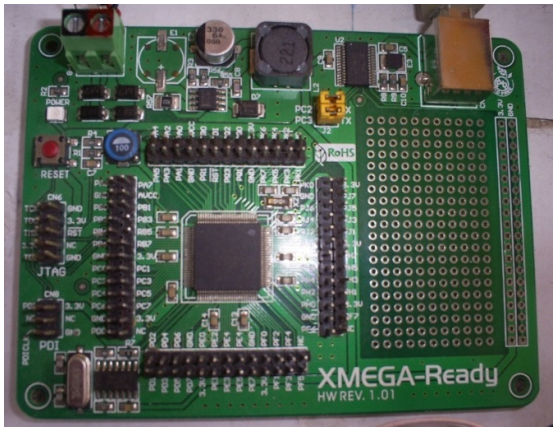


Figura 3. Fotografía de la tarjeta de desarrollo XMEGA-Ready.

Elaborado por: Diego Jarrín.

La tarjeta XMEGA-Ready cuenta con su propio circuito de acondicionamiento para alimentación, asegurando un voltaje de 3.6 V para que funcione el microcontrolador, además tiene conexión USB-UART para programar el AVR con el Bootloader de MikroElektronika (MikroElektronika, 2012, pág. 3).

2.1.2 **Convertidor análogo a digital**

El microcontrolador XMEGA posee dos módulos ADC de 12 bits, uno en el puerto A denotado como ADC A y otro en el puerto B denotada como ADC B, con ocho pines de entrada en cada puerto (Atmel, 2013, pág. 49).

La configuración del ADC en el AVR XMEGA se muestra en la Figura 4:

Configuración para funcionamiento del ADC

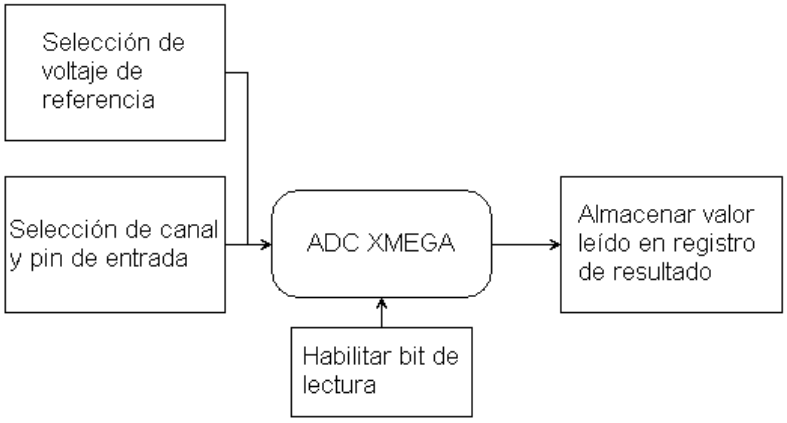


Figura 4. Configuración del ADC en el AVR XMEGA

Fuente: (Atmel, 2013, pág. 49)

El módulo ADC se configura seleccionando el voltaje de referencia, el pin de entrada donde realizará la lectura de voltaje y el canal donde se almacenará el resultado, existen cuatro canales por cada pin de entrada; se debe habilitar el bit de lectura para leer la entrada correspondiente y guardar el resultado. La entrada leída se encuentra en un rango de 0 a 4095, estos 4096 valores se deben a los 12 bits del ADC (Atmel, 2013, pág. 49).

2.1.2.1 Registros para configurar el módulo ADC

Los registros para configurar un módulo del ADC según la hoja de datos del microcontrolador XMEGA (Atmel, 2013, pág. 22) son:

1. ADCx_CTRLA para habilitar el ADC y seleccionar el voltaje de referencia, donde x indica el puerto x=A o x=B respectivamente.
2. ADCx_CHn_MUXCTRL para seleccionar el pin correspondiente a la entrada física que utilizara ese canal.
3. ADCx_CHn_CTRL para habilitar uno de los cuatro canales donde se almacenará la lectura del ADC; n indica el número de canal, n es un número entre 0 y 3.

4. El resultado de 12 bits se guarda en los registros de 8 bits ADCx_CHnRESL y ADCx_CHnRESH.

2.1.3 Convertidor digital a analógico

El microcontrolador XMEGA posee dos módulos DAC de 12 bits, uno en el puerto A denotado como DAC A y otro en el puerto B denotada como DAC B, con dos pines de salida en cada puerto (Atmel, 2013, pág. 51).

La configuración del DAC en el AVR XMEGA se muestra en la Figura 5:

Configuración para funcionamiento del DAC

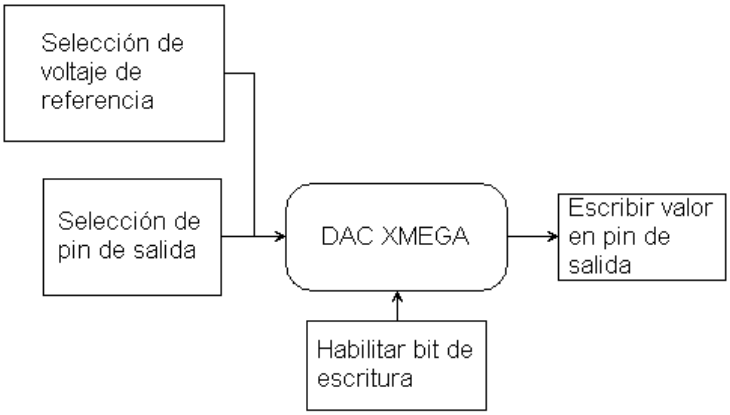


Figura 5. Configuración del DAC en el AVR XMEGA

Fuente: (Atmel, 2013, pág. 51)

La configuración del DAC se realiza seleccionando el voltaje de referencia, el pin de salida a utilizar y es necesario habilitar el bit de escritura para escribir el voltaje en la salida respectiva. La salida reconstruida se encuentra en un rango de 0 a 4095, estos 4096 valores se deben a los 12 bits del DAC (Atmel, 2013, pág. 51).

2.1.3.1 Registros para configurar el módulo DAC

El registro de 8 bits denominado DACx_CTRLA permite la configuración del DAC, donde x indica el puerto x=A o x=B respectivamente, dicho registro habilita el uso del DAC, selecciona el voltaje de referencia y se direcciona el canal donde se envía el valor a escribirse, existen dos canales, canal 0 y canal 1, cada uno utiliza un pin de salida. La escritura en el respectivo canal n se realiza a través de los registros de 8 bits DACx_CHnDATAL y DACx_CHnDATAH (Atmel, 2008, pág. 6).

2.2 Procesamiento digital de señales

Un sistema dedicado al procesamiento digital de señales (DSP) es el que adquiere, procesa y reconstruye señales utilizando algoritmos digitales. La Figura 6 representa un sistema DSP donde una señal analógica representada como x(t) es adquirida y digitalizada en una señal discreta x(n) por el ADC, tratada por el procesador digital de señales obteniendo una señal discreta y(n) y reconstruida por el DAC como una señal analógica y(t). (Proakis & Manolakis, 2007, pág. 5).

Esquema del procesamiento digital de señales

Figura 6. Diagrama de bloques de un sistema DSP

Fuente: (Proakis & Manolakis, 2007, pág. 5)

La función de transferencia H(n) de un sistema define la relación entre la salida y(n) y la entrada x(n), el procesamiento de una señal consiste en calcular la señal de salida del sistema mediante una ecuación en diferencia, la cual expresa la salida como un producto o convolución entre la señal de entrada y la función de transferencia, $y(n)=x(n)\cdot H(n)$ (Clavijo, 2011, pág. 191).

Las señales de audio son señales analógicas con frecuencias de hasta 20 kHz (Tan, 2007, pág. 15).

Los microcontroladores AVR de 8 bits se presentan como una solución económica que cuenta con los módulos necesarios para ser utilizados como procesadores digitales de señales (Atmel, 2008, pág. 1).

La adquisición y reconstrucción de señales requieren un acondicionamiento de voltajes para que el microcontrolador pueda procesarlos, se requiere de filtros antialiasing y de reconstrucción (Proakis & Manolakis, 2007, pág. 746).

2.2.1.1 Filtro antialiasing y de reconstrucción

El aliasing es un problema que afecta al procesamiento digital de señales, ocurre cuando el tiempo de procesamiento del sistema DSP no es suficiente para reconstruir las señales, debido a que estas poseen frecuencias mayores a las que el sistema puede procesar, generando una reconstrucción incorrecta de las señales procesadas. Para evitar el aliasing, se utiliza un filtro pasa-bajos denominado filtro antialiasing, el cual

limita las frecuencias de las señales a un rango que el sistema puede procesar, la frecuencia límite se denomina frecuencia de corte y debe ser menor o igual a la mitad de la frecuencia de muestreo (Proakis & Manolakis, 2007, pág. 27).

El filtro pasa-bajos que se muestra en la Figura 7, es un filtro activo de segundo orden y ganancia unitaria denominado filtro Sallen Key, es utilizado como filtro antialiasing, para determinar la frecuencia de corte (Fc) del filtro se utiliza la Ecuación 1 (Tan, 2007, pág. 26).

$$Fc=\frac{1}{2\cdot\pi\cdot\sqrt{R_1\cdot R_2\cdot C_1\cdot C_2}} \qquad \text{Ecuación 1}$$

Donde R1, R2 representan resistencias eléctricas y C1, C2, representan los capacitores cerámicos.

Filtro analógico activo pasa-bajos de segundo orden

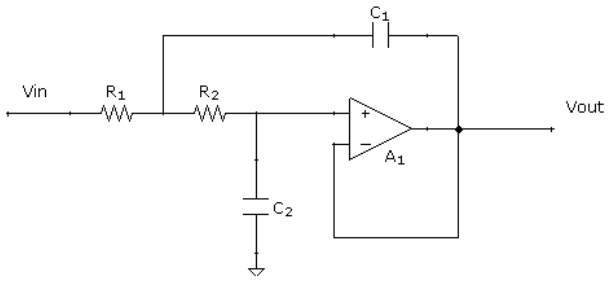


Figura 7. Filtro Sallen Key de ganancia unitaria

Fuente: (Tan, 2007, pág. 26)

2.3 Filtros digitales

Un filtro digital es un algoritmo de procesamiento digital de señales donde se extrae información dejando pasar o rechazando una banda de frecuencias de la señal procesada. Las principales configuraciones para filtros son: pasa-bajos, pasa-altos, pasa-banda y rechaza-banda; el nombre de cada uno se debe al tipo de respuesta de frecuencia que se busca conseguir con tales filtros (Clavijo, 2011, pág. 191).

La Figura 8 indica el diagrama de bloques para un filtro digital, el cual depende de las muestras actuales y pasadas de la entrada y salida del sistema DSP y la función de transferencia del filtro, representada como un conjunto de coeficientes o impulsos. Un filtro digital puede ser implementado a través de la ecuación en diferencias que lo define (Clavijo, 2011, pág. 191).

Esquema del filtrado digital de una señal discreta

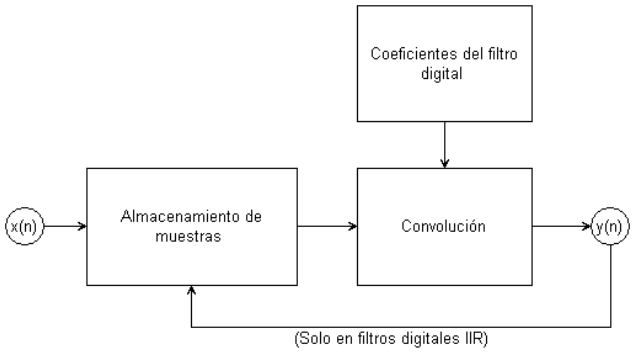


Figura 8. Diagrama de bloques para los filtros digitales

Fuente: (Clavijo, 2011, pág. 191)

La ecuación en diferencias para los filtros digitales IIR se muestra en la Ecuación 2 (Clavijo, 2011, pág. 191):

$$y(0)=\sum_{0}^{M} b_k \cdot x(n-k) - \sum_{1}^{N} a_k \cdot y(n-k) \qquad \text{Ecuación 2}$$

Donde b_k,a_k son los coeficientes del filtro digital, x(n-k) y y(n-k) son las señales discretas de la entrada y salida respectivamente, desplazadas k veces para representar las muestras almacenadas, denominadas muestras pasadas.

En el caso de los filtros digitales FIR, no se considera las muestras de la salida, la Ecuación 3 muestra la ecuación en diferencia de los filtros digitales FIR (Clavijo, 2011, pág. 191).

$$y(0)=\sum_{0}^{M} b_k \cdot x(n-k) \qquad \text{Ecuación 3}$$

El mayor de los índices M o N en las sumatorias, determina el orden del filtro digital, para filtros simétricos se considera M=N y M es un número par, existen M+1 coeficientes. El diseño de un filtro digital se resume en calcular sus coeficientes y con ellos determinar la respuesta de frecuencia para simular su comportamiento (Tan, 2007, pág. 160).

2.3.1 **Filtros digitales FIR**

El método de la transformada de Fourier simplifica el diseño de los filtros digitales FIR pasa-bajos, pasa-altos, pasa-banda, rechaza-banda utilizando a las fórmulas resumidas en la Tabla 1 para el cálculo de sus coeficientes (Tan, 2007, pág. 220):

Tabla 1.

Coeficientes de un filtro digital FIR por el método de la transformada de Fourier

Muestra	Coeficientes $h(n)$ de un filtro digital FIR no causal			
$-\frac{M}{2} \leq n \leq \frac{M}{2}$	Pasa bajos	Pasa altos	Pasa banda	Rechaza banda
n=0	$\frac{\Omega_L}{\pi}$	$\frac{\pi-\Omega_H}{\pi}$	$\frac{\Omega_H-\Omega_L}{\pi}$	$\frac{\pi-\Omega_H+\Omega_L}{\pi}$
n≠0	$\frac{\sin(\Omega_L \cdot n)}{n \cdot \pi}$	$-\frac{\sin(\Omega_H \cdot n)}{n \cdot \pi}$	$\frac{\sin(\Omega_H \cdot n)}{n \cdot \pi} - \frac{\sin(\Omega_L \cdot n)}{n \cdot \pi}$	$-\frac{\sin(\Omega_H \cdot n)}{n \cdot \pi} + \frac{\sin(\Omega_L \cdot n)}{n \cdot \pi}$

Nota: Se muestran las fórmulas necesarias para calcular los coeficientes de cada tipo de filtro.
Fuente (Tan, 2007, pág. 220).

Las fórmulas empleadas en la Tabla 1 son utilizadas para diseñar un filtro no causal porque consideran en sus cálculos las muestras presentes, pasadas y futuras, pero para la implementación en tiempo real donde se desconoce las muestras futuras, se debe introducir un desplazamiento, tal como se muestra en la Ecuación 4, transformando al filtro no causal en un filtro causal que depende de las muestras presentes y pasadas (Tan, 2007, pág. 220):

$$b_n = h\left(n - \frac{M}{2}\right) \qquad \text{Ecuación 4}$$

Donde n tiene valores entre 0 y M

La Ecuación 5 representa la frecuencia de corte normalizada (Ω) en radianes, en función de la frecuencia de muestreo (Fs) y la frecuencia de corte (f) en Hz, el subíndice L indica si es una frecuencia de corte baja y el subíndice H indica si una frecuencia de corte alta (Tan, 2007, pág. 220).

$$\Omega = 2 \cdot \pi \cdot \frac{f}{F_s} \qquad \text{Ecuación 5}$$

2.3.2 **Filtros digitales IIR**

El método de la transformada bilineal considera el diseño de los filtros digitales IIR en base a la adaptación de filtros analógicos. Para simplificar el diseño y reducir el número de operaciones aritméticas y cálculos matemáticos, se utilizará filtros de segundo orden. La Tabla 2 muestra la función de transferencia de filtros analógicos de segundo orden (Clavijo, 2011, pág. 213):

Tabla 2.

Función de transferencia de filtros analógicos de segundo orden

Filtro analógico	Función de transferencia $H(s)$
Pasa bajos	$\frac{\omega_L^2}{s^2+2\cdot\zeta\cdot\omega_L\cdot s+\omega_L^2}$
Pasa altos	$\frac{s^2}{s^2+2\cdot\zeta\cdot\omega_H\cdot s+\omega_H^2}$
Pasa banda	$\frac{s\cdot\omega_L}{s^2+\omega_L\cdot s+\omega_H^2}$
Rechaza banda	$\frac{s^2+\omega_H^2}{s^2+\omega_L\cdot s+\omega_H^2}$

Nota: Se muestran las funciones de transferencia de distintos filtros analógicos de segundo orden.

Fuente: (Clavijo, 2011, pág. 213).

Donde ω representa la frecuencia análoga normalizada, su relación con la frecuencia digital normalizada **Ω** es $\omega=2\cdot f_s\cdot \tan\left(\frac{\Omega}{2}\right)$, en los filtros digitales pasa-bajos y pasa-altos se asigna al factor de amortiguamiento un valor de $\zeta=\frac{\sqrt{2}}{2}$ para no desplazar la frecuencia de corte. La transformada bilineal realiza la conversión del plano s el cual representa la frecuencia compleja para un sistema en tiempo continuo, al plano z el cual representa la frecuencia compleja para un sistema en tiempo discreto, utilizando la Ecuación 6, donde (Fs) representa la frecuencia de muestreo (Clavijo, 2011, pág. 213).

$$s=2\cdot F_s\cdot\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \quad \text{Ecuación 6}$$

Se despeja los coeficientes b_k,a_k obteniendo las fórmulas resumidas en la Tabla 3 (Clavijo, 2011, págs. 213, 223)

Tabla 3.

Coeficientes de los filtros digitales IIR de segundo orden

Coeficientes	Pasa bajos	Pasa altos
$\mathbf{b_0}$	$\frac{\omega_L^2}{4\cdot Fs^2+4\cdot\zeta\cdot\omega_L\cdot Fs+\omega_L^2}$	$\frac{4\cdot Fs^2}{4\cdot Fs^2+4\cdot\zeta\cdot\omega_H\cdot Fs+\omega_H^2}$
$\mathbf{b_1}$	$2\cdot b_0$	$-2\cdot b_0$
$\mathbf{b_2}$	b_0	
$\mathbf{a_0}$	1	
$\mathbf{a_1}$	$\frac{2\cdot\omega_C^2-8\cdot Fs^2}{4\cdot Fs^2+4\cdot\zeta\cdot\omega_C\cdot Fs+\omega_C^2}$	
$\mathbf{a_2}$	$\frac{4\cdot Fs^2-4\cdot\zeta\cdot\omega_C Fs+\omega_C^2}{4\cdot Fs^2+4\cdot\zeta\cdot\omega_C\cdot Fs+\omega_{LC}^2}$	
Coeficientes	Pasa banda	Rechaza banda
$\mathbf{b_0}$	$\frac{2\cdot Fs\cdot\omega_H}{4\cdot Fs^2+2\cdot\omega_H\cdot Fs+\omega_L^2}$	$\frac{4\cdot Fs^2+\omega_L^2}{4\cdot F_s^2+2\cdot\omega_H\cdot Fs+\omega_L^2}$
$\mathbf{b_1}$	0	$\frac{2\cdot\omega_L^2-8\cdot Fs^2}{4\cdot Fs^2+2\cdot\omega_H\cdot Fs+\omega_L^2}$
$\mathbf{b_2}$	$-b_0$	b_0
$\mathbf{a_0}$	1	
$\mathbf{a_1}$	$\frac{2\cdot\omega_L^2-2\cdot Fs\cdot\omega_H}{4\cdot Fs^2+2\cdot\omega_H\cdot Fs+\omega_L^2}$	b_1
$\mathbf{a_2}$	$\frac{4\cdot Fs^2-2\cdot\omega_H\cdot Fs+\omega_L^2}{4\cdot Fs^2+2\cdot\omega_H\cdot Fs+\omega_L^2}$	

Nota: Coeficientes de los filtros digitales IIR obtenidos por el método de la transformada bilineal.

Fuente: (Clavijo, 2011, págs. 213, 223).

Aplicando la transformada z inversa en H(z) se obtiene la función de transferencia H(n) para poder implementar la ecuación en diferencias con la que se programa los filtros digitales IIR (Tan, 2007, pág. 165).

2.3.3 **Respuesta en frecuencia**

La respuesta en frecuencia permite predecir el comportamiento de un filtro, una vez conocidos los coeficientes en la función de transferencia H(z) del filtro digital se realiza el reemplazo $z=e^{j\cdot\Omega}$. La Tabla 4 muestra las fórmulas utilizadas en el cálculo de la magnitud y fase de la respuesta de frecuencia en base a la función de transferencia H(z) de un filtro digital IIR (Tan, 2007, págs. 179, 182):

Tabla 4.

Respuesta en frecuencia de los filtros digitales

Función de transferencia	$H(z)=\frac{\sum_0^M b_k \cdot z^{-k}}{1+\sum_1^N a_k \cdot z^{-k}}$
Respuesta en frecuencia	$H(e^{j\cdot\Omega})=\frac{\sum_0^M b_k \cdot e^{-j\cdot\Omega\cdot k}}{1+\sum_1^N a_k \cdot e^{-j\cdot\Omega\cdot k}}=p+j\cdot q$
Magnitud	$\lvert H(e^{-j\cdot\Omega})\rvert=\sqrt{p^2+q^2}$
Fase	$\angle H(e^{-j\cdot\Omega})=\tan^{-1}\left(\frac{q}{p}\right)$

Nota: Se muestra las fórmulas para calcular la magnitud y fase de la respuesta de frecuencia.

Fuente: (Tan, 2007, págs. 179, 182).

2.3.4 Escalamiento de los coeficientes de los filtros digitales

Los filtros digitales pueden generar desbordamiento en la memoria del microcontrolador, cuando la respuesta impulso obtenida del filtro digital considera coeficientes que alcanzan un valor en la salida que está fuera del rango que el microcontrolador puede reconstruir. Para el microcontrolador XMEGA de 8 bits, el rango de valores a la salida es de 0 a 4095, se escala los coeficientes en un rango de $2^8 = 256$ valores debido a los 8 bits que representan el tamaño de memoria de sus registros, los coeficientes escalados se encuentran en un rango comprendido entre -128 a 127 debido a que los coeficientes son positivos y negativos. Se calcula el factor de escalamiento y cada coeficiente se escala dividiendo para dicho factor. El factor de escalamiento se representa como un número en potencias de 2. El escalamiento produce perdidas en la información (Atmel, 2008, pág. 7).

Para un conjunto b(n) de coeficientes, el factor de escalamiento se calcula con la Ecuación 7, Vmax representa el valor máximo que el microcontrolador puede reconstruir, en el caso del microcontrolador XMEGA el máximo valor es de 4095, la sumatoria de coeficientes representa su respuesta al impulso (Tan, 2007, pág. 442).

$$\mathbf{S=V_{max}\cdot\sum_0^M b_k} \qquad \text{Ecuación 7}$$

Capítulo 3
Diseño e implementación

En este capítulo se describe los cálculos utilizados en el diseño de la tarjeta de filtros antialiasing, de reconstrucción y acondicionamiento de la señal a procesar, el procedimiento para determinar la máxima frecuencia de adquisición y reconstrucción con el microcontrolador AVR XMEGA, el diseño de software para implementar los filtros digitales FIR e IIR en Mikro C para AVR y en MATLAB.

3.1 Diseño de la tarjeta de filtros antialiasing, de reconstrucción y acondicionamiento de las señales a procesar

Para realizar en forma adecuada el procesamiento digital de señales se requiere elaborar un filtro antialiasing y ubicarlo a la entrada del ADC, un filtro de reconstrucción colocado a la salida del DAC y un circuito de acondicionamiento para asegurar voltajes positivos en el microcontrolador.

3.1.1 Filtro antialiasing y de reconstrucción

Para el filtro Sallen Key de segundo orden con ganancia unitaria se calcula el valor de las resistencias si los capacitores cerámicos utilizados son asumidos con valor de 100 nF y la máxima frecuencia de muestreo a utilizar es de 16 kHz. La frecuencia de corte del filtro es la frecuencia de muestreo divida para dos, en este caso es de 8 kHz. Despejando el valor de la resistencia R en la Ecuación 2.

$$R=\frac{1}{2\cdot\pi\cdot 8\text{ kHz}\cdot 100\text{ nF}}=198.9437\ \Omega$$

3.1.2 Acondicionamiento de la señal

Se utiliza un sumador no inversor con ganancia unitaria implementado a través del amplificador operacional como circuito para acondicionar las señales de audio a un rango de voltajes positivo necesarios para el microcontrolador XMEGA, la Figura 9 indica el circuito acondicionador, la señal de audio que posee voltajes positivos y negativos en un rango de voltaje de -0.5 V a 0.5 V se representa en la entrada E1, si se le suma un voltaje continuo de 0.5 V en la entrada E2, se consigue a la salida el rango de 0 V a 1 V necesario para el microcontrolador.

Sumador no inversor como circuito acondicionador

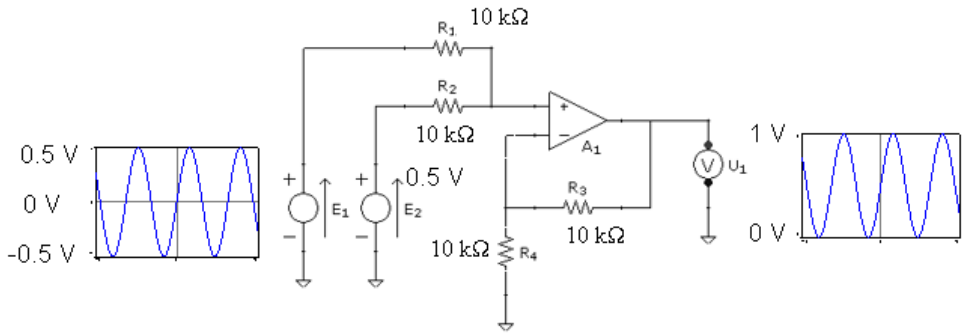


Figura 9. Acondicionador de la señal de audio.

Elaborado por: Diego Jarrín.

3.2 Máxima frecuencia de adquisición y reconstrucción

Para determinar la máxima frecuencia que permite la reconstrucción de una señal digital, se adquiere y reconstruye la señal incrementado la frecuencia de un generador de señales y observando en un osciloscopio la señal reconstruida por el DAC, hasta que la señal análoga reconstruida tenga forma similar a la del generador, si la forma se distorsiona se considera que se sobrepasó la frecuencia de muestreo del microcontrolador.

3.3 Códigos en MATLAB y Mikro C para AVR para diseñar e implementar los filtros digitales FIR e IIR

A continuación, se detalla a través de diagramas de flujo los códigos en MATLAB y Mikro C para AVR para el diseño e implementación de los filtros digitales FIR e IIR en el microcontrolador XMEGA.

3.3.1 Diseño de los filtros digitales en MATLAB

Según el número de coeficientes del filtro digital, el cálculo de coeficientes y el cálculo de la respuesta de frecuencia se vuelve largo, por eso se lo realiza con la ayuda de un software como es MATLAB. Adicionalmente al cálculo de coeficientes de los filtros digitales, se debe garantizar que estos quepan en la memoria y registros del microcontrolador donde van a ser implementados. Se desarrolla el cálculo matemático considerando el respectivo escalamiento a la memoria del microcontrolador. Los códigos en MATLAB para calcular los coeficientes y simular la respuesta en frecuencia de los filtros digitales FIR e IIR se encuentran en el Anexo 2.

Para los filtros digitales FIR e IIR se ha elaborado una lógica de diseño similar tal como se muestra en la Figura 10.

Lógica de diseño de filtros digitales en MATLAB

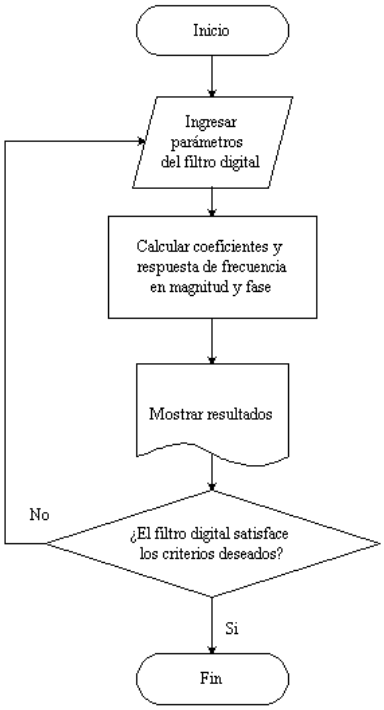


Figura 10. Diagrama de flujo del diseño de los filtros digitales en MATLAB.

Elaborado por: Diego Jarrín.

1. Inicia la programación del diseño de filtros digitales FIR e IIR en MATLAB
2. El usuario ingresa parámetros como número de coeficientes, frecuencia de muestreo y tipo de filtro.
3. En MATLAB se calcula los coeficientes y se analiza si es necesario realizar su escalamiento a 8 bits, la simulación del comportamiento del filtro se realiza en base a la gráfica de magnitud y fase de la respuesta en frecuencia para los coeficientes calculados.
4. Se muestra los resultados de los coeficientes escalados para que puedan ser utilizados por el compilador Mikro C para AVR.
5. El usuario puede terminar el proceso o puede continuar calculando nuevos coeficientes para otros filtros.
6. Finaliza la programación del diseño de filtros digitales FIR e IIR en MATLAB.

3.3.2 **Códigos en Mikro C para AVR**

Utilizando el compilador Mikro C para AVR, se crea un nuevo proyecto, se especifica el AVR ATxmega128A1 y la velocidad de reloj a utilizarse, en este caso la velocidad más alta permitida es de 32 MHz, se especifica el directorio donde se guardará el proyecto y en el archivo C se programa el algoritmo DSP para los filtros digitales, siguiendo la lógica mostrada en la Figura 11. Los códigos en Mikro C para AVR para implementar los filtros digitales FIR e IIR se encuentran en el Anexo 2.

Lógica de programación en el AVR XMEGA para el sistema DSP de los filtros digitales

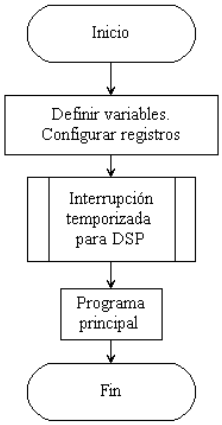


Figura 11. Diagrama de flujo del programa utilizado para DSP con la XMEGA.

Elaborado por: Diego Jarrín.

1. Inicia la programación orientado a DSP con el microcontrolador AVR XMEGA.
2. Se define variables donde se almacenarán las muestras tomadas por el ADC y los valores que se van a escribir en el DAC, también las variables donde se almacenarán los coeficientes y el orden del filtro digital; además se configuran los registros: para el reloj interno con el que se trabajará, el ADC, el DAC y la interrupción temporizada.
3. La interrupción temporizada es una subrutina que se ejecuta continuamente según el tiempo en la que fue configurada, en ella se realiza el algoritmo DSP de los filtros digitales.
4. Se define un bucle infinito a través del comando de programación “while” en el programa principal para que el microcontrolador se mantenga funcionando indefinidamente.

5. Finaliza la programación orientado a DSP con el microcontrolador XMEGA.

3.3.2.1 Interrupción temporizada para los filtros digitales FIR

La interrupción temporizada para DSP orientada a los filtros digitales FIR obedece el diagrama de flujo mostrado en la Figura 12:

Lógica de programación de un filtro digital FIR

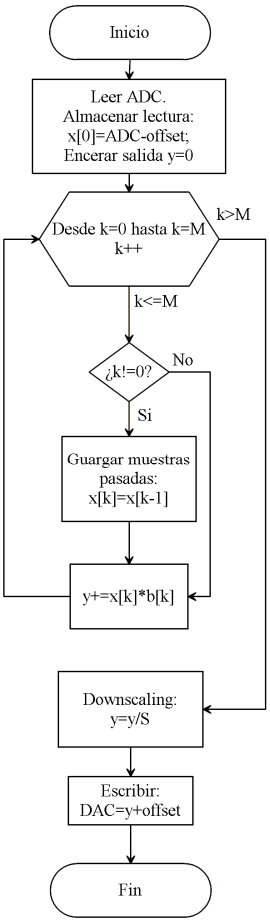


Figura 12. Diagrama de flujo para filtros digitales FIR.

Elaborado por: Diego Jarrín.

1. Inicia el programa del filtro digital FIR durante la interrupción temporizada.
2. Se lee la entrada del ADC y almacena el valor leído en una variable x[0], a la que para evitar problemas de desbordamiento, se le resta un valor denominado offset con el fin de mantener el valor en un rango entre -2048 y 2047.
3. Según el orden del filtro digital FIR definido en la constante M, se ejecuta un bucle a través del comando de programación “for”, desde k=0 hasta M para

almacenar muestras pasadas de la muestra presente (k=0) tomada por el ADC en el vector X; por medio de una sentencia condicional "if" se analiza las muestras y realiza el almacenamiento respectivo, la muestra presente se guarda en la posición x[0] y cada vez que se adquiere una nueva muestra, se desplazan las muestras pasadas en el vector X tal que x[k]=x[k-1].

4. Con los valores almacenados se procede a efectuar las operaciones aritméticas de multiplicación de las muestras por los coeficientes, X*B, obteniendo la ecuación en diferencias del filtro digital FIR.
5. Para evitar desbordamiento en la salida, se divide la salida para el factor de escalamiento S.
6. Al dato de que se escribir en el DAC se le suma el valor de offset restado al valor de la muestra tomado por el ADC, los valores deben estar en un rango de 0 a 4095.

3.3.2.2 Interrupción temporizada para los filtros digitales IIR

La interrupción temporizada para DSP orientada a los filtros digitales IIR obedece el diagrama de flujo mostrado en la Figura 13:

Lógica de programación de un filtro digital IIR

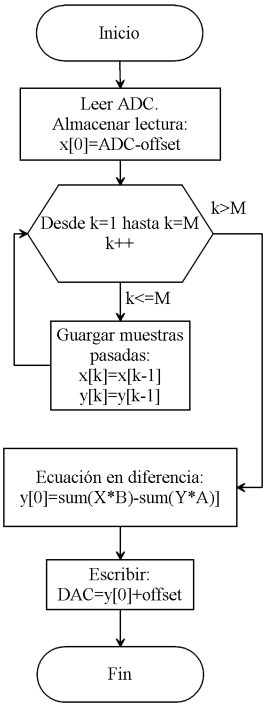


Figura 13. Diagrama de flujo para filtros digitales IIR.

Elaborado por: Diego Jarrín.

1. Inicia el programa del filtro digital IIR durante la interrupción temporizada.
2. Se lee la entrada del ADC y almacena el valor leído en una variable x[0] a la que para evitar problemas de desbordamiento, se le resta un valor de offset para mantener el dato en un rango entre -2048 y 2047.
3. Según el orden del filtro digital IIR definido por la constante M, se ejecuta un bucle for desde k=0 hasta M en el que se almacena muestras tomadas por el ADC en el vector X y los datos que se van a escribir en el DAC en el vector Y; por medio de una sentencia condicional "if" se analiza las muestras y realiza el almacenamiento respectivo, las muestras presentes se guarda en la posición x[0] y y[0] para la entrada y la salida respectivamente, cada vez que se adquiere una nueva muestra, se desplazan las muestras pasadas en el vector X tal que x[k]=x[k-1] y y[k]=y[k-1].
4. Con los valores almacenados se procede a efectuar las operaciones aritméticas de multiplicación de las muestras por los coeficientes, X*B para la entrada, Y*A obteniendo la ecuación en diferencia del filtro digital IIR.
5. Al dato que se va a escribir en el DAC se le suma el valor restado al dato tomado por el ADC, el valor que se escribirá debe estar en un rango de 0 a 4095.
6. Se finaliza el programa del filtro digital IIR.

Capítulo 4
Análisis de resultados

En este capítulo se describe la implementación del circuito de filtros antialiasing, de reconstrucción y acondicionamiento de las señales a ser procesadas con el microcontrolador XMEGA, obteniendo los siguientes resultados: la máxima frecuencia de adquisición y reconstrucción obtenida con el microcontrolador AVR XMEGA, la respuesta de frecuencia de cada filtro digital implementado en el AVR XMEGA respecto a la simulación de MATLAB.

4.1 Implementación de la tarjeta de filtros antialiasing, reconstrucción y acondicionamiento de las señales

La Figura 14 muestra la PCB implementada para filtros antialiasing, de reconstrucción y acondicionamiento de señales, el esquemático y los detalles para elaborar este circuito electrónico se muestran en el Anexo 1. Una primera bornera es utilizada para la alimentación a ±15V DC, una segunda bornera es utilizada para conexión con la tarjeta XMEGA-Ready y una tercera bornera es utilizada para conexión con el osciloscopio y la entrada de audio. Los filtros y el acondicionador se diseñaron utilizando cuatro amplificadores operacionales del circuito integrado TL084, el cual posee tecnología JFET que lo vuelve menos vulnerable al ruido externo (Texas Instruments, 2016, pág. 14).

Implementación de la PCB para filtros analógicos

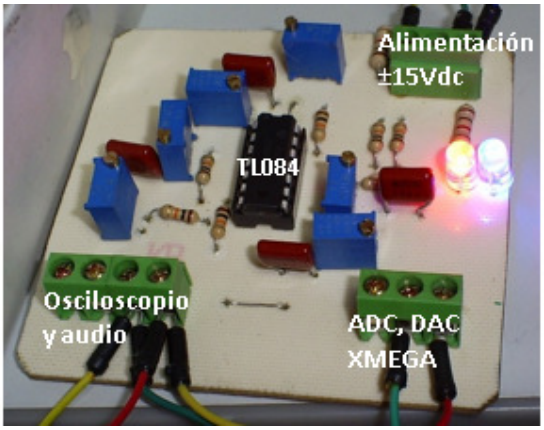


Figura 14. PCB de los filtros antialiasing y de reconstrucción.

Elaborado por: Diego Jarrín.

4.2 Determinación de la frecuencia máxima de adquisición y reconstrucción

Utilizando el generador de señales disponible en los Laboratorios de Electrónica de la Universidad Politécnica Salesiana, se ingresó al microcontrolador señales con forma de onda cuadrada y señales con forma de onda sinusoidal, ambas con frecuencias entre 0 Hz y 5 MHz y amplitud entre 0 V y 1 V. Observando la reconstrucción de las señales procesadas en el osciloscopio, se determinó experimentalmente que en el microcontrolador la máxima frecuencia de adquisición y reconstrucción para señales con forma de onda cuadrada es de 200 kHz y para señales con forma de onda sinusoidal, la máxima frecuencia de adquisición y reconstrucción es de 100 kHz.

En la Figura 15 se observa la señal análoga de una onda cuadrada de 100 kHz adquirida a la entrada del ADC, conectada al canal 1 del osciloscopio, mostrada en color rojo y la señal reconstruida a la salida del DAC, conectada al canal 2 del osciloscopio, mostrada en color amarillo:

Máxima frecuencia para adquirir y reconstruir una señal analógica cuadrada

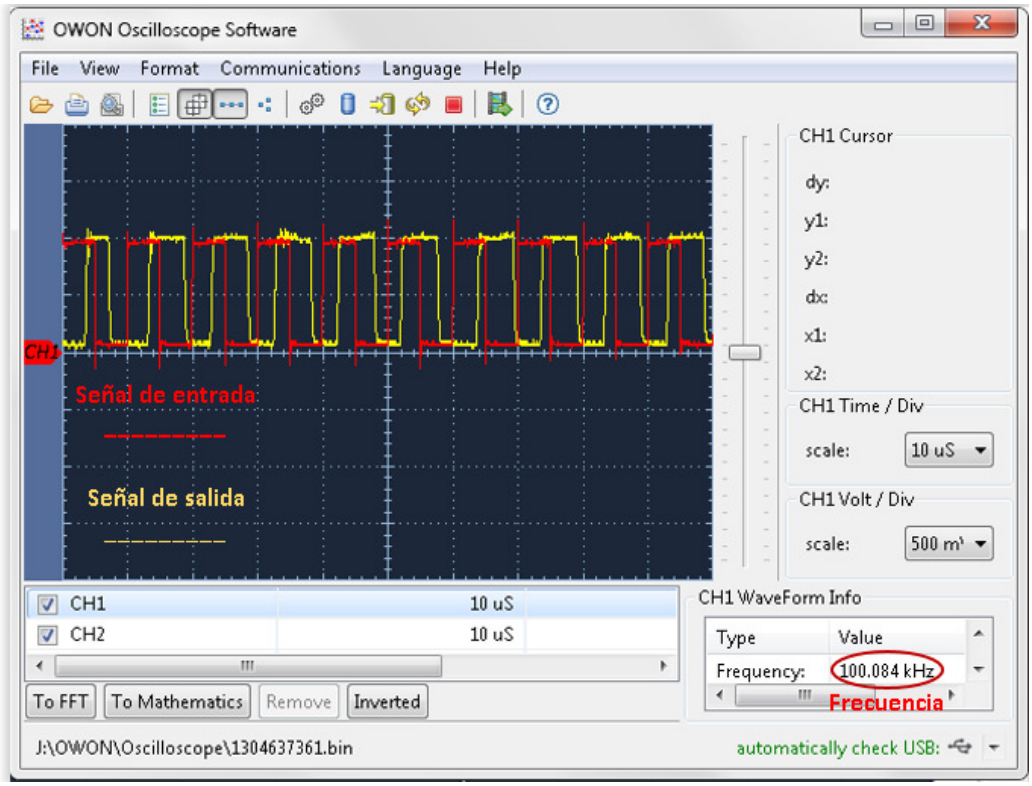


Figura 15. Adquisición y reconstrucción de una señal analógica cuadrada de 100 kHz.

Elaborado por: Diego Jarrín.

Se logró adquirir y reconstruir una señal analógica sinusoidal de 50 kHz como se observa en la Figura 16, la señal adquirida a la entrada del ADC y conectada al canal 1 del osciloscopio es mostrada en color rojo y la señal reconstruida a la salida del DAC conectado en el canal 2 del osciloscopio se muestra en color amarillo:

Máxima frecuencia para adquirir y reconstruir una señal analógica sinusoidal

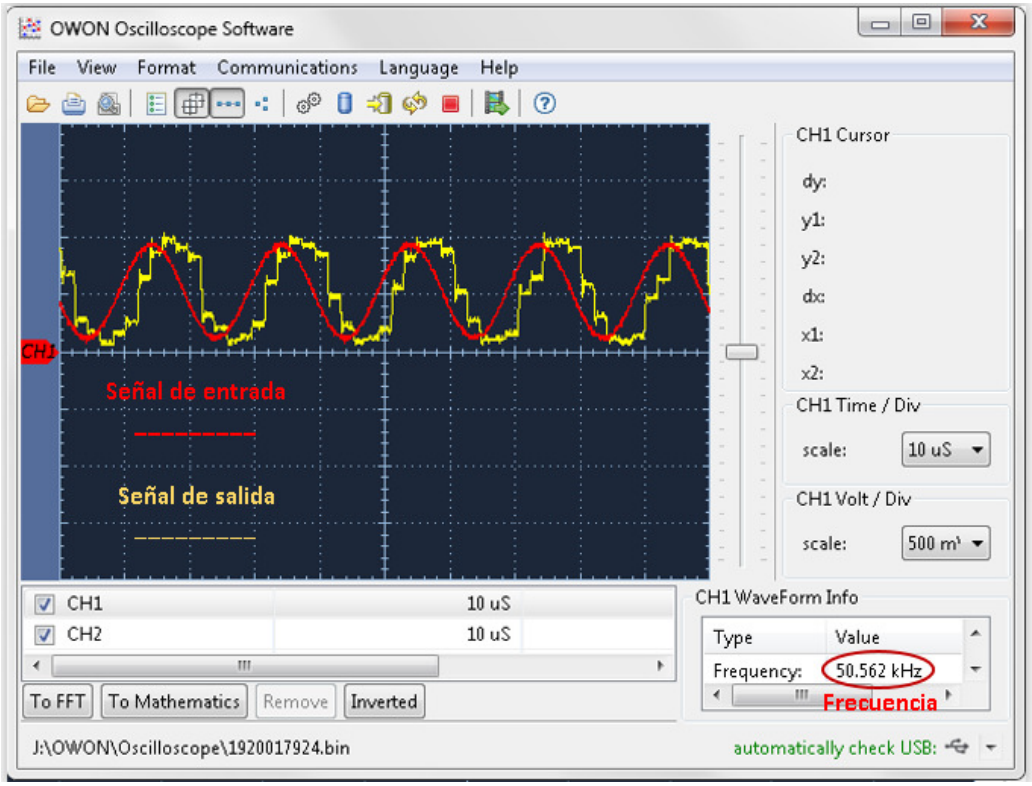


Figura 16. Adquisición y reconstrucción de una señal analógica sinusoidal de 50 kHz.

Elaborado por: Diego Jarrín.

4.2.1 Frecuencia de muestreo en los filtros digitales FIR e IIR y número de coeficientes

Cuando se implementa filtros digitales, además de la adquisición y reconstrucción de la señal, se usan algoritmos de procesamiento digital de señales que necesitan realizar operaciones aritméticas.

Experimentalmente se determinó que la máxima frecuencia de adquisición y reconstrucción es de 16 kHz para los filtros digitales FIR y que se puede programar filtros digitales FIR hasta de octavo orden, el cual además debe ser escalado a 8 bits para evitar problemas de desbordamiento en la salida. Con 16 kHz de frecuencia de

muestreo, el microcontrolador AVR XMEGA puede procesar señales en un rango de 0 a 8 kHz, porque al procesar frecuencias mayores a 8 kHz se observó distorsiones y aliasing de las señales; la Figura 17 muestra un filtro digital FIR pasa-altos de octavo orden, la señal de salida se distorsiona al llegar a los 8 kHz y a frecuencias mayores.

Limitaciones en el procesamiento utilizando filtros digitales FIR de octavo orden

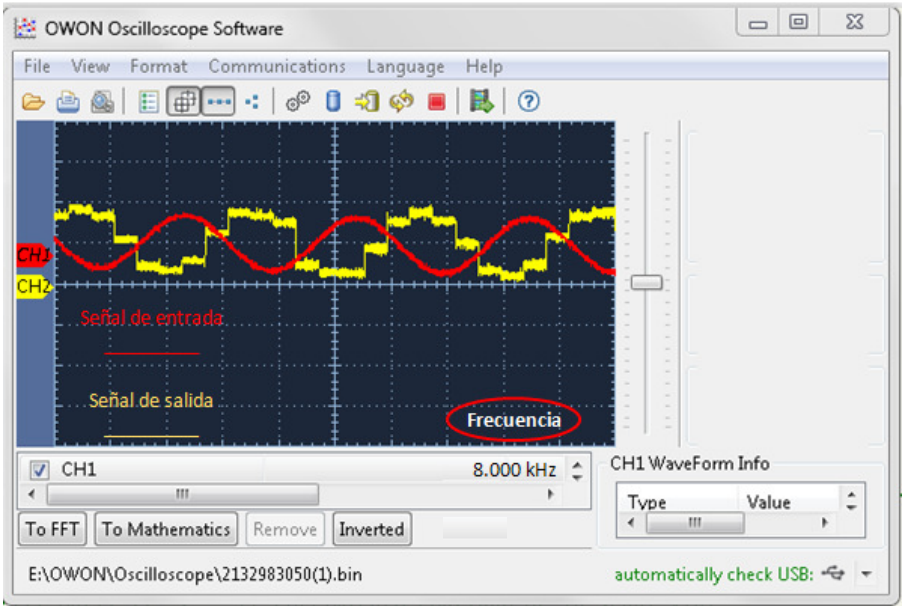


Figura 17. Máxima frecuencia de adquisición y reconstrucción para los filtros IIR.

Elaborado por: Diego Jarrín.

Para los filtros digitales IIR la máxima frecuencia de adquisición y reconstrucción encontrada experimentalmente es de 8 kHz, se pueden procesar señales con frecuencias en un rango entre 0 y 4 kHz; la Figura 18 muestra un filtro digital IIR pasa-altos de segundo orden donde la señal reconstruida empieza a distorsionarse a una frecuencia de 4 kHz, procesar señales con frecuencias mayores a 4 kHz genera salidas distorsionadas y con aliasing. Los filtros digitales IIR se realizaron utilizando la transformada bilineal en base de filtros analógicos de segundo orden, entonces los filtros digitales obtenidos son de segundo orden y poseen seis coeficientes, tres en el numerador y tres en el denominador de su función de transferencia, por tal razón la salida reconstruida presenta menos escalones que la salida de los filtros FIR que utilizo más coeficientes en el procesamiento de las señales.

Limitaciones en el procesamiento de señales con filtros digitales IIR de segundo orden

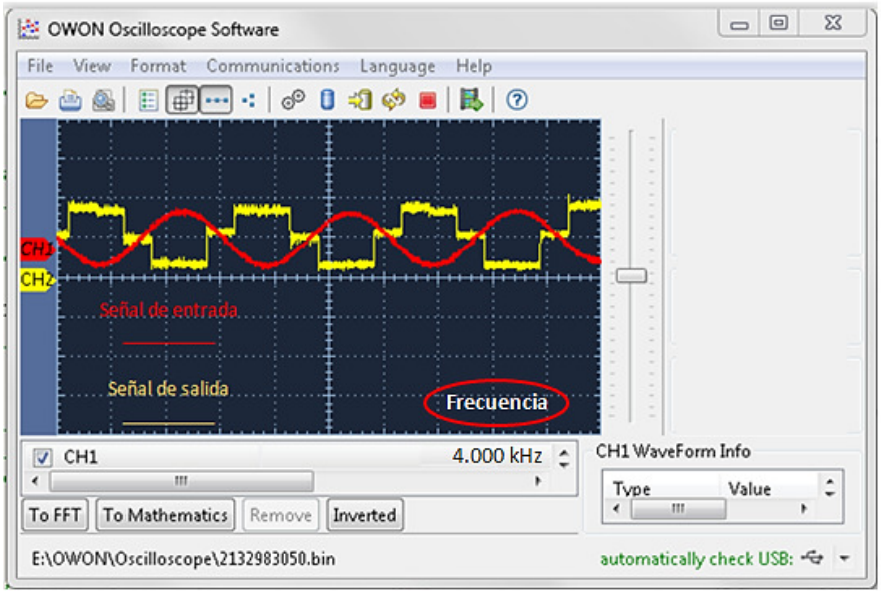


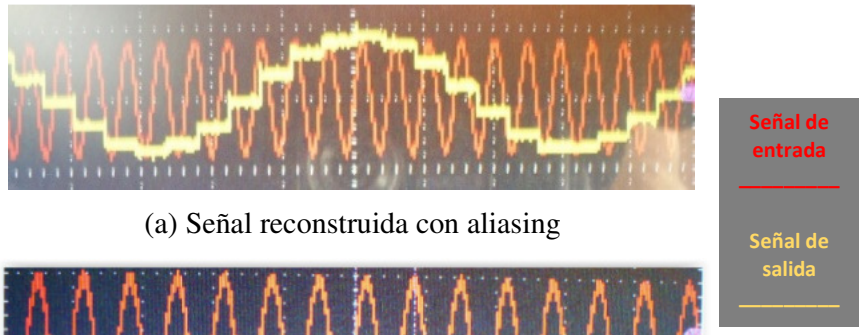
Figura 18. Máxima frecuencia de adquisición y reconstrucción para los filtros IIR.

Elaborado por: Diego Jarrín.

4.3 Resultados obtenidos con el filtro antialiasing y el filtro de reconstrucción

La Figura 19 (a) muestra el aliasing al procesar una señal con una frecuencia mayor a la que el microcontrolador AVR XMEGA permite con una frecuencia de muestreo de 16 kHz, la señal con una frecuencia de 20 kHz que ingresa al ADC se muestra en color rojo, la señal con aliasing que sale del DAC se muestra en color amarillo. La Figura 19 (b) muestra el efecto del filtro antialiasing que bloquea el paso de señales con frecuencias que no cumplen con el teorema del muestreo, dichas señales no se reconstruyen.

Filtro antialiasing para evitar reconstruir señales con aliasing



(a) Señal reconstruida con aliasing

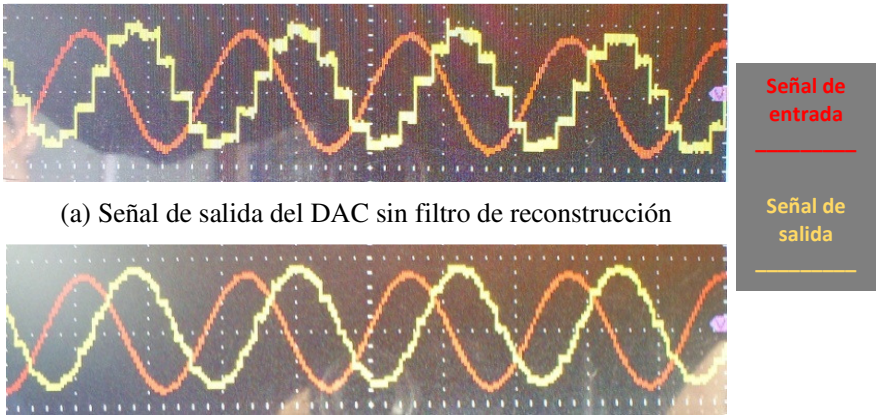
(b) No se reconstruye la señal con aliasing

Figura 19. Ejemplo de aliasing mostrado en el osciloscopio.

Elaborado por: Diego Jarrín.

El microcontrolador AVR XMEGA puede procesar señales con frecuencia de hasta 8 KHz y con frecuencia de muestreo de 16 kHz, se debe suavizar la forma de la señal de la salida del DAC, para lo cual se utiliza el filtro de reconstrucción, la Figura 20 (a) muestra una señal reconstruida directamente por el DAC, la Figura 20 (b) muestra la aplicación del filtro de reconstrucción para suavizar la señal.

Filtro de reconstrucción a la salida del DAC



(a) Señal de salida del DAC sin filtro de reconstrucción

(b) Señal de salida del DAC con filtro de reconstrucción

Figura 20. Suavizado de una señal utilizando el filtro de reconstrucción.

Elaborado por: Diego Jarrín.

4.4 Respuesta en magnitud y fase de los filtros digitales

Se implementó los filtros digitales FIR e IIR a varios rangos de frecuencias, pero se documentó los filtros cuya magnitud y fase de respuesta de frecuencia eran apreciables y permitían realizar mediciones de amplitud y desfase en el osciloscopio. Con los datos teóricos calculados y experimentales medidos se calculó el error relativo y la varianza para la magnitud y la fase de la respuesta de frecuencia de cada tipo de filtro digital. Para cada filtro digital FIR e IIR se indica la frecuencia de muestreo, la frecuencia de corte y los resultados de la respuesta de frecuencia.

4.4.1 Filtro digital pasa-bajos FIR

Con una frecuencia de muestreo de 16 kHz y una frecuencia de corte inferior de 2 kHz se registró los resultados mostrados en la Tabla 5, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz, se puede comparar la magnitud y fase de la respuesta de frecuencia experimental con los valores teóricos y determinar el error relativo y la varianza. Para la magnitud se obtuvo un error relativo promedio de 13.65% con una varianza de 3.82% y para la fase se obtuvo un error relativo promedio de 0.06% con una varianza de 1.70%.

Tabla 5.

Valores experimentales y teóricos para el filtro digital FIR pasa-bajos

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	980	980	1	0.9902	0.0099	-3	-1.9531	-2.0000	0.0235
200	980	980	1	0.9823	0.0180	-13	-9.3136	-22.5000	0.5861
400	980	980	1	0.9587	0.0431	-27	-38.7911	-45.0000	0.1380
600	980	940	0.9592	0.9203	0.0423	-28	-60.4125	-67.5000	0.1050
800	980	860	0.8776	0.8685	0.0105	-30	-85.0682	-90.0000	0.0548
1000	980	800	0.8163	0.805	0.0140	-30	-108.3315	-112.5000	0.0371
1200	980	740	0.7551	0.732	0.0316	-30	-130.0139	-135.0000	0.0369
1400	980	680	0.6939	0.6518	0.0646	-31	-155.2085	-157.5000	0.0145
1600	980	580	0.5918	0.5673	0.0432	-31	-177.3763	-180.0000	0.0146
1800	980	520	0.5306	0.4808	0.1036	-31	-199.965	-202.5000	0.0125
2000	980	440	0.449	0.3952	0.1361	-31	-226.4362	-225.0000	0.0064
2200	980	320	0.3265	0.3128	0.0438	-32	-252.1273	-247.5000	0.0187
2400	980	240	0.2449	0.2357	0.0390	-32	-280.6012	-270.0000	0.0393
2600	980	200	0.2041	0.1659	0.2303	-32	-299.1765	-292.5000	0.0228
2800	980	120	0.1224	0.1048	0.1679	-32	-323.8402	-315.0000	0.0281
3000	980	80	0.0816	0.0533	0.5310	-32	-346.1965	-337.5000	0.0258
3200	980	40	0.0408	0.032	0.2750	-32	-370.2063	-360.0000	0.0284
3400	980	40	0.0408	0.0391	0.0435	-17	-205.4672	-202.5000	0.0147
3600	980	10	0.0102	0.0403	0.7469	-17	-216.9542	-225.0000	0.0358
Promedio			0.5129	0.4914	0.1365	Promedio	-183.5494	-183.6579	0.0654
				Varianza	0.0382			Varianza	0.0170

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital FIR pasa-bajos.

Elaborado por: Diego Jarrín.

La Figura 21 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul en este caso las gráficas son similares.

Comparación de los valores experimentales con los teóricos

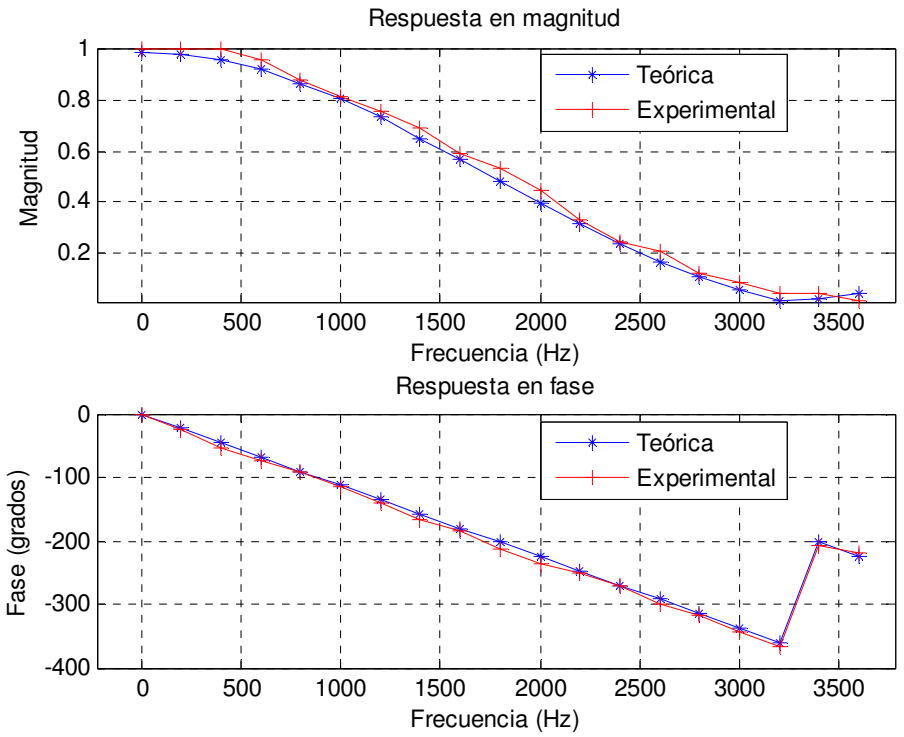


Figura 21. Respuesta de frecuencia para filtro digital FIR pasa-bajos.

Elaborado por: Diego Jarrín.

4.4.2 **Filtro digital pasa-altos FIR**

Con una frecuencia de muestreo de 16 kHz y una frecuencia de corte superior de 2 kHz se registró los resultados mostrados en la Tabla 6, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz, se puede comparar la magnitud y fase de la respuesta de frecuencia experimental con los valores teóricos y determinar el error relativo y la varianza que en este caso para la magnitud se obtuvo un error relativo promedio de 15.89% con una varianza de 5.01% y para la fase se obtuvo un error relativo promedio de 6.26% con una varianza de 0.21%.

Tabla 6.

Valores experimentales y teóricos para el filtro digital FIR pasa-altos

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	940	10	0.0106	0.1719	0.9383	250	180.335	180.0000	0.0019
200	940	160	0.1702	0.1624	0.0480	204	146.7604	157.5000	0.0682
400	940	160	0.1702	0.1344	0.2664	85	122.911	135.0000	0.0895
600	940	120	0.1277	0.0889	0.4364	48	104.0556	112.5000	0.0751
800	940	60	0.0638	0.0473	0.3488	26	73.492	90.0000	0.1834
1000	940	40	0.0426	0.0481	0.1143	68	245.0553	247.5000	0.0099
1200	940	150	0.1596	0.1348	0.1840	49	210.1138	225.0000	0.0662
1400	940	260	0.2766	0.2299	0.2031	38	189.6136	202.5000	0.0636
1600	940	340	0.3617	0.3303	0.0951	30	174.4236	180.0000	0.0310
1800	940	400	0.4255	0.4329	0.0171	25	159.6546	157.5000	0.0137
2000	940	500	0.5319	0.5346	0.0051	19	136.1757	135.0000	0.0087
2200	940	560	0.5957	0.6324	0.0580	15	120.4273	112.5000	0.0705
2400	940	640	0.6809	0.7238	0.0593	11	99.1249	90.0000	0.1014
2600	940	720	0.766	0.8066	0.0503	7	64.5977	67.5000	0.0430
2800	940	800	0.8511	0.8792	0.0320	4	42.251	45.0000	0.0611
3000	940	870	0.9255	0.9402	0.0156	2	19.7245	22.5000	0.1234
3200	940	900	0.9574	0.9892	0.0321	1	5.9965	5.8956	0.0171
3400	940	920	0.9787	1.026	0.0461	-2	-21.2359	-22.5000	0.0562
3600	940	920	0.9787	1.0511	0.0689	-3	-40.2854	-45.0000	0.1048
Promedio			0.4776	0.4928	0.1589	Promedio	107.0101	110.4419	0.0626
				Varianza	0.0501			Varianza	0.0021

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital FIR pasa-altos.

Elaborado por: Diego Jarrín.

La Figura 22 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. La magnitud y la fase de la respuesta de frecuencia son similares entre datos teóricos y experimentales.

Comparación de los valores experimentales con los teóricos

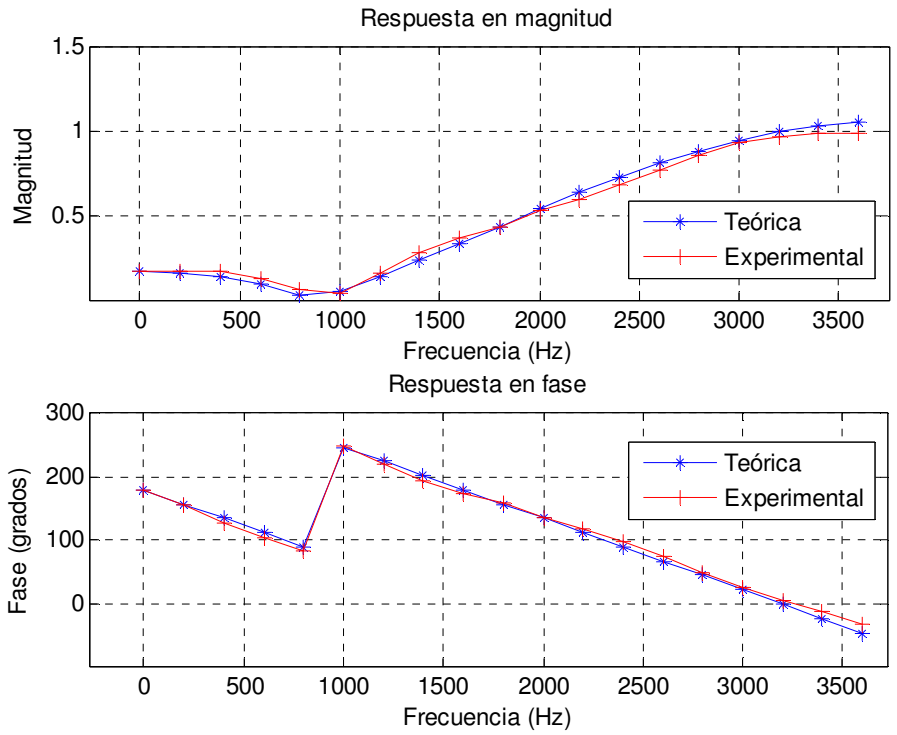


Figura 22. Respuesta de frecuencia para filtro digital FIR pasa-altos.

Elaborado por: Diego Jarrín.

4.4.3 **Filtro digital pasa-banda FIR**

Con una frecuencia de muestreo de 16 kHz y una frecuencia de corte inferior de 2 kHz y una frecuencia de corte superior de 3 kHz se registró los resultados mostrados en la Tabla 7, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz, se puede comparar la magnitud y fase de la respuesta de frecuencia experimental con los valores teóricos y determinar el error relativo y la varianza que en este caso son: la magnitud se obtuvo un error relativo promedio de 11.93% con una varianza de 5.24% y para la fase se obtuvo un error relativo promedio de 4.14% con una varianza de 0.26%.

Tabla 7.

Valores experimentales y teóricos para el filtro digital FIR pasa-banda

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	980	10	0.0102	0.2188	0.9534	253	182.33	180.0000	0.0129
200	980	200	0.2041	0.2039	0.0010	219	157.8322	157.5000	0.0021
400	980	180	0.1837	0.1605	0.1445	91	130.7011	135.0000	0.0318
600	980	100	0.102	0.0919	0.1099	51	110.7481	112.5000	0.0156
800	980	20	0.002	0.0033	0.3939	30	86.7766	90.0000	0.0358
1000	980	128	0.1306	0.0986	0.3245	67	240.9922	247.5000	0.0263
1200	980	192	0.1959	0.2063	0.0504	53	227.0494	225.0000	0.0091
1400	980	312	0.3184	0.3117	0.0215	40	201.2332	202.5000	0.0063
1600	980	416	0.4245	0.4074	0.0420	31	176.7137	180.0000	0.0183
1800	980	472	0.4816	0.4867	0.0105	24	158.6838	157.5000	0.0075
2000	980	540	0.551	0.5442	0.0125	19	133.8485	135.0000	0.0085
2200	980	580	0.5918	0.5767	0.0262	14	109.6841	112.5000	0.0250
2400	980	590	0.602	0.5827	0.0331	10	84.2763	90.0000	0.0636
2600	980	560	0.5714	0.5628	0.0153	7	61.8383	67.5000	0.0839
2800	980	520	0.5306	0.5196	0.0212	4	38.8052	45.0000	0.1377
3000	980	450	0.4592	0.4573	0.0042	2	22.1417	22.5000	0.0159
3200	980	380	0.3878	0.3813	0.0170	0	-1.3545	-1.3698	0.0112
3400	980	300	0.3061	0.2976	0.0286	-2	-24.3389	-22.5000	0.0817
3600	980	220	0.2245	0.2125	0.0565	-4	-53.723	-45.0000	0.1938
Promedio			0.3304	0.3328	0.1193	Promedio	107.5915	110.0595	0.0414
				Varianza	0.0524			Varianza	0.0026

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital FIR pasa-banda.

Elaborado por: Diego Jarrín.

La Figura 23 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. En este caso las gráficas de la respuesta de frecuencia en magnitud y fase son similares.

Comparación de los valores experimentales con los teóricos

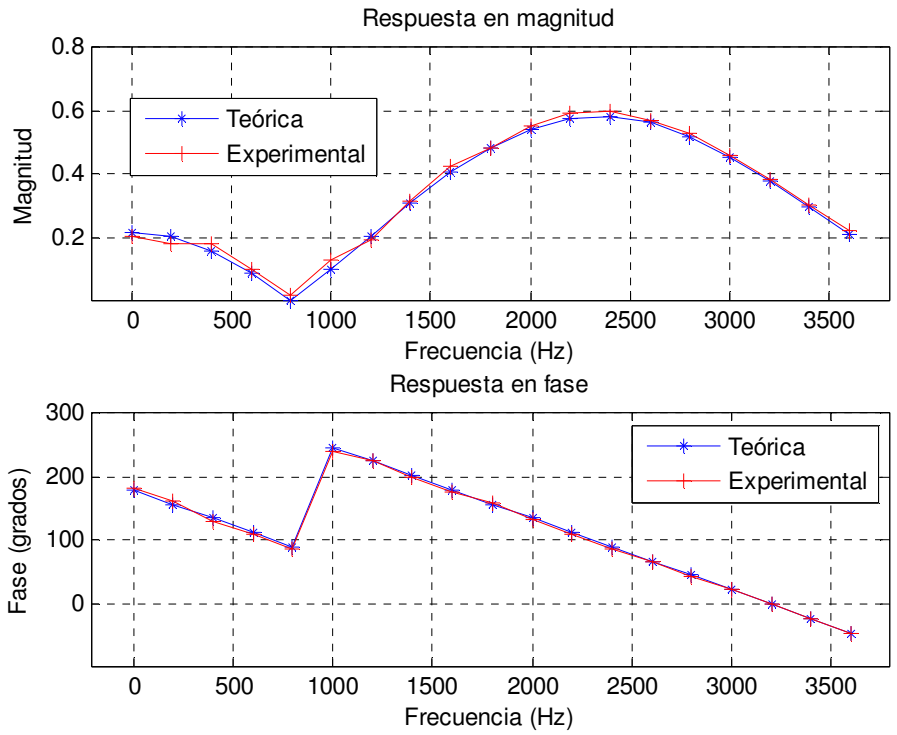


Figura 23. Respuesta de frecuencia para filtro digital FIR pasa-banda.

Elaborado por: Diego Jarrín.

4.4.4 **Filtro digital rechaza-banda FIR**

Con una frecuencia de muestreo de 16 kHz y una frecuencia de corte inferior de 2 kHz y una frecuencia de corte superior de 3 kHz se registró los resultados mostrados en la Tabla 8, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz. Para la magnitud se obtuvo un error relativo promedio de 3.31% con una varianza de 0.04% y para la fase se obtuvo un error relativo promedio de 2.58% con una varianza de 0.03%.

Tabla 8.

Valores experimentales y teóricos para el filtro digital FIR rechaza-banda

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	980	980	1	0.9883	0.0118	2	1.7372	1.6532	0.0508
200	980	960	0.9796	0.9765	0.0032	-32	-23.1275	-22.5000	0.0279
400	980	940	0.9592	0.9419	0.0184	-30	-42.5429	-45.0000	0.0546
600	980	900	0.9184	0.8872	0.0352	-30	-63.9398	-67.5000	0.0527
800	980	830	0.8469	0.8165	0.0372	-33	-94.6506	-90.0000	0.0517
1000	980	760	0.7755	0.7353	0.0547	-30	-109.7937	-112.5000	0.0241
1200	980	680	0.6939	0.6494	0.0685	-32	-136.1358	-135.0000	0.0084
1400	980	580	0.5918	0.5653	0.0469	-32	-160.966	-157.5000	0.0220
1600	980	500	0.5102	0.489	0.0434	-32	-182.4006	-180.0000	0.0133
1800	980	420	0.4286	0.4258	0.0066	-31	-200.2129	-202.5000	0.0113
2000	980	360	0.3673	0.3799	0.0332	-33	-234.73	-225.0000	0.0432
2200	980	350	0.3571	0.3541	0.0085	-33	-258.3209	-247.5000	0.0437
2400	980	360	0.3673	0.3494	0.0512	-32	-277.6757	-270.0000	0.0284
2600	980	380	0.3878	0.3654	0.0613	-32	-296.9752	-292.5000	0.0153
2800	980	400	0.4082	0.4	0.0205	-31	-316.6244	-315.0000	0.0052
3000	980	460	0.4694	0.45	0.0431	-32	-341.3692	-337.5000	0.0115
3200	980	500	0.5102	0.5111	0.0018	-31	-359.8213	-360.0000	0.0005
3400	980	550	0.5612	0.5784	0.0297	-31	-379.0722	-382.5000	0.0090
3600	980	600	0.6122	0.6469	0.0536	-31	-398.0757	-405.0000	0.0171
Promedio			0.6181	0.6058	0.0331	Promedio	-203.9314	-202.4130	0.0258
				Varianza	0.0004			Varianza	0.0003

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital FIR rechaza-banda.

Elaborado por: Diego Jarrín.

La Figura 24 muestra la magnitud y fase de la respuesta de frecuencia graficada a partir de los datos de la Tabla 8. Los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. La magnitud de la respuesta de frecuencia tiene una mayor ganancia en los datos experimentales, las fases de la respuesta de frecuencia son similares.

Comparación de los valores experimentales con los teóricos

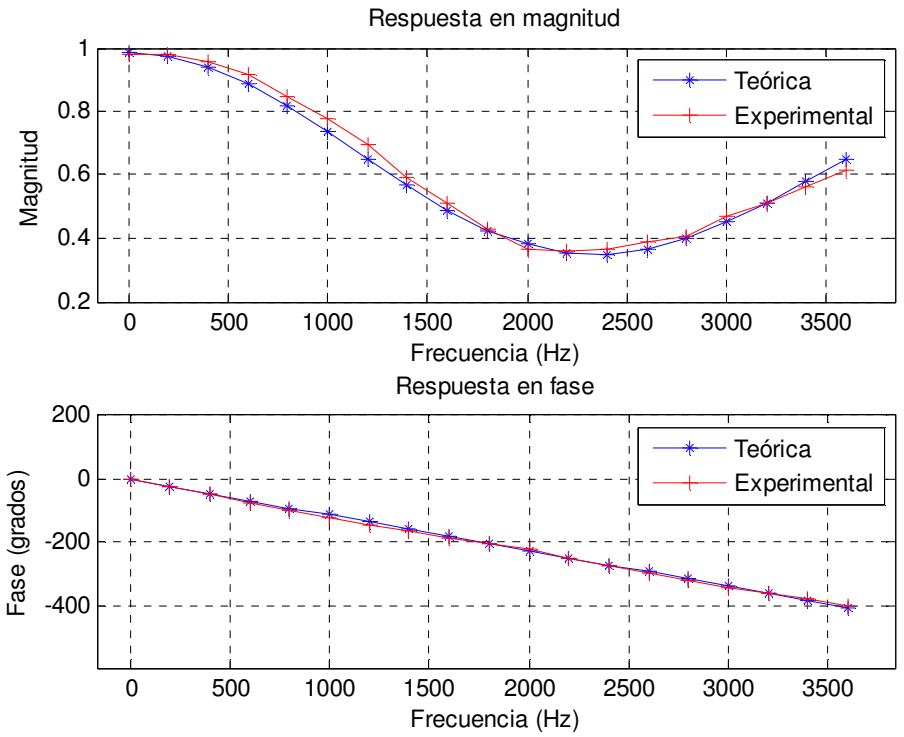


Figura 24. Respuesta de frecuencia para filtro digital FIR rechaza-banda.

Elaborado por: Diego Jarrín.

4.4.5 **Filtro digital pasa-bajos IIR**

Con una frecuencia de muestreo de 8 kHz y una frecuencia de corte inferior de 1 kHz se registró los resultados mostrados en la Tabla 9, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz. Para la magnitud se obtuvo un error relativo promedio de 17.49% con una varianza de 6.40% y para la fase se obtuvo un error relativo promedio de 2.85% con una varianza de 0.04%.

Tabla 9.

Valores experimentales y teóricos para el filtro digital IIR pasa-bajos

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	920	920	1	1	0.0000	0	0.002	0.0020	0.0000
200	920	920	1	0.9993	0.0007	-22	-15.6419	-15.5768	0.0042
400	920	900	0.9783	0.9895	0.0113	-22	-31.2267	-32.3486	0.0347
600	920	840	0.913	0.9479	0.0368	-22	-47.4978	-50.9874	0.0684
800	920	740	0.8043	0.8517	0.0557	-23	-66.1421	-70.8755	0.0668
1000	920	600	0.6522	0.7071	0.0776	-23	-84.5067	-90.0000	0.0610
1200	920	460	0.5	0.5513	0.0931	-23	-101.3001	-106.4350	0.0482
1400	920	340	0.3696	0.4156	0.1107	-23	-115.0031	-119.6034	0.0385
1600	920	260	0.2826	0.3091	0.0857	-22	-127.284	-129.9345	0.0204
1800	920	180	0.1957	0.229	0.1454	-21	-134.7835	-138.1142	0.0241
2000	920	160	0.1739	0.1691	0.0284	-20	-145.2167	-144.7356	0.0033
2200	920	140	0.1522	0.1242	0.2254	-19	-153.0343	-150.2355	0.0186
2400	920	120	0.1304	0.0902	0.4457	-18	-158.9449	-154.9212	0.0260
2600	920	100	0.1087	0.0643	0.6905	-17	-163.4485	-159.0086	0.0279
2800	920	80	0.087	0.0445	0.9551	-17	-166.8976	-162.6518	0.0261
3000	920	60	0.0652	0.0594	0.0976	-16	-169.543	-165.9638	0.0216
3200	920	50	0.0543	0.0581	0.0654	-15	-172.6533	-169.0296	0.0214
3400	920	40	0.0435	0.0399	0.0902	-14	-175.2748	-171.9158	0.0195
3600	920	20	0.0217	0.0243	0.1070	-14	-176.42	-174.6765	0.0100
Promedio			0.3965	0.4039	0.1749	Promedio	-116.0430	-116.1585	0.0285
				Varianza	0.0640			Varianza	0.0004

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital IIR pasa-bajos.

Elaborado por: Diego Jarrín.

La Figura 25 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. La magnitud de la respuesta de frecuencia en los datos experimentales tiene una mayor atenuación con respecto a los experimentales entre 500 Hz y 1500 Hz, y tiene menor atenuación entre 2000 HZ y 3500 Hz. Son similares las dos curvas de la fase de la respuesta de frecuencia.

Comparación de los valores experimentales con los teóricos

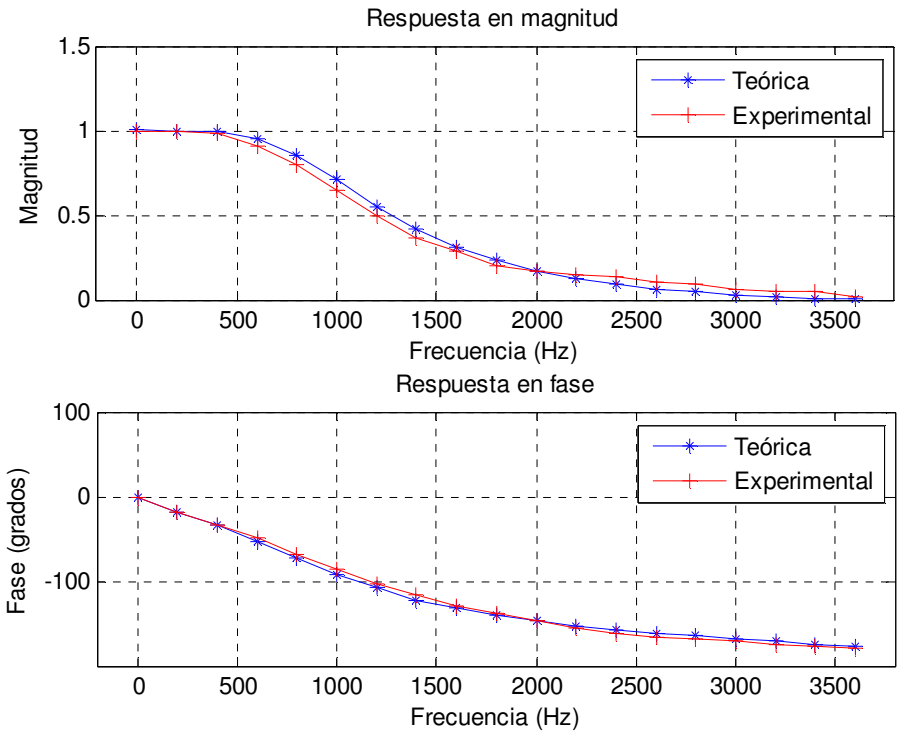


Figura 25. Respuesta de frecuencia para filtro digital IIR pasa-bajos.

Elaborado por: Diego Jarrín.

4.4.6 **Filtro digital pasa-altos IIR**

Con una frecuencia de muestreo de 8 kHz y una frecuencia de corte superior de 0.6 kHz se registró los resultados mostrados en la Tabla 10, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 1800 Hz. Para la magnitud se obtuvo un error relativo promedio de 13.26% con una varianza de 11.26% y para la fase se obtuvo un error relativo promedio de 5.63% con una varianza de 0.30%.

Tabla 10.

Valores experimentales y teóricos para el filtro digital IIR pasa-altos

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	900	10	0.0111	0.0101	0.0990	448	161.4202	160.5278	0.0056
100	900	60	0.0667	0.0268	1.4888	452	162.6302	166.6228	0.0240
200	900	120	0.1333	0.1068	0.2481	208	149.9065	152.5517	0.0173
300	900	260	0.2889	0.2362	0.2231	122	132.08	137.3531	0.0384
400	900	400	0.4444	0.3991	0.1135	81	116.6552	121.1882	0.0374
500	900	520	0.5778	0.5659	0.0210	58	103.7955	104.9807	0.0113
600	900	620	0.6889	0.7071	0.0257	43	91.8207	90.0000	0.0202
700	900	720	0.8	0.8098	0.0121	31	78.0911	77.1161	0.0126
800	900	780	0.8667	0.8777	0.0125	23	67.6153	66.5142	0.0166
900	900	820	0.9111	0.9208	0.0105	18	58.909	57.9360	0.0168
1000	900	840	0.9333	0.9479	0.0154	15	52.4514	50.9874	0.0287
1100	900	860	0.9556	0.9651	0.0098	12	46.2541	45.3005	0.0211
1200	900	860	0.9556	0.9762	0.0211	10	42.6433	40.5802	0.0508
1300	900	860	0.9556	0.9835	0.0284	8	39.3965	36.6036	0.0763
1400	900	860	0.9556	0.9884	0.0332	7	36.4876	33.2051	0.0989
1500	900	860	0.9556	0.9918	0.0365	6	33.8786	30.2616	0.1195
1600	900	860	0.9556	0.9941	0.0387	5	31.5304	27.6812	0.1391
1700	900	860	0.9556	0.9957	0.0403	5	29.4068	25.3941	0.1580
1800	900	860	0.9556	0.9969	0.0414	4	27.476	23.3467	0.1769
Promedio			0.7035	0.7105	0.1326	Promedio	76.9710	76.2185	0.0563
				Varianza	0.1126			Varianza	0.0030

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital IIR pasa-altos.

Elaborado por: Diego Jarrín.

La Figura 26 muestra la magnitud y fase de la respuesta de frecuencia graficada, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. En la magnitud de la respuesta de frecuencia los valores experimentales tienen menor atenuación que los datos teóricos entre 0 Hz y 600 Hz, y se tiene una mayor atenuación en los datos experimentales con respecto a los datos teóricos para frecuencias mayores a 600 Hz. La fase de la respuesta de frecuencia entre los datos teóricos y experimentales es similar.

Comparación de los valores experimentales con los teóricos

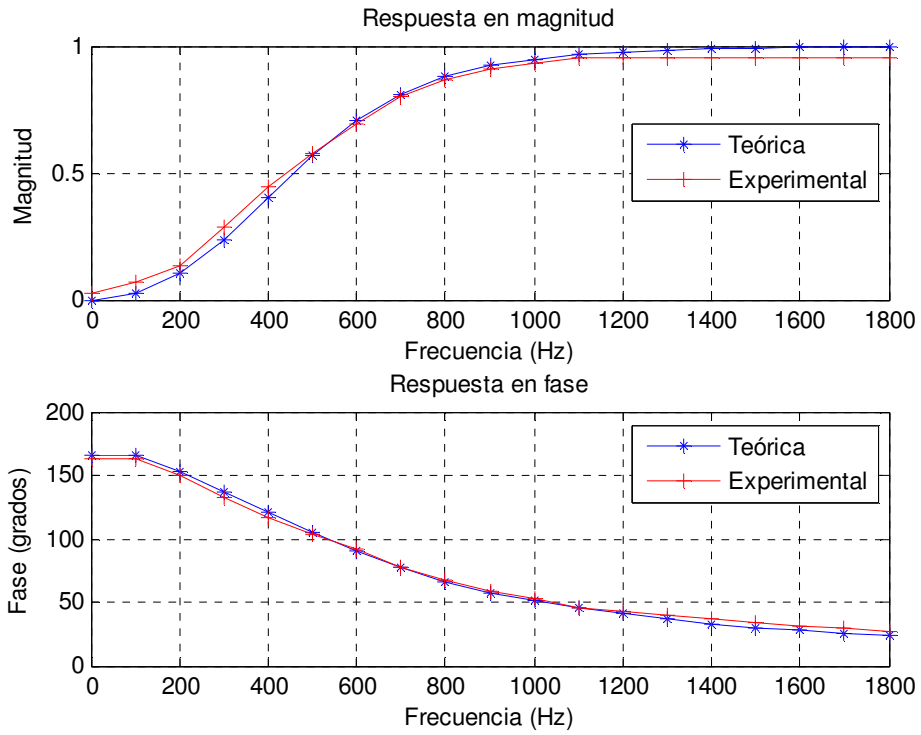


Figura 26. Respuesta de frecuencia para filtro digital IIR pasa-altos.

Elaborado por: Diego Jarrín.

4.4.7 **Filtro digital pasa-banda IIR**

Con una frecuencia de muestreo de 8 kHz y una frecuencia de corte inferior de 0.9 kHz y una frecuencia de corte superior de 1.4 kHz se registró los resultados mostrados en la Tabla 11, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 3600 Hz. Para la magnitud se obtuvo un error relativo promedio de 18.11% con una varianza de 3.66% y para la fase se obtuvo un error relativo promedio de 11.28% con una varianza de 6.05%.

Tabla 11.

Valores experimentales y teóricos para el filtro digital IIR pasa-banda

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	920	10	0.0108	0.0160	0.3250	123	88.5463	85.2145	0.0391
200	920	10	0.0109	0.0372	0.7070	126	90.4988	87.8672	0.0299
400	920	100	0.1087	0.0758	0.4340	57	82.3637	85.6517	0.0384
600	920	140	0.1522	0.1174	0.2964	37	79.7783	83.2575	0.0418
800	920	180	0.1957	0.1641	0.1926	27	77.3931	80.5577	0.0393
1000	920	220	0.2391	0.2187	0.0933	21	75.3289	77.3650	0.0263
1200	920	300	0.3261	0.2861	0.1398	16	69.3756	73.3750	0.0545
1400	920	340	0.3696	0.3738	0.0112	14	68.4711	68.0477	0.0062
1600	920	420	0.4565	0.4949	0.0776	11	64.1753	60.3399	0.0636
1800	920	600	0.6522	0.6677	0.0232	8	49.6603	48.1098	0.0322
2000	920	740	0.8043	0.888	0.0943	5	35.7476	27.3798	0.3056
2200	920	880	0.9565	0.9978	0.0414	0	0.3454	-3.7863	1.0912
2400	920	720	0.7826	0.832	0.0594	-3	-28.7643	-33.6999	0.1465
2600	920	540	0.587	0.6043	0.0286	-5	-51.0935	-52.8224	0.0327
2800	920	380	0.413	0.4352	0.0510	-6	-61.9815	-64.2018	0.0346
3000	920	320	0.3478	0.3166	0.0985	-7	-74.6628	-71.5427	0.0436
3200	920	220	0.2391	0.229	0.0441	-7	-77.7717	-76.7618	0.0132
3400	920	180	0.1957	0.1598	0.2247	-7	-84.3925	-80.8039	0.0444
3600	920	140	0.1522	0.1015	0.4995	-7	-89.2411	-84.1732	0.0602
Promedio			0.3684	0.3693	0.1811	Promedio	16.5146	16.2828	0.1128
				Varianza	0.0366			Varianza	0.0605

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital IIR pasa-banda.

Elaborado por: Diego Jarrín.

La Figura 27 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. Para la magnitud de la respuesta de frecuencia los datos experimentales tienen menor atenuación en los rangos entre 0 Hz a 1500 Hz y entre 3000 a 3600 Hz, la respuesta de fase entre los datos experimentales y teóricos es prácticamente similar.

Comparación de la respuesta experimental con la teórica

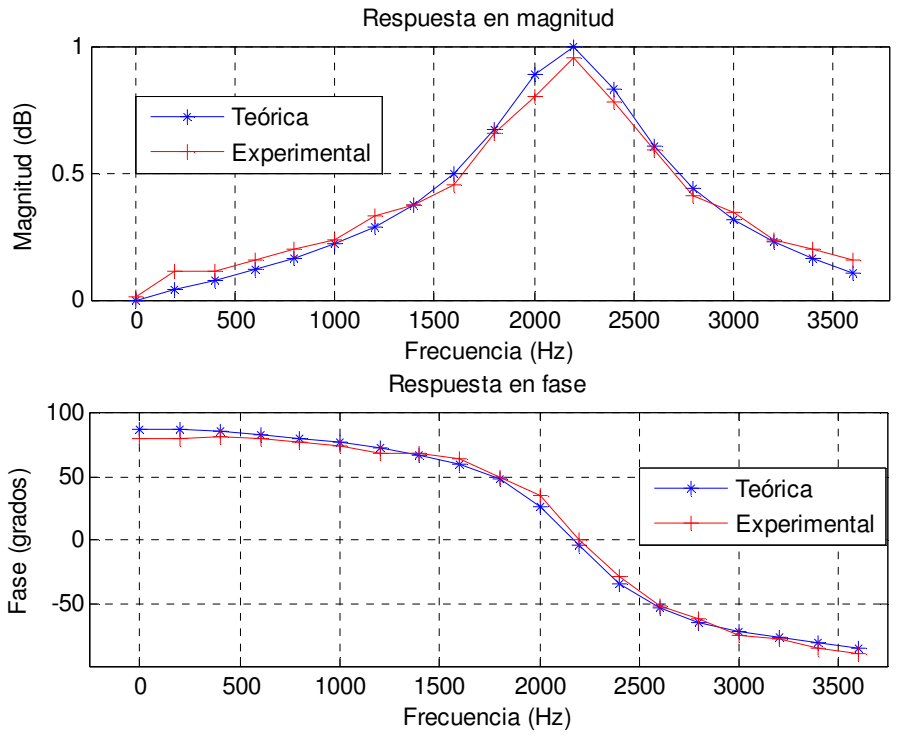


Figura 27. Respuesta de frecuencia para filtro digital IIR pasa-banda.

Elaborado por: Diego Jarrín.

4.4.8 **Filtro digital rechaza-banda IIR**

Con una frecuencia de muestreo de 8 kHz y una frecuencia de corte inferior de 1 kHz y una frecuencia de corte superior de 1.5 kHz se registró los resultados mostrados en la Tabla 12, se registran la amplitud y el desfase de la señal de salida respecto a la señal de entrada, en un rango de frecuencias de 0 a 2700 Hz. Para la magnitud se obtuvo un error relativo promedio de 3.29% con una varianza de 0.10% y para la fase se obtuvo un error relativo promedio de 8.22% con una varianza de 0.23%.

Tabla 12.

Valores experimentales y teóricos para el filtro digital IIR rechaza-banda

Frecuencia	Amplitud (mV)		Magnitud		Error	Desfase	Fase		Error
F (Hz)	Entrada	Salida	Hexp	Hteor	Erel H	d (us)	ωexp °	ωteor °	Erel ω
0	920	920	1	1	0.0000	0	0.0009	0.0010	0.1000
150	920	920	1	0.9985	0.0015	-6	-3.3079	-3.1563	0.0480
300	920	920	1	0.9936	0.0064	-7	-7.1042	-6.4678	0.0984
450	920	920	1	0.9845	0.0157	-7	-11.6711	-10.1164	0.1537
600	920	880	0.9565	0.9688	0.0127	-6	-13.1167	-14.3462	0.0857
750	920	840	0.913	0.9425	0.0313	-6	-16.571	-19.5194	0.1510
900	920	820	0.8913	0.8971	0.0065	-8	-25.6296	-26.2141	0.0223
1050	920	720	0.7826	0.8153	0.0401	-8	-32.1076	-35.3798	0.0925
1200	920	560	0.6087	0.6629	0.0818	-10	-43.0643	-48.4813	0.1117
1350	920	340	0.3696	0.3904	0.0533	-13	-64.9429	-67.0228	0.0310
1500	920	10	0.0109	0.01	0.0900	25	133.913	135.0000	0.0081
1650	920	380	0.413	0.3743	0.1034	11	64.144	68.0199	0.0570
1800	920	580	0.6304	0.6247	0.0091	8	50.7653	51.3428	0.0112
1950	920	660	0.7174	0.7683	0.0663	6	44.8891	39.8012	0.1278
2100	920	760	0.8261	0.8503	0.0285	5	34.1774	31.7605	0.0761
2250	920	820	0.8913	0.8994	0.0090	3	26.7312	25.9248	0.0311
2400	920	880	0.9565	0.9305	0.0279	2	18.8865	21.4914	0.1212
2550	920	900	0.9783	0.9512	0.0285	2	15.2702	17.9797	0.1507
2700	920	900	0.9783	0.9655	0.0133	1	13.8189	15.0957	0.0846
Promedio			0.7855	0.7909	0.0329	Promedio	9.7411	9.2480	0.0822
				Varianza	0.0010			Varianza	0.0023

Nota: Comparación de valores experimentales y teóricos para calcular el error relativo y la varianza en el filtro digital IIR rechaza-banda.

Elaborado por: Diego Jarrín.

La Figura 28 muestra la magnitud y fase de la respuesta de frecuencia, los valores experimentales están graficados en color rojo y los valores teóricos están graficados en color azul. En este caso la magnitud y la fase de la respuesta de frecuencia del filtro son similares.

Comparación de los valores experimentales con los teóricos

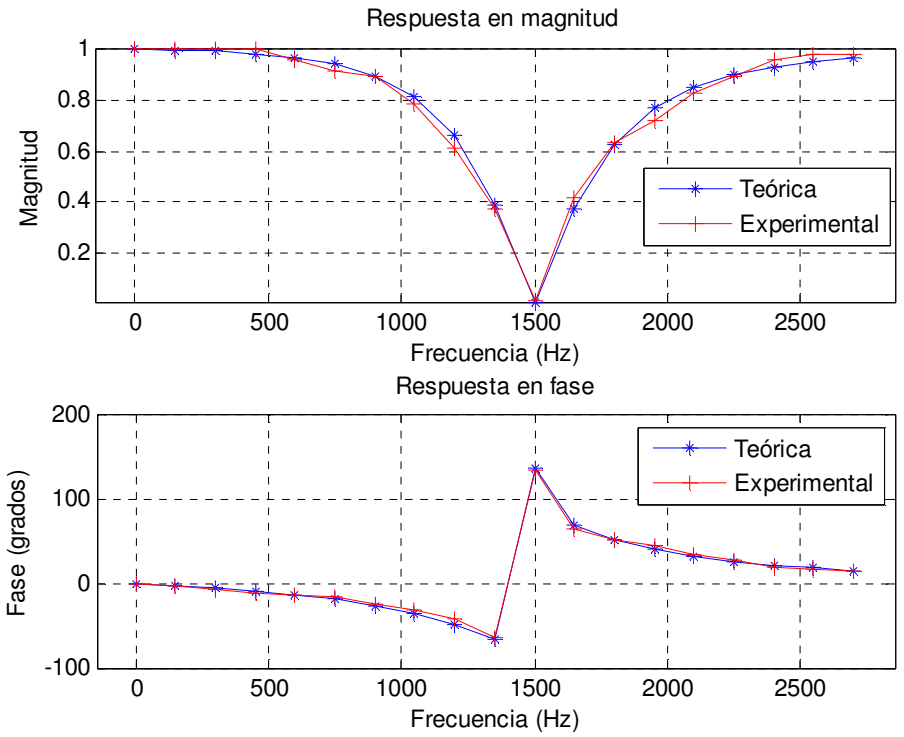


Figura 28. Respuesta de frecuencia para filtro digital IIR rechaza-banda.

Elaborado por: Diego Jarrín.

Conclusiones

Se diseñó e implementó filtros digitales FIR e IIR, pasa-bajos, pasa-altos, pasa-banda y rechaza-banda en el microcontrolador XMEGA de Atmel, la programación se realizó con Mikro C para AVR de MikroElektronika, se utilizó MATLAB para calcular los coeficientes, la magnitud y la fase de la respuesta de frecuencia de cada filtro. Se elaboró una placa de circuito impreso para los filtros antialiasing y de reconstrucción necesarios para analizar las señales a procesar.

Para el circuito de los filtros antialiasing y de reconstrucción se debe utilizar filtros activos, los cuales se implementaron con el circuito integrado TL084 que incorpora cuatro amplificadores operacionales con tecnología JFET, lo cual le permite ser menos vulnerable al ruido; para alimentación de los filtros, debe utilizarse una fuente bipolar con voltajes mayores al de las señales a filtrar, en este caso se utilizó una fuente de laboratorio de ±15 V. Para implementación de los filtros digitales FIR es necesario realizar el escalamiento de los coeficientes, para así evitar desbordamiento en la salida del microcontrolador.

La máxima frecuencia de adquisición y reconstrucción con el microcontrolador AVR XMEGA es de 200 kHz, se puede reconstruir una señal discreta de forma de onda cuadrada de una frecuencia máxima de 100 KHz y una señal analógica de forma de onda seno de una frecuencia máxima de 50 kHz. La máxima frecuencia de muestreo para los filtros digitales FIR de octavo orden es de 16 kHz, para los filtros digitales IIR de segundo orden la máxima frecuencia de muestreo es 8 kHz. Estos resultados se deben a que el microcontrolador XMEGA limita sus operaciones matemáticas a valores con tamaño de memoria de 8 bits, lo cual no le permite procesar señales con frecuencias más altas.

Los filtros digitales FIR presentan los siguientes errores relativos y varianzas de sus valores experimentales de magnitud y fase de la respuesta de frecuencia respecto a los valores teóricos obtenidos en la simulación en MATLAB: los filtros digitales FIR pasa-bajos implementados tienen un error relativo de 13.65% con una varianza de 3.82% para la magnitud de la respuesta de frecuencia y un error relativo de 0.06% con una varianza de 1.70% para la fase de la respuesta de frecuencia. Los filtros digitales FIR pasa-altos implementados tienen un error

relativo de 15.89% con una varianza de 5.01% para la magnitud de la respuesta de frecuencia y un error relativo de 6.26% con una varianza de 0.21% para la fase de la respuesta de frecuencia. Los filtros digitales FIR pasa-banda implementados tienen un error relativo de 11.93% con una varianza de 5.24% para la magnitud de la respuesta de frecuencia y un error relativo de 4.14% con una varianza de 0.26% para la fase de la respuesta de frecuencia. Los filtros digitales FIR rechaza-banda implementados tienen un error relativo de 3.31% con una varianza de 0.04% para la magnitud de la respuesta de frecuencia y un error relativo de 2.58% con una varianza de 0.03% para la fase de la respuesta de frecuencia. Los errores obtenidos en los filtros digitales FIR se deben al escalamiento a 8 bits efectuado para evitar desbordamiento en la memoria del microcontrolador.

Los filtros digitales IIR presentan los siguientes errores relativos y varianzas de sus valores experimentales de magnitud y fase de la respuesta de frecuencia respecto a los valores teóricos obtenidos en la simulación en MATLAB: Los filtros digitales IIR pasa-bajo implementados tienen un error relativo de 17.49% con una varianza de 6.40% para la magnitud de la respuesta de frecuencia y un error relativo de 2.85% con una varianza de 0.04% para la fase de la respuesta de frecuencia. Los filtros digitales IIR pasa-altos implementados tienen un error relativo de 13.26% con una varianza de 11.26% para la magnitud de la respuesta de frecuencia y un error relativo de 5.63% con una varianza de 0.30% para la fase de la respuesta de frecuencia. Los filtros digitales IIR pasa-banda implementados tienen un error relativo de 18.11% con una varianza de 3.66% para la magnitud de la respuesta de frecuencia y un error relativo de 11.28% con una varianza de 6.05% para la fase de la respuesta de frecuencia. Los filtros digitales IIR rechaza-banda implementados tienen un error relativo de 3.29% con una varianza de 0.10% para la magnitud de la respuesta de frecuencia y un error relativo de 8.22% con una varianza de 0.23% para la fase de la respuesta de frecuencia. En el caso de los filtros digitales IIR no fue necesario realizar escalamiento de coeficientes a 8 bits, porque la respuesta al impulso analizada en MATLAB no generaba desbordamiento.

Los errores obtenidos para los filtros digitales FIR e IIR, tal como manifiesta (Tan, 2007, págs. 283, 377) se deben al escalamiento y al truncamiento que

realiza el microcontrolador para que los valores de entrada y salida de las señales quepan en los registros del microcontrolador y puedan ser procesados; al no utilizar el microcontrolador AVR XMEGA como instrumento de medición normado, el criterio de ingeniería de tolerar un error del 3% no se cumple, los errores obtenidos no representan malfuncionamiento o mala práctica en el experimento.

Recomendaciones

Mejorar la tarjeta de los filtros analógicos antialiasing y de reconstrucción implementando conectores de audio y trabajar preferiblemente con señales de voz ya que se encuentran en un rango de frecuencias entre 0 y 4000 Hz y se pueden muestrear a 8 y 16 kHz que son las frecuencias óptimas que se determinó para implementar los filtros digitales en el microcontrolador AVR XMEGA.

Utilizar osciloscopios digitales para analizar las señales procesadas, ya que representan las señales en colores lo que permite una fácil identificación y tienen conexión con software para PC, lo que permite guardar información de las señales. El Software disponible libremente en Internet y compatible con los osciloscopios disponibles en los Laboratorios de Electrónica de la Universidad Politécnica Salesiana es Owon Oscilloscope y como Hardware generador de señales se puede utilizar un PC que cuente con el Software gratuito Scope de Zeinits 2015.

Utilizar un microcontrolador más robusto como lo es el SMT32 que posee registros de 32 bits y procesador de punto flotante para evitar problemas de desbordamiento al momento de implementar los filtros digitales FIR e IIR.

Aumentar un capacitor de desacoplo con un valor mínimo de capacitancia de 10µF para filtrar ruido de alta frecuencia en la entrada de alimentación de la tarjeta XMEGA-Ready.

Referencias

Atmel. (2008). AVR XMEGA. Recuperado el 22 de Mayo de 2015, de Atmel: http://www.atmel.com/Images/doc7925.pdf

Atmel. (Abril de 2008). AVR1301: Using the XMEGA DAC. Recuperado el 14 de Septiembre de 2015, de Atmel Application notes: http://www.atmel.com/Images/doc8033.pdf

Atmel. (Julio de 2008). AVR223: Digital Filters on tinyAVR and megaAVR devices. Recuperado el 25 de Agosto de 2015, de Atmel: http://www.atmel.com/Images/doc2527.pdf

Atmel. (2013). ATxmega128A1 / ATxmega64A1 Preliminary. Recuperado el 24 de Mayo de 2015, de Atmel: http://www.atmel.com/Images/Atmel-8067-8-and-16-bit-AVR-Microcontrollers-ATxmega64A1-ATxmega128A1_Datasheet.pdf

Atmel. (Junio de 2013). AVR1300: Using the Atmel AVR XMEGA ADC. Recuperado el 25 de Agosto de 2015, de Atmel Application notes: http://www.atmel.com/Images/Atmel-8032-Using-the-Atmel-AVR-XMEGA-ADC_Application-Note_AVR1300.pdf

Atmel. (2016). Atmel Maker Faire. Recuperado el 27 de 04 de 2016, de Makerplace: http://www.atmel.com/atmel-makes/

Chicaiza, N., & Reyes, F. (Junio de 2013). Diseño y construcción de un módulo de entrenamiento basado en el microcontrolador XMEGA de ATMEL para los laboratorios de sistemas microprocesadores de la Universidad Politécnica Salesiana. Recuperado el 14 de Mayo de 2015, de dspace ups: http://dspace.ups.edu.ec/bitstream/123456789/5166/1/UPS-ST000981.pdf

Clavijo, J. (2011). Diseño y simulación de sistemas microcontrolados en lenguaje C. Recuperado el 14 de Junio de 2015, de mikroe: http://learn.mikroe.com/ebooks/disenoysimulaciondesistemasmicrocontroladosenlenguaje/

MikroElektronika. (2012). Ready for XMEGA User Manual. Recuperado el 18 de Mayo de 2015, de MikroElektronika: http://www.mikroe.com/downloads/get/1706/ready_xmega_manual_v101.pdf

Proakis, J., & Manolakis, D. (2007). Digital Signal Processing. Toronto: Pearson Prentice Hall.

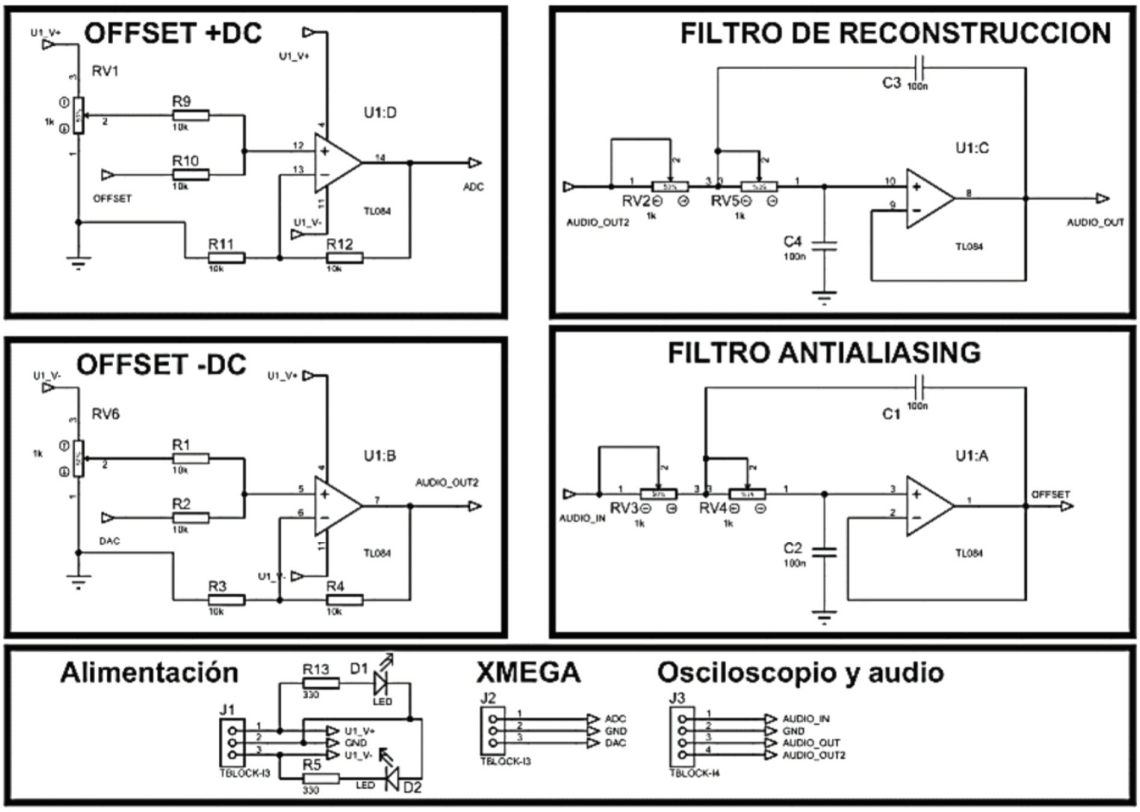
Tan, L. (2007). Digital Signal Processing: Fundamentals and Applications. Decatur: Academic Press.

Texas Instruments. (28 de 04 de 2016). Amplificadores de propósito general. Obtenido de Texas Instruments: http://www.ti.com.cn/cn/lit/ds/symlink/tl084.pdf

Anexos

Anexo 1. Esquemático y fotolito de la PCB para filtrado y acondicionamiento de las señales de audio

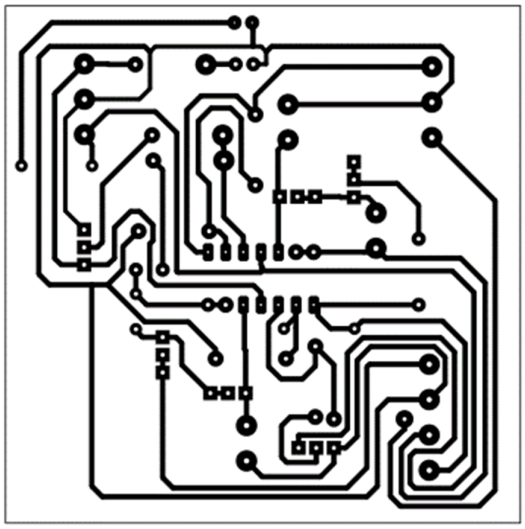
1. Esquemático del circuito:



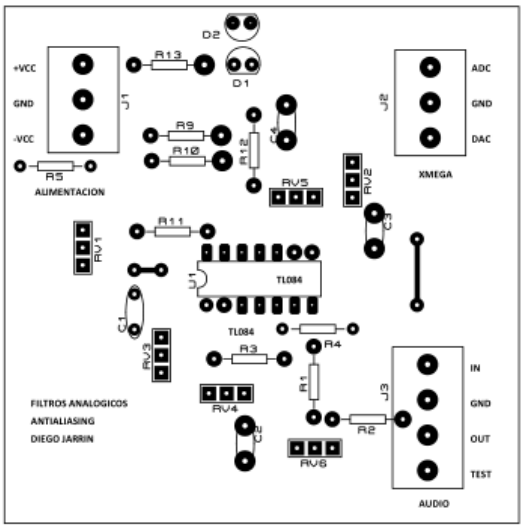
2. Lista de elementos

Cantidad	Elemento	Descripción	Característica
8	R1-R4, R9-R12	Resistor	10kΩ
2	R5, R13	Resistor	330Ω
4	C1-C4	Capacitor	100nF
1	U1	Amplificador operacional	TL084
2	D1, D2	Diodo	LED
2	J1, J2	Bornera	TBLOCK-I3
1	J3	Bornera	TBLOCK-I4
6	RV1-RV6	Potenciómetro de precisión	1kΩ
1	PCB	Baquelita	10x10cm²

3. Fotolito en tamaño real 10x10 cm² y ubicación de los elementos, para ser impresos con una impresora láser.



Nota: imprimir en papel fotográfico.



Nota: imprimir en acetato.

Anexo 2. Códigos de los programas en MATLAB y MikroC para AVR

1. Cálculo de los coeficientes de los filtros digitales FIR en MATLAB

```
% Filtros digitales FIR
% Diego Jarrín
continuar='s';
while continuar=='s'
   clear;
   clc;
   clc
   clear
   disp('Universidad Politécnica Salesiana')
   disp('DSP con XMEGA')
   disp('Filtros digitales FIR')
   disp('Programación: Diego Jarrín')
   %  Pedir datos del filtro pasabajos al usuario
   fs=input('Ingrese la frecuencia de muestreo o sampling: ');
   N=input('Ingrese el número de coeficientes: ')
   %Selección del filtro
   disp('Seleccione un tipo de filtro:')
   disp('1: Pasabajos')
   disp('2: Pasaaltos')
   disp('3: Pasabanda')
   disp('4: Rechazabanda')
   Ftype = input('Opción: ');
   M = (N - 1)/2;
   if Ftype == 1
      disp('Filtro pasa-bajos')
      WnL=input('Ingrese la frecuencia de corte inferior: ');
      WnL=WnL*2*pi/fs;
      hL = sin(WnL * [ - M:1:-1])./([ - M:1:-1] * pi);
      hL(M + 1) = WnL/pi;
      hL(M + 2:1:N) = hL(M:-1:1);
      h(1:N) = hL(1:N);
   end
   if Ftype == 2
      disp('Filtro pasa-altos')
      WnH=input('Ingrese la frecuencia de corte superior: ');
      WnH=WnH*2*pi/fs;
      hH = sin(WnH * [ - M:1:-1])./([ - M:1:-1] * pi);
      hH(M + 1) = WnH/pi;
      hH(M + 2:1:N) = hH(M:-1:1);
      h(1:N) = -hH(1:N);
      h(M + 1) = 1 + h(M + 1);
   end
   if Ftype ==3
      disp('Filtro pasa-banda')
      WnL=input('Ingrese la frecuencia de corte inferior: ');
      WnL=WnL*2*pi/fs;
      WnH=input('Ingrese la frecuencia de corte superior: ');
      WnH=WnH*2*pi/fs;
      hH = sin(WnH * [ - M:1:-1])./([ - M:1:-1] * pi);
      hH(M + 1) = WnH/pi;
      hH(M + 2:1:N) = hH(M:-1:1);
      hL = sin(WnL * [ - M:1:-1])./([ - M:1:-1] * pi);
      hL(M + 1) = WnL/pi;
      hL(M + 2:1:N) = hL(M:-1:1);
      h(1:N) = hH(1:N) - hL(1:N);
   end
   if Ftype == 4
      disp('Filtro rechazabanda')
      WnL=input('Ingrese la frecuencia de corte inferior: ');
      WnL=WnL*2*pi/fs;
      WnH=input('Ingrese la frecuencia de corte superior: ');
      WnH=WnH*2*pi/fs;
      hH = sin(WnH * [ - M:1:-1])./([ - M:1:-1] * pi);
      hH(M + 1) = WnH/pi;
      hH(M + 2:1:N) = hH(M:-1:1);
      hL = sin(WnL * [ - M:1:-1])./([ - M:1:-1] * pi);
      hL(M + 1) = WnL/pi;
      hL(M + 2:1:N) = hL(M:-1:1);
      h(1:N) = hL(1:N) - hH(1:N);
```

```
    h(M + 1) = 1 + h(M + 1);
  end
  % Mostrar matriz de coeficientes
  format short
  % Escalar coeficientes
  gain=round(min(abs(-128/min(h)), abs(127/max(h))));
  gain=2^(length(dec2binvec(gain)));
  numf=round(gain*h);
  % Evitar overflow
  if abs(sum(numf))>gain
    aux=gain/sum(numf);
    for n = 1:1:N
      numf(1,n)=fix(numf(1,n)*aux);
    end
  end
  % Mostrar matriz de coeficientes
  disp('Los coeficientes del filtro son:')
  h
  disp('Para evitar overflow, utilice los siguientes coeficientes:')
  for n = 1:1:N
    fprintf(strcat(num2str(numf(1,n)),','))
  end
  fprintf('\n El factor de escalamiento es:')
  S=gain
  if sum(numf)<0
    disp('Para evitar underflow, reste a la entrada y sume a la salida:')
    offset=2048
  end
  % Gráficas de magnitud y fase en ventana rectangular
  figure
  [hz,w]=freqz(h,1,512);
  phi=180*unwrap(angle(hz))/pi;
  subplot(2,1,1)
  plot(w,abs(hz));
  grid
  title('Respuesta en magnitud')
  xlabel('Frecuencia (rad)');
  ylabel('Magnitud');
  subplot(2,1,2)
  plot(w,phi);
  grid
  title('Respuesta en fase')
  xlabel('Frecuencia (rad)');
  ylabel('Fase (grados)');
  continuar=input('\n ¿Desea continuar? Si(s)/No(n): ','s')
end
```

2. Cálculo de los coeficientes de los filtros digitales IIR en MATLAB

```
% Filtros digitales IIR: Diego Jarrín
continuar='s';
while continuar=='s'
   clear;
   clc;
   % Coeficiente de amortiguamiento para filtros pasabajos y pasaaltos
   zeta=sqrt(2)/2; % Factor de amortiguamiento
   res=12; % Resolución del DAC
   orden=2;
   disp('Universidad Politécnica Salesiana')
   disp('DSP con XMEGA')
   disp('Filtros digitales IIR')
   disp('Programación: Diego Jarrín')
   aux=0;
   %  Pedir datos del filtro pasabajos al usuario
   fs=input('Ingrese la frecuencia de muestreo o sampling en Hz: ')
   %Selección del filtro
   disp('Seleccione un tipo de filtro:')
   disp('1: Pasa-bajos')
   disp('2: Pasa-altos')
   disp('3: Pasa-banda')
   disp('4: Rechaza-banda')
   filtro = input('Opción: ');
   switch filtro
      case {1}
         disp('Filtro pasa-bajos')
         fl=input('Ingrese la frecuencia de corte inferior: ')
         if 0<fl && fl<fs
            % Frecuencia normalizada
            wd=2*pi*fl/fs;
            % Cálculo de coeficientes
            omega=2*fs*tan(wd/2);
            b0=omega^2/(4*fs^2+4*zeta*omega*fs+omega^2);
            b1=2*b0;
            b2=b0;
            a0=1;
            a1=(2*omega^2-8*fs^2)/(4*fs^2+4*zeta*omega*fs+omega^2);
            a2=(4*fs^2-4*zeta*omega*fs+omega^2)/(4*fs^2+4*zeta*omega*fs+omega^2);
         else
            disp('0<<fl<<fs')
            break
         end
      case {2}
         disp('Filtro pasa-altos')
         fh=input('Ingrese la frecuencia de corte superior: ')
         if 0<fh && fh<fs
            % Frecuencia normalizada
            wd=2*pi*fh/fs;
            % Cálculo de coeficientes
            omega=2*fs*tan(wd/2);
            b0=(4*fs^2)/(4*fs^2+4*zeta*omega*fs+omega^2);
            b1=-2*b0;
            b2=b0;
            a0=1;
            a1=(2*omega^2-8*fs^2)/(4*fs^2+4*zeta*omega*fs+omega^2);
            a2=(4*fs^2-4*zeta*omega*fs+omega^2)/(4*fs^2+4*zeta*omega*fs+omega^2);
         else
            disp('0<<fh<<fs')
            break
         end
      case {3}
         disp('Filtro pasa-banda')
         fl=input('Ingrese la frecuencia de corte inferior: ')
         fh=input('Ingrese la frecuencia de corte superior: ')
         if 0<fl && fl<fh && fh<fs
            % Frecuencia normalizada
            wdl=2*pi*fl/fs;
            wdh=2*pi*fh/fs;
            % Cálculo de coeficientes
            omegal=2*fs*tan(wdl/2);
            omegah=2*fs*tan(wdh/2);
            b0=(2*fs*omegal)/(4*fs^2+2*omegal*fs+omegah^2);
```

```
            b1=0;
            b2=-b0;
            a0=1;
            a1=(2*omegah^2-2*fs*omegal)/(4*fs^2+2*omegal*fs+omegah^2);
            a2=(4*fs^2-2*omegal*fs+omegah^2)/(4*fs^2+2*omegal*fs+omegah^2);
         else
            disp('0<<fl<<fh<<fs')
            break
         end
      case {4}
         disp('Filtro rechazabanda')
         fl=input('Ingrese la frecuencia de corte inferior: ')
         fh=input('Ingrese la frecuencia de corte superior: ')
         if 0<fl && fl<fh && fh<fs
            % Frecuencia normalizada
            wdl=2*pi*fl/fs;
            wdh=2*pi*fh/fs;
            % Cálculo de coeficientes
            omegal=2*fs*tan(wdl/2);
            omegah=2*fs*tan(wdh/2);
            b0=(omegah^2+4*fs^2)/(4*fs^2+2*omegal*fs+omegah^2);
            b1=(2*omegah^2-8*fs^2)/(4*fs^2+2*omegal*fs+omegah^2);
            b2=b0;
            a0=1;
            a1=b1;
            a2=(4*fs^2-2*omegal*fs+omegah^2)/(4*fs^2+2*omegal*fs+omegah^2);
         else
            disp('0<<fl<<fh<<fs')
            break
         end
      otherwise
         disp('!Seleccione una opción válida¡')
         break
   end
   % Mostrar matriz de coeficientes
   disp('Los coeficientes del filtro digital IIR son:')
   disp('Numerador')
   b=[b0 b1 b2]
   disp('Denominador')
   a=[a0 a1 a2]
   % Evitar desbordamiento
   [h,t]=impz(b,a);
   if(sum(h)<0)
      disp('Para evitar underflow, reste a la entrada y sume a la salida:')
      offset=2048
   end
   % Gráficas de magnitud y fase en ventana rectangular
   figure
   [hz,w]=freqz(b,a,512);
   phi=180*unwrap(angle(hz))/pi;
   subplot(2,1,1)
   plot(w,abs(hz));
   grid
   title('Respuesta en magnitud')
   xlabel('Frecuencia (rad)');
   ylabel('Magnitud');
   subplot(2,1,2)
   plot(w,phi);
   grid
   title('Respuesta en fase')
   xlabel('Frecuencia (rad)');
   ylabel('Fase (grados)');
   continuar=input('\n ¿Desea continuar? Si(s)/No(n): ','s')
end
```

3. Simulación de la respuesta de frecuencia de los filtros digitales FIR en MATLAB

```
% Filtros digitales FIR: Diego Jarrín
  clc
  clear
  disp('Universidad Politécnica Salesiana')
  disp('DSP con XMEGA')
  disp('Filtros digitales FIR')
  disp('Programación: Diego Jarrín')
  %  Pedir datos del filtro al usuario
  fs=input('Ingrese la frecuencia de muestreo o sampling en Hz: ');
  fl=input('Ingrese desde que frecuencia tomo los datos, en Hz: ');
  wl=2*pi*fl/fs;
  fh=input('Ingrese hasta que frecuencia tomo los datos, en Hz: ');
  wh=2*pi*fh/fs;;
  cont=input('Cada cuanto tomo los datos, en Hz: ');
  salto=2*pi*cont/fs;;
  num=input('Ingrese coeficientes del filtro digital \n');
  % Inicialización de variables
  orden=length(num);
  aux=0;
  % Respuesta de frecuencia
  for(p=wl:salto:wh)
     aux=aux+1;
     f(aux,1)=p;
     for k=1:1:orden
        e(1,k)=exp(-1i*p*k);
     end
     c=dot(num,e);
     h(aux,1)=c;
     mag(aux,1)=abs(c);
     fase(aux,1)=angle(c);
  end
  % Mostrar respuesta de frecuencia
  mag
  grad=unwrap(fase)*180/pi
```

4. Simulación de la respuesta de frecuencia de los filtros digitales IIR en MATLAB

```
% Filtros digitales IIR: Diego Jarrín
  clc
  clear
  disp('Filtros digitales IIR')
  disp('Programación: Diego Jarrín')
  %  Pedir datos del filtro al usuario
  fs=input('Ingrese la frecuencia de muestreo o sampling en Hz: ');
  fl=input('Ingrese desde que frecuencia tomo los datos, en Hz: ');
  wl=2*pi*fl/fs;
  fh=input('Ingrese hasta que frecuencia tomo los datos, en Hz: ');
  wh=2*pi*fh/fs;;
  cont=input('Cada cuanto tomo los datos, en Hz: ');
  salto=2*pi*cont/fs;;
  b=input('Ingrese coeficientes del numerador \n');
  a=input('Ingrese coeficientes del denominador \n');
  % Inicialización de variables
  orden=length(b);
  aux=0;
  % Respuesta de frecuencia
  for(p=wl:salto:wh)
     aux=aux+1;
     f(aux,1)=p;
     for k=1:1:orden
        e(1,k)=exp(-1i*p*k);
     end
     c=dot(b,e);
     d=dot(a,e);
     h(aux,1)=c/d;
     mag(aux,1)=abs(c/d);
     fase(aux,1)=angle(c/d);
  end
  % Moatrar respuesta de frecuencia
  mag
  grad=unwrap(fase)*180/pi
```

5. Código de los filtros digitales FIR en Mikro C para AVR

```
/* Filtros digitales FIR
 Autor: Diego Jarrín
 MCU: ATxmega128A1
 Dev.Board: Ready for XMEGA,
 Oscilador: Reloj interno 32 MHz
 Compilador: mikroC PRO for AVR*/

// Declaración de varíables.
#define M 9 // Número de coeficientes
#define G 256  // Factor de escalamiento: 512 pasa-bajos, 256 pasa-altos, 1024 pasa-banda, 256 rechaza-banda
signed long out, in[M];    // Entrada in[n], salida out;
unsigned short i; // Variable auxiliar para bucles for

// Coeficientes del filtro FIR
int h[M] = {
        //0,32,69,98,109,98,69,32,0  // Pasa-bajos fl=2000Hz;
        //0,-19,-41,-58,192,-58,-41,-19,0 // Pasa-altos fh=2000Hz;
        //-81,-118,-48,71,128,71,-48,-118,-81 // Pasa-banda fl=2000Hz, fh=3000Hz;
        16,24,9,-14,183,-14,9,24,16// Rechaza-banda fl=2000Hz, fh=3000Hz;
};

//Interrupción para realizar adquisición, procesamiento y reconstrucción de la señal
void TimerTCC0Overflow_ISR() org IVT_ADDR_TCC0_OVF {
// Adquisición
  ADCA_CH2_CTRL.B7=1;   // Iniciar conversión, mantener lectura del canal 2.
  in[0]= ADCA_CH2RES-2048; // Entrada menos offset
// Algoritmo DSP para un filtro digital FIR
  for( out=0;i=0; i<=M; i++ )    {
              if(i==0){out = in[0]*h[0];}
              else
              {in[i]=in[i-1]; // Almacenamiento de muestras
              out += in[i]*h[i];}    // Convolución
// Reconstrucción
  DACB_CH0DATA= out/G;+2048;      // Salida más offset
}

// Bloque de programa principal
void main() {
//  Configuración del Reloj
  OSC_CTRL = 0x02;        // 32MHz
  while(RC32MRDY_bit == 0);
  CPU_CCP = 0xD8;
  CLK_CTRL = 1;           // Habilitar oscilador interno

// Configuración del Timer TCC0 para Fs=16000 Hz; CLK/fs=TCC0_PER en 2 registros
  TCC0_PERL = 0xD0;
  TCC0_PERH = 0x07;
  TCC0_CTRLA = 0x01; // Prescaler 0
  TCC0_INTCTRLA = 0x02; // Habilitar overflow interrupt
  PMIC_CTRL = 0x02;     // Habilitar medium level interrupts
  CPU_SREG.B7 = 1;      // Habilitar global interrupts

// Configuración del ADCA en Canal 2, pin PA1
  ADCA_CH2_CTRL.B0=1;   // Single-ended unsigned; 0 a 4095, Vref=1V
  ADCA_CH2_MUXCTRL.B3=1;  //Canal 2 para ADCA
  ADCA_CTRLA=0x01;   // Habilito ADC en puerto A, bit de enable.

// Configuración del DACB, pin PB2
  DACB_CTRLA=0x05;  //Inicializar DAC bit 0 en canal 0, bit 2 en B, pin PB2

// Bucle infinito del bloque main
  while (1);
}
```

6. Código de los filtros digitales IIR en MikroC para AVR

```
/* Filtros digitales IIR
 Autor: Diego Jarrín
 MCU: ATxmega128A1
 Dev.Board: Ready for XMEGA,
 Oscilador: Reloj interno 32 MHz
 Compilador: mikroC PRO for AVR*/

// Declaración de varíales.
#define M 3 // Número de coeficientes
float out[M], in[M];    // Entrada in[n], salida out[n];
unsigned short i; // Variable auxiliar para bucles for

//Interrupción para realizar adquisición, procesamiento y reconstrucción de la señal
void TimerTCC0Overflow_ISR() org IVT_ADDR_TCC0_OVF {
// Adquisición
  ADCA_CH2_CTRL.B7=1;   // Iniciar conversión, mantener lectura del canal 2.
  in[0]= ADCA_CH2RES-2048; // Entrada más offset
  // Ecuación en diferencias para el filtro IIR
      //out[0] = (in[0]+in[2])* 0.7885-0.7160*(in[1]- out[1])- out[2]*0.5770;   // Rechaza-banda fs=8000Hz, fl=1000Hz,
fh=1500Hz
      //out[0] = (in[0]-in[2])* 0.2115+0*in[1]-0.2191*out[1]-out[2]*0.5770; // Pasabanda fs=8000Hz y fl=900Hz, fh=1400Hz
      out[0] = (in[0]+in[2])*0.6575-in[1]*1.3149+1.1939*out[1]-out[2]*0.4359; // Pasaaltas fs=8000Hz y fh=600Hz
      //out[0] = (in[0]+in[2])*0.2929+in[1]* 0.5858+0*out[1]-out[2]*0.1716; // Pasabajos 8000Hz y Fl=1000Hz
  for( i=M-1; i>0; i-- ){in[i]=in[i-1]; out[i]=out[i-1]; }    // Almacenamiento de muestras
// Reconstrucción
  DACB_CH0DATA= out[0]+2048;//+2048; // Salida menos offset
}

// Bloque de programa principal
void main() {
//  Configuración del Reloj
  OSC_CTRL = 0x02;        // 32MHz
  while(RC32MRDY_bit == 0);
  CPU_CCP = 0xD8;
  CLK_CTRL = 1;           // Habilitar oscilador interno

// Configuración del Timer TCC0 para Fs;
  TCC0_PERL = 0xD0;
  TCC0_PERH = 0x07;
  TCC0_CTRLA = 0x01; // Prescaler 0
  TCC0_INTCTRLA = 0x02; // Habilitar overflow interrupt
  PMIC_CTRL = 0x02;     // Habilitar medium level interrupts
  CPU_SREG.B7 = 1;      // Habilitar global interrupts

// Configuración del ADCA en Canal 2, pin PA1
  ADCA_CH2_CTRL.B0=1;   // Single-ended unsigned; 0 a 4095, Vref=1V
  ADCA_CH2_MUXCTRL.B3=1;  //Canal 2 para ADCA
  ADCA_CTRLA=0x01;   // Habilito bit de enable.

// Configuración del DACB, pin PB2
  DACB_CTRLA=0x05;  //Inicializar DAC bit 0 en canal 0, bit 2, pin PB2
// Bucle infinito del bloque main
  while (1);
}
```

Anexo 3. Coeficientes calculados para los filtros digitales FIR e IIR utilizando MATLAB

1. Coeficientes para los filtros digitales FIR

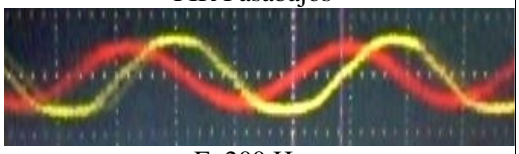
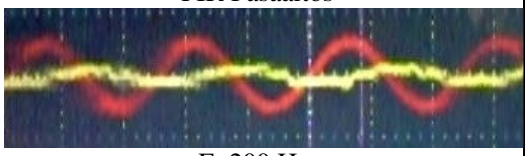
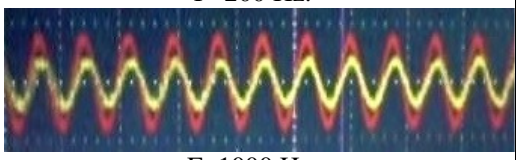
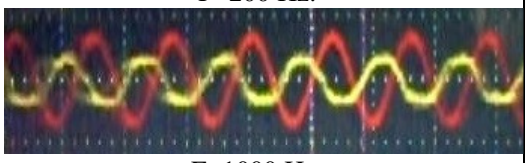
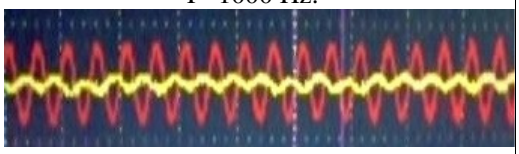
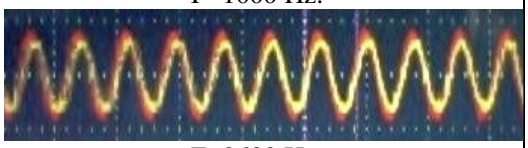
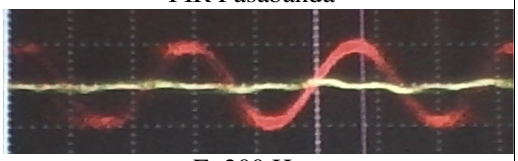
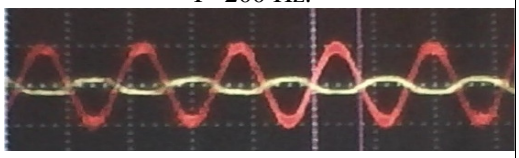
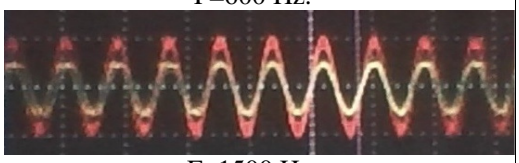
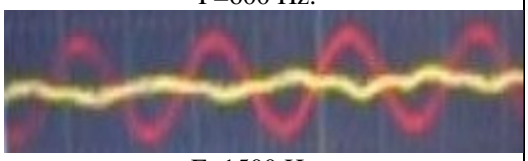
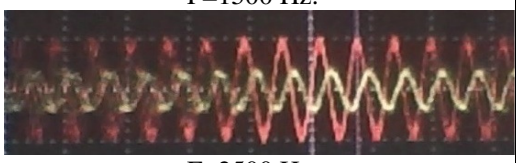
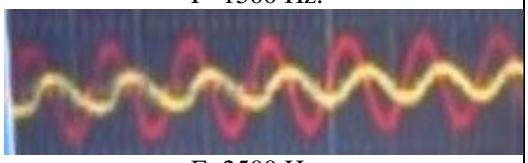
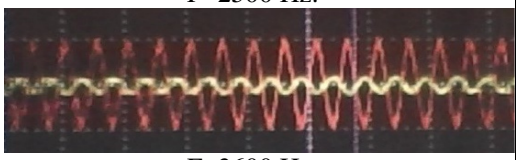
Fs= 16 kHz	**F de corte**	$b(0)$	$b(1)$	$b(2)$	$b(3)$	$b(4)$	$b(5)$	$b(6)$	$b(7)$	$b(8)$
Pasa-bajo	2 kHz	0.0000	0.0750	0.1592	0.2251	0.2500	0.2251	0.1592	0.0750	0.0000
Escalamiento	512	0	32	69	98	109	98	69	32	0
Pasa-alto	2 kHz	0.0000	-0.0750	-0.1592	-0.2251	0.7500	-0.2251	-0.1592	-0.0750	0.0000
Escalamiento	256	0	-19	-41	-58	192	-58	-41	-19	0
Pasa-banda	2 kHz a 3 kHz	-0.0796	-0.1156	-0.0466	0.0690	0.1250	0.0690	-0.0466	-0.1156	-0.0796
Escalamiento	1024	-81	-118	-48	71	128	71	-48	-118	-81
Rechaza-banda	2 kHz a 3 kHz	0.0796	0.1156	0.0466	-0.0690	0.8750	-0.0690	0.0466	0.1156	0.0796
Escalamiento	256	16	24	9	-14	183	-14	9	24	16

2. Coeficientes para los filtros digitales IIR

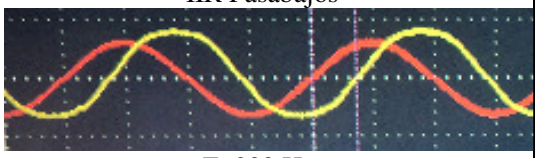
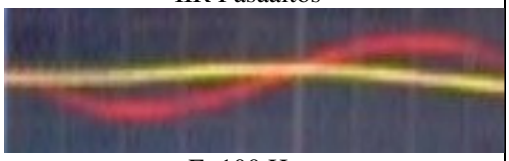
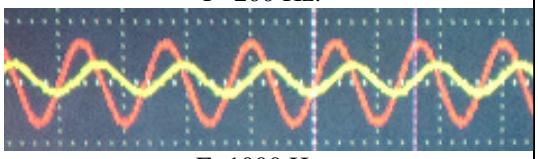
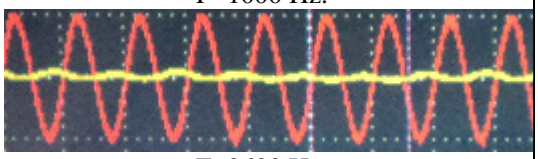
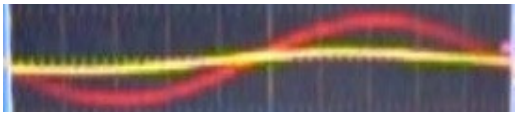
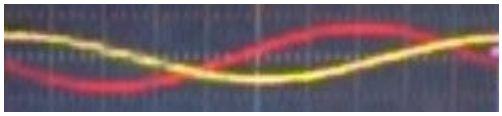
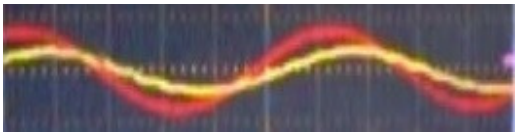
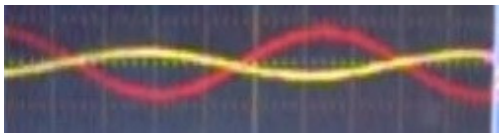
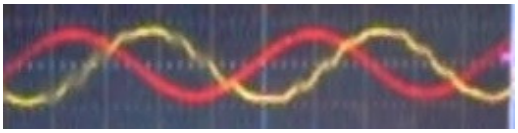
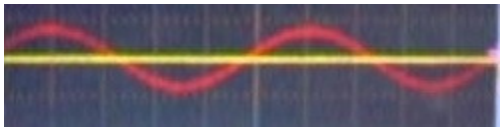
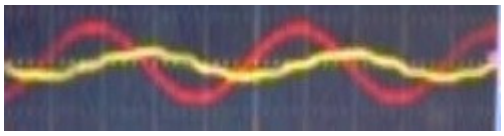
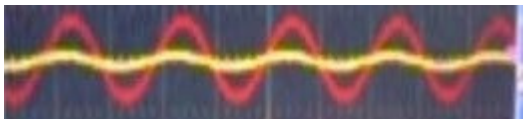
Fs= 8 kHz	**F de corte**	**b(0)**	**b(1)**	**b(2)**	**a(0)**	**a(1)**	**a(2)**
Pasa-bajo	1 kHz	0.0976	0.1953	0.0976	1.0000	-0.9428	0.3333
Pasa-alto	0.6 kHz	0.7157	-1.4315	0.7157	1.0000	-1.3490	0.5140
Pasa-banda	0.9 kHz a 1.4 kHz	0.2115	0	-0.2115	1.0000	0.2191	0.5770
Rechaza-banda	1 kHz a 1.5 kHz	0.7774	-0.5950	0.7774	1.0000	-0.5950	0.5548

Anexo 4. Documentación fotográfica del comportamiento de los filtros digitales FIR e IIR

1. Fotografías de los filtros digitales FIR: en rojo se muestra la señal de entrada y en amarillo la señal de salida.

FIR Pasabajos	FIR Pasaaltos
	
F=200 Hz.	F=200 Hz.
	
F=1000 Hz.	F=1000 Hz.
	
F=3600 Hz.	F=3600 Hz.
FIR Pasabanda	FIR Rechazabanda
	
F=200 Hz.	F=200 Hz.
	
F=600 Hz.	F=600 Hz.
	
F=1500 Hz.	F=1500 Hz.
	
F=2500 Hz.	F=2500 Hz.
	
F=3600 Hz.	F=3600 Hz.

2. Fotografías de los filtros digitales IIR: en rojo se muestra la señal de entrada y en amarillo la señal de salida.

IIR Pasabajos	IIR Pasaaltos
	
F=200 Hz.	F=100 Hz.
	
F=1000 Hz.	F=500 Hz.
	
F=3600 Hz.	F=1800 Hz.
IIR Pasabanda	**IIR Rechazabanda**
	
F=200 Hz.	F=150 Hz.
	
F=600 Hz.	F=600 Hz.
	
F=1500 Hz.	F=1200 Hz.
	
F=2500 Hz.	F=1800 Hz.
	
F=3600 Hz.	F=2700 Hz.