Universidad del Valle de Guatemala

Facultad de Ingeniería

Diseño de un afinador digital para guitarra eléctrica bajo el estándar E

Trabajo de graduación presentado por
Victor Orlando Fuentes Araujo
para optar al grado académico de Licenciado en Ingeniería en Mecatrónica

Guatemala,
2016

Diseño de un afinador digital para guitarra eléctrica bajo el estándar E

Universidad del Valle de Guatemala

Facultad de Ingeniería

Diseño de un afinador digital para guitarra eléctrica bajo el estándar E

Trabajo de graduación presentado por
Victor Orlando Fuentes Araujo
para optar al grado académico de Licenciado en Ingeniería en Mecatrónica

Guatemala,
2016

Vo. Bo. :

(f)_____

Ing. Miguel Zea

Asesor

Tribunal Examinador:

(f)_____

Ing. Miguel Zea

Asesor

(f)_____

Ing. Pablo Mazariegos

(f)_____

M.Sc. Carlos Esquit

Fecha de aprobación: Guatemala, 5 de diciembre del 2,016.

ÍNDICE

LISTA DE CUADROS vi
LISTA DE FIGURAS vii
RESUMEN ix
I. INTRODUCCIÓN 1
II. OBJETIVOS 3
A. Objetivo general 3
B. Objetivos específicos 3
III. JUSTIFICACIÓN 4
IV. MARCO TEÓRICO 6
A. PROCESAMIENTO DE SEÑALES DIGITALES 6
1. Señales en tiempo discreto. 6
2. Sistemas lineales en invariantes en el tiempo. 7
a. Linealidad e invariante en el tiempo. 8
b. Suma de convolución. 8
3. Discretización de señales continuas 8
a. Teorema de muestreo. 8
4. Transformada discreta de Fourier. 9
5. Transformada rápida de Fourier. 10
6. Filtros digitales. 11
a. Filtros anti-alias. 11
b. Filtros pasa-banda 12
c. Filtros de respuesta infinita al impuso. 12
1) Método del Impulso Invariante 13
2) Transformación bilineal. 13
3) Filtros Notch. 13
d. Filtros de respuesta finita al impulso. 14
1) Método de ventanas. 14
a) Ventana Kaiser. 15
e. Herramienta de diseño de filtros de Matlab. 15
7. Microcontroladores de 32 bits Microchip 16
a. PIC32mx. 16
1) Núcleo. 17
2) Organización de memoria 17
3) Módulos. 19

a) Módulo de convertidor analógico-digital. 19
b) Módulo de entradas y salidas digitales. 19
b. Pantalla LCD 1602a. 20
c. Módulo DAC PCM1795 de *Texas Instruments*. 20
B. INSTRUMENTO MUSICAL DE GUITARRA 22
1. Clasificaciones de las guitarras. 22
2. Estándar de afinación E. 24
a. Métodos de afinación de guitarra eléctrica. 24
1) Afinación manual. 24
2) Afinación con afinador. 25
3. Afinadores digitales. 25
a. Captación por micrófono. 25
b. Captación por vibración. 25
4. Señales de audio. 26
5. Numeración de mano. 26
6. Acordes del círculo de Do. 26
IV. ANTECEDENTES 29
V. METODOLOGÍA 31
A. Captación de señal de guitarra 31
1. Acondicionamiento de señal de guitarra. 31
2. Módulo convertidor analógico-digital 32
B. Elección de frecuencias para clasificación de notas y acordes 32
C. Filtrado de señal por medio de filtros de tipo pasa-banda 34
1. Filtro con respuesta al impulso finita. 34
2. Filtro con respuesta al impulso infinita tipo Notch. 35
a. Inversión de filtros Notch. 35
D. Análisis en dominio de frecuencia por medio de transformada discreta de Fourier 35
E. Diseño de módulo de afinador 36
1. Implementación filtros FIR. 36
a. Modo de operación para afinación. 37
b. Modo de operación para acodes. 37
2. Implementación filtros Notch Inverso. 38
3. Implementación FFT. 38
a. Modo de operación para afinación. 38
b. Modo de operación para acodes. 39
VII. RESULTADOS 40
A. Filtros FIR. 40

B. Filtros IIR Notch. 44
C. Filtro IIR Notch Inverso. 49
D. Transformada Rápida de Fourier. 54
E. Afinador digital 55
VIII. ANÁLISIS DE RESULTADOS 56
IX. CONCLUSIONES 59
X. RECOMENDACIONES 60
XI. BIBLIOGRAFÍA 61
XII. ANEXOS 64
A. Código microcontrolador pantalla 64
B. Código microcontrolador procesamiento de señal 72
C. Manual de usuario de afinador 80
XIII. GLOSARIO 81

LISTA DE CUADROS

Cuadro No. 1. Frecuencias de las cuerdas tocadas sin presionar algún traste según el estándar de A4=440Hz. 24
Cuadro No. 2. Frecuencias teóricas de notas musicales. 28
Cuadro No. 3. Notas musicales en idioma inglés y español. 28
Cuadro No. 4. Voltajes máximos y mínimos de la señal de la guitarra y del pin de entrada permitido en el microcontrolador. 31
Cuadro No. 5. Frecuencias de diseño para notas de afinación. 32
Cuadro No. 6. Frecuencias de diseño para notas de acorde Do Mayor. 33
Cuadro No. 7. Frecuencias de diseño para notas de acorde La Menor. 33
Cuadro No. 8. Frecuencias de diseño para notas de acorde Re Menor. 33
Cuadro No. 9. Frecuencias de diseño para notas de acorde Sol Mayor. 33
Cuadro No. 10. Orden de filtros FIR pasa-banda para afinación. 43
Cuadro No. 11. Orden de filtros IIR Notch para afinación. 47
Cuadro No. 12. Orden de filtros IIR Notch para acordes. 47
Cuadro No. 13. Constantes de implementación del filtro IIR Notch Inverso para 82Hz en función de transferencia. 53

LISTA DE FIGURAS

Figura No.1 Sistema de procesamiento de señales digitales. 6
Figura No. 2. Secuencia de datos para una señal discreta arbitraria e impulso unitario. 7
Figura No. 3. Representación con diagrama de bloques de una transformación aplicada a una señal de entrada para obtener una señal de salida. 7
Figura No. 4. Señal reconstruida con aliasing 9
Figura No. 5. Señal reconstruida sin aliasing, utilizando un periodo de $1fmuestreo$ 9
Figura No. 6. Representación de FFT para N=8. 10
Figura No. 7. Efecto de alias en dominio de frecuencia, al muestrear a fm. 11
Figura No. 8. Filtro anti-alias aplicado a una señal a frecuencia de corte fm 11
Figura No. 9. Respuesta en frecuencia del filtro Pasa-Banda con frecuencias de corte y frecuencias de ancho de banda. 12
Figura No. 10. Método de ventanas para el diseño de un filtro FIR. 15
Figura No. 11. Arquitectura de familia PIC32mx. 16
Figura No. 12. Diagrama de bloques del núcleo de la familia PIC32mx. 17
Figura No. 13. Mapa de memoria de la familia PIC32mx150/250 18
Figura No. 14. Mapa de direcciones de registros especiales de funciones. 18
Figura No. 15. Nombre de pines para microcontrolador PIC32mx250F128B 19
Figura No. 16. Diagrama de bloques de pantalla LCD con 16 columnas y 2 filas. 20
Figura No. 17. Diagrama de bloques de DAC PCM1795. 21
Figura No. 18. Partes de la guitarra 22
Figura No. 19. Partes de guitarra eléctrica. 23
Figura No. 20. Diagrama de trastes y cuerdas en la guitarra acústica o eléctrica. 23
Figura No. 21. Pastillas o “Pick-ups” marca Dogear P90 23
Figura No. 22. Convención de la numeración de los dedos de la mano izquierda para ejecutar la guitarra. 26
Figura No. 23. Circuito de anti-alias, acondicionamiento y amplificación de señal de guitarra. 32
Figura No. 24. Interfaz de la herramienta de diseño de filtros de respuesta al impulso finito en software Matlab. 34
Figura No. 25. Diagrama de bloques del afinador. 36
Figura No. 26. Diseño de afinador con pantalla LCD 16x2 modo de operación para afinación. 37
Figura No. 27. Diseño de afinador con pantalla LCD 16x2 modo de operación para acordes. 37
Figura No. 28. Diseño de afinador con pantalla GLCD modo de operación para afinación. 38
Figura No. 29. Diseño de afinador con pantalla GLCD modo de operación para acordes 39
Figura No. 30. Respuesta en frecuencia del filtro FIR pasa-banda para 82Hz. 40
Figura No. 31. Respuesta en frecuencia del filtro FIR pasa-banda para 110Hz 40
Figura No. 32. Respuesta en frecuencia del filtro FIR pasa-banda para 146Hz 41
Figura No. 33. Respuesta en frecuencia del filtro FIR pasa-banda para 196Hz. 41
Figura No. 34. Respuesta en frecuencia del filtro FIR pasa-banda para 246Hz. 42
Figura No. 35. Respuesta en frecuencia del filtro FIR pasa-banda para 329Hz. 42
Figura No. 36. Retraso de grupo de filtros FIR pasa-banda para afinación. 43
Figura No. 37. Respuesta en frecuencia del filtro IIR Notch para 82Hz. 44
Figura No. 38. Respuesta en frecuencia del filtro IIR Notch para 110Hz 44
Figura No. 39. Respuesta en frecuencia del filtro IIR Notch para 149Hz. 45
Figura No. 40. Respuesta en frecuencia del filtro IIR Notch para 196Hz 45
Figura No. 41. Respuesta en frecuencia del filtro IIR Notch para 247Hz. 46
Figura No. 42. Respuesta en frecuencia del filtro IIR Notch para 330Hz 46
Figura No. 43. Retraso de grupo de filtros IIR Notch para afinación 47
Figura No. 44. Respuesta en frecuencia de filtro IIR Notch para acordes 48
Figura No. 45. Retraso de grupo para filtros IIR Notch para acordes 48
Figura No. 46. Respuesta en frecuencia del filtro IIR Notch Inverso para 82Hz. 49

Figura No. 47. Respuesta en frecuencia del filtro IIR Notch Inverso para 110Hz 49
Figura No. 48. Respuesta en frecuencia del filtro IIR Notch Inverso para 146Hz. 50
Figura No. 49. Respuesta en frecuencia del filtro IIR Notch Inverso para 196Hz. 50
Figura No. 50. Respuesta en frecuencia del filtro IIR Notch Inverso para 246Hz. 51
Figura No. 51. Respuesta en frecuencia del filtro IIR Notch Inverso para 329Hz 51
Figura No. 52. Retraso de grupo de filtros IIR Notch Inverso para afinación 52
Figura No. 53. Digitalización de señal de 82Hz por medio del módulo ADC. 52
Figura No. 54. Respuesta del filtro Notch Inverso para 82Hz en dominio de tiempo discreto para 53
Figura No. 55. Transformada Rápida de Fourier de señal de entrada de 82Hz. 54
Figura No. 56. Transformada Rápida de Fourier de señal de entrada con componentes de 222Hz, 296Hz y 394Hz 54
Figura No. 57. Prototipo del circuito anti-alias, acondicionamiento y amplificación de señal de guitarra. 55
Figura No. 58. Prototipo del circuito afinador en placa de pruebas 55

RESUMEN

En el presente trabajo se recopila el proceso de diseño de un afinador digital, el cual es un dispositivo que ayuda al músico a asegurar que su instrumento se encuentra afinado según un estándar de notas musicales. El instrumento que se afinará será la guitarra eléctrica, la cual por medio de la señal generada de la misma se determinará si se encuentra afinada o no, al igual que se incluye el reconocimiento de acordes del círculo de Do, conformado por las notas: Do mayor, La menor, Re menor y Sol mayor.

Esta señal eléctrica fue procesada digitalmente para poder identificar las frecuencias y así determinar si la nota tocada o el acorde se encuentra afinado según el estándar de afinación E o conocido como La_4=440Hz.

La adquisición de la señal de la guitarra se realizó mediante el circuito de la Figura No. 23, el cual amplifica la señal para poder ser captada por el microcontrolador PIC32mx250F128B. Se propusieron dos métodos para la identificación de frecuencias, los cuales consistieron en: filtros pasa-banda de respuesta al impulso de duración finita, filtros Notch y Transformada Discreta de Fourier. Según el Cuadro No .11, los filtros de tipo pasa-banda de respuesta al impulso de duración finita no lograron ser implementados debido al alto orden de su función de transferencia. Los filtros Notch no lograron detectar la frecuencia de afinación, ver Figura No. 54, debido a la precisión en sus constantes al momento de realizar la implementación. El método que sí logró identificar la frecuencia de afinación es la Transformada Discreta de Fourier, ver Figura No. 55 y 56.

El afinador digital logró ser implementado en una tarjeta de pruebas, ver Figura No. 58, el cual no requiere de un computador para realizar el procesamiento de la señal, y es factible establecer la afinación del instrumento a partir de la señal eléctrica de la guitarra.

I. INTRODUCCIÓN

Actualmente existen gran variedad de instrumentos de cuerda, los cuales requieren de cuidados y mantenimiento para su funcionamiento óptimo. Uno de los instrumentos de cuerda más antiguos es la guitarra, la cual evolucionó hasta nuestro tiempo actual, dividiéndose en varias clasificaciones, nos enfocaremos en la guitarra eléctrica.

La guitarra eléctrica es una de las clasificaciones de la guitarra que utiliza dispositivos mecánicos y eléctricos para generar el sonido, a diferencia de la guitarra acústica que utiliza la resonancia de su cuerpo para producir el sonido. Uno de los mantenimientos necesarios en la guitarra es la afinación del instrumento. (La Cuerda, 2016)

La afinación asegura que las notas que se ejecuten en el instrumento obedecen un estándar de frecuencia, sonido y tonalidad. La afinación de la guitarra eléctrica se puede realizar por medio de una referencia o por medio de un afinador. La referencia puede ser la experiencia que obtenga el músico al reproducir el sonido con la guitarra, sonido que él puede comparar para afinar su instrumento. Sin embargo, para un músico que se inicia en el arte este puede resultar ser un método poco exacto. Es por lo anterior que surgen los afinadores de guitarra. (Guitar Book, 2012)

En la actualidad existen gran variedad de afinadores de guitarra eléctrica, hay de los cuales extraen el sonido de la guitarra por medio del aire o de las vibraciones en el cuerpo de la guitarra. En el ámbito académico guatemalteco ya se han creado afinador de guitarra, utilizando como unidad de procesamiento el computador. (Vargas, 2013)

El requerir un computador personal limita la tarea de afinación del instrumento, así también no han utilizado la salida de línea de la guitarra para realizar la afinación. El afinador propuesto utilizará esta señal generada por la guitarra para indicar la afinación del mismo. Esto conlleva a las ventajas de reducción del ruido que otros afinadores presentan y que requieren de componentes adicionales para suprimirlas.

El diseño del afinador incluirá varias fases de prueba para determinar el método por el cual se detectará la afinación del instrumento. Es importante mencionar que la afinación del instrumento se determina por medio del sonido que emite, en nuestro caso al ser cuerdas las cuales vibran, la afinación se asegura por medio de la frecuencia de oscilación de la cuerda. Y debido a que la señal emitida por la guitarra eléctrica es una señal eléctrica esta puede ser analiza por medios digitales.

Debido a lo anterior se plantea la digitalización de la señal eléctrica proporcionada por la guitarra, el procesamiento de las señales digitales es una técnica utilizada que dio inicios en los años 60. Consiste en la discretización de una señal continua, la cual pueda ser procesada de manera digital en un computador o procesador. El esquema básico de un sistema de procesamiento de digital es: la obtención de la señal continúa, que posteriormente se ingresa a un módulo de procesamiento digital. Este módulo se encarga de la extracción de información de la señal, aunque también pude incluir ciertas modificaciones a la señal. Por último, la señal digital vuelve a ser convertida a una señal analógica. (Marve & Ewers, 1996)

El módulo de procesamiento digital constará de métodos que identifiquen la frecuencia de la cuerda. Los métodos propuestos son: filtros pasa-banda de respuesta al impulso de duración finita, filtros tipo Notch y el análisis en el domino de frecuencia proporcionado por la transformada discreta de Fourier. El afinador además de indicar la afinación del instrumento, se plantea la identificación de los acordes del círculo de Do, los cuales son: Do mayor, La menor, Re menor y Sol Mayor.

Finalmente, el afinador propuesto deberá de utilizar un microcontrolador para realizar el procesamiento digital de la señal de la guitarra, esto con el fin de asegurar que el dispositivo es portable y no requerirá de un computador personal para su funcionamiento.

II. OBJETIVOS

A. Objetivo general

- Crear un afinador digital para guitarra eléctrica bajo el estándar E que reconozca los acordes del círculo de Do, utilizando la salida de línea de la guitarra y especificando la cuerda a afinar, mediante la implementación de un método de detección de frecuencias en un microcontrolador.

B. Objetivos específicos

- Diseñar un algoritmo que detecte las frecuencias que conforman la señal de la salida de línea de la guitarra, y que muestre si las frecuencias corresponden a las frecuencias de afinación del estándar de afinación, mediante el uso de técnicas de procesamiento digital, como lo son: la Transformada de Fourier y filtros digitales.
- Lograr una afinación de la cuerda cuyo error no sea perceptible al oído humano utilizando un algoritmo de detección de frecuencias, mediante su implementación en un microcontrolador que realice el procesamiento digital de la señal de la salida de línea de la guitarra eléctrica.
- Implementar una función de reconocimiento de acordes del círculo de Do en el algoritmo de detección de frecuencia, mediante el uso de técnicas de procesamiento digital como lo son: la Transformada de Fourier y filtros digitales, luego que la guitarra haya sido afinada e indicando al afinador qué acorde se desea comprobar.
- Realizar un diseño de hardware portátil, cómodo y práctico del prototipo, que no requiera de componentes adicionales para su funcionamiento, mediante el uso de componentes electrónicos de estado sólido implementados en una placa para circuitos impresos.

III. JUSTIFICACIÓN

Actualmente en el mercado se encuentran gran variedad de afinadores para instrumentos de cuerda, los cuales por medio del sonido que emiten o de las vibraciones del instrumento determinan si el instrumento se encuentra afinado. Estos afinadores logran recibir la señal emitida por la cuerda de la guitarra mediante transductores mecánico-eléctricos, que traducen las vibraciones del cuerpo de la guitarra en señales eléctricas. Las cuales por medio del procesamiento digital de la señal logran establecer cuales frecuencias conforman la señal y de esta forma determinan si la cuerda se encuentra afinada a la frecuencia establecida por el estándar de afinación. Este diseño de afinador presenta un problema, toda vibración que se genere en la guitarra se traducirá en información que adquiera el afinador, pudiendo incorporarse señales parásitas como: ruido del ambiente o hacer contacto con el cuerpo del instrumento. Lo que involucra implementar filtros adicionales para eliminar estas señales.

El diseño del afinador que capta el ruido de la guitarra utiliza un micrófono para poder traducir las ondas en el ambiente a ondas eléctricas. Este afinador también presenta el problema de que se introduzcan señales parásitas, pudiéndose generar del ruido del ambiente. Lo anterior implica que este afinador únicamente se puede utilizar cuando el ruido del ambiente sea lo suficientemente escaso para no afectar a la onda de la cuerda.

Sin embargo, no existen afinadores para guitarra eléctrica que procesen la señal de línea de salida de la guitarra para determinar la afinación del instrumento. Utilizar esta señal presenta una ventaja, será menos susceptible al ruido ambiental e interferencias electromagnéticas ya que posee elementos mecánicos y eléctricos ya implementados en la guitarra que se encargan de eliminar este tipo de señales indeseables. Estos elementos mecánicos se denominan “Pastillas”, o por sus siglas en ingles “Pickups”. Las pastillas consisten en embobinados de cobre alrededor de un imán, el cual mediante la vibración de la cuerda de metal inducen una corriente en el embobinado, la cual genera la señal eléctrica de la salida de línea de la guitarra. Estas pastillas poseen ciertos recubrimientos cerámicos que aíslan la pastilla del cuerpo de la guitarra, evitando así la adición de señales indeseables.

Dado lo anterior, el utilizar la señal de línea de la guitarra disminuirá la necesidad de filtros digitales para reducir estas señales parásitas y por lo tanto el proceso de afinación será más rápido y requerirá de menos componentes electrónicos. Además de la reducción de componentes, implementar el procesamiento de la señal en un microcontrolador en vez de en un computador personal y el tamaño compacto de los componentes, proporciona la ventaja de generar un prototipo de un afinador de guitarra eléctrica compacto y portátil, y el cual podrá ser utilizado en ambientes con gran cantidad de ruido ambiental o en lugares con poco ruido ambiental, algo que el afinador que utiliza el micrófono para captar la señal dela guitarra carece.

Al agregar la posibilidad del reconocimiento de acordes del círculo de Do se brinda un beneficio sobre los afinadores que exclusivamente han sido construidos para determinar si la guitarra se encuentra afinada. Por lo tanto, se generará un prototipo que es portable en comparación a utilizar un computador personal para procesar la información y que presenta ventajas sobre un afinador común que exclusivamente indica si la guitarra se encuentra afinada.

IV. MARCO TEÓRICO

A. PROCESAMIENTO DE SEÑALES DIGITALES

El procesamiento digital de las señales forma un sistema compuesto por: la conversión de una señal continúa o analógica a discreta o digital, llamada conversión A/D, un proceso de lectura, modificación o almacenamiento de dicha señal, una conversión de la señal discreta o digital, llamada conversión D/A, y finalmente la traducción de esta señal continua a un actuador o medio físico capaz de reproducirla. En la Figura No. 1 se puede apreciar este sistema de manera visual.

Figura No.1 Sistema de procesamiento de señales digitales.

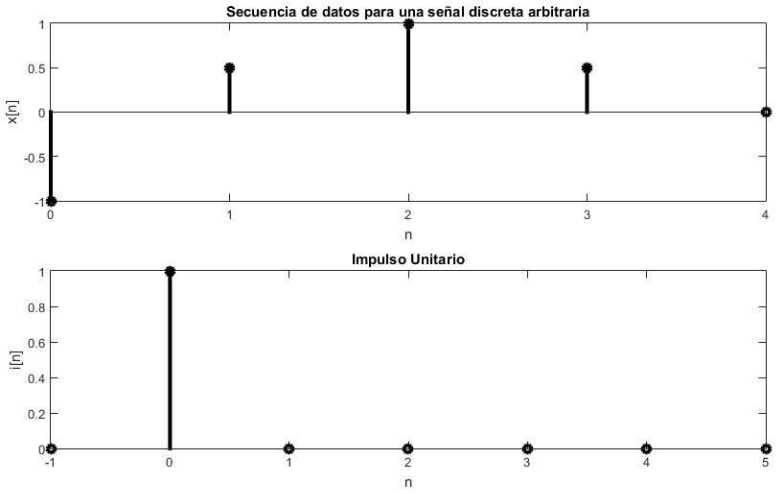
(*Marven & Ewers, 1996, pp3*)

El desarrollo de las técnicas para realizar procesamiento digital de señales o DSP fue de la mano con el desarrollo de las computadoras y de la matemática alrededor de los años 1950 a 1960. Estas técnicas o conocimientos matemáticos desarrollados dieron la pauta para el desarrollo de las computadoras capaces de realizar procesamiento digital, las cuales alrededor de 1980, fundaron las bases para el DSP que se conoce actualmente. Estas computadoras, que posteriormente se les dio el nombre de microcomputadoras o microcontroladores, que realizaban cálculos sobre la señal digital se les conoce como “Procesadores de señales digitales” o DSPs.

El objetivo principal para el desarrollo del procesamiento digital de señales era la creación de simulaciones sobre señales continúas del mundo real, sin embargo, se requeriría de una velocidad lo suficientemente rápida de los DSPs para que se pudiera realizar estos calculos sin afectar el tiempo de respuesta del sistema. Lo que conduciría al desarrollo del procesamiento de señales digitales en tiempo real, el cual será el enfoque de este trabajo. (Marve & Ewers, 1996)

1. Señales en tiempo discreto. Las señales en tiempo discreto son conjuntos de datos o secuencias que contiene información sobre algún fenómeno físico, pudiendo ser la digitalización de una señal continua, o una secuencia de datos originaria desde una computadora. Sea cual sea el caso, las señales discretas, como así nos referiremos en el presente trabajo, contienen información y podrán ser graficadas como se puede observar en la Figura No. 2, para el caso de una señal arbitraria y el impulso unitario.

Figura No. 2. Secuencia de datos para una señal discreta arbitraria e impulso unitario.



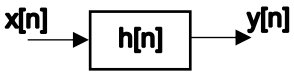
Se resalta que estas señales solo se encuentran definidas para valores en el eje “X” a cada “n” muestras, siendo $n \in \mathbb{Z}$. Con las señales antes mencionadas es posible componer cualquier señal discreta mediante el uso de la señal del impulso unitario discreto, siendo esta escalada y/o atrasada y sumándolas para formar una nueva secuencia. La Ecuación No. 1 ejemplifica esta composición de cualquier señal discreta como la suma de impulsos unitarios desfasados, una forma de expresar la señal que conducirá a definir los sistemas lineales invariantes en el tiempo. (Oppenhaim & Schafer, 1975)

$$x[n] = \sum_{K=-\infty}^{\infty} x[k]\delta[n-k] = \cdots x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \cdots \quad (1)$$

2. Sistemas lineales en invariantes en el tiempo.

Los sistemas son transformaciones matemáticas que se aplican a una señal o función de entrada x[n] para obtener una señal y[n]. Esta transformación se define como una función de transferencia h[n]. En la Figura No. 3 se puede visualizar lo anterior.

Figura No. 3. Representación con diagrama de bloques de una transformación aplicada a una señal de entrada para obtener una señal de salida.



(*Oppenhaim & Schafer, 1975, pp11*)

Esta transformación tiene algunas restricciones para poderse definir como un sistema lineal e invariante en el tiempo.

a. Linealidad e invariante en el tiempo. Un sistema es lineal si cumple con tener una transformación homogénea, si es posible la superposición de entradas para generar la superposición de salidas, Implicando que el sistema tiene un orden de 1. Cumple con ser un sistema invariante en el tiempo si un desplazamiento de tiempo en la entrada se refleja en un desplazamiento de igual tiempo en la salida. (Oppenhaim, A & Schafer, 1975)

b. Suma de convolución. La suma de convolución indica que la salida “y[n]” de un sistema lineal e invariante en el tiempo puede caracterizarse por medio de la convolución de su respuesta al impulso “h[n]” y la entrada “x[n]” del sistema. Lo anterior puede visualizarse en la Ecuación No. 2. (Oppenhaim, A & Schafer, 1975)

$$y[n] = \sum_{K=-\infty}^{\infty} x[k]h[n-k] = x[k] * h[n] \quad (2)$$

3. Discretización de señales continuas

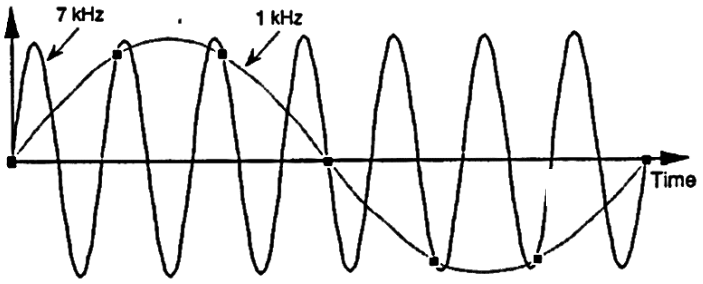
a. Teorema de muestreo. El teorema de muestreo es un postulado matemático establecido originalmente por Harry Nyquist en 1928, posteriormente se formalizo por Claude Shannon en el año de 1949. El teorema establece que la frecuencia mínima de muestreo, o frecuencia de Nyquist, para representar una señal analógica debe de ser al menos el doble de la frecuencia máxima que posee la señal analógica original. El término de frecuencia máxima hace referencia al ancho de banda de la señal original. En la Ecuación No. 3 se puede visualizar el teorema de Nyquist. (Marve & Ewers, 1996)

$$f_{Nyquist} \geq 2f_{máxima} \quad (3)$$

Esta frecuencia de muestreo se establece al momento de realizar el convertidor A/D, el cual puede ser un circuito de muestreo implementado en un microcontrolador mediante un módulo de ADC, el cual se discutirá posteriormente. Sin embargo, existe una consideración más antes de convertir la señal analógica a digital, la cual es el efecto de aliasing.

El efecto de aliasing se entiende como la trasposición de una señal de mayor frecuencia a una de menor frecuencia debido a que la frecuencia de muestreo no supera en doble cantidad al ancho de banda de la señal que se desea muestrear. En la Figura No. 4 se puede observar el efecto de aliasing al tener un periodo entre muestras erróneo para nuestra señal, cuando se desee digitalizarla vemos que la señal sufrió modificaciones en el dominio de frecuencia. (Marve & Ewers, 1996)

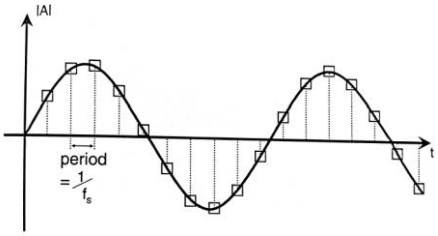
Figura No. 4. Señal reconstruida con aliasing.



(*Lyons, 2011, pp37*)

En la Figura No. 5 se puede observar que la señal digital ya no sufre del efecto de aliasing debido a que ahora la frecuencia de muestreo fue modificada, en este caso la señal reconstruida si se asemeja a la señal original debido a que las muestras fueron estuvieron menos espaciadas.

Figura No. 5. Señal reconstruida sin aliasing, utilizando un periodo de $\frac{1}{f_{muestreo}}$



(*Marven & Ewers, 1996. pp39*.)

4. Transformada discreta de Fourier. La transformada discreta de Fourier es una secuencia de valores o serie finita que indica los componentes en frecuencia que posee una secuencia discreta. Supongamos que tenemos una secuencia finita $\tilde{x}[n]$, de duración "N" cuya transformación usando la Transformada discreta de Fourier (DFT) se puede ver en la Ecuación No. 4.

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n]\, e^{-j\left(\frac{2\pi k}{N}\right)n} = \sum_{n=0}^{N-1} \tilde{x}[n]\, W_N^{kn} \quad (4)$$

Así también se tiene la fórmula para Transformada inversa discreta de Fourier, Ecuación No. 5, para obtener la secuencia discreta $\tilde{x}[n]$, a partir de la secuencia $\tilde{X}[k]$ en dominio de frecuencia discreto. (Oppenhaim, A & Schafer, 1975)

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k]\, e^{+j\left(\frac{2\pi n}{N}\right)k} = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k]\, W_N^{-kn} \quad (5)$$

La implementación de la Transformada discreta de Fourier requiere de un orden de valor N^2, debido a que los valores de “n” y “k” se deben de sumar y multiplicar “N” veces.

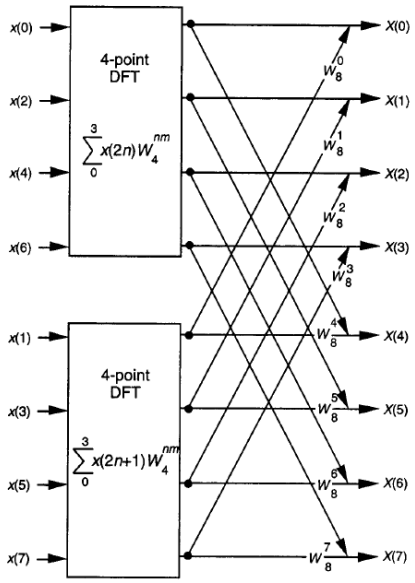
5. Transformada rápida de Fourier. Debido a que la implementación de la transformada discreta de Fourier requiere de excesivos ciclos de procesamiento, existe un algoritmo para reducir la cantidad de ciclos de procesamiento que requiere el realizar la transformada discreta de Fourier, llamado Transformada Rápida de Fourier (FFT). El algoritmo de Cooley-Tukey o “Radix – 2” rescribirá la FFT como la suma de dos funciones, una par y otra impar, lo cual reducirá la cantidad de ciclos de procesamiento para computar el algoritmo. En la Ecuación No. 6 se puede observar la fórmula que se utiliza para implementar el algoritmo FFT. (Marven & Ewers, 1996)

$$\tilde{X}[k] = \sum_{r=0}^{\frac{N}{2}-1} \tilde{x}[2r]\, W_{\frac{N}{2}}^{kr} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} \tilde{x}[2r+1]\, W_{\frac{N}{2}}^{kr} \quad (6)$$

Se puede observar que el largo de la sumatorias es $\frac{N}{2} - 1$ lo que implica que se reducen la cantidad de ciclos de procesamiento, el algoritmo FFT es de complejidad $O(Nlog_{10}N)$.

El algoritmo se puede representar mediante los diagramas “Mariposa”, los cuales indican el orden en que se realizan las operaciones para la obtención de la respuesta en frecuencia X[k]. En la Figura No. 6 se puede visualizar el algoritmo FFT por medio del diagrama de “Mariposa”. Los bloques representan las dos sumas que se realizan de DFT, los cuales se pueden seguir subdividiendo para obtener cuatro sumas de DFT. (Lyons, 2011)

Figura No. 6. Representación de FFT para N=8.

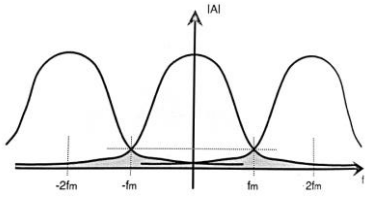


(*Lyons, 2011, pp139*)

6. Filtros digitales. El procesamiento digital de señales se puede entender como la modificación o alteración de las señales digitales, en tiempo discreto, para obtener una señal discreta con diferentes propiedades o características. Tal es el caso de los filtros digitales, los cuales se encargan de modificar el contenido en frecuencia de una señal digital, entre otras modificaciones.

a. Filtros anti-alias. Anteriormente se observó que el efecto de aliasing puede ser eliminado mediante una selección correcta de la frecuencia de muestreo, sin embargo, existe otro método por el cual se elimina este efecto. Este filtro obtiene su nombre debido a la función que cumple al ser implementado, evita el efecto conocido como aliasing. Pudiendo ser implementando como un filtro analógico o digital, mayormente se implementa de forma analógica. Conocido como un filtro pasa bajas, el cual atenuará las frecuencias por encima de su frecuencia de corte y amplificará las frecuencias menores a su frecuencia de corte. Expongamos el caso en que se muestrea una señal a una frecuencia f_m, si la señal posee un ancho de banda que supere esta frecuencia, entonces se genera aliasing, como se puede observar en la Figura No. 7. (Marven & Ewers, 1996)

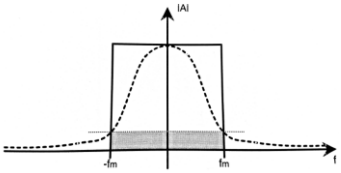
Figura No. 7. Efecto de alias en dominio de frecuencia, al muestrear a f_m.



(*Marven & Ewers, 1996. pp40*)

Es claro que se puede eliminar este efecto mediante la modificación de la frecuencia de muestreo, f_m, sin embargo, existe la posibilidad de aplicar un filtro pasa bajas a la señal con frecuencia de corte f_m, para evitar que frecuencias mayores a la frecuencia de muestreo se traslapen a frecuencias bajas. El efecto de este filtro anti-alias se puede ver en la Figura No. 8. (Marven & Ewers, 1996)

Figura No. 8. Filtro anti-alias aplicado a una señal a frecuencia de corte f_m.

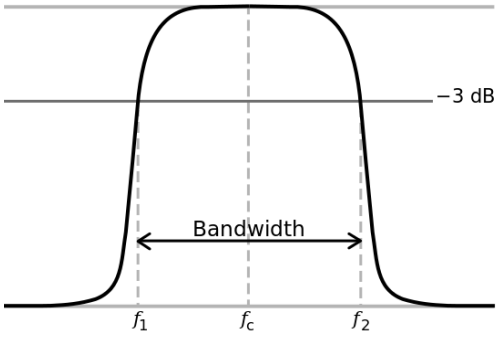


(*Marven & Ewers, 1996. pp41*)

b. Filtros pasa-banda. El filtro de tipo pasa-banda es un filtro que se define mediante la banda de paso, la cual dejará pasar todas las frecuencias que se encuentren dentro de esta banda de paso, y atenuará las frecuencias que se encuentren afuera de esta banda de paso. En la Figura No. 9 se puede observar la respuesta en frecuencia de un filtro Pasa-banda. La banda de paso se define entre dos frecuencias de corte, f_1 y f_2, por lo tanto, el ancho de banda será la diferencia de estas dos frecuencias. El diseño de estos filtros incluye un factor de calidad Q, el cual establece si el filtro es de banda angosta o de banda ancha. El factor Q se puede visualizar en la Ecuación No. 7. (Lyons, 1997)

$$Q = \frac{f_c}{|f_1 - f_2|} \quad (7)$$

Figura No. 9. Respuesta en frecuencia del filtro pasa-banda con frecuencias de corte y frecuencias de ancho de banda.



(*Agarwal & Lang, 2005, En: https://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/Bandwidth.svg/542px-Bandwidth.svg.png*)

c. Filtros de respuesta infinita al impuso. Los filtros se pueden clasificar por medio de la duración de la repuesta al impulso, para el caso de los filtros con respuesta infinita al impulso la duración es infinita. Esto quiere decir que la función “h[n]” posee infinitos términos no ceros a lo largo del eje “n”. Los filtros también son llamadas “IIR” por sus siglas en inglés: Infinite Impulse Response. El procedimiento para la creación de los filtros IIR es a partir de filtros analógicos que fueron discretizados o digitalizados. (Lyons, 2011)

Por lo tanto, existen consideraciones ya que se requiere que la respuesta al impulso del filtro analógico “$H_a(s)$” se convierta a una señal discreta “$H_a(z)$”. Lo anterior implica que existen dos consideraciones al realizar la transformación.

- El mapeo del plano “s” hacia el interior del circulo unitario en el plano “z” debe ser posible para poder asegurar que la respuesta en frecuencia del filtro analógico no sufra modificaciones el realizar la transformación.

- El filtro analógico debe de ser estable y, por lo tanto, al realizar la transformación aún debe de ser estable en el nuevo dominio.

Algunos de los métodos por los cuales se diseñan los filtros IIR a partir del dominio continuo son los siguientes:

1) Método del Impulso Invariante. Este método cumple con la Ecuación No. 8, cuya idea es la discretización de la respuesta al impulso analógica del filtro, "$H_a(s)$", mediante el muestreo y digitalización para obtener la respuesta al impulso digital "h[n]".

$$h[n] = h_a(nT) \quad (8)$$

Esta nueva respuesta al impulso discreta implica que se debe relacionar la transformada de Laplace con la transformada en Z, lo que se evidencia con la Ecuación No. 9. (Oppenhaim, A & Schafer, 1975)

$$H(z)|_{z=e^{sT}} = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(s + j\frac{2\pi}{T}k\right) \quad (9)$$

La expresión $z = e^{sT}$ es la encargada de realizar el mapeo desde el plano "s" al plano "z", mapeando el lado izquierdo del plano "s" al interior del circulo unitario en el plano "z", y el lado derecho del plano "s" al exterior del circulo unitario en el plano "z".

2) Transformación bilineal. Este método es un mapeo del dominio "s" al dominio "z". Utilizando la Ecuación No. 10 para realizar dicho mapeo.

$$z = \frac{1+\frac{2}{T}s}{1-\frac{2}{T}s} \quad (10)$$

3) Filtros Notch. Los filtros tipo Notch son filtros de tipo elimina-banda angosta. Lo que implica que atenúan la señal a cierto valor de frecuencia de corte. Su función de transferencia se puede ver en la Ecuación No. 11, e implica que su respuesta al impulso es infinita. El filtro Notch se utiliza en casos donde se requiera eliminar una frecuencia en específico, por lo cual la banda de rechazo es angosta en comparación a filtros pasa-banda en la que la banda de paso es mayor. (Padmanabham, 2000)

$$N(s) = \frac{s^2+w_n^2}{s^2+2\xi w_n s+w_n^2} \quad (11)$$

El diseño del filtro Notch se realiza según la Ecuación No. 11, eligiendo la frecuencia w_n que se desea eliminar de la señal. El factor "ξ" se relaciona con la calidad del filtro, su valor normalmente se debe asemejar a 1.

d. Filtros de respuesta finita al impulso. Los filtros cuya respuesta al impulso sea de duración finita, serán conocidos como FIR, implicando que poseen una duración infinita de términos que son ceros en su respuesta al impulso. Relacionados a la estabilidad, todos los filtros FIR son BIBO estables. Los filtros FIR se pueden describir mediante la Ecuación No. 12.

$$H(z) = \sum_{n=0}^{N-1} h[n]z^{-n} \quad (12)$$

La Ecuación No. 11 además de indicar que los filtros FIR son causales por tener el índice inicial de la sumatoria en n=0, indican que serán polinomios de "z^{-n}", llegando hasta un grado de N-1, siendo N el número de muestras que se tienen y que provienen de la transformada discreta de Fourier. Realizando la sustitución, $z = e^{jw}$, se genera la Ecuación No. 13 para obtener la respuesta en frecuencia del filtro. (Oppenhaim, A & Schafer, 1975)

$$H(e^{jw}) = \sum_{n=0}^{N-1} h[n]e^{-jwn} \quad (13)$$

Por medio de la Ecuación No. 14, se puede establecer si el filtro FIR es de fase lineal.

$$h[n] = h[N-1-n] \quad (14)$$

1) Método de ventanas. Este método realiza el diseña del filtro FIR mediante el truncamiento de una respuesta al impulso infinita ideal hacia una repuesta finita, por medio de una convolución entre una señal de "ventana" que realiza el truncamiento para poder generar el filtro FIR. Supongamos que se tiene una respuesta infinita al impulso ideal, Ecuación No. 15.

$$H_d(e^{jw}) = \sum_{n=-\infty}^{\infty} h_d[n]e^{-jwn} \quad (15)$$

La Ecuación No. 14, por medio de la fórmula de transformada de Fourier se puede encontrar la secuencia de la respuesta al impulso, Ecuación No. 16.

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{jw})e^{jwn}dw \quad (16)$$

La secuencia que se representa con la Ecuación No. 15 es de duración infinita, para poder obtener una función finita, se puede redefinir por partes esta secuencia, como se aprecia a continuación en la Ecuación No. 17.

$$h[n] = \begin{cases} h_d[n], & 0 \le n < N-1 \\ 0, & otro\ caso \end{cases} \quad (17)$$

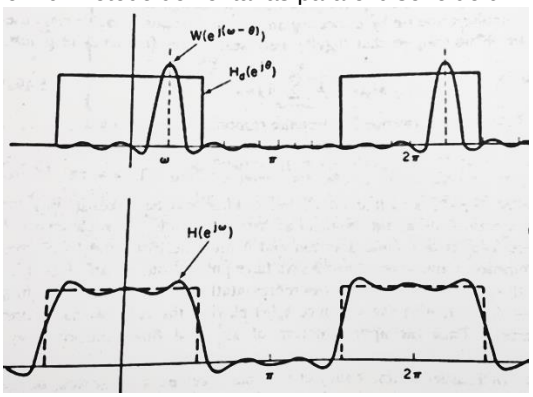
$$w[n] = \begin{cases} 1, & 0 \le n < N-1 \\ 0, & otro\ caso \end{cases} \quad (18)$$

Por lo tanto, la nueva función será: “h[n]=h_d[n]w[n], en el que la función w[n], Ecuación No.19, será la función ventana con la que se realizará la convolución. La fórmula que se generará para la convolución entre estas dos señales será la siguiente, Ecuación No. 19.

$$H(e^{jw}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\theta}\right)W\left(e^{j(w-\theta)}\right)d\theta \quad (19)$$

El efecto de estas ecuaciones se puede visualizar en la Figura No. 10. Se puede apreciar que teniendo “H_d” como una secuencia infinita, el realizar la convolución con “W” o ventana se genera nuestra respuesta al impulso finita “H”. (Oppenhaim, A & Schafer, 1975)

Figura No. 10. Método de ventanas para el diseño de un filtro FIR.



(*Oppenhaim, A & Schafer, 1975*)

a) Ventana Kaiser. La ventana de Kaiser se encuentra definida por la Ecuación No. 20. En la cual se puede definir la amplitud de la banda lateral de frecuencias y el ancho de la banda principal. Las características de diseño de esta ventana permiten una mayor variación en el diseño de filtros. Sin embargo, todo este manejo se realizará por medio de la herramienta de diseño. (Paolini, & Chierchie,2016)

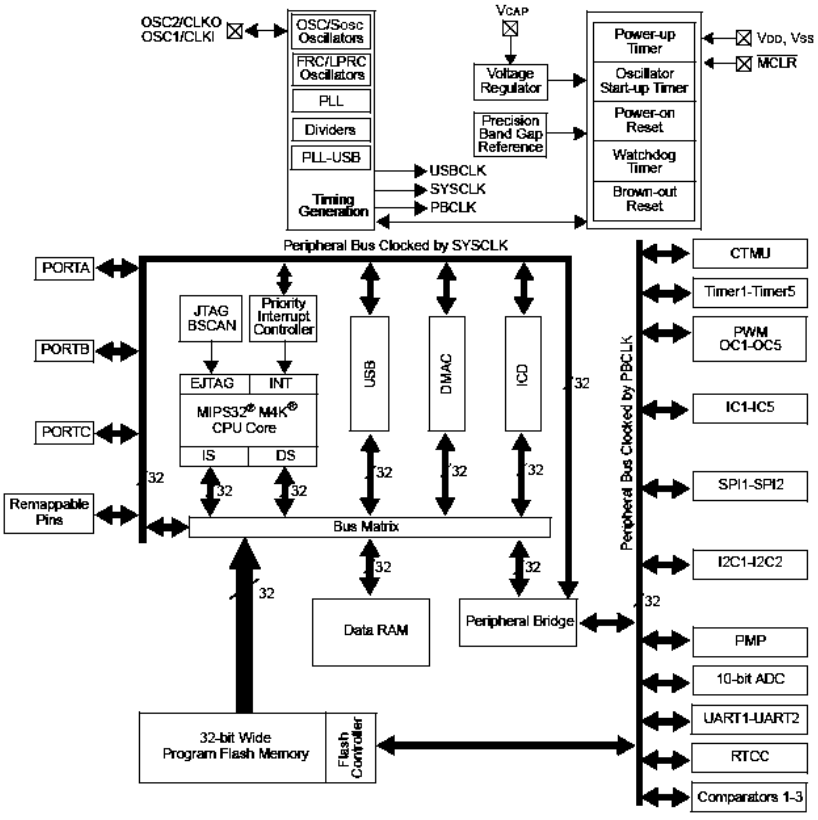
$$w[n] = \frac{I_o\left[B\sqrt{1-\left(\frac{n-\frac{N}{2}}{\frac{N}{2}}\right)^2}\right]}{I_o(B)}, 0 \leq n \leq N-1 \quad (20)$$

e. Herramienta de diseño de filtros de Matlab. El software de Matlab contiene una herramienta para el diseño de filtros digitales llamado “Filter Design and Analysis”. En esta herramienta se puede especificar los tipos de filtros a diseñarse, de tipo FIR o IIR, así como también los métodos por los cuales se desea construir el filtro. Para el caso de filtros FIR se posee la opción de especificación del método de “Ventanas” y para el caso de los filtros IIR se especifica la opción de filtros Notch. La herramienta también contiene la opción para exportar los coeficientes de los filtros diseñados para su implementación.

7. Microcontroladores de 32 bits Microchip. La serie PIC32, de la empresa Microchip, contiene microcontroladores con un set de instrucciones de 32bits. Las propiedades de esta familia radican en módulos especializados para el procesamiento de información de: audio, gráficos, interconectividad inalámbrica y alámbrica, módulos periféricos digitales y analógicos, entre otros. Así como también de una mayor velocidad de reloj y un set de instrucciones. En este trabajo nos enfocaremos en la familia PIC32MX.

a. PIC32mx. Esta familia de microcontroladores trabaja con un set de instrucciones de 32 bits, y posee una arquitectura basada en un bus periférico de 32bits, el cual conecta: los módulos periféricos, comunicaciones en serial, y una matriz de bus principal de 32bits que conecta: el bus periférico, el núcleo CPU, la memoria FLASH del programa, módulos de conversión analógico-digital, módulos USB, memoria RAM, ver en Figura No. 11. (Microchip, 2016)

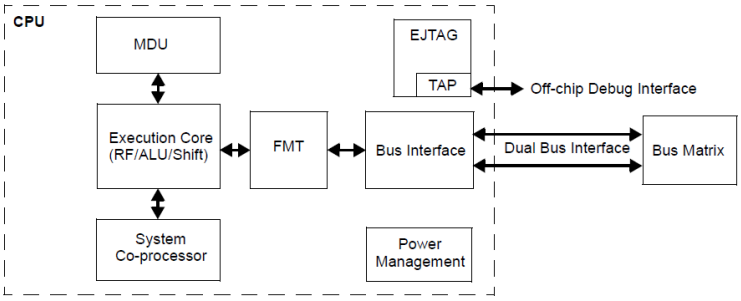
Figura No. 11. Arquitectura de familia PIC32mx.



(*Microchip, 2016, pp19*)

1) Núcleo. El núcleo de la familia PIC32mx se encuentra compuesto por: el CPU, la unidad aritmética y lógica, 32 registros de propósito general de 32 bits cada uno, un pipeline de cinco etapas, una arquitectura de instrucciones MIPS32 con versión de lanzamiento 2. El CPU es de versión MIPS32 M4K. En la Figura No. 12 se encuentra el diagrama de bloques del núcleo de la familia PIC32mx. El núcleo también cuenta con el protocolo de depuración EJTAG de módulos de hardware. El sistema de control de coprocesadores se encarga del mapeo de direcciones virtuales a físicas. (Microchip, 2016)

Figura No. 12. Diagrama de bloques del núcleo de la familia PIC32mx.



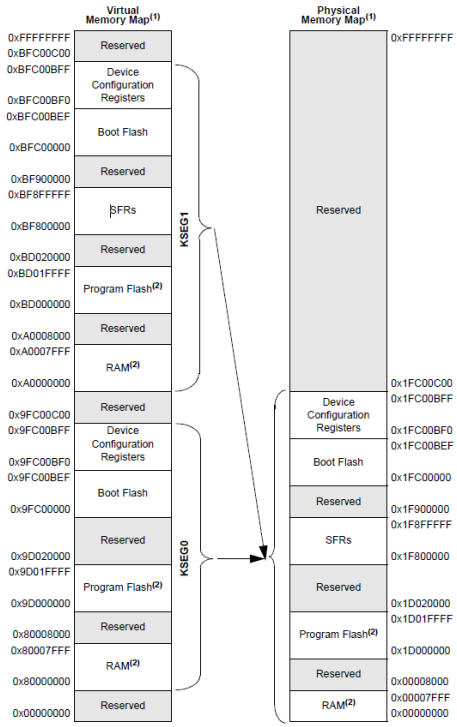
(*Microchip, 2016, pp33*)

2) Organización de memoria. La memoria del microcontrolador se encuentra dividida en memoria virtual y memoria física. El mapeo de la memoria virtual a la física se realiza por medio del sistema de control de coprocesadores. El microcontrolador PIC32mx250F128B cuenta con 128KB de memoria flash para el almacenamiento del programa y de 32KB para memoria de datos. En la figura No. 13 se encuentra el mapa de memoria virtual y física para el microcontrolador.

Para el control de la matriz de bus, existe un arreglo de registros de control, los cuales indican si existen errores en el direccionamiento y acceso de la memoria del microcontrolador. Estos registros se encuentran a partir de la dirección virtual 0xBF882000.

Los registros especiales de funciones o SFR, por sus siglas en inglés, se encargan del manejo del microcontrolador y de los módulos, en la Figura No. 14 se puede visualizar las direcciones virtuales para el acceso de los registros especiales de funciones.

Figura No. 13. Mapa de memoria de la familia PIC32mx150/250



(*Microchip, 2016, pp41*)

Figura No. 14. Mapa de direcciones de registros especiales de funciones.

Peripheral	Virtual Address	
	Base	Offset Start
Watchdog Timer	0xBF80	0x0000
RTCC		0x0200
Timer1-5		0x0600
Input Capture 1-5		0x2000
Output Compare 1-5		0x3000
IC1 and IC2		0x5000
SPI1 and SPI2		0x5800
UART1 and UART2		0x6000
PMP		0x7000
ADC		0x9000
CVREF		0x9800
Comparator		0xA000
CTMU		0xA200
Oscillator		0xF000
Device and Revision ID		0xF220
Peripheral Module Disable		0xF240
Flash Controller		0xF400
Reset		0xF600
PPS		0xFA04
Interrupts	0xBF88	0x1000
Bus Matrix		0x2000
DMA		0x3000
USB		0x5050
PORTA-PORTC		0x6000
Configuration	0xBFC0	0x0BF0

(*Microchip, 2016, pp44*).

3) Módulos. La familia PIC32mx posee módulos que permiten la intercomunicación del microcontrolador con el resto de dispositivos de hardware. Los siguiente s módulos son los relevantes para el desarrollo del prototipo.

a) Módulo de convertidor analógico-digital. El módulo de conversión analógico-digital, ADC, realiza la conversión de una señal analógica continua a datos digitales que se almacenan en el microcontrolador. Para la configuración de los puertos analógica, se debe de ubicar los registros Anselx y Trisx de los puertos A, B o C. La conversión es de resolución de 10-bits y con velocidad de 1 millón de muestras por segundo.

Se encuentran disponibles 13 canales de recepción analógica, con un único circuito de muestreo. Le elección del reloj para realizar el muestreo proviene de multiplicador por hardware del microcontrolador, los registros para ajustar el tiempo de muestreo son ADCS, ADCR, PBDIV, PLLMULT, COSCC. (Microchip, 2016)

b) Módulo de entradas y salidas digitales. El módulo periférico de la familia PIC32mx consta de tres puertos: puerto A, puerto B y Puerto C, cada uno es de 32 bits de tamaño y poseen los registros: Anselx, Trisx, Latx, ODCx, Cstatx, entre otros. Estos registros se encargan del control analógico o digital del pin, el cambio de estado del pin, la habilitación de interrupciones debido al cambio de estado en el pin, entre otras funciones. Las direcciones de los registros virtuales A, B y C son: 0xBF886000, 0xBF886100 y 0xBF886200 respectivamente. Los voltajes de funcionamiento de los pines son de 3.3voltios y el total de pines de entrada o salida es de 28, entre los cuales se debe de descartar los pines de alimentación o pines especiales. En la Figura No. 15 se puede observar el mapeo de los pines del microcontrolador PIC32mx.

Figura No. 15. Nombre de pines para microcontrolador PIC32mx250F128B.

Pin #	Full Pin Name	Pin #	Full Pin Name
1	MCLR	15	VBUS
2	PGED3/VREF+/CVREF+/AN0/C3INC/RPA0/CTED1/PMD7/RA0	16	TDI/RPB7/CTED3/PMD5/INT0/RB7
3	PGEC3/VREF-/CVREF-/AN1/RPA1/CTED2/PMD6/RA1	17	TCK/RPB8/SCL1/CTED10/PMD4/RB8
4	PGED1/AN2/C1IND/C2INB/C3IND/RPB0/PMD0/RB0	18	TDO/RPB9/SDA1/CTED4/PMD3/RB9
5	PGEC1/AN3/C1INC/C2INA/RPB1/CTED12/PMD1/RB1	19	VSS
6	AN4/C1INB/C2IND/RPB2/SDA2/CTED13/PMD2/RB2	20	VCAP
7	AN5/C1INA/C2INC/RTCC/RPB3/SCL2/PMWR/RB3	21	PGED2/RPB10/D+/CTED11/RB10
8	VSS	22	PGEC2/RPB11/D-/RB11
9	OSC1/CLKI/RPA2/RA2	23	VUSB3V3
10	OSC2/CLKO/RPA3/PMA0/RA3	24	AN11/RPB13/CTPLS/PMRD/RB13
11	SOSCI/RPB4/RB4	25	CVREFOUT/AN10/C3INB/RPB14/VBUSON/SCK1/CTED5/RB14
12	SOSCO/RPA4/T1CK/CTED9/PMA1/RA4	26	AN9/C3INA/RPB15/SCK2/CTED6/PMCS1/RB15
13	VDD	27	AVSS
14	TMS/RPB5/USBID/RB5	28	AVDD

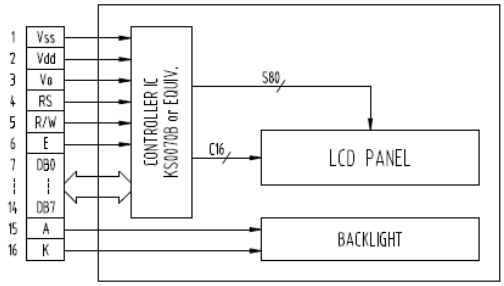
(*Microchip, 2016, pp5*)

En la familia PICe32mx se tiene la posibilidad de realizar un mapeo de pines periféricos, también llamado Selección de Pines Periféricos, PPS. Los pines que indiquen un prefijo “RPn” son los pines que se les puede realizar un mapeo. Este mapeo permite, por medio de software, modificar la salida física de algún pin, evitando el rediseño por medio de hardware. (Microchip, 2016)

b. Pantalla LCD 1602a. Se utilizará la pantalla LCD 1602a monocromática de la empresa East Rising. Esta pantalla cuenta con dos filas y 16 columnas para el despliegue de los caracteres. En la Figura No. 16 se puede ver el diagrama de bloques de la pantalla LCD 1602a. (EastRising, 2016)

La comunicación que admite la pantalla es a través de comunicación en paralelo por medio de los pines DB0 a DB7 en el controlador KS0070B de la pantalla LCD. Esta pantalla requerirá de un puerto de 8 bits para el manejo de la misma, así también de comandos para los pines de Enable, Reset, Read/Write.

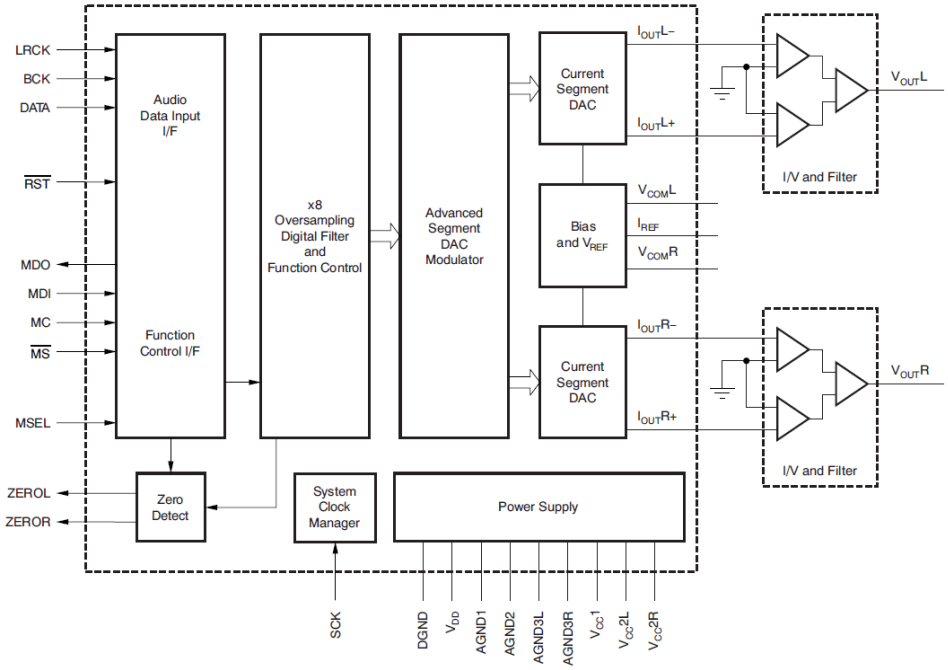
Figura No. 16. Diagrama de bloques de pantalla LCD con 16 columnas y 2 filas.



(*East Rising, 2016, pp3*)

c. Módulo DAC PCM1795 de *Texas Instruments*. Después de generar la interfaz con el usuario para que interactué con el afinador se considera un módulo para generar una salida del afinador que se pueda conectar a un amplificador, una salida denominada “Line Out”. La conversión digital se realizará mediante el circuito integrado PCM1795. La comunicación se realiza mediante el protocolo I^2C o SPI el cual reduce la cantidad de pines requeridos en el microcontrolador. Así también se tiene una resolución de 32 bits para la conversión Digital-Analógica y un máximo de 200KHz de frecuencia de muestreo. En la Figura No. 17 se muestra el diagrama de bloques del chip PCM1795. (*Texas Instruments*, 2015)

Figura No. 17. Diagrama de bloques de DAC PCM1795.



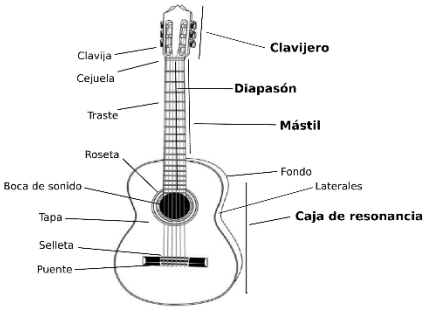
(*Texas Instruments, 2015, pp17*)

B. INSTRUMENTO MUSICAL DE GUITARRA

El instrumento musical conocido como guitarra proviene desde el año 1000 a.C., la primera pieza arqueológica estaba compuesta por una caja de resonancia con un vestigio de cuerda que amplifica el sonido de la cuerda. El cuál es la base de las guitarras actuales. La guitarra costa de 14 partes o secciones, las cuales cumplen con diferentes propósitos.

En la Figura No. 18 se puede apreciar las partes de la guitarra acústica.

Figura No. 18. Partes de la guitarra



(*CursosXniveles, 2013, en:https://cursoxniveles.files.wordpress.com/2013/03/claspart.*png)

1. Clasificaciones de las guitarras. Las guitarras se pueden clasificar por los componentes por que utilizan para amplificar el sonido de las cuerdas.

Guitarra acústica. Este tipo de guitarra utiliza únicamente la caja de resonancia para poder amplificar el sonido de las cuerdas vibrando.

Guitarra electroacústica. Posee las mismas partes que la guitarra acústica, pero incluye componentes electrónicos para poder obtener una señal eléctrica de las cuerdas tocadas.

Guitarra eléctrica. Esta guitarra posee las mismas partes que la Figura No. 18, sin embargo, carece de una caja de resonancia que amplifica la señal de las cuerdas mediante un dispositivo eléctrico. También posee un fondo con menor tamaño y son más esbeltas que el resto de guitarras, además de tener controles mecánicos para modificar la señal eléctrica de la guitarra. En la Figura No. 19 se puede apreciar una guitarra eléctrica. (Guitarristas, 2011)

Figura No. 19. Partes de guitarra eléctrica.

(*Lacuerda, 2016, en: http://lacuerda.net/Recursos/cursoguitarra/?page=2*)

La guitarra cuenta con un mástil, sobre el cual se colocan las seis cuerdas, la 6ta. cuerda es la que tiene mayor grosor, y la 1ra. cuerda es la que tiene el menor grosor. El primer traste es el que se encuentra más cercano a la tapa del alma, y aumentan los trastes con forme se acercan a las pastillas de la guitarra. Para visualizar la organización de los trastes y cuerdas, la Figura No. 20 muestra un diagrama de trastes y cuerdas en la guitarra. (Molina, 2016)

Figura No. 20. Diagrama de trastes y cuerdas en la guitarra acústica o eléctrica.

(*Molina, 2016*)

Generación de señales eléctricas de guitarra eléctrica. Como se expuso en secciones pasadas, las guitarras eléctricas no utilizan una caja de resonancia para amplificar la vibración de las cuerdas. En la Figura No. 21 se puede visualizar las “pastillas” o “PickUps”, estos elementos son eléctricos. Consisten en bobinas con cobre alrededor de un imán. Al momento de vibrar alguna cuerda, por ser metálica induce una corriente en el cable, por lo tanto, se genera una corriente eléctrica, la cual es una señal analógica de voltaje, que es la salida de la línea de la guitarra eléctrica. (Dawsons, 2013)

Figura No. 21. Pastillas o “Pick-ups” marca Dogear P90.

(*Dawsons, 2013*)

2. Estándar de afinación E. Actualmente se ha definido un estándar que estable las frecuencias que deben de tener las notas de una guitarra para sonar adecuadamente. En este trabajo se estará utilizando el estándar que comúnmente se denomina E, debido a que en el idioma ingles las notas que se ejecutan al tocar las seis cuerdas al aire son, empezando por la primera cuerda: E, B, G, D, A, E. En otras palabras, se ejecutan las notas: Mi, Si, Sol, Re, La, y Mi. El estándar también define que la nota A4 debe de sonar a 440Hz, lo que equivale a que las frecuencias de las notas al aire de las cuerdas sean las del Cuadro No. 1. (Guitar Book, 2012)

Cuadro No. 1. Frecuencias de las cuerdas tocadas sin presionar algún traste según el estándar de A4=440Hz.

Cuerda/nota	Frecuencia (Hz)
1ra. /Mi	329.63
2da. /Si	246.94
3ra. /Sol	196.00
4ta. /Re	146.83
5ta. /La	110.00
6ta. /Mi	82.41

(*Guitar Book, 2012*)

a. Métodos de afinación de guitarra eléctrica. La afinación de la guitarra es un proceso indispensable al momento de ejecutar el instrumento, ya que asegura que las notas que suenen cumplen con cierto estándar definido para cada nota. Se expondrán dos métodos para afinación de la guitarra.

1) Afinación manual. Este método requiere que la 6ta. cuerda se encuentre afinada según el estándar de afinación para A4=440Hz. Se utiliza la Figura No. 20 para referencia de los trastes y de las cuerdas, ejecutadas sin presionar sobre algún traste.

- Paso 1: con la 6ta. cuerda afinada, se toca en el 5to. traste, y esta debe de sonar igual que la 5ta. cuerda tocada sin presionar algún traste. Mover la clavija de la 5ta. cuerda hasta que suenen con la misma tonalidad.
- Paso 2: con la 5ta. cuerda afinada, se toca en el 5to. traste, y esta debe de sonar igual que la 4ta. cuerda tocada sin presionar algún traste. Volver a girar la clavija de la 4ta. cuerda hasta que suenen igual.
- Paso 3: con la 4ta. cuerda afinada, se toca en el 5to. traste, y esta debe de sonar igual que la 3ra. cuerda tocada sin presionar algún traste. Ajustar clavija de la 3ra. cuerda para que suenen igual.

- Paso 4: con la 3ra. cuerda afinada, se toca en el 4to. traste, y esta debe de sonar igual que la 2da. cuerda tocada sin presionar algún traste. Girar la clavija de la 2da. cuerda hasta que suenen igual.
- Paso 5: con la 2da. cuerda afinada, se toca en el 5to. traste, y esta debe de sonar igual que la 1ra. cuerda tocada sin presionar algún traste. Ajustar la clavija de la 1ra. cuerda hasta que suenen igual. (Guitar Book, 2012)

2) Afinación con afinador. Este método no requiere que la 6ta. cuerda se encuentre afinada.

- Paso 1: selección de la cuerda que se desea afinar en el afinador.
- Paso 2: tocar la cuerda sin presionar algún traste.
- Paso 3: observar el afinador para establecer si la cuerda se encuentra afinada.
- Paso4: en caso no se encuentre afinada la cuerda, girar la clavija de la cuerda que se tocó y observar el afinador para establecer si se requiere de algún otro ajuste en la clavija.
- Paso 5: si el afinador indica que la cuerda se encuentra afinada, seleccionar otra cuerda a afinar, y tocar si presionar algún traste dicha cuerda, repitiendo los pasos anteriores hasta afinar las seis cuerdas de la guitarra. (Guitar Book, 2012)

3. Afinadores digitales. Actualmente existen afinadores digitales los cuales por medio de las ondas de sonido o de las vibraciones logran determinar si las cuerdas de la guitarra se encuentran afinadas o no. Los métodos por los cuales logran determinar la afinación de la guitarra es mediante el análisis de las frecuencias de la señal de la cuerda que está vibrando.

Los afinadores digitales se encuentran se pueden clasificar por el método de captación de la señal de audio.

a. Captación por micrófono. Este afinador utiliza las ondas sonoras generadas por la caja de resonancia de la guitarra para captar la señal a analizar. El micrófono utilizado deberá de cumplir con requerimientos funcionales para poder captar el sonido de la guitarra, siendo un transductor que traduce una señal de vibración sonora hacia una señal eléctrica. (Guitar Book, 2012)

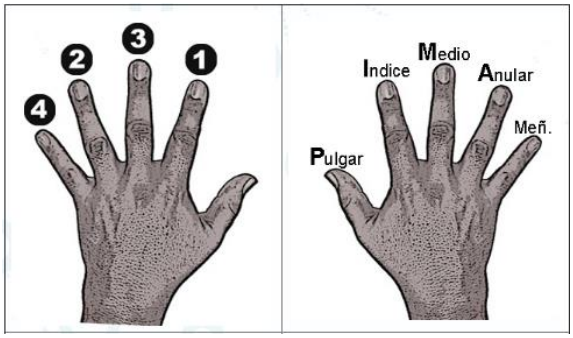
b. Captación por vibración. El afinador requiere de un transductor que se pegue a la guitarra, normalmente se pega al mástil de la guitarra, donde se encuentran las clavijas. Este transductor traduce las vibraciones hacia una señal eléctrica. Las vibraciones de las cuerdas se

propagan en el aire, pero también se propagan hacia el cuerpo de la guitarra, el cual como se encuentra conectado al mástil de la guitarra, logra llegar hacia el transductor. (Guitar Book, 2012)

4. Señales de audio. Las señales de audio que son utilizadas y que el oído humano puede captar se encuentran en el rango de 20Hz hasta 20,000Hz. Las señales pueden ser graficadas en el domino del tiempo, o en el dominio de frecuencia. La señal que se obtiene de la guitarra eléctrica es una señal de voltaje continua en tiempo. Esta señal continua o analógica porta toda la información referente a las vibraciones de la cuerda, o de las cuerdas que se estén tocando en ese instante de tiempo. (Unad, 2016)

5. Numeración de mano. Debido a la variedad de posiciones que se pueden ejecutar al momento de tocar alguna nota en la guitarra se ha definido una numeración de los dedos de la mano izquierda. De esta manera se facilita la expresión de las notas y/o acordes al momento de ejecutar alguna pieza musical. En la Figura No. 22 se puede observar esta convención. (Molina, 2016)

Figura No. 22. Convención de la numeración de los dedos de la mano izquierda para ejecutar la guitarra.



(*Molina, 2016, En: http://aprendeenlinea.udea.edu.co/lms/moodle/mod/page/view.php?id=105867&inpopup=1*)

6. Acordes del círculo de Do. Una nota tocada en guitarra es presionar una única cuerda en un único traste, y tocar la cuerda en la boca de la guitarra. Sin embargo, se ha establecido el acorde, que es el tocar más de una cuerda a la vez, al mismo tiempo que se presionas diferentes notas en los trastes. El propósito de los círculos es agrupar acordes que tengan semejanza alguna para facilitar la ejecución de alguna pieza musical. (Lopez, 2005)

El círculo de Do se compone de cuatro acordes, los cuales son:

- Do Mayor:
  - 6ta. cuerda: no se toca.
  - 5ta. cuerda: tercer traste.
  - 4ta. cuerda: segundo traste.
  - 3ra. cuerda: tocar sin presionar algún traste.
  - 2da. cuerda: primer traste.
  - 1ra. cuerda: tocar sin presionar algún traste.

- La Mayor:
  - 6ta. cuerda: no se toca.
  - 5ta. cuerda: tocar sin presionar algún traste.
  - 4ta. cuerda: segundo traste.
  - 3ra. cuerda: segundo traste.
  - 2da. cuerda: segundo traste.
  - 1ra. cuerda: tocar sin presionar algún traste.

- Mi Mayor:
  - 6ta. cuerda: tocar sin presionar algún traste.
  - 5ta. cuerda: segundo traste.
  - 4ta. cuerda: segundo traste.
  - 3ra. cuerda: primer traste.
  - 2da. cuerda: tocar sin presionar algún traste.
  - 1ra. cuerda: tocar sin presionar algún traste.

- Sol Mayor:
  - 6ta. cuerda: tercer traste.
  - 5ta. cuerda: segundo traste.
  - 4ta. cuerda: tocar sin presionar algún traste.
  - 3ra. cuerda: tocar sin presionar algún traste.
  - 2da. cuerda: tocar sin presionar algún traste.
  - 1ra. cuerda: tercer traste.

En el Cuadro No. 2 se puede observar las frecuencias para las notas de la guitarra obtenidas según la afinación de la nota “La” a 440Hz, que es el estándar E. A partir de esta información y de los acordes se puede establecer los Cuadros No. 5, 6, 7, 8 y 9 para los acordes del círculo de Do y la afinación para las cuerdas según las frecuencias máximas y mínimas que se aceptan antes de que se interfiera con la frecuencia de otra nota.

Cuadro No. 2. Frecuencias teóricas de notas musicales.

Nota	Frecuencia (Hz)	Largo de onda (cm)	Nota	Frecuencia (Hz)	Largo de onda (cm)	Nota	Frecuencia (Hz)	Largo de onda (cm)
C_0	16.35	2109.89	C_3	130.81	263.74	$C^{\#}_6/D^{b}_6$	1108.73	31.12
$C^{\#}_0/D^{b}_0$	17.32	1991.47	$C^{\#}_3/D^{b}_3$	138.59	248.93	D_6	1174.66	29.37
D_0	18.35	1879.69	$D^{\#}_3/E^{b}_3$	155.56	221.77	$D^{\#}_6/E^{b}_6$	1244.51	27.72
$D^{\#}_0/E^{b}_0$	19.45	1774.2	E_3	164.81	209.33	E_6	1318.51	26.17
E_0	20.6	1674.62	F_3	174.61	197.58	F_6	1396.91	24.7
F_0	21.83	1580.63	$F^{\#}_3/G^{b}_3$	185	186.49	$F^{\#}_6/G^{b}_6$	1479.98	23.31
$F^{\#}_0/G^{b}_0$	23.12	1491.91	G_3	196	176.02	G_6	1567.98	22
G_0	24.5	1408.18	$G^{\#}_3/A^{b}_3$	207.65	166.14	$G^{\#}_6/A^{b}_6$	1661.22	20.77
$G^{\#}_0/A^{b}_0$	25.96	1329.14	A_3	220	156.82	A_6	1760	19.6
A_0	27.5	1254.55	$A^{\#}_3/B^{b}_3$	233.08	148.02	$A^{\#}_6/B^{b}_6$	1864.66	18.5
$A^{\#}_0/B^{b}_0$	29.14	1184.13	B_3	246.94	139.71	B_6	1975.53	17.46
B_0	30.87	1117.67	C_4	261.63	131.87	C_7	2093	16.48
C_1	32.7	1054.94	$C^{\#}_4/D^{b}_4$	277.18	124.47	$C^{\#}_7/D^{b}_7$	2217.46	15.56
$C^{\#}_1/D^{b}_1$	34.65	995.73	D4	293.66	117.48	D_7	2349.32	14.69
D_1	36.71	939.85	$D^{\#}_4/E^{b}_4$	311.13	110.89	$D^{\#}_7/E^{b}_7$	2489.02	13.86
$D^{\#}_1/E^{b}_1$	38.89	887.1	E_4	329.63	104.66	E_7	2637.02	13.08
E1	41.2	837.31	F_4	349.23	98.79	F_7	2793.83	12.35
F_1	43.65	790.31	$F^{\#}_4/G^{b}_4$	369.99	93.24	$F^{\#}_7/G^{b}_7$	2959.96	11.66
$F^{\#}_1/G^{b}_1$	46.25	745.96	G_4	392	88.01	G_7	3135.96	11
G_1	49	704.09	$G^{\#}_4/A^{b}_4$	415.3	83.07	$G^{\#}_7/A^{b}_7$	3322.44	10.38
$G^{\#}_1/A^{b}_1$	51.91	664.57	A_4	440	78.41	A_7	3520	9.8
A_1	55	627.27	$A^{\#}_4/B^{b}_4$	466.16	74.01	$A^{\#}_7/B^{b}_7$	3729.31	9.25
$A^{\#}_1/B^{b}_1$	58.27	592.07	B_4	493.88	69.85	B_7	3951.07	8.73
B_1	61.74	558.84	C_5	523.25	65.93	C_8	4186.01	8.24
C_2	65.41	527.47	$C^{\#}_5/D^{b}_5$	554.37	62.23	$C^{\#}_8/D^{b}_8$	4434.92	7.78
$C^{\#}_2/D^{b}_2$	69.3	497.87	D_5	587.33	58.74	D_8	4698.63	7.34
D_2	73.42	469.92	$D^{\#}_5/E^{b}_5$	622.25	55.44	$D^{\#}_8/E^{b}_8$	4978.03	6.93
$D^{\#}_2/E^{b}_2$	77.78	443.55	E_5	659.25	52.33	E_8	5274.04	6.54
E_2	82.41	418.65	F_5	698.46	49.39	F_8	5587.65	6.17
F_2	87.31	395.16	$F^{\#}_5/G^{b}_5$	739.99	46.62	$F^{\#}_8/G^{b}_8$	5919.91	5.83
$F^{\#}_2/G^{b}_2$	92.5	372.98	G5	783.99	44.01	G_8	6271.93	5.5
G_2	98	352.04	$G^{\#}_5/A^{b}_5$	830.61	41.54	$G^{\#}_8/A^{b}_8$	6644.88	5.19
$G^{\#}_2/A^{b}_2$	103.83	332.29	A_5	880	39.2	A_8	7040	4.9
A_2	110	313.64	$A^{\#}_5/B^{b}_5$	932.33	37	$A^{\#}_8/B^{b}_8$	7458.62	4.63
$A^{\#}_2/B^{b}_2$	116.54	296.03	B_5	987.77	34.93	B_8	7902.13	4.37
B_2	123.47	279.42	C_6	1046.5	32.97			

(*Suits, 2015*)

Debido a que las notas se encuentran en idioma inglés se necesita de su equivalente en idioma español, en el Cuadro No. 3 se visualiza la traducción de notas al idioma español.

Cuadro No. 3. Notas musicales en idioma inglés y español.

Nota en inglés	Nota en español
Do	C
Re	D
Mi	E
Fa	F
Sol	G
La	A
Si	B

(*Avila, 2016, En: http://curso-de-guitarra.guitarsimple.com/acordes-en-ingles/*)

IV. ANTECEDENTES

Actualmente existe una gran variedad de afinadores en la industria, los cuales utilizan diferentes medios de captación de las señales de la guitarra para poder establecer si se encuentra afinado o no el instrumento. Estos recolectan la señal de las cuerdas de la guitarra por medio de las ondas sonoras que estas producen al tocar el instrumento, o bien por medio de las vibraciones en el cuerpo de la guitarra al momento de que vibren las cuerdas.

En el ámbito académico guatemalteco ya se ha creado un afinador automático, el cual utiliza servomotores para ajustar las clavijas para poder afinar las cuerdas (Vargas, 2013) y un micrófono para poder captar las ondas sonoras de la guitarra. Sin embargo, el afinador fue implementado en una computadora por lo cual no resulta ser portátil. Este programa utiliza algoritmos de detección de frecuencias como lo es la transformada discreta de Fourier, para poder reconocer la frecuencia de la señal emitida por la guitarra. También se puede destacar el afinador de Chafchalaf (2013), el cual utiliza una computadora para generar un algoritmo de detección de frecuencias empleando la transformada de Fourier, para obtener la respuesta en frecuencia de la señal de la guitarra. El método por el cual se obtuvo la señal de la guitarra en este afinador fue mediante un micrófono, el cual a través de una tarjeta de sonido logró ingresar la información a la computadora para su posterior procesamiento digital.

En otros afinadores se encuentra el de Zurale (2016) el cual implementa un afinador de guitarra automático utilizando servomotores para poder ajustar las clavijas. A diferencia del afinador de guitarra de Chafchalaf, éste se implementó en un microcontrolador, aunque no se generó un afinador portátil o un módulo compacto que tuviera todos los componentes encapsulados. El método por el cual captaban la señal de la guitarra fue mediante un transductor de vibraciones de señales eléctricas, llamado un sensor piezoeléctrico. Otro afinador realizado es el de Lourde (2009), el cual no cuenta con un sistema de afinación automático y la implementación del algoritmo para detección de frecuencias esta realizado en un computador, por lo que el afinador no se considera portátil.

Entre los afinadores comerciales se encuentra el “Min-ETune” de la compañía Gibson (Gibson, 2016), el cual utiliza servomotores para controlar las clavijas y así poder afinar las cuerdas. Sin embargo, este afinador no puede ser removido de la guitarra, lo que implica que únicamente la guitarra en la cual fue instalado puede ser afinada. Incluye una gran variedad de tonos y estándares de afinación, aunque el costo del instrumento incluye el costo del afinador.

Resumiendo, la información anterior se puede resaltar que los afinadores no han sido construidos o diseñados para obtener la señal de línea de la guitarra eléctrica, sino que utilizan las vibraciones en ambiente, como el sonido, o las vibraciones en el cuerpo de la guitarra para capturar

la señal de la cuerda y así determinar la afinación del instrumento. Los afinadores determinan la afinación de la cuerda por medio de métodos de identificación de frecuencias en las señales, lo cual puede ser realizado mediante el análisis frecuencias proporcionado por la transformada de Fourier, así como también una combinación de distintos filtros.

Con referencia a la portabilidad de los afinadores, varios no cumplen dada su dependencia de un computador personal, algo que difícilmente puede ser llevado dentro de un bolsillo. Por otro lado, a pesar que algunos afinadores si fueron implementados con hardware más compacto, no lograron integrar un módulo portátil. Finalmente, la gran mayoría de afinadores cumple la función de indicar la afinación del instrumento exclusivamente, por lo que agregar la función del reconocimiento de acordes sería una ventaja sobre los afinadores antes mencionados.

V. METODOLOGÍA

Para poder alcanzar los objetivos planteados y afirmar que la guitarra se encuentra afinada, es necesario comprobar que la frecuencia de la guitarra para las notas tocadas es igual a la frecuencia teórica de la nota establecida en el Cuadro No. 1.

Primeramente, se requiere de un método para captar la señal de la guitarra, lo que implica digitalizar dicha señal.

A. Captación de señal de guitarra

La captación de la señal de la guitarra requiere de un circuito que permita realizar el muestreo de la señal eléctrica, el acondicionamiento de la señal de la guitarra para que pueda ser obtenida por el hardware del microcontrolador y finalmente un circuito que almacene esta información digitalizada.

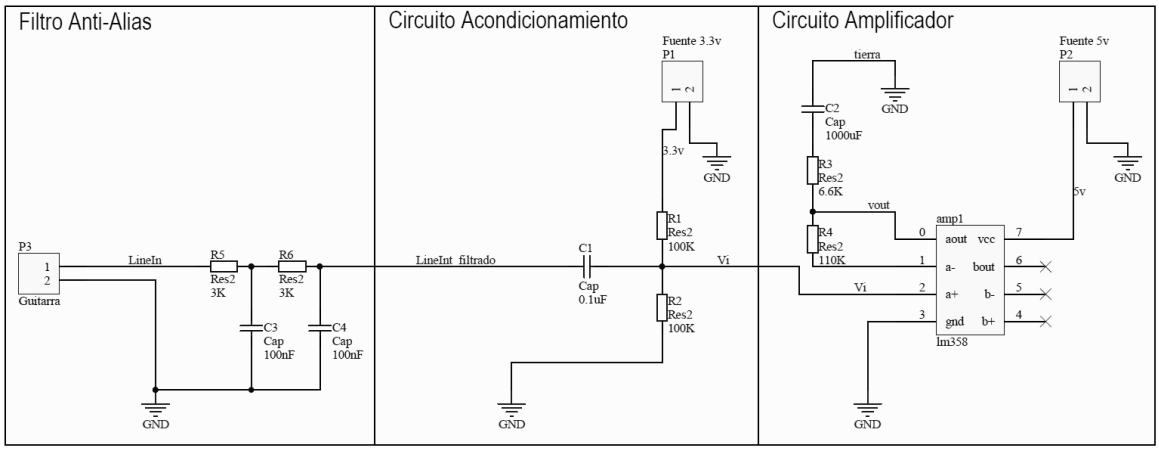
1. Acondicionamiento de señal de guitarra. La señal de la guitarra fue medida mediante un osciloscopio, con un voltaje pico a pico de 200 milivoltios (mV). Por lo tanto, se requiere que esta señal se acerque al valor aceptable de referencia del módulo ADC del microcontrolador. En el Cuadro No. 4 se puede visualizar lo voltajes máximos y mínimos de la señal de la guitarra y de la señal deseada en el pin de entrada del microcontrolador.

Cuadro No. 4. Voltajes máximos y mínimos de la señal de la guitarra y del pin de entrada permitido en el microcontrolador.

Señal	Voltaje Máximo (voltios)	Voltaje Mínimo (voltios)
Señal de guitarra	0.1	-0.1
Señal permitida en pin	3.3	0

Debido a esta diferencia de voltajes, se requiere de un acondicionamiento de la señal de la guitarra, en la Figura No. 23 se encuentra el esquemático para modificar esta señal y hacerla coincidir dentro de los parámetros permitidos para que ingrese al pin del microcontrolador. El filtro anti-alias también se encuentra dentro del circuito de amplificación y acondicionamiento de señal. Éste fue diseñado con una frecuencia de corte de 500Hz, dado que la mayor frecuencia que se leerá será de 400Hz, según el Cuadro No. 9.

Figura No. 23. Circuito de anti-alias, acondicionamiento y amplificación de señal de guitarra.



2. Módulo convertidor analógico-digital. Ahora ya se cuenta con una señal acondicionada al pin del PIC, se requiere digitalizar la señal analógica para que pueda ser procesada por el PIC. El módulo ADC del PIC se encarga de esta tarea. Por lo tanto, ya se puede continuar al procesamiento digital de la señal. a continuación, se presentan los métodos propuestos de identificación de frecuencias.

B. Elección de frecuencias para clasificación de notas y acordes

Después de haber modificado la señal para adecuarse a los pines de entrada del microcontrolador es necesario definir las frecuencias que establecen las notas de afinación y de acordes. Así como también el rango de frecuencias permisibles en cada nota antes de que la nota se traslape con otra. Utilizando la información de los Cuadros No. 1 y 2, así como también de los acordes y notas de afinación se generaron los Cuadros No. 5, 6, 7, 8 y 9, en los cuales se especificó la frecuencia central de la nota, y las notas máximas y mínimas permisibles para cada nota.

Cuadro No. 5. Frecuencias de diseño para notas de afinación.

Cuerda	Nota Español	Nota Inglés	Frecuencia Central (Hz)	Frecuencia Mínima (Hz)	Frecuencia Máxima (Hz)
1era.	Mi	E_4	329.63	311.13	349.23
2nda.	Si	B_3	246.94	233.08	261.63
3era.	Sol	G_3	196.00	185.00	207.65
4ta.	Re	D_3	146.83	138.83	155.56
5ta.	La	A_2	110.00	103.83	116.54
6ta.	Mi	E_2	82.41	77.78	87.31

Cuadro No. 6. Frecuencias de diseño para notas de acorde Do Mayor.

Cuerda	Nota Español	Nota Inglés	Frecuencia Central (Hz)	Frecuencia Mínima (Hz)	Frecuencia Máxima (Hz)
1era.	Mi	E_4	329.63	311.13	349.23
2nda.	Do	C	262	246.64	277
3era.	Sol	G_3	196.00	185.00	207.65
4ta.	Mi	E	165	155.56	174.61
5ta.	Do	C	131	123.47	138.59
6ta.	Mi	E_2	82.41	77.78	87.31

Cuadro No. 7. Frecuencias de diseño para notas de acorde La Menor.

Cuerda	Nota Español	Nota Inglés	Frecuencia Central (Hz)	Frecuencia Mínima (Hz)	Frecuencia Máxima (Hz)
1era.	Mi	E_4	329.63	311.13	349.23
2nda.	Do	C	262	246.64	277
3era.	La	A	220	208	233
4ta.	Mi	E	165	155.56	174.61
5ta.	La	A_2	110.00	103.83	116.54
6ta.	Mi	E_2	82.41	77.78	87.31

Cuadro No. 8. Frecuencias de diseño para notas de acorde Re Menor.

Cuerda	Nota Español	Nota Inglés	Frecuencia Central (Hz)	Frecuencia Mínima (Hz)	Frecuencia Máxima (Hz)
1era.	Fa	F	349	330	370
2nda.	Re	D	294	277	311
3era.	La	A	220	208	233
4ta.	Re	D_3	146.83	138.83	155.56
5ta.	La	A_2	110.00	103.83	116.54
6ta.	Mi	E_2	82.41	77.78	87.31

Cuadro No. 9. Frecuencias de diseño para notas de acorde Sol Mayor.

Cuerda	Nota Español	Nota Inglés	Frecuencia Central (Hz)	Frecuencia Mínima (Hz)	Frecuencia Máxima (Hz)
1era.	Sol	G	392	370	415
2nda.	Re	D	294	277	311
3era.	Sol	G_3	196.00	185.00	207.65
4ta.	Re	D_3	146.83	138.83	155.56
5ta.	Si	B	123	117	131
6ta.	Sol	G	98	92	104

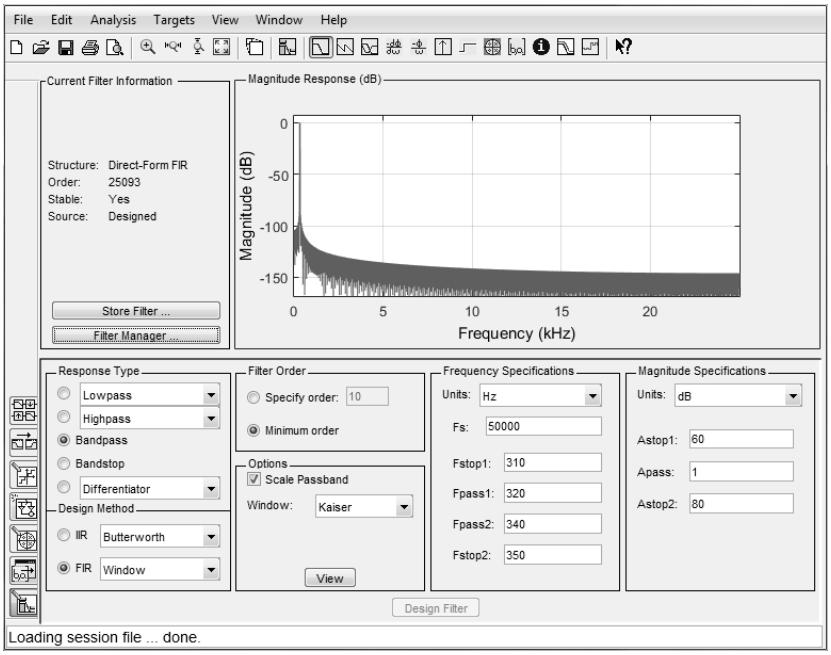
C. Filtrado de señal por medio de filtros de tipo pasa-banda

La identificación de frecuencias por este método consistió mediante la evaluación de la magnitud de la señal de salida del filtro, la cual se vería atenuada severamente de no haberse detectado la frecuencia de interés o, en el caso contrario, no se vería atenuada significativamente al ser detectada la frecuencia de interés o que por lo menos quedase dentro de la banda de paso del filtro.

1. Filtro con respuesta al impulso finita. El diseño de filtro con respuesta finita se realizó con la herramienta de diseño de filtros de Matlab. Esta herramienta permite especificar el tipo de filtro, el ancho de banda para el filtro pasa-banda, y la atenuación en decibeles. En la Figura No. 24 se puede visualizar la interfaz para el diseño de los filtros de respuesta al impuso finita. En el Cuadro No. 5, muestra las frecuencias de diseño para las frecuencias de afinación.

El diseño se hizo utilizando el método de “ventanas” y utilizando la ventana “Kaiser” descrita anteriormente en la sección de marco teórico. Para el diseño se debe de cuidar que el ancho de banda del filtro pasa-banda no exceda los límites de frecuencias de diseño en el Cuadro No. 5.

Figura No. 24. Interfaz de la herramienta de diseño de filtros de respuesta al impulso finito en software Matlab.



2. Filtro con respuesta al impulso infinita tipo Notch. Para probar el otro método propuesto se realizó el diseño de filtros con respuesta al impulso infinita los cuales pueden generarse con la herramienta de diseño de Matlab. Debido a las frecuencias de diseño de las notas, descritas en los Cuadros No. 5, 6, 7, 8 y 9, y que el ancho de banda es angosto, será necesario especificar el filtro tipo “Notch” en la herramienta de diseño.

Es también necesario especificar que el diseño del filtro tipo Notch será por medio de un Notch único, y además la frecuencia de diseño será la frecuencia central de la nota, esta frecuencia central se encuentra en los Cuadros No. 5, 6, 7, 8 y 9.

a. Inversión de filtros Notch. Debido a que el método para identificar la frecuencia de interés es la evaluación de la magnitud de la salida del filtro, es necesario invertir la respuesta del filtro para que únicamente al presentarse la frecuencia de interés la señal de salida no se vea atenuada. Para obtener la respuesta del filtro inversa se utilizó la Ecuación No. 21, la cual genera un filtro Notch Inverso “$N^{-1}[n]$” a partir de un filtro Notch “N[n]”.

$$N^{-1}[n] = 1 - N[n] \quad (21)$$

Para ambos procedimientos del diseño de filtros, la herramienta de diseño ofrece la opción de exportación de las constantes de la función de transferencia de los filtros. Estas constantes son las que se implementaran como funciones recursivas para implementar los filtros diseñados.

D. Análisis en dominio de frecuencia por medio de transformada discreta de Fourier

El método de la transformación discreta de Fourier de la señal digital, mapea la misma desde el dominio discreto de tiempo hacia el dominio discreto de frecuencia. En este nuevo dominio se puede conocer las frecuencias que componen la señal analizada, comparando estas frecuencias con las indicadas en los Cuadros No. 5, 6, 7, 8 y 9 se puede establecer si la guitarra se encuentra afinada.

La transformada discreta de Fourier se realizó mediante el algoritmo FFT, o por sus siglas en inglés “Fast Fourier Transform”. En la Ecuación No. 22 se encuentra la expresión matemática para la transformada discreta de Fourier sobre una función digital x[n], de duración N. Esta implementación se realiza mediante librerías en código de lenguaje apropiado para el microcontrolador PIC32mx.

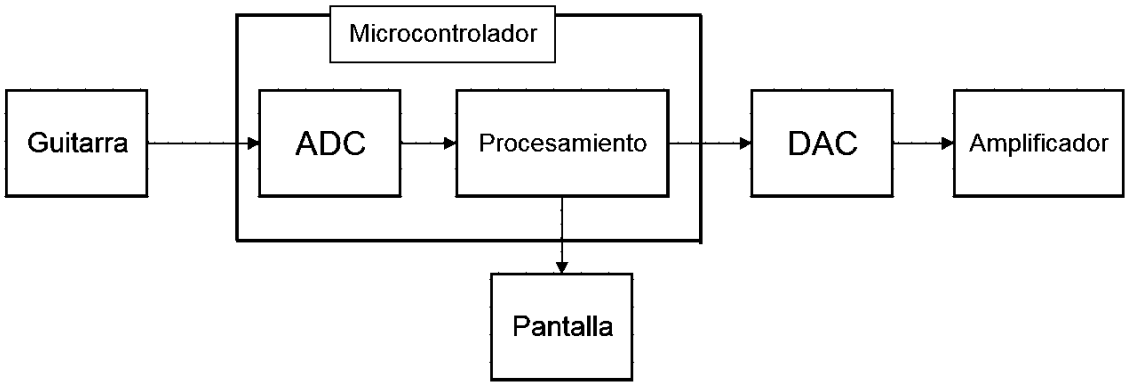
$$\tilde{X}[k] = \sum_{r=0}^{\frac{N}{2}-1} \tilde{x}[2r]\, W_{\frac{N}{2}}^{kr} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} \tilde{x}[2r+1]\, W_{\frac{N}{2}}^{kr} \quad (22)$$

E. Diseño de módulo de afinador

Después de realizar pruebas con los métodos propuestos para la identificación de frecuencias fue posible conceptualizar el diseño del hardware del afinador, utilizando los módulos de ADC del microcontrolador, el circuito de amplificación y el circuito DAC como se propone en la Figura No. 25. En este diagrama de bloques se presenta la organización de bloques del afinador, en donde la interfaz de conexión provee al usuario la información necesaria para afinar la cuerda o el reconocimiento de acordes y el método de detección de frecuencias que se implementa modificará la estructura para informar al usuario sobre la afinación.

El bloque del DAC incluye el circuito de suavización de tipo pasa-baja para asegurar que la señal digital logre ser lo más parecida a su contraparte analógica, así como también con una impedancia de salida baja para que logre ingresar al amplificador sin pérdidas de voltaje significativas.

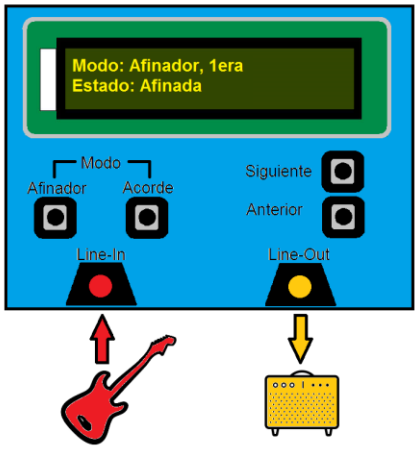
Figura No. 25. Diagrama de bloques del afinador.



1. Implementación filtros FIR. Para el caso en que se decida utilizar los filtros FIR como método de detección de frecuencias, la pantalla en la Figura No. 25 se implementará por medio de una LCD de 16x2 filas. Lo anterior se debe a que la información a desplegar será: el modo de operación del afinador, afinación o reconocimiento de acordes, y el acorde actual con el que se esté trabajando o la cuerda a afinar según sea el modo de operación seleccionado. Toda esta información se presentará al usuario como se puede observar en la Figura No. 26, en la cual se puede visualizar cómo sería la conexión del afinador hacia la guitarra y como sería la interfaz de usuario que se presentará.

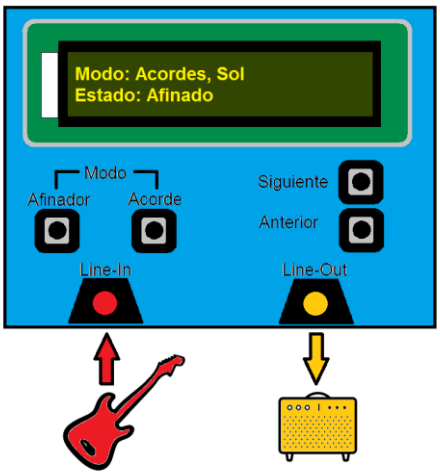
a. Modo de operación para afinación. En el modo de operación de afinación, la afinación se indicará por medio de un mensaje, el cual avisa al usuario que la cuerda se encuentra afinada, o indicando que ajuste las clavijas para lograr la afinación. En la Figura No. 26 se puede observar este modo de operación en la pantalla LCD.

Figura No. 26. Diseño de afinador con pantalla LCD 16x2 modo de operación para afinación.



b. Modo de operación para acodes. Este modo de operación solo debe ser accedido si las cuerdas ya se encuentran afinadas. En este caso el usuario podrá seleccionar el acorde que desea comprobar por medio de los botones de la Figura No. 26. Una vez seleccionado el acorde a comprobar deberá ejecutar el acorde y esperar el mensaje que se despliegue en la pantalla. Si el acorde es correcto entonces en la pantalla se desplegará el mensaje indicando una ejecución satisfactoria del acorde, en caso contrario se le indicará que el acorde requiere de afinación. En la Figura No. 27 se puede ver el modo de operación para acordes con pantalla LCD.

Figura No. 27. Diseño de afinador con pantalla LCD 16x2 modo de operación para acordes.

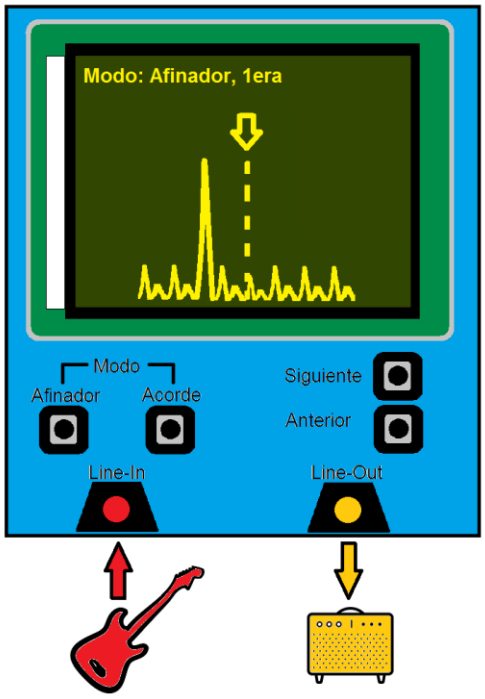


2. Implementación filtros Notch Inverso. Para el caso en que se decida utilizar los filtros Notch Inversos como el método para la detección de frecuencias, la interfaz de usuario seria la misma a la interfaz de la Figura No. 26. Esto se debe a que la información que se extraiga de la señal por medio de los filtros Notch Inversos es equivalente a la información que se extraiga con los filtros FIR. Por lo tanto, la pantalla a utilizar sería la pantalla LCD 16x2, y los mensajes de información serían los mismos que para la implementación con filtros FIR, de igual manera los modos de operación se ejecutaran de la misma manera.

3. Implementación FFT. En el caso en que se utilice la FFT como método para la detección de frecuencias, se implementará una pantalla GLCD, como se puede ver en la Figura No. 28. En este caso la información que se presentará al usuario será el espectro de frecuencias de la señal medida por el afinador. Se presentan los modos de operación para afinación y para acordes.

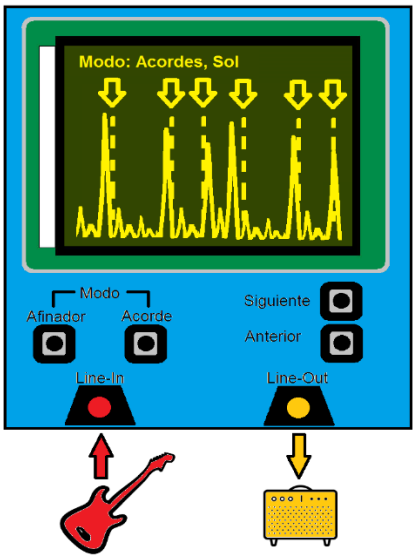
a. Modo de operación para afinación. En este caso se presentará al usuario, por medio de un indicador en la pantalla, la frecuencia de afinación de la cuerda, así mismo la frecuencia medida por el afinador. El usuario deberá de modificar la clavija de la cuerda para modificar la frecuencia de la cuerda y aproximar lo mejor posible la frecuencia de la cuerda a la frecuencia de afinación. Ver Figura No. 28 para observar el modo de operación para afinación en la pantalla GLCD.

Figura No. 28. Diseño de afinador con pantalla GLCD modo de operación para afinación.



b. Modo de operación para acodes. Para el modo de operación de acordes las cuerdas ya deben de estar afinadas. La información en la pantalla GLCD será la misma que la del modo de operación para afinación. Sin embargo, se desplegarán las frecuencias del acorde que haya seleccionado el usuario, por lo tanto, existirán seis indicadores en la pantalla correspondientes a las seis cuerdas que exija el acorde, o en su defecto cinco cuerdas. Al ejecutar el acorde se graficará las frecuencias y deberá de manera visual comprobar que las frecuencias de las cuerdas sean las más cercanas a las frecuencias teóricas indicadas en el afinador.

Figura No. 29. Diseño de afinador con pantalla GLCD modo de operación para acordes.



VII. RESULTADOS

A. Filtros FIR.

Figura No. 30. Respuesta en frecuencia del filtro FIR pasa-banda para 82Hz.

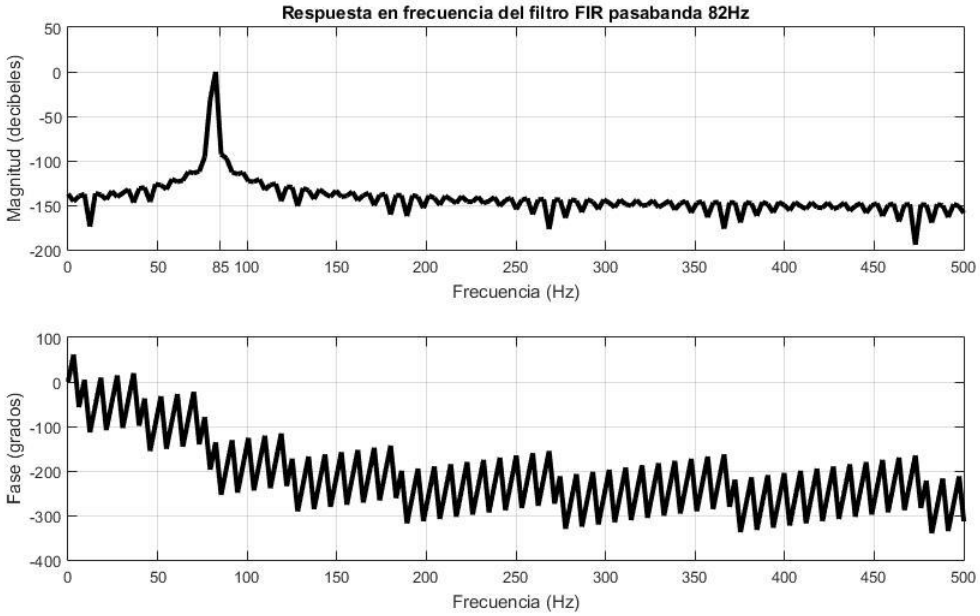


Figura No. 31. Respuesta en frecuencia del filtro FIR pasa-banda para 110Hz.

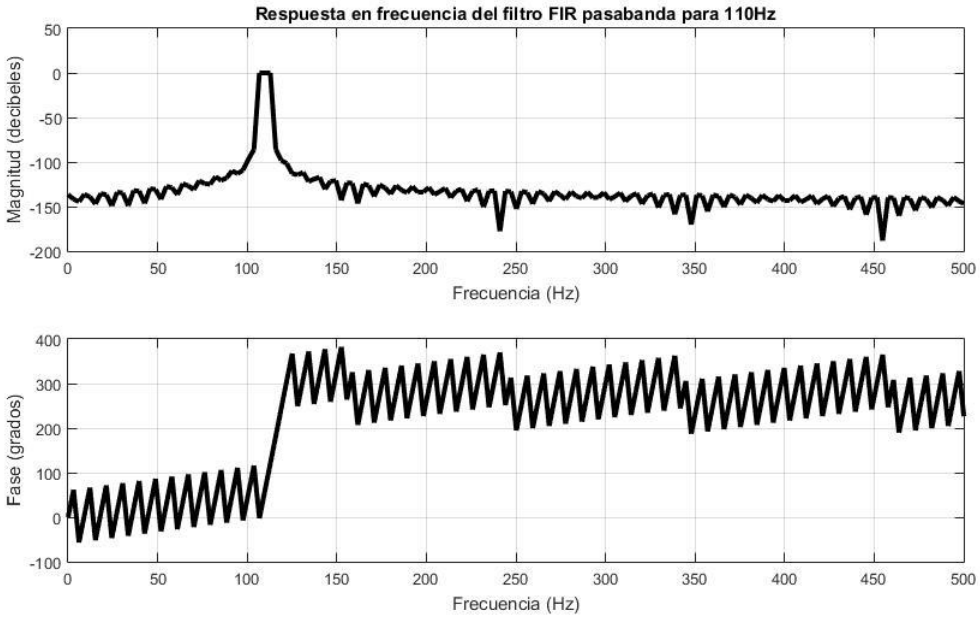


Figura No. 32. Respuesta en frecuencia del filtro FIR pasa-banda para 146Hz.

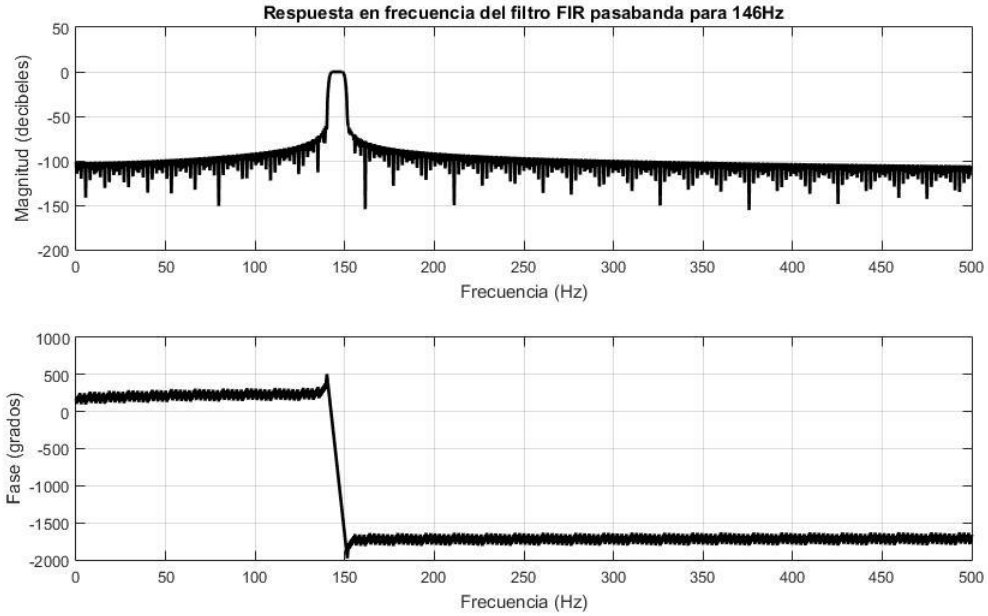


Figura No. 33. Respuesta en frecuencia del filtro FIR pasa-banda para 196Hz.

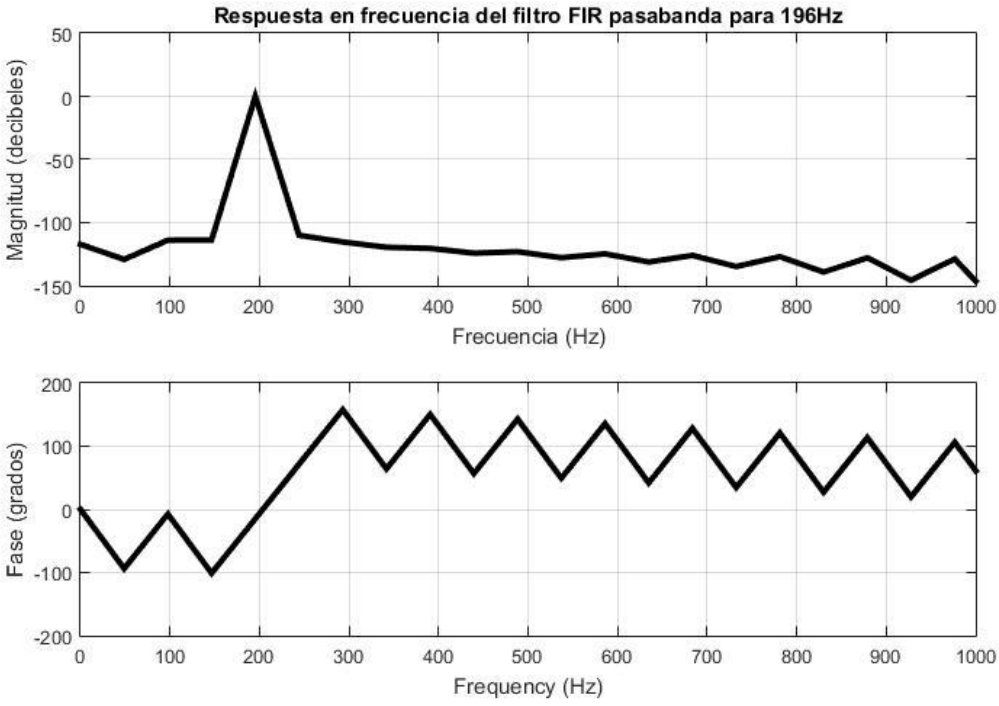


Figura No. 34. Respuesta en frecuencia del filtro FIR pasa-banda para 246Hz.

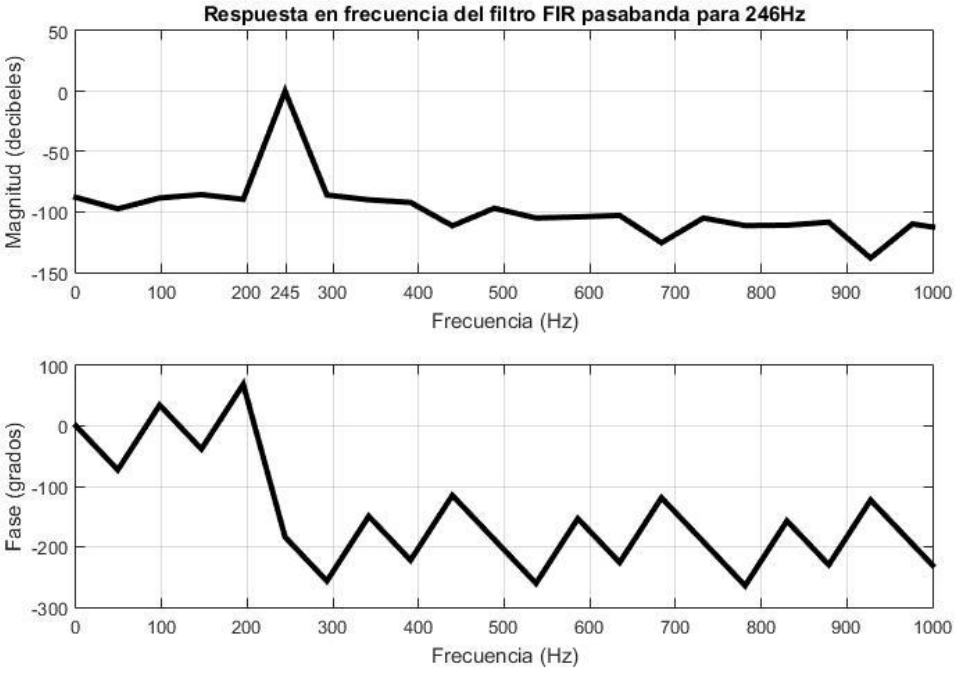


Figura No. 35. Respuesta en frecuencia del filtro FIR pasa-banda para 329Hz.

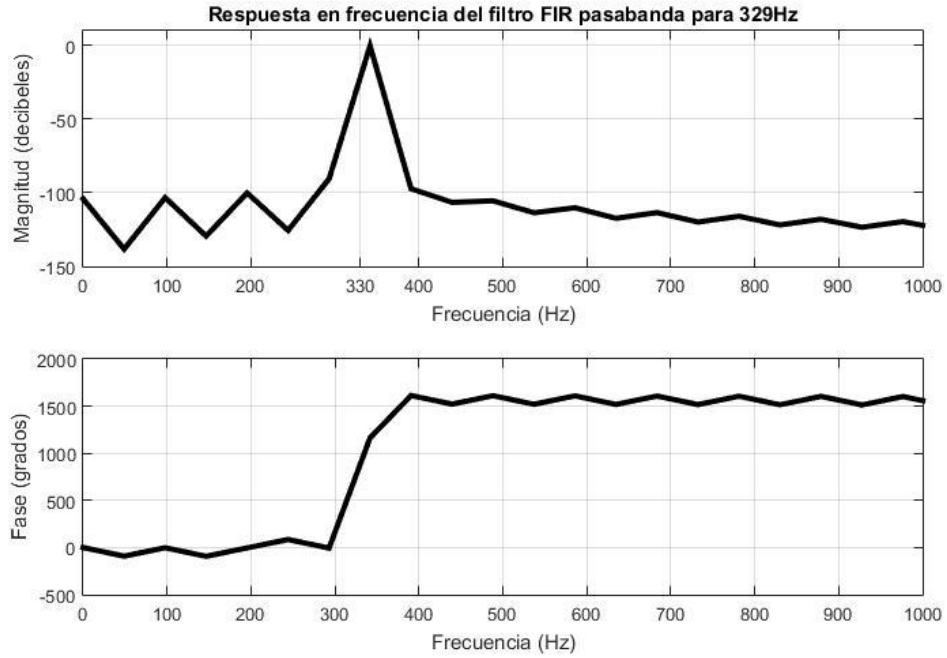
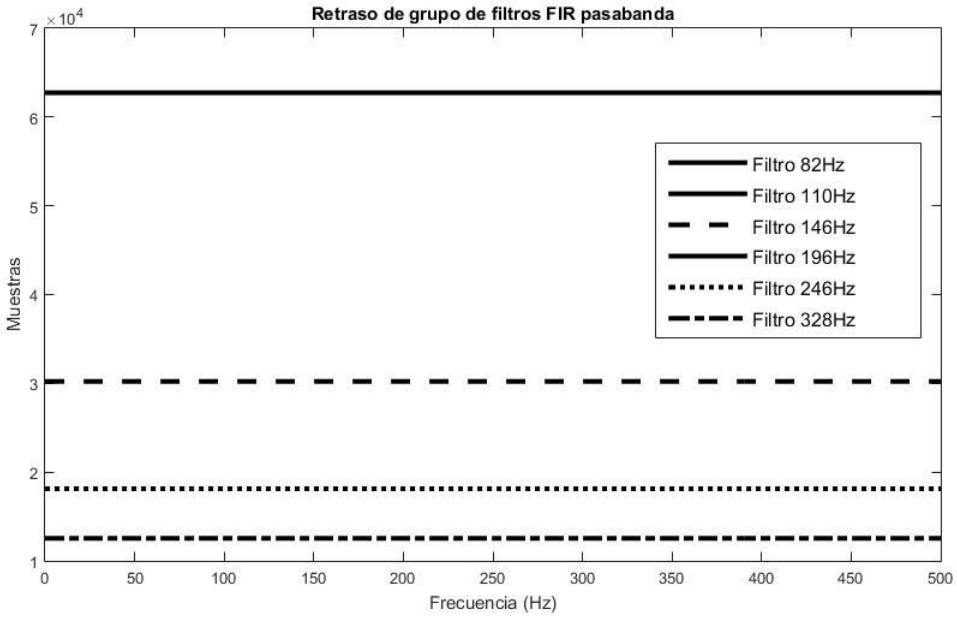


Figura No. 36. Retraso de grupo de filtros FIR pasa-banda para afinación.



Cuadro No. 10. Orden de filtros FIR pasa-banda para afinación.

Filtro	Orden
82Hz	125,461
110Hz	125,461
146Hz	60,424
196Hz	125,461
246Hz	36,254
328Hz	25,093

B. Filtros IIR Notch.

Figura No. 37. Respuesta en frecuencia del filtro IIR Notch para 82Hz.

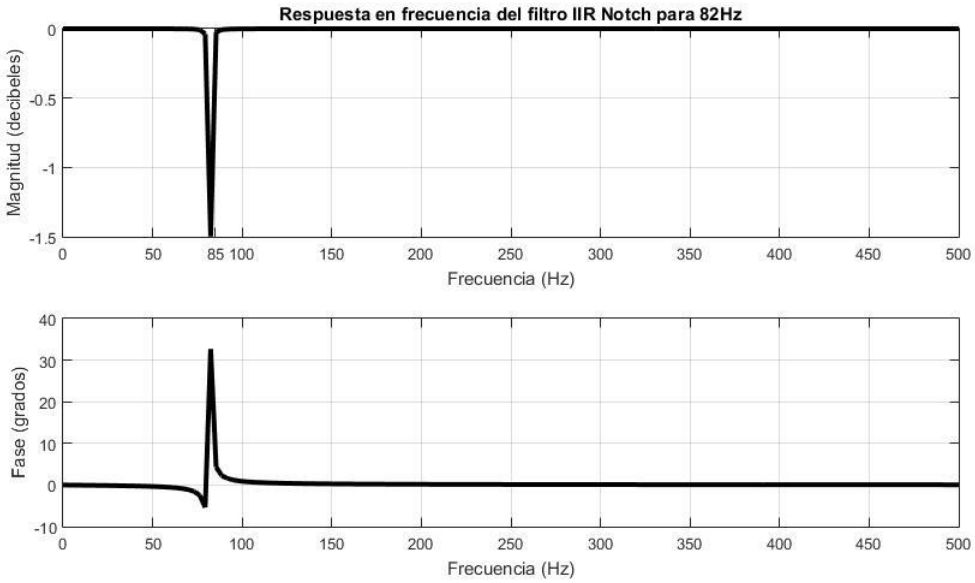


Figura No. 38. Respuesta en frecuencia del filtro IIR Notch para 110Hz

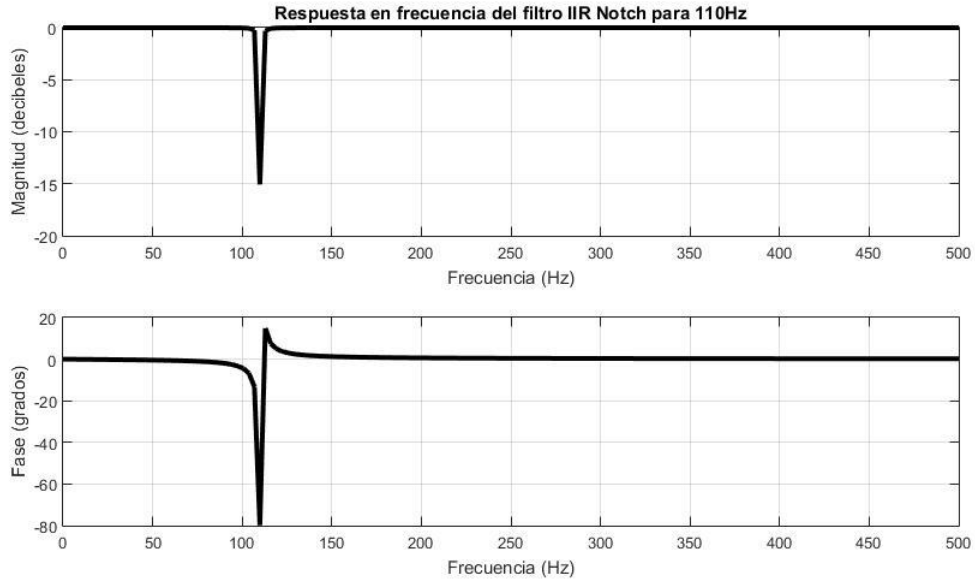


Figura No. 39. Respuesta en frecuencia del filtro IIR Notch para 149Hz.

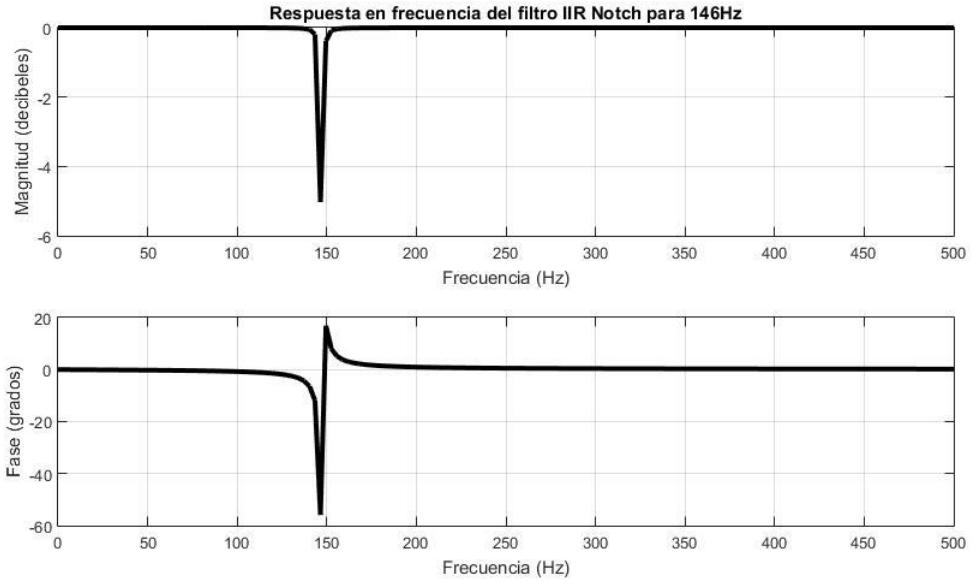


Figura No. 40. Respuesta en frecuencia del filtro IIR Notch para 196Hz.

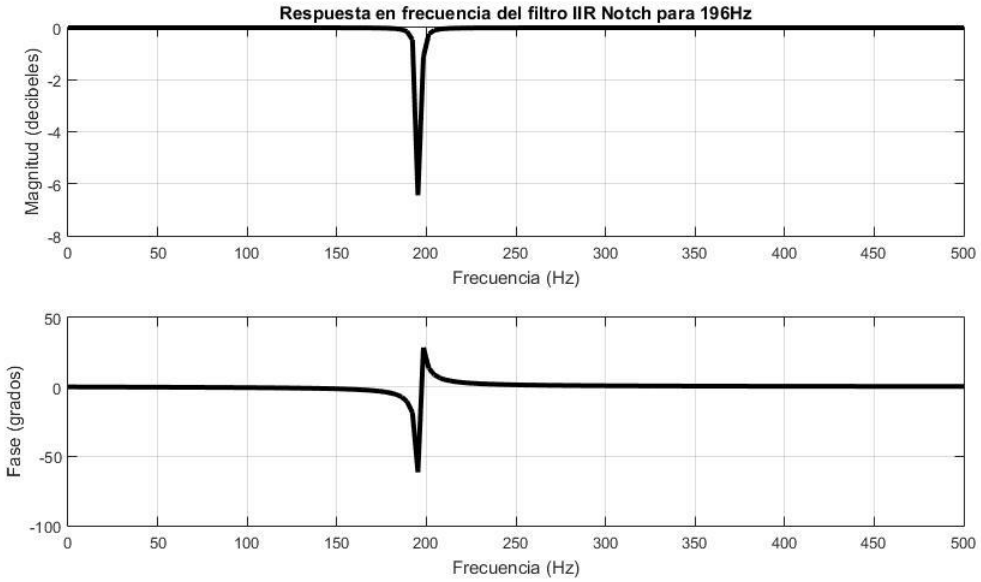


Figura No. 41. Respuesta en frecuencia del filtro IIR Notch para 247Hz.

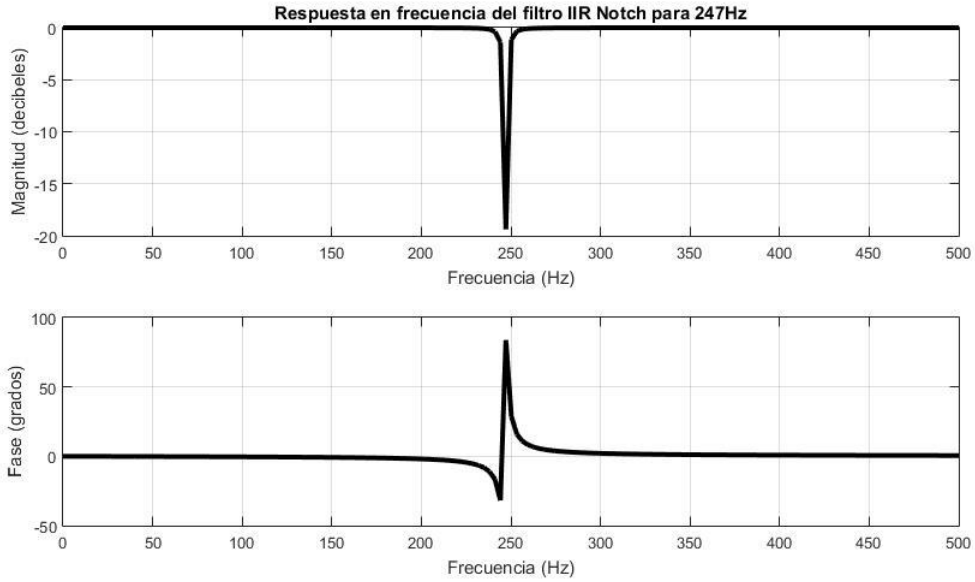


Figura No. 42. Respuesta en frecuencia del filtro IIR Notch para 330Hz.

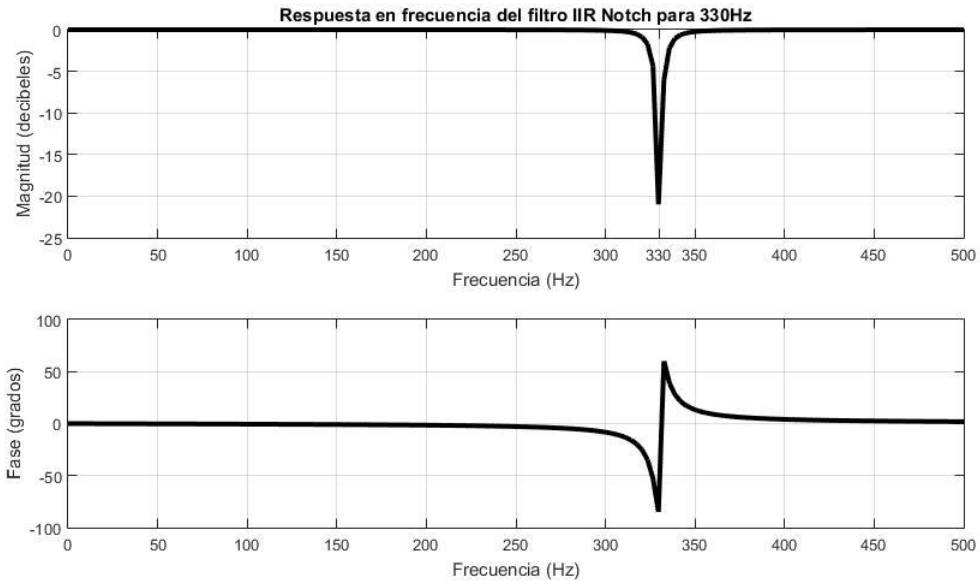
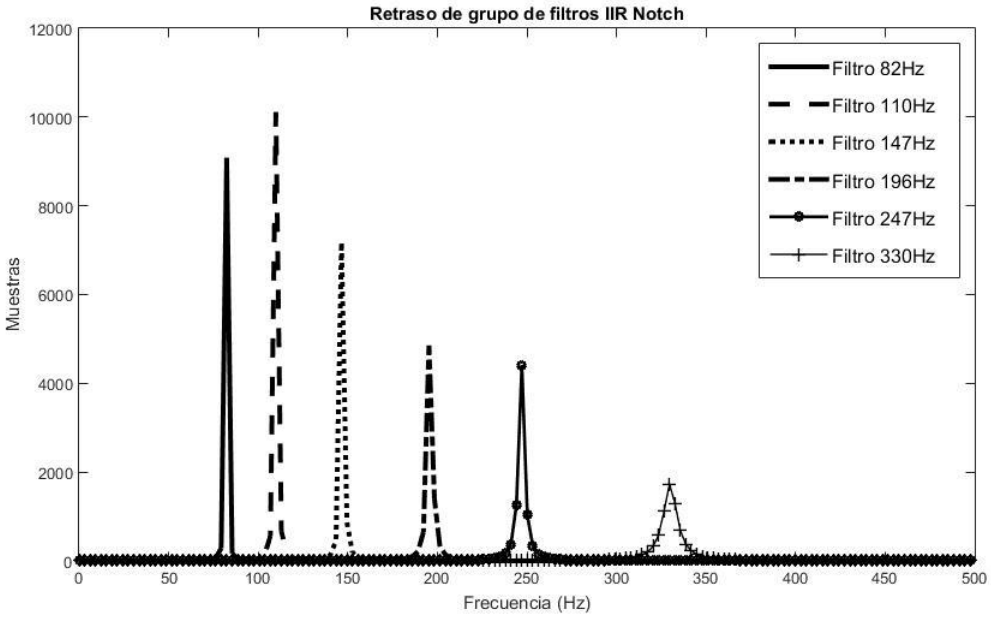


Figura No. 43. Retraso de grupo de filtros IIR Notch para afinación.



Cuadro No. 11. Orden de filtros IIR Notch para afinación.

Filtro	Orden
82Hz	2
110Hz	2
147Hz	2
196Hz	2
247Hz	2
330Hz	2

Cuadro No. 12. Orden de filtros IIR Notch para acordes.

Filtro	Orden
98Hz	2
123Hz	2
131Hz	2
165Hz	2
220Hz	2
262Hz	2
294Hz	2
350Hz	2
392Hz	2

Figura No. 44. Respuesta en frecuencia de filtro IIR Notch para acordes.

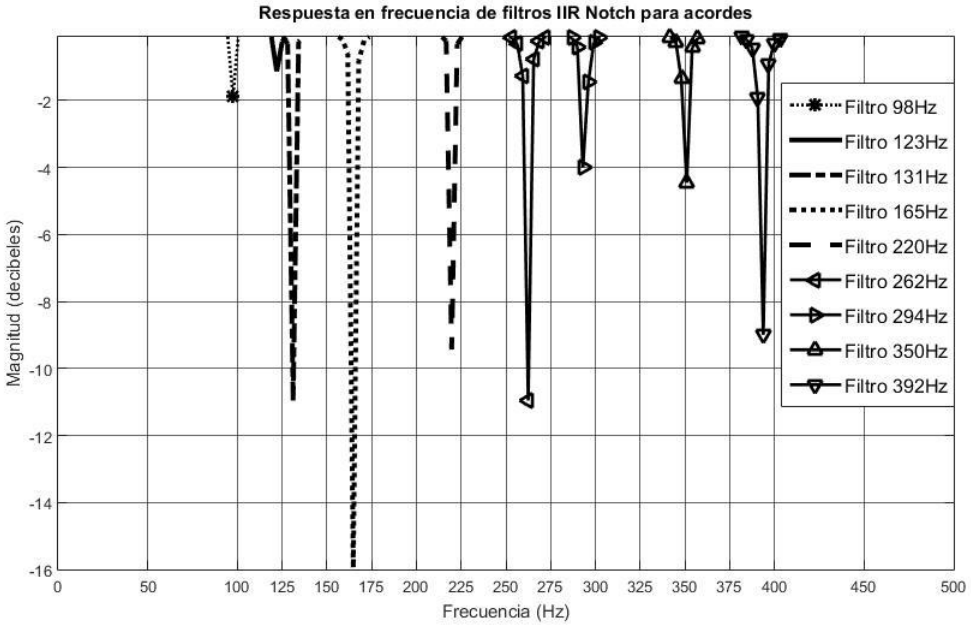
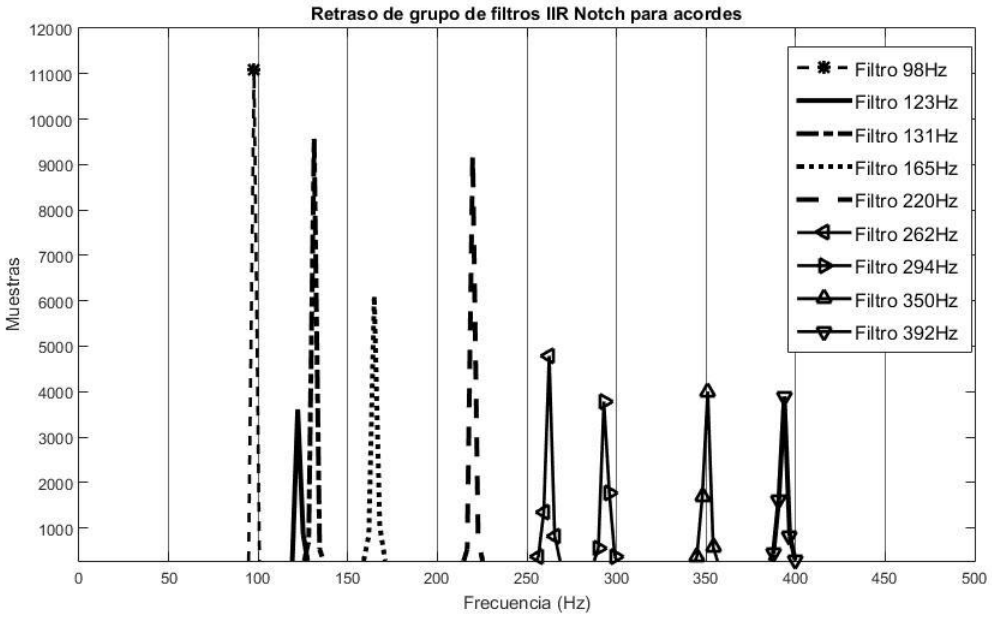


Figura No. 45. Retraso de grupo para filtros IIR Notch para acordes.



C. Filtro IIR Notch Inverso.

Figura No. 46. Respuesta en frecuencia del filtro IIR Notch Inverso para 82Hz.

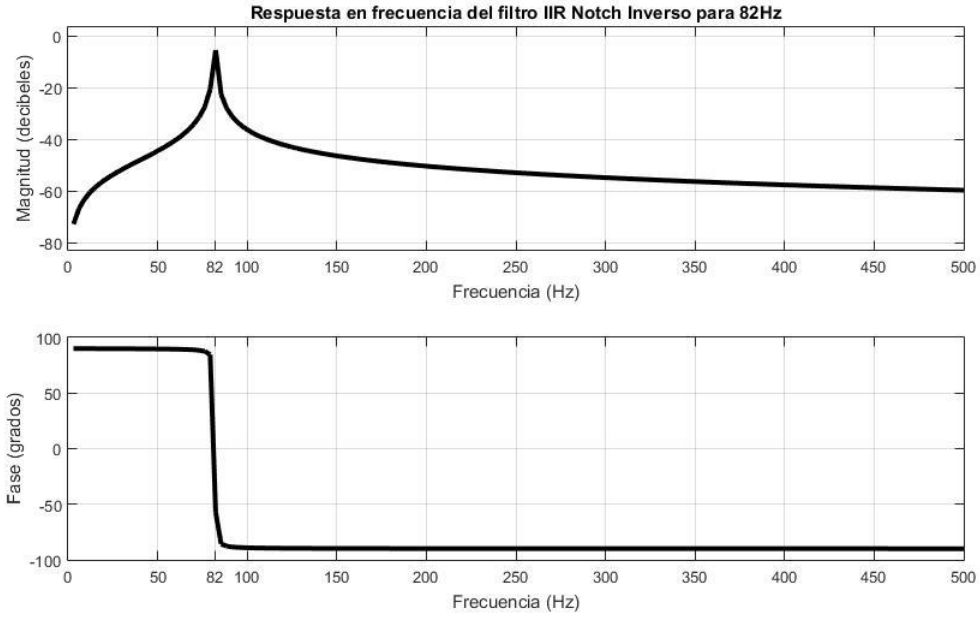


Figura No. 47. Respuesta en frecuencia del filtro IIR Notch Inverso para 110Hz.

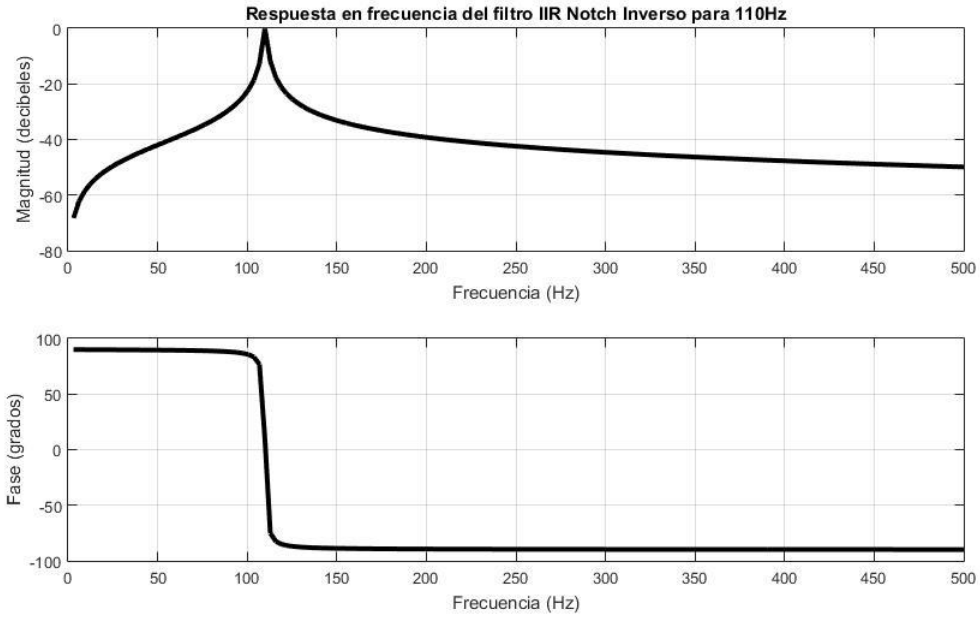


Figura No. 48. Respuesta en frecuencia del filtro IIR Notch Inverso para 146Hz.

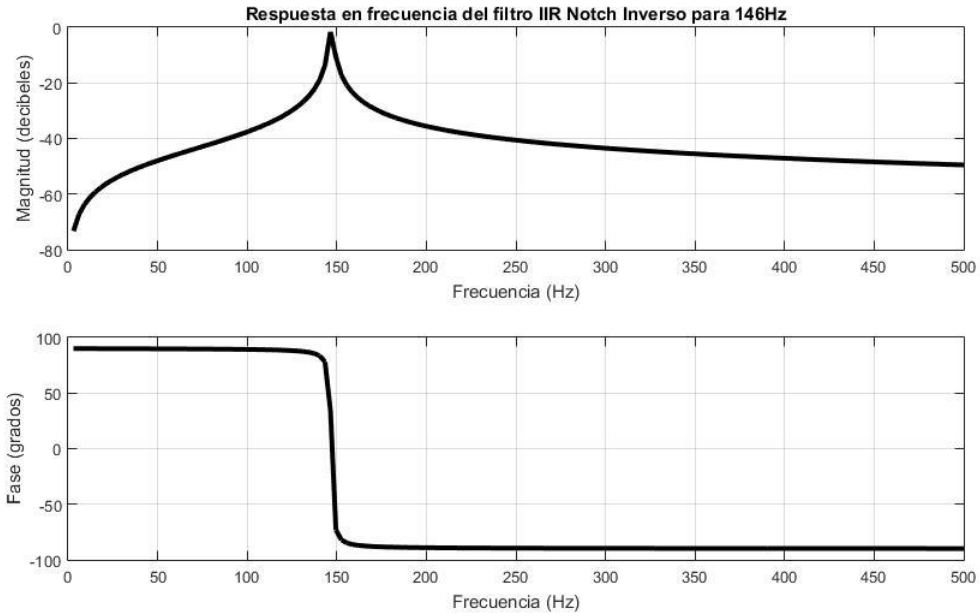


Figura No. 49. Respuesta en frecuencia del filtro IIR Notch Inverso para 196Hz.

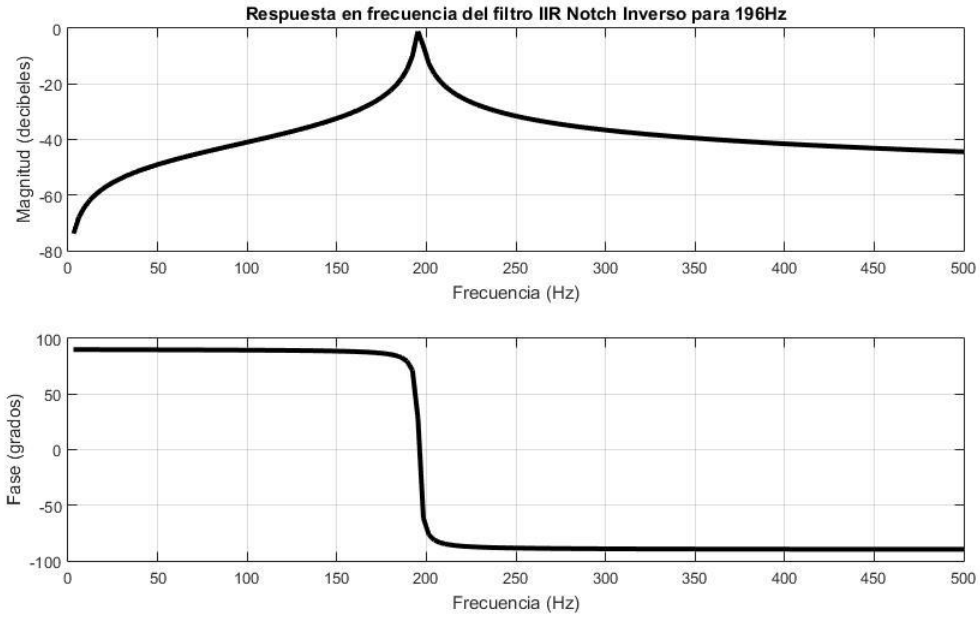


Figura No. 50. Respuesta en frecuencia del filtro IIR Notch Inverso para 246Hz.

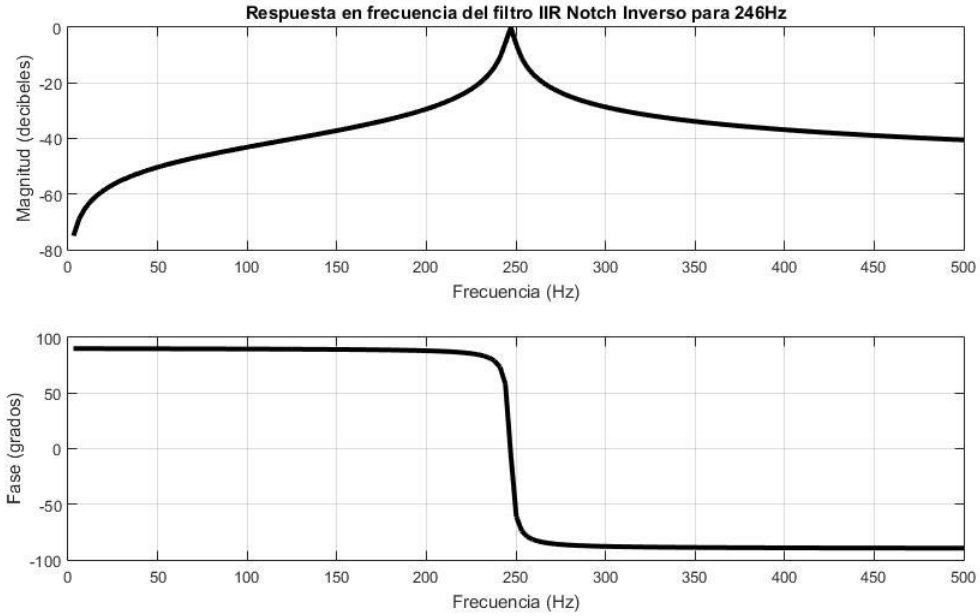


Figura No. 51. Respuesta en frecuencia del filtro IIR Notch Inverso para 329Hz.

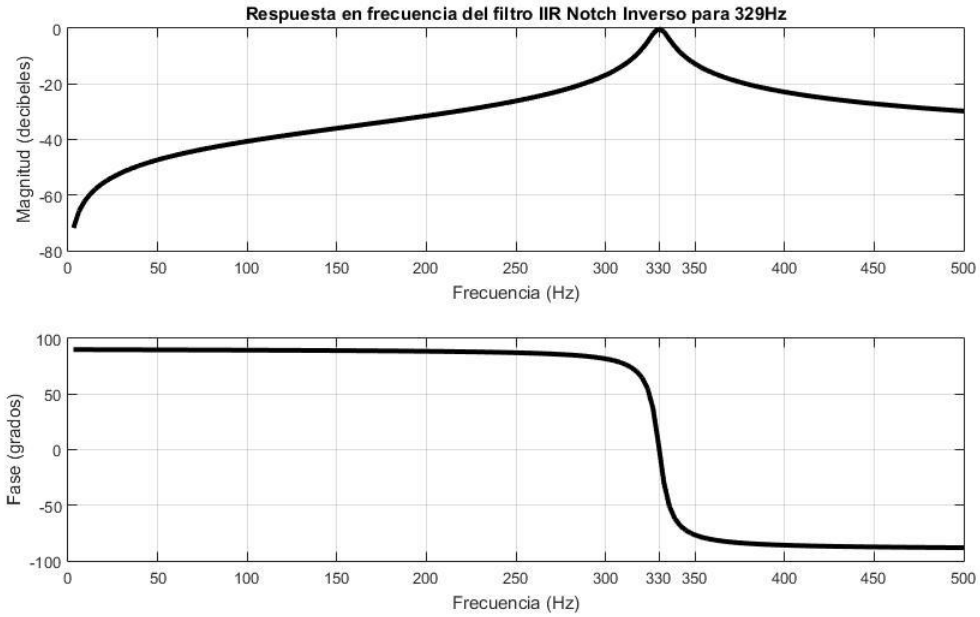


Figura No. 52. Retraso de grupo de filtros IIR Notch Inverso para afinación

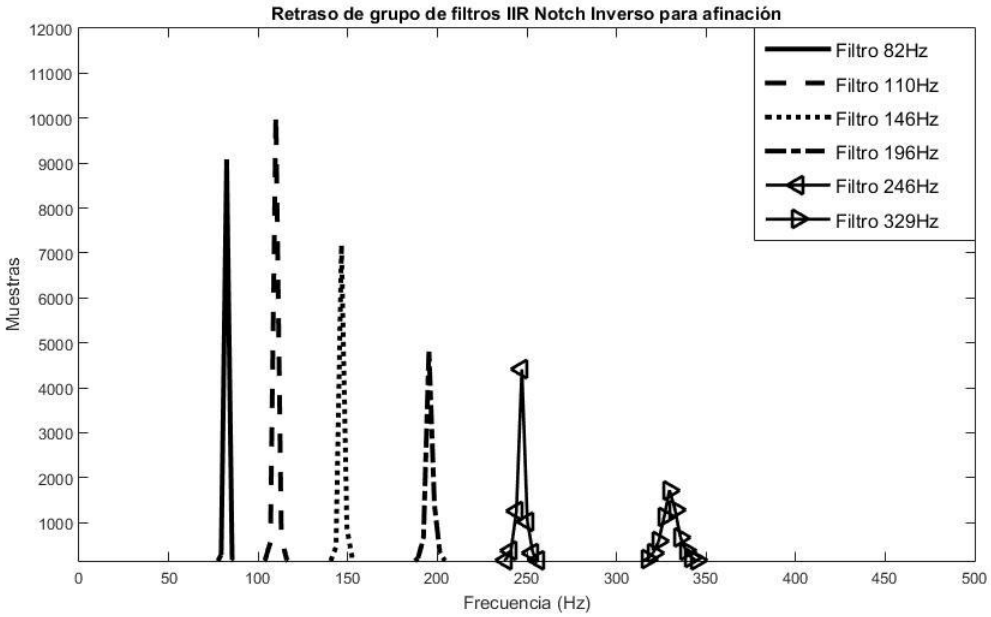


Figura No. 53. Digitalización de señal de 82Hz por medio del módulo ADC.

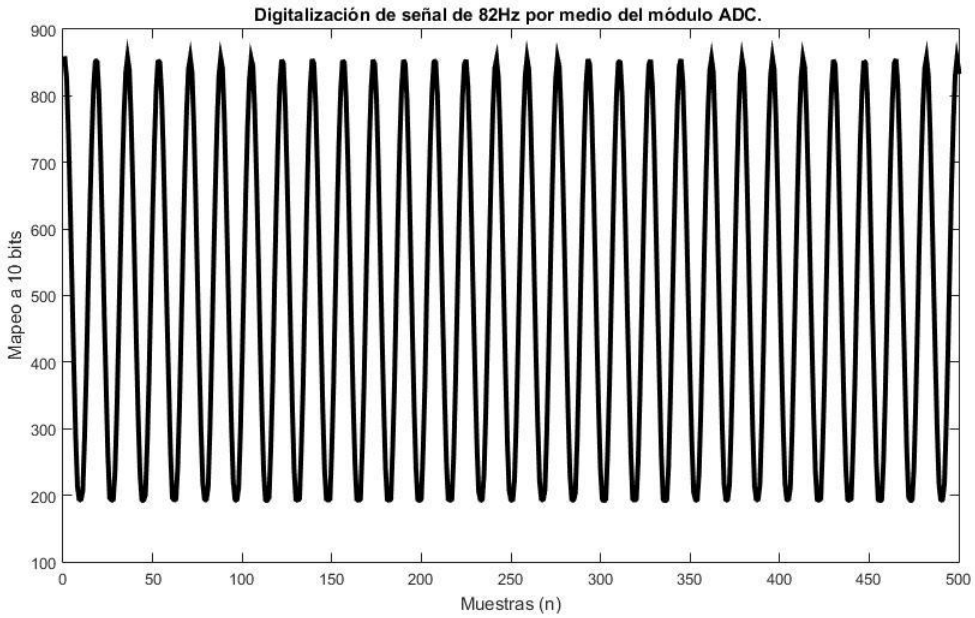
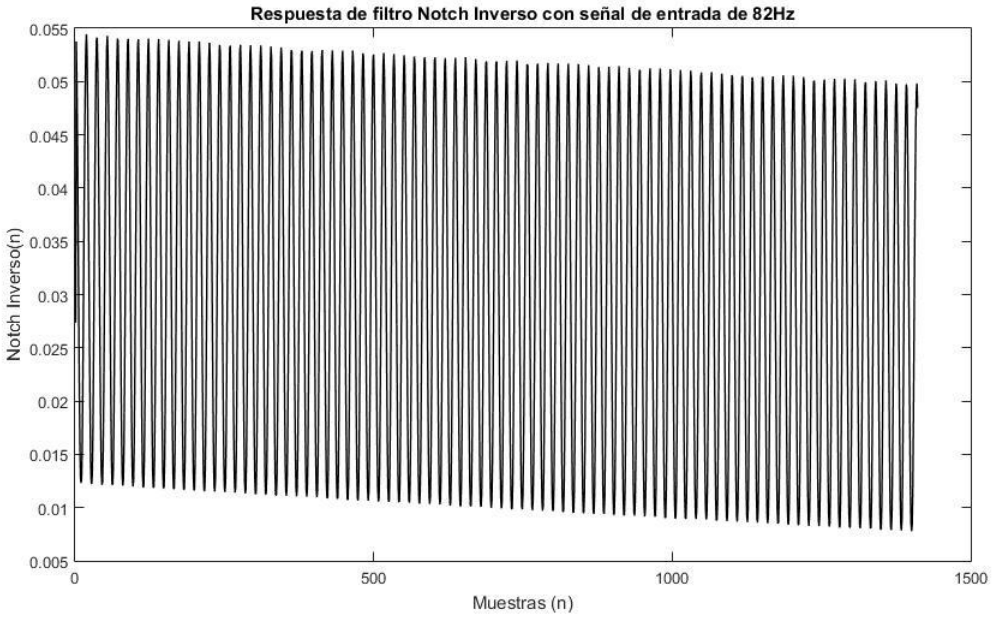


Figura No. 54. Respuesta del filtro Notch Inverso para 82Hz en dominio de tiempo discreto para una señal de entrada de 82Hz.



Cuadro No. 13. Constantes de implementación del filtro IIR Notch Inverso para 82Hz en función de transferencia.

Constantes		Valor
Numerador	N0	3.197078660899244e-05
	N1	0
	N2	-3.197078660899244e-05
Denominador	D0	1.000000000000000
	D1	-1.999829881608991
	D2	0.999936058426782
Función de Transferencia		
$Notch_{Inverso}[z] = \frac{3.19707^{-5}z^0 + 0.0z^{-1} - 3.19707^{-5}z^{-2}}{1.0z^0 - 1.999z^{-1} - 0.999z^{-2}}x[z]$		

D. Transformada Rápida de Fourier.

Figura No. 55. Transformada Rápida de Fourier de señal de entrada de 82Hz.

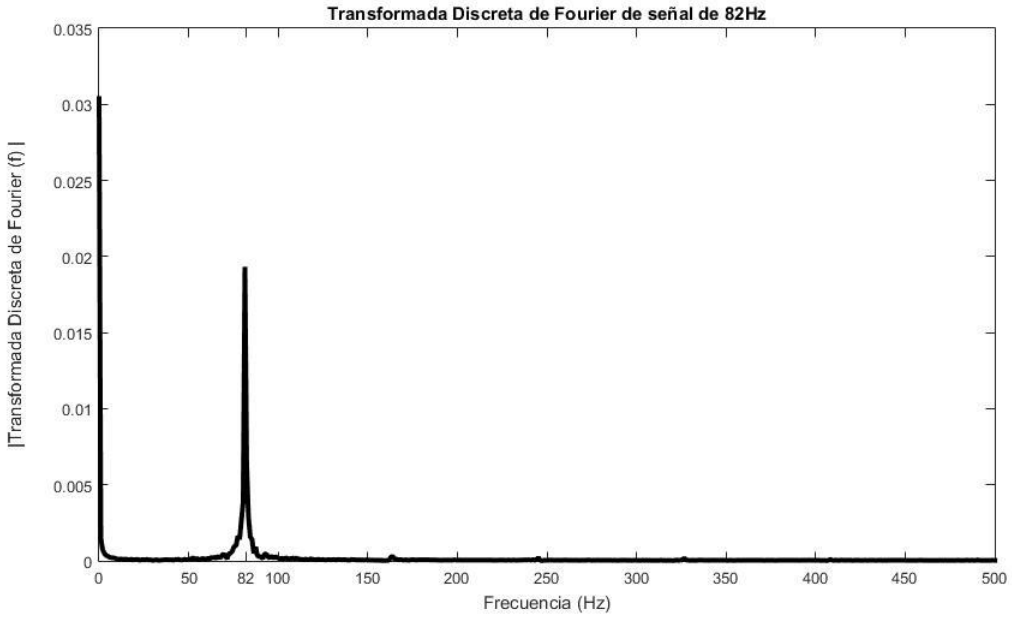
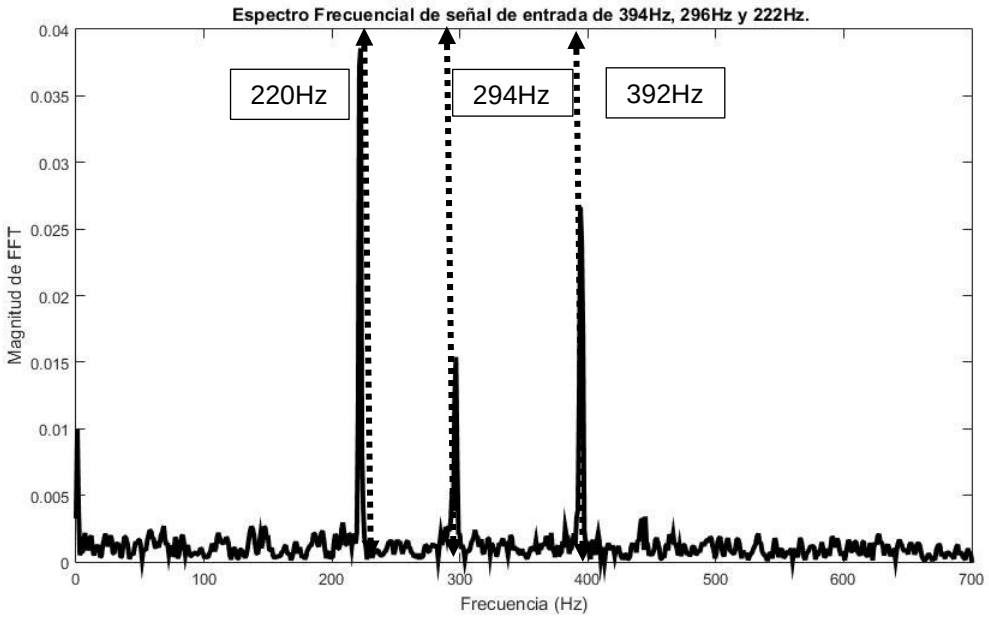


Figura No. 56. Transformada Rápida de Fourier de señal de entrada con componentes de 222Hz, 296Hz y 394Hz.



E. Afinador digital

Figura No. 57. Prototipo del circuito anti-alias, acondicionamiento y amplificación de señal de guitarra.

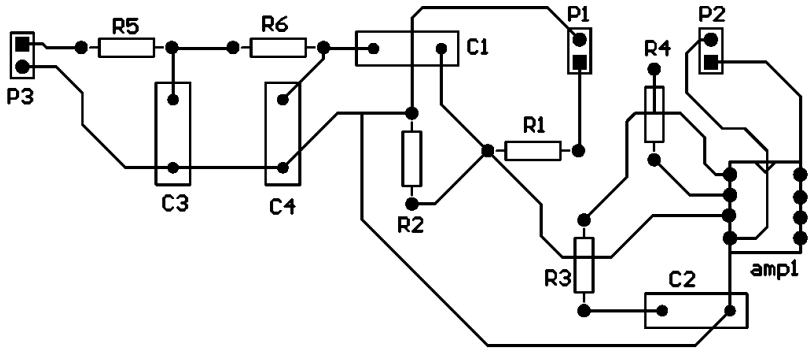
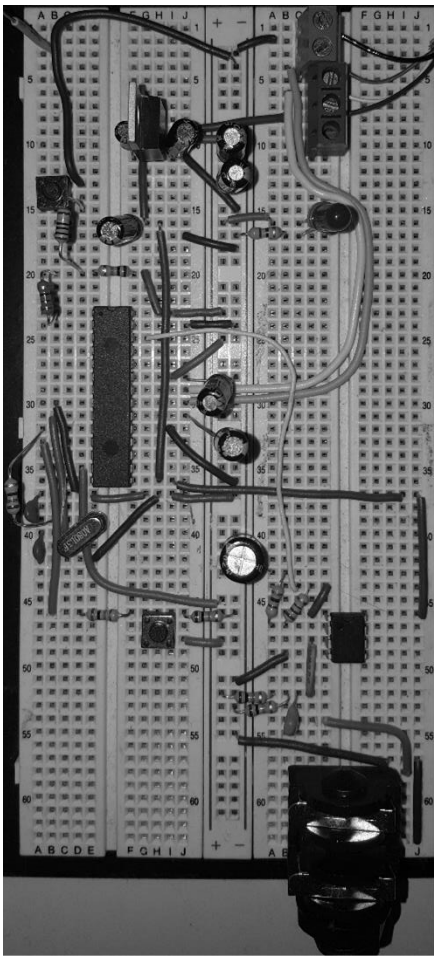


Figura No. 58. Prototipo del circuito afinador en placa de pruebas.



VIII. ANÁLISIS DE RESULTADOS

Después de haber realizado los procedimientos de diseño de filtros para las frecuencias de afinación de la guitarra, descritas en el Cuadro No. 3, fue necesario comprobar que el diseño cumpliera con las especificaciones y verificar su implementación en el microcontrolador.

Las respuestas en frecuencia de los filtros FIR pasa-banda para afinación se pueden observar en las Figuras No. 30, 31, 32, 33, 34 y 35. En estas figuras es posible determinar que la banda de paso de los filtros concuerda con los valores especificados en el Cuadro No. 5. Por lo tanto, el filtro si logró determinar si la señal digital contiene la frecuencia de afinación correcta para la cuerda deseada. Sin embargo, al observar la Figura No. 36 el retraso de grupo de los filtros FIR es cercano al orden de 10^4, lo que implica que al tener una frecuencia de muestreo de 50KHz el filtro con mayor retraso de grupo tardaría alrededor de 1.4 segundos en procesar la información. Esto demostró que los filtros FIR diseñados responderán en un tiempo inaceptable para la aplicación del afinador. Otro aspecto es el orden de los filtros FIR, el cual puede ser observado en el Cuadro No. 10., en el cual el orden mayor es de 125, 461 para los filtros de: 82Hz, 110Hz y 196Hz. Debido al orden se requerían de 125,461 espacios en memoria para lograr obtener la respuesta de los filtros FIR. Lo anterior implica que los filtros FIR no podrían ser implementados en el microcontrolador debido a que la memoria de datos no excede 32KB.

Al haber finalizado el análisis de los filtros FIR, se optó por utilizar filtros IIR, dado que estos generalmente son de orden menor. Al ver las Figuras No. 30, 31, 32, 33, 34 y 35 los filtros resultaron ser de banda angosta, dada las características de diseño del Cuadro No. 5. Por lo tanto, se decidió utilizar los filtros Notch.

La respuesta en frecuencia de los filtros IIR tipo Notch para afinación se pueden observar en las Figuras No. 37, 38, 39, 40, 41 y 42. En las que se puede evidenciar que son de banda angosta, también es posible implementarlos en el microcontrolador debido a que el orden de los filtros es de 2, lo que se puede comprobar al ver el Cuadro No. 11 y 12. Por lo tanto, se requerirían de dos espacios en memoria para computar la respuesta de los filtros. En relación al retraso de grupo, Figura No. 43, el retraso máximo sería de 10,000 muestras conociendo que la frecuencia de muestro es de 50Khz, el tiempo de computo será de 0.2 segundos, lo que claramente es una reducción de tiempo en comparación al tiempo de respuesta de los filtros FIR. También se puede observar la respuesta en frecuencia de los filtros para acordes en la Figura No. 44, en la que evidencia que al ser de banda angosta evitan la interferencia con otro filtro, y únicamente cuando se esté en la frecuencia de diseño se obtendría una señal válida del filtro. En la Figura No. 45 se encuentra el retraso de grupo para los filtros IIR para acordes, se observa que el valor máximo de muestras es cercano al valor máximo de retraso para los filtros IIR para afinación, ver Figura No. 43. Por lo tanto, los filtros para acordes no representan un retraso significativo al utilizarse.

Entonces los filtros IIR Notch pueden ser implementados en el microcontrolador y responden con mayor rapidez.

Debido a que se invirtieron los filtros Notch para lograr obtener una respuesta positiva al momento de detectar la frecuencia de afinación, fue necesario reevaluar el filtro resultante, Notch inverso, para asegurar que el comportamiento siguiese siendo aceptable.

En la Figura No. 46, 47, 48, 49, 50 y 51 se puede observar la respuesta en frecuencia de los filtros IIR Notch Inversos, diseñados para la afinación de la guitarra. Como se puede observar las respuestas fueron positivas al momento que las frecuencias centrales de diseño del Cuadro No. 5 ingresaron al filtro y debido a que es de banda angosta no se traslapó con otra nota, lo que indica que la precisión del filtro fue lo suficiente para asegurar que se encuentre afinada la nota. Al ver la Figura No.52, el retraso de grupo de los filtros IIR Notch Inverso para afinación no fue excesivo en comparación a los filtros IIR Notch, dado que el tiempo de retardo fue de 0.2 segundos.

Sin embargo, al momento de realizar la implementación del filtro IIR Notch Inverso no se tuvo el resultado esperado. En la Figura No. 54, se obtiene la respuesta en tiempo discreto del filtro para 82Hz al momento de ingresar una señal de 82Hz. Al analizar la imagen nos revela que el resultado del filtro es prácticamente nulo, lo que implica que no logró detectar la frecuencia de entrada. Al observar la respuesta en frecuencia de este filtro, ver Figura No. 46, la ganancia para la frecuencia de 82Hz es de aproximadamente -10dB, lo que implicaría un factor multiplicativo de 0.316 para una señal de entrada. Al ver la Figura No. 53, la señal ingresada oscilaba en el rango de 850 y 200 aproximadamente, lo que daría como resultado una señal de rango 268 y 63 en la salida del filtro. Este no fue el resultado del filtro, observar la Figura No. 54. Lo anterior puede deberse a la precisión del filtro, en el Cuadro No. 13 se observa la función de transferencia del filtro. Al momento de realizar la implementación en el microcontrolador, este carece de unidad de punto flotante lo que implica que al momento de la codificación se reduzca cierto valor de precisión en estos valores. Por lo tanto, no es factible realizar la implementación con el microcontrolador elegido, la precisión de las constantes influye en el comportamiento del filtro debido al diseño del mismo, por ser de banda angosta cualquier cambio en estas modifica las propiedades del mismo.

Debido a lo anterior, se decidió utilizar otro método para la identificación de frecuencias en las señales de entrada, mediante el uso de la FFT. En la Figura No. 55 se puede observar que se logra detectar que la señal de entrada contiene una señal de 82Hz, lo cual concuerda con la señal de entrada que tenía una componente de 82Hz. Es posible determinar la frecuencia de la señal y se puede comprobar que esta concuerda con la afinación requerida para la cuerda especificada. En la Figura No. 56 se ingresó una señal compuesta de varias sinusoides, y fue exitoso el reconocimiento de las componentes de frecuencia.

Debido a que la FFT si logró presentar una respuesta válida para las señales de entrada y contiene mayor información en dominio de frecuencia, la implementación de una interfaz por medio de una pantalla con una resolución mayor se consideró en el prototipo. Así también se consideró un DAC con una resolución de al menos la cantidad máxima de bits del ADC del microcontrolador, para evitar pérdidas de calidad en la señal. Todo esto se debe a que los métodos de filtros FIR y Notch Inverso no dieron resultados válidos o no fue posible su implementación, además por medio de filtros no es posible determinar todo el contenido en frecuencia de una señal, por lo que poseer un LCD de mayor resolución no hubiera sido necesario en estos métodos. Sin embargo, debido a que la FFT fue el método que se seleccionó para realizar el procesamiento, la pantalla gráfica es la mejor opción para presentar la información al usuario.

IX. CONCLUSIONES

- No es posible implementar filtros FIR de tipo pasa-banda en el microcontrolador a causa de su orden, el cual supera las limitaciones de memoria disponibles y el tiempo de respuesta es inaceptable para el afinador.
- Es factible utilizar filtros IIR para la detección de frecuencias, sin embargo, dada las restricciones en la arquitectura del microcontrolador evitan obtener resultados precisos que son necesarios para poder afinar la cuerda.
- Se evidenció que el método de detección de frecuencias mediante la FFT es válido para determinar las frecuencias de afinación de guitarra y para los acordes del Círculo de Do.
- Dado que existe una implementación de la FFT en el microcontrolador y los circuitos utilizados no requieren de componentes externos se logró implementar un afinador portátil que no requiera de un computador personal.
- Debido a que la afinación ser realizará de forma visual al comparar las frecuencias medidas con las de afinación el error no será perceptible al oído humano.

X. RECOMENDACIONES

- Se recomienda para el desarrollo del afinador por medio de filtros FIR utilizar otro método de diseño además del de “ventanas”, para generar filtros con menor orden.
- Se recomienda utilizar un microcontrolador con unida de punto flotante para evitar afectar la precisión de las constantes de los filtros al momento de realizar la implementación.
- Se recomienda que, al momento de realizar la conversión analógica a digital, la implementación en un microcontrolador se con una mayor precisión de conversión para poder tomar muestras de mayor fidelidad de la guitarra.
- Se recomienda utilizar diferentes topologías de circuitos anti-alias para evitar la atenuación de señales de interés.
- Se recomienda la utilización de microcontroladores con mayor capacidad de almacenamiento o implementar una memoria externa para el almacenamiento de los datos temporales.

XI. BIBLIOGRAFÍA

B. H. Suits. 2015. *Física de la Música – Notas.* Universidad Tecnológica de Michigan. Estados Unidos de América. [Extraído el: 4/06/16]. En:

Bygone Tones, 2014. *Common problems with vintage guitar speakers & how to test for them. Bygone Tones*. Vintage Guitar Speakers & Cabinets. En: http://www.bygonetones.com/home/how-to-test-vintage-guitar-speakers

Chafchalaf, Detlev. 2013. *Diseño de un afinador musical digital por medio de Matlab*. Ciudad de Guatemala. Universidad de San Carlos de Guatemala. 114 págs.

Dawsons, 2013. *A guide to the main electric guitar pickup types*. Dawsons, UK. Inglaterra. En; http://www.dawsons.co.uk/blog/main-electric-guitar-pickup-types

EastRising, 2016. *ERM1602SBS-4 SPECIFICATIONS LCD MODULE*. EastRising. 17 págs.

Engelberg, S. 2008. *Digital Signal Proccesing, An Experimental Approach*. Inglaterra. Springer. 212 págs.

Gibson. 2016. *Min-ETune*. Estados Unidos de América. En: http://www.gibsonguitar.es/Productos/Min-ETune.aspx

Gomes, Luis. 2016. *Círculos de guitarra.* En; http://www.actiweb.es/animacion-escoge/archivo4.pdf

Guitar Book. 2012. *How to use a guitar tuner*. Guitar Book. En: http://www.guitar-book.com/how-to-tune-a-guitar.htm

Guitarristas. 2011. *Tipos de guitarra*. Guitarristas. España. En: http://www.guitarristas.info/foros/tipos-guitarras/131928.

La cuerda. 2016. *Estructura de la guitarra*. La cuerda. España. En: http://lacuerda.net/Recursos/cursoguitarra/?page=2

López, G & Gil, W. 2005. *Acordes Del Círculo de Do*. Método Actualizado para Guitarra.

Lopez, Javi. 2016. *Curso de Guitarra Eléctrica*. IES Poeta Paco Mollá. España. 43 págs. En: http://iespoetapacomolla.edu.gva.es/bach2016/javilopez/wp-content/uploads/2016/04/Curso-de-Guitarra-electrica.pdf

Lourde, M, *et-al*. 2009. *A Digital Guitar Tuner*. Dubai. Departamento de Ingeniería Eléctrica y Eletrónica BITS. 7 págs. En: https://arxiv.org/ftp/arxiv/papers/0912/0912.0745.pdf

Lyons, Richard. 1997. *Understanding digital signal processing*. Nueva Jersey: Bergen. Prentice Hall. 507 págs.

Lyons, Richard. 2011. *Understanding digital signal processing*. Nueva Jersey: Bergen. 3era Edición. Prentice Hall. 954 págs.

Marven, C & Ewers, G. 1996. *A Simple Approach to Digital Signal Proccesing.* Nueva York. Texas Instruments. 236 págs.

Microchip. 2016. *PIC32MX1XX/2XX 28/36/44-PIN*. Estados Unidos de América. Microchip Technology Inc. 344págs.

Molina, Leonel. 2016*. Asignación de nombres y numeración de dedos*. Universidad de Antioquia. Colombia. En: http://aprendeenlinea.udea.edu.co/lms/moodle/mod/page/view.php?id=105867&inpopup=1

Molina, Leonel. 2016. *Numeración de cuerdas y trastes*. Universidad de Antioquia. Colombia. En: http://aprendeenlinea.udea.edu.co/lms/moodle/mod/page/view.php?id=105868

Oppenhaim, A & Schafer, R. 1975. *Digital Signal Proccesing*. Nueva Jersey: Englewood Cliffs. Prentice Hall. 585 págs.

Padmanabham, T. 2000. *Industrial Instrumentation, principles and design.* Reino Unido. Springer. 649 págs.

Paolini, Eduardo & Chierchie, Fernando. 2016. *Filtros FIR*. Departamento de Ingeniería Eléctrica y de Computadoras, Universidad Nacional Del Sur. Argentina. En: http://www.ingelec.uns.edu.ar/pds2803/materiales/cap10/12-cap12.pdf

Texas Instruments. 2015. *PCM1795 32-bit. Texas Instruments*. Estados Unidos de América. 63págs.

UNAD. 2016. *El espectro de frecuencia audible*. Universidad Nacional Abierta y a Distancia. Colombia. En: http://datateca.unad.edu.co/contenidos/208037/MODULO_208037_TECNICAS_DE_GRABACION/leccin_4_el_espectro_de_frecuencia_audible.html

Vargas, Pedro. 2013. *Diseño y fabricación de una guitarra con cuerdas poliméricas, cuerpo sólido y sistema de captación de señales por un medio electromagnético*. Ciudad de Guatemala. Universidad del Valle de Guatemala. 63 págs.

Zurale, D, *et-al*. 2016. Automatic Guitar Tuner. Mumbai. Colegio de Ingeniería Dwarkadas. 6 págs. En: http://www.technofocus-djscoe.org/papers/V5I1/Automatic_Guitar_Tuner.pdf

XII. ANEXOS

A. Código microcontrolador pantalla

```
//8 de diciembre del 2016
//Codigo de Pantalla para Trabajo de Graduacion
//Diseno de afinador digital para guitarra electrica bajo el estandar e
//Victor Fuentes
//Carne 12298

//Codigo de libreria Adafruit GFX
//Codigo de libreria Adafruit TFTLCD
#include <Adafruit_GFX.h>
#include <Adafruit_TFTLCD.h>
#include <TouchScreen.h>
#include <Wire.h>

#if defined(__SAM3X8E__)
    #undef __FlashStringHelper::F(string_literal)
    #define F(string_literal) string_literal
#endif

#ifndef USE_ADAFRUIT_SHIELD_PINOUT
 #error "This sketch is intended for use with the TFT LCD Shield. Make sure that
USE_ADAFRUIT_SHIELD_PINOUT is #defined in the Adafruit_TFTLCD.h library file."
#endif


#define YP A1  // pin analogico
#define XM A2  // pin analogico
#define YM 7   // puede ser pin digital
#define XP 6   // puede ser pin digital

#ifdef __SAM3X8E__
  #define TS_MINX 125
  #define TS_MINY 170
  #define TS_MAXX 880
  #define TS_MAXY 940
#else
  #define TS_MINX  150
  #define TS_MINY  120
  #define TS_MAXX  920
  #define TS_MAXY  940
#endif

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

#define LCD_CS A3
#define LCD_CD A2
```

```
#define LCD_WR A1
#define LCD_RD A0

//Creacion de color en resolucion de 16 bits
#define BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF


Adafruit_TFTLCD tft;

#define BOXSIZE   40
#define PENRADIUS  2
int oldcolor, currentcolor, ingreso,nota,tiempo=0, tiempo1=0;
int envio[520];
int puntero=0;int copia=0;

//variables para validar que se dejo de apachar pantalla para
//modificar nota a afinar o acorde
bool subirnota=false;bool bajarnota=false;bool lleno=false;
String msn="";

void setup(void) {

  tft.reset();

  uint16_t identifier = tft.readID();

  tft.begin(identifier);

  tft.fillScreen(BLACK);

  currentcolor = WHITE;

  pinMode(13, OUTPUT);
  pantallaInicio();
  flechas();
  Wire.begin(40);               // Iniciar Esclavo I2C con direccion 40
  Wire.onReceive(receiveEvent); // Iniciar evento de esclavo I2C
  Serial.begin(115200);
  ingreso=7;
  nota=1;
}

#define MINPRESSURE 10
#define MAXPRESSURE 1000
```

```
void loop()
{
  digitalWrite(13, HIGH);
  TSPoint p = ts.getPoint();
  digitalWrite(13, LOW);


  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  //Presion minima de pantalla
  if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {



    p.x = map(p.x, TS_MINX, TS_MAXX, tft.height(),0);
    p.y = map(p.y, TS_MINY, TS_MAXX, 0,tft.width());

    int x=p.y;
    int y=p.x;

    //Validar posicion sobre botones de cambio de cuerda o acorde
    if(x>10 && x<45 && y>25 && y<55){//bajar nota
        //nota--;
        bajarnota=true;
        tiempo=millis();

     }else if (x>55 && x<90 && y>25 && y<55){//subir nota
                //nota++;
                subirnota=true;
                tiempo=millis();
     }else if (x>95 && x<267 && y>25 && y<55){//borrar lectura
                tft.fillRect(0, 100, tft.width(), tft.height(), BLACK);



       }
  }

  //Cuando no se este apachando la pantalla realizar modificaion de nota
  if(p.z==0 && subirnota==true || bajarnota==true){
     tiempo1=millis()-tiempo;
     if(tiempo1>250){
        if(subirnota==true){
          nota++;
          subirnota=false;
        }else if(bajarnota==true){
           nota--;
           bajarnota=false;
          }

        //Borrar pantalla, quitar marcas anteriores
```

```
        tft.fillRect(0, 56, tft.width(), tft.height(), BLACK);

        if (nota<=1){nota=1;}else if(nota>=10){nota=10;}
        if(nota==1){msn="1era cuerda";tft.drawLine(168,80,168,100,GREEN);}else
if(nota==2){msn="2nda cuerda";tft.drawLine(126,80,126,100,GREEN);}else
        if(nota==3){msn="3era
cuerda";tft.drawLine(99,80,99,100,RED);tft.drawLine(200,80,200,100,RED);}else
if(nota==4){msn="4ta
cuerda";tft.drawLine(75,80,75,100,RED);tft.drawLine(149,80,149,100,RED);}else
        if(nota==5){msn="5ta
cuerda";tft.drawLine(56,80,56,100,RED);tft.drawLine(113,80,113,100,RED);tft.drawLine(168,80,1
68,100,RED);}else if(nota==6){msn="6ta
cuerda";tft.drawLine(84,80,84,100,RED);tft.drawLine(42,80,42,100,RED);tft.drawLine(126,80,126,
100,RED);tft.drawLine(167,80,167,100,RED);}else
        if(nota==7){msn="Do mayor";
                tft.drawLine(135,90,135,100,GREEN);
                tft.drawLine(170,90,170,100,GREEN);
                tft.drawLine(69,90,69,100,GREEN);

                tft.drawLine(134,90,134,100,GREEN);
                tft.drawLine(169,90,169,100,GREEN);
                tft.drawLine(67,90,67,100,GREEN);

                tft.drawLine(133,90,133,100,GREEN);
                tft.drawLine(168,90,168,100,GREEN);
                tft.drawLine(68,90,68,100,GREEN);


                tft.drawLine(134,80,134,90,RED);
                tft.drawLine(201,80,201,90,RED);
                tft.drawLine(170,70,170,80,WHITE);
                tft.drawLine(100,70,100,80,WHITE);
                tft.drawLine(41,70,41,80,WHITE);
                tft.drawLine(41,80,41,90,RED);
                tft.drawLine(84,70,84,80,WHITE);
                tft.drawLine(84,80,84,90,RED);}else
        if(nota==8){msn="La menor";
                tft.drawLine(135,90,135,100,GREEN);
                tft.drawLine(115,90,115,100,GREEN);
                tft.drawLine(170,90,170,100,GREEN);//ya es doble de 165hz,

                tft.drawLine(134,90,134,100,GREEN);
                tft.drawLine(113,90,113,100,GREEN);
                tft.drawLine(169,90,169,100,GREEN);//ya es doble de 165hz,

                tft.drawLine(133,90,133,100,GREEN);
                tft.drawLine(114,90,114,100,GREEN);
                tft.drawLine(168,90,168,100,GREEN);//ya es doble de 165hz,

                tft.drawLine(41,70,41,80,WHITE);
                tft.drawLine(41,80,41,90,RED);
                tft.drawLine(126,70,126,80,WHITE);
```

```
        tft.drawLine(126,80,126,90,RED);

        tft.drawLine(83,70,83,80,WHITE);
        tft.drawLine(83,80,83,90,RED);
        tft.drawLine(56,70,56,80,WHITE);
        tft.drawLine(56,80,56,90,RED);
        tft.drawLine(113,70,113,80,WHITE);
        tft.drawLine(113,80,113,90,RED);
        tft.drawLine(170,70,170,80,WHITE);}else
if(nota==9){msn="Re menor";
        tft.drawLine(180,90,180,100,GREEN);
        tft.drawLine(149,90,149,100,GREEN);
        tft.drawLine(114,90,114,100,GREEN);

        tft.drawLine(179,90,179,100,GREEN);
        tft.drawLine(151,90,151,100,GREEN);
        tft.drawLine(113,90,113,100,GREEN);

        tft.drawLine(178,90,178,100,GREEN);
        tft.drawLine(150,90,150,100,GREEN);
        tft.drawLine(112,90,112,100,GREEN);

        tft.drawLine(41,70,41,80,WHITE);
        tft.drawLine(41,80,41,90,RED);

        tft.drawLine(56,70,56,80,WHITE);
        tft.drawLine(56,80,56,90,RED);

        tft.drawLine(75,70,75,80,WHITE);
        tft.drawLine(75,80,75,90,RED);


        tft.drawLine(83,70,83,80,WHITE);
        tft.drawLine(83,80,83,90,RED);

        tft.drawLine(113,70,113,80,WHITE);
        tft.drawLine(113,80,113,90,RED);

        tft.drawLine(149,70,149,80,WHITE);
        tft.drawLine(149,80,149,90,RED);


        tft.drawLine(126,70,126,80,WHITE);
        tft.drawLine(126,80,126,90,RED);

        tft.drawLine(169,70,169,80,WHITE);
        tft.drawLine(169,80,169,90,RED);}else
if(nota==10){msn="Sol mayor";
        tft.drawLine(200,90,200,100,GREEN);
        tft.drawLine(201,90,201,100,GREEN);
        tft.drawLine(199,90,199,100,GREEN);
```

```
                tft.drawLine(63,90,63,100,GREEN);
                tft.drawLine(62,90,62,100,GREEN);
                tft.drawLine(61,90,61,100,GREEN);

                tft.drawLine(50,90,50,100,GREEN);
                tft.drawLine(49,90,49,100,GREEN);
                tft.drawLine(48,90,48,100,GREEN);

                tft.drawLine(125,80,125,90,RED);
                tft.drawLine(189,80,189,90,RED);
                tft.drawLine(100,80,100,90,RED);
                tft.drawLine(150,80,150,90,RED);
                tft.drawLine(200,80,200,90,RED);}

      tft.fillRect(120, 0, 250, 20, BLACK);
      tft.setCursor(130, 0);
      tft.setTextColor(GREEN);  tft.setTextSize(2);
      tft.println(msn);
      tft.setCursor(270, 7);
      tft.setTextColor(RED);  tft.setTextSize(1);
      tft.println("Opcional");
      tft.setCursor(270, 0);
      tft.setTextColor(WHITE);
      tft.println("Aire");
      tft.setCursor(270, 15);
      tft.setTextColor(GREEN);
      tft.println("Obligado");



    }

 }

//en ciclo principal se grafica vector recibido por i2c
if (lleno==true){
   int var=0;

   if (nota>=1 && nota<=6 && copia!=2){
        tft.drawLine(envio[1],100,envio[1],150,YELLOW);
      }else if(copia!=2){
           while(var<copia){
                int asd=envio[var];

                if (asd<206 && asd >1){
                   tft.drawLine(asd,100,asd,150,YELLOW);
                }
                var=var+1;
           }
      }
   lleno=false;
 }
```

```
}

//Funcion para recibir datos por medio de i2c
void receiveEvent(int howMany)
{
  if(Wire.available() >0)
  {
    while(0 < Wire.available())
    {

      if (lleno==false){
          int c=Wire.read();
          envio[puntero]=c;
          puntero++;
          if (c>=253){lleno=true;copia=puntero;puntero=0;}
          }
    }
  }
}



//metodo para iniciar presentacion de pantalla
void pantallaInicio(){
  tft.fillScreen(BLACK);
  tft.setRotation(3);

  int alto=tft.height();
  int largo=tft.width();

  int i=0;
  tft.setCursor(0, 0);
  tft.setTextColor(BLUE);  tft.setTextSize(2);
  tft.println("Afinador V1.0");
  tft.print("Cargando");
  delay(500);
  tft.print(" . ");
  delay(500);
  tft.print(" . ");
  delay(500);
  tft.print(" . ");
  delay(1000);


  while(i<=largo){
    tft.drawLine(i,0, largo-i, alto, BLUE);
    i++;
  }
  i=0;
  while(i<=alto){
    tft.drawLine(0,i, largo, alto-i, BLUE);
    i++;
```

```
  }


  tft.fillScreen(BLACK);
  tft.setCursor(0, 0);
  tft.setTextColor(GREEN);  tft.setTextSize(2);
  tft.println("Afinando:");
}

//metodo para dibujar flechas en pantalla
void flechas(){
  //tft.drawRect(10,25,45,55, BLUE);
  tft.drawLine(10,25,45,25,GREEN);
  tft.drawLine(45,25,45,55,GREEN);
  tft.drawLine(10,25,10,55,GREEN);
  tft.drawLine(10,55,45,55,GREEN);

  tft.drawLine(12,40,30,26,GREEN);
  tft.drawLine(12,40,30,54,GREEN);

  //tft.drawRect(55,25,90,55, BLUE);
  tft.drawLine(55,25,90,25,GREEN);
  tft.drawLine(55,25,55,55,GREEN);
  tft.drawLine(90,25,90,55,GREEN);
  tft.drawLine(55,55,90,55,GREEN);

  tft.drawLine(88,40,73,26,GREEN);
  tft.drawLine(88,40,73,54,GREEN);

  //cuadro borrar
  tft.drawLine(95,25,267,25,GREEN);
  tft.drawLine(95,25,95,55,GREEN);
  tft.drawLine(267,25,267,55,GREEN);
  tft.drawLine(95,55,267,55,GREEN);

  tft.setCursor(97,30);
  tft.setTextColor(BLUE);  tft.setTextSize(2);
  tft.println("BORRAR LECTURA");

}
```

B. Código microcontrolador procesamiento de señal

```
//8 de diciembre del 2016
//Codigo de microcontrolador de procesamiento para Trabajo de Graduacion
//Diseno de afinador digital para guitarra electrica bajo el estandar e
//Victor Fuentes
//Carne 12298

#include <Wire.h>

//Definicion para ciclo de reloj
#pragma config FNOSC =PRIPLL
#pragma config FSOSCEN =OFF
#pragma config POSCMOD =XT
#pragma config OSCIOFNC =OFF
#pragma config FPBDIV =DIV_2

#pragma config FPLLIDIV =DIV_2
#pragma config FPLLMUL =MUL_20
#pragma config FPLLODIV =DIV_2
#pragma config UPLLIDIV =DIV_2
#pragma config UPLLEN =ON

#pragma config FWDTEN = OFF
#pragma config FCKSM = CSECMD

//Definicion de constantes para interrupcion
#define T3CON_ENABLE_BIT 0x8000
#define T3CON_PRESCALER_BITS 0x0070
#define T3_SOURCE_INT 0

//Definicion de prescalers
#define T3_PRESCALE_1_1 0
#define T3_PRESCALE_1_2 1
#define T3_PRESCALE_1_4 2
#define T3_PRESCALE_1_8 3
#define T3_PRESCALE_1_16 4
#define T3_PRESCALE_1_32 5
#define T3_PRESCALE_1_64 6
#define T3_PRESCALE_1_256 7

//Definir prescaler 1:2
#define PRESCALE T3_PRESCALE_1_2

//Ciclo de reloj de 40MHz
#define CLOCK_FREQ 40000000

//Definir periodo de interrupcion
#define INT_FREQUENCY 1000

//Definir pin para interrupcion, ver frecuencia de muestreo, rpb0, pin 4 de pic
```

```
#define IntOut 11

// This sets which bit LED 2 represents
// 0 sets it to the least significant bit
// 1 sets it to the next
// LEDs 3 and 4 will represent the next two bits
#define BIT_SHIFT 0

//Variables
volatile uint32_t count = 0;
volatile unsigned int flag = 0;
int mask = 0;
int lectura=0;

#define LOG2_N_WAVE 9                // 9, 10, 11, 12 bits
#define N_WAVE (1 << LOG2_N_WAVE)      // 512, 1024, 2048, 4096 muestras
#define N_WAVE34 (N_WAVE - N_WAVE / 4) // Tabla del Seno (3/4 onda)
short int real[N_WAVE];
short int imag[N_WAVE];
int16_t Sinewave[N_WAVE34];
short int envio[N_WAVE];
short int envio2[N_WAVE];


//Definicion de la Subrutina de Interrupcion
void __attribute__((interrupt)) myISR()
{

  flag = 1;
  //realizar lectura del pin analogico, rpb15, pin26 de pic
  lectura=analogRead(A2);
  if(count<N_WAVE){
    real[count]=lectura;
    imag[count]=0;
    count++;

  }
  //Limpiar bandera de interrpcion
  clearIntFlag(_TIMER_3_IRQ);
}

//Tomado de: http://www.instructables.com/id/Timer-Interrupts-on-the-DP32/
//Definir el timer e Interrupcion
void start_timer_3(uint32_t frequency)
{
  uint32_t period;

  // Calcular periodo de interrupcion
  if (PRESCALE == 7) period = 256; // 1:256 is a special case
  else period = 1 << PRESCALE;
  period = period * frequency;
  period = CLOCK_FREQ / period;
```

```
  // Configurar Timer3
  T3CONCLR = T3CON_ENABLE_BIT;           // Apagar Timer
  T3CONCLR = T3CON_PRESCALER_BITS;       // Borrar Prescaler
  mask = PRESCALE << 4;                  // Configurar Prescaler
  mask = mask | T3CON;                   // Mask our prescaler
  T3CON = mask;                          // Configurar Prescaler
  TMR3 = 0;                              // Limpiar contador
  PR3 = period;                          // Configurar Periodo
  T3CONSET = T3CON_ENABLE_BIT;           // Encender Timer
}

void setup(){

  // Iniciar Timer
  start_timer_3(INT_FREQUENCY);

  //Configurar el vector de interrupcion con parametros del Timer y la Subrutina de Interrupcion
  setIntVector(_TIMER_3_VECTOR, myISR);

  //Configurar prioridad de interrupcion
  setIntPriority(_TIMER_3_VECTOR, 4, 0);

  //Limpiar bandera de interrupcion
  clearIntFlag(_TIMER_3_IRQ);

  //Iniciar interrupcion
  setIntEnable(_TIMER_3_IRQ);

  //Definir pin de interrupcion para ver frecuencia de muestreo
  pinMode(IntOut,OUTPUT);

  //Iniciar el modulo i2c
  //pin 18 SDA
  //pin 17 SCL
  Wire.begin();

  //Iniciar modulo serial
  Serial0.begin(115200);

  //Generar tabla de Seno
  for(uint16_t i = 0; i < N_WAVE34; i++){
    Sinewave[i] = int(sin(i * 2 * PI / N_WAVE) * 32767);
    }
}


void loop(){

  if(Serial0.available() || lectura>630 )
    {
```

```
int serial=Serial0.read();
int puntero=0;
Serial0.println((char)serial);
//si envia 1 ascii.
if (serial==49 || lectura>630){
    Serial0.println("fue 1");
      count=0;
      int i =0;
      while(count<N_WAVE){
         i=0;
        }

      uint32_t s=disableInterrupts();

      Serial0.println("TRANSFORMADA1");
      //Iniciar algoritmo de FFT
      fix_fft( real, imag,LOG2_N_WAVE,0);
      i=0;
      Serial0.println("TRANSFORMADA");

      while(i<N_WAVE/2-3){

        int intensity = sqrt(pow(real[i], 2) + pow(imag[i], 2));

        if (intensity>10){

          //Serial0.print(i*0.4882); //1000/2048
          //Serial0.print(i*0.9765);  //1000/1024
          Serial0.print(i*1.9531);  //1000/512
          Serial0.print(" Hz, ");
          Serial0.print(i);
          Serial0.print(", ");
          Serial0.println(intensity);
          envio[puntero]=i;
          if(intensity>100){intensity=99;}
          envio2[puntero]=intensity;
          puntero++;
          }
        i++;
      }

      count=0;
      restoreInterrupts(s);

  }

  //Envio i2c
  Wire.beginTransmission(40);
  for (int i=0;i<puntero;i++){

     Wire.write(envio[i]);
    }
```

```
      //Caracter de fin
      Wire.write(254);
      Wire.endTransmission();
      Serial0.flush();
  }

  //Encender pin para mostrar interrupcion
   if (flag){
     digitalWrite(IntOut,HIGH);
     flag=0;
   } else {digitalWrite(IntOut,LOW);}


}

//Codigo de Transformada Rapida de Fourier
/* fix_fft.c - Fixed-point in-place Fast Fourier Transform  */
/*
  All data are fixed-point short integers, in which -32768
  to +32768 represent -1.0 to +1.0 respectively. Integer
  arithmetic is used for speed, instead of the more natural
  floating-point.

  For the forward FFT (time -> freq), fixed scaling is
  performed to prevent arithmetic overflow, and to map a 0dB
  sine/cosine wave (i.e. amplitude = 32767) to two -6dB freq
  coefficients. The return value is always 0.

  For the inverse FFT (freq -> time), fixed scaling cannot be
  done, as two 0dB coefficients would sum to a peak amplitude
  of 64K, overflowing the 32k range of the fixed-point integers.
  Thus, the fix_fft() routine performs variable scaling, and
  returns a value which is the number of bits LEFT by which
  the output must be shifted to get the actual amplitude
  (i.e. if fix_fft() returns 3, each value of fr[] and fi[]
  must be multiplied by 8 (2**3) for proper scaling.
  Clearly, this cannot be done within fixed-point short
  integers. In practice, if the result is to be used as a
  filter, the scale_shift can usually be ignored, as the
  result will be approximately correctly normalized as is.

  Written by:  Tom Roberts  11/8/89
  Made portable:  Malcolm Slaney 12/15/94 malcolm@interval.com
  Enhanced:  Dimitrios P. Bouras  14 Jun 2006 dbouras@ieee.org
*/

/*
  FIX_MPY() - fixed-point multiplication & scaling.
  Substitute inline assembly for hardware-specific
  optimization suited to a particluar DSP processor.
  Scaling ensures that result remains 16-bit.
*/
```

```
inline short FIX_MPY(short a, short b)
{
  /* shift right one less bit (i.e. 15-1) */
  int c = ((int)a * (int)b) >> 14;
  /* last bit shifted out = rounding-bit */
  b = c & 0x01;
  /* last shift + rounding bit */
  a = (c >> 1) + b;
  return a;
}

/*
  fix_fft() - perform forward/inverse fast Fourier transform.
  fr[n],fi[n] are real and imaginary arrays, both INPUT AND
  RESULT (in-place FFT), with 0 <= n < 2**m; set inverse to
  0 for forward transform (FFT), or 1 for iFFT.
*/
int fix_fft(short fr[], short fi[], short m, short inverse)
{
  int mr, nn, i, j, l, k, istep, n, scale, shift;
  short qr, qi, tr, ti, wr, wi;

  n = 1 << m;

  /* max FFT size = N_WAVE */
  if (n > N_WAVE)
    return -1;

  mr = 0;
  nn = n - 1;
  scale = 0;

  /* decimation in time - re-order data */
  for (m=1; m<=nn; ++m) {
    l = n;
    do {
      l >>= 1;
    } while (mr+l > nn);
    mr = (mr & (l-1)) + l;

    if (mr <= m)
      continue;
    tr = fr[m];
    fr[m] = fr[mr];
    fr[mr] = tr;
    ti = fi[m];
    fi[m] = fi[mr];
    fi[mr] = ti;
  }

  l = 1;
  k = LOG2_N_WAVE-1;
```

```
while (l < n) {
  if (inverse) {
    /* variable scaling, depending upon data */
    shift = 0;
    for (i=0; i<n; ++i) {
      j = fr[i];
      if (j < 0)
        j = -j;
      m = fi[i];
      if (m < 0)
        m = -m;
      if (j > 16383 || m > 16383) {
        shift = 1;
        break;
      }
    }
    if (shift)
      ++scale;
  } else {
    /*
      fixed scaling, for proper normalization --
      there will be log2(n) passes, so this results
      in an overall factor of 1/n, distributed to
      maximize arithmetic accuracy.
    */
    shift = 1;
  }
  /*
    it may not be obvious, but the shift will be
    performed on each data point exactly once,
    during this pass.
  */
  istep = l << 1;
  for (m=0; m<l; ++m) {
    j = m << k;
    /* 0 <= j < N_WAVE/2 */
    wr =  Sinewave[j+N_WAVE/4];
    wi = -Sinewave[j];
    if (inverse)
      wi = -wi;
    if (shift) {
      wr >>= 1;
      wi >>= 1;
    }
    for (i=m; i<n; i+=istep) {
      j = i + l;
      tr = FIX_MPY(wr,fr[j]) - FIX_MPY(wi,fi[j]);
      ti = FIX_MPY(wr,fi[j]) + FIX_MPY(wi,fr[j]);
      qr = fr[i];
      qi = fi[i];
      if (shift) {
        qr >>= 1;
```

```
          qi >>= 1;
        }
        fr[j] = qr - tr;
        fi[j] = qi - ti;
        fr[i] = qr + tr;
        fi[i] = qi + ti;
      }
    }
    --k;
    l = istep;
  }
  return scale;
}
```

C. Manual de usuario de afinador

Universidad del Valle de Guatemala Enero 2017
Tesis de Trabajo de Graduación
Diseño de un afinador digital para guitarra eléctrica bajo el estándar E
Victor Orlando Fuentes Araujo

Manual de Usuario

Indicaciones de uso:

- La guitarra eléctrica debe tener tonalidad natural o "sonido limpio".

Conexión al afinador:

- Utilice cable "Jack 6.3 milímetros macho" para conectar la salida de línea de la guitarra eléctrica al puerto hembra del afinador, ubicado a un costado de la carcasa.

Modos de operación:

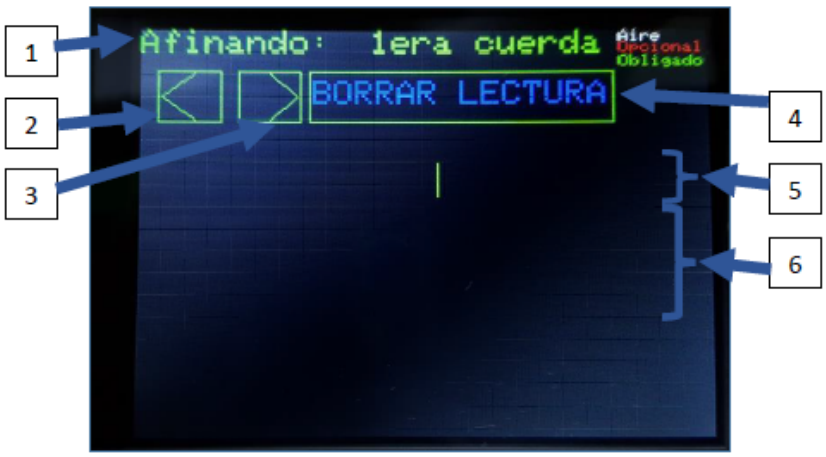
- Cuerdas: 1era., 2nda., 3era., 4ta., 5ta. y 6ta.
- Acordes del círculo de DO: Do mayor, La mayor, Mi mayor y Sol mayor.

Instrucciones de uso:

1. Realice la conexión al afinador explicada en la sección "Conexión al afinador".
2. Utilice las flechas derecha o izquierda, explicadas en la sección "Partes del afinador", para seleccionar el modo de operación del afinador.
3. Ejecute el acorde o la cuerda y verifique que la lectura de "Pantalla secundaria" concuerde con las marcas de "Pantalla principal".
4. Si la lectura se encuentra a la izquierda de las marcas, afloje la cuerda.
5. Si la lectura se encuentra a la derecha de las marcas, apriete la cuerda.
6. Utilice el botón "Borrar Lectura" para limpiar la pantalla.

Partes del afinador:

1. Mensaje principal: indica en qué modo se encuentra operando el afinador.
2. Flecha izquierda: permite desplazarse por los distintos modos de operación del afinador.
3. Flecha derecha: permite desplazarse por los distintos modos de operación del afinador.
4. Botón "Borrar Lectura": permite borrar la lectura anterior del afinador.
5. Pantalla principal: se muestran las marcas guía, blancas, rojas y verdes.
    a. Blancas: indican cuerdas tocadas "al aire".
    b. Rojas: indican marcas opcionales para la cuerda o el acorde.
    c. Verdes: indican marcas obligatorias para la nota o el acorde.
6. Pantalla secundaria: se muestra la lectura actual de la guitarra.



XIII. GLOSARIO

Aliasing: efecto que se genera en una señal digital debido al traslape de frecuencias altas que contiene una señal a frecuencias bajas debido a una selección de frecuencia de muestreo errónea.

Ancho de banda: espectro de todas las frecuencias de interés que se deseen analizar o procesar.

Comunicaciones en serial: comunicación basada en pulsos por una única línea de transmisión, utiliza una línea secundaria para realizar sincronización entre dispositivos.

DAC: abreviatura para el conversor de digital a analógico, módulo implementado en un circuito integrado. Se encarga de realizar la conversión de una señal digital a una señal continua de voltaje.

Filtro pasa bajas: filtro que amplificará las frecuencias menores a su frecuencia de corte, y atenuará las frecuencias por encima de su frecuencia de corte.

Filtros de respuesta al impulso de duración finita o FIR:

Filtros de respuesta al impulso de duración infinita o IIR:

Frecuencia de corte: frecuencia de diseño para realizar la atenuación desde un filtro.

Hardware: capa física de las computadoras que opera a nivel de electrónica y que no puede ser modificada.

Memoria RAM: memoria de acceso aleatorio que almacena información de manera volátil.

Microcontrolador: dispositivo electrónico que incluye un núcleo de procesamiento, unidad lógica y unidad de memoria encapsulado.

Mips: acrónimo para millón de instrucciones procesadas en un segundo.

Pastillas o pickups: transductor de la guitarra eléctrica que genera la señal eléctrica respondiendo a la vibración de la cuerda.

Pipeline: módulo de memoria dentro del núcleo de procesamiento por el cual son analizadas y ejecutadas las instrucciones del microcontrolador.

Software: capa virtual de las computadoras que puede ser modificada.