

Diseño e implementación de un prototipo portátil de ecualizador de audio de bajo costo que funcione en tiempo real para compensar las deficiencias auditivas humanas

Pablo Leonidas Cevallos Duque, Jorge Mauricio Quitama Haro,

Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito - Ecuador

Resumen. Este trabajo toma resultados de una tesis anterior "Implementación de un audiómetro basado en software y su empleo en el diseño de un ecualizador para compensar las deficiencias auditivas en tiempo real". Nos interesan los siguientes resultados de dicho trabajo: primero, el resultado de la audiometría para determinar las deficiencias auditivas a través del software implementado. Segundo, el diseño del filtro digital (creación de coeficientes) para compensar las deficiencias auditivas a través del software implementado. Nuestro trabajo consiste en tomar los coeficientes de este filtro digital para procesarlos y diseñar el ecualizador de audio digital, con la gran diferencia que lo implementamos en software y hardware.

I. INTRODUCCIÓN

Planteamiento del problema

Diseñar e implementar un prototipo portátil de ecualizador de audio a bajo costo y que funcione en tiempo real para compensar las deficiencias auditivas humanas.

Solución

En nuestros tiempos hay muchas personas que van perdiendo el sentido del oído, sea cual fuere la causa, hasta ahora una solución era comprar dispositivos auriculares electrónicos, pero son muy caros debido a su miniaturización, pues nuestra propuesta es implementar un prototipo con tecnología digital y en tiempo real que permita solucionar éste problema, pero a un precio asequible. Tal es así que después de analizar varias opciones existentes en el mercado, se ha escogido la

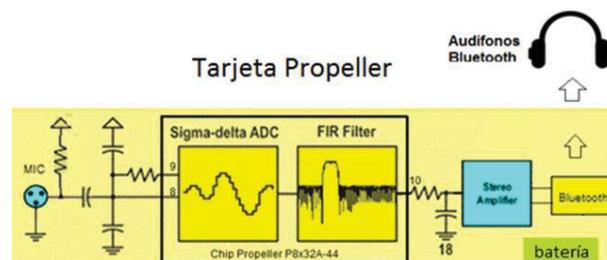
tarjeta demo Propeller de la compañía Parallax que nos permitirá implementar nuestro prototipo con los requerimientos de bajo costo que sea fácilmente adaptable al usuario y además con el uso de audífonos inalámbricos (bluetooth) que pueden remplazar perfectamente aquellos dispositivos electrónicos usados detrás de la oreja para suplir la deficiencia auditiva.

II. MATERIAL

Implementación en Hardware

Para la ecualización del audio se diseñó un sistema basado en algoritmos de filtros digitales FIR (Respuesta Finita al Impulso), Aritmética Distribuida (DA) con Dígito Canónico Signado (CSD), que fueron implementados en la Tarjeta Demo Propeller de Parallax en donde se captura el sonido de audio a través del micrófono, luego los datos son enviados hacia el chip Propeller P8X32A-44 para realizar la conversión análoga-digital (A/D), la ecualización y la conversión digital-análoga (D/A), los datos ecualizados son enviados a una salida estéreo amplificada y finalmente son enviados al Transmisor de audio Bluetooth BTT005.

Fig. 1. Esquema Ecualización de audio a través de la Tarjeta Demo Propeller



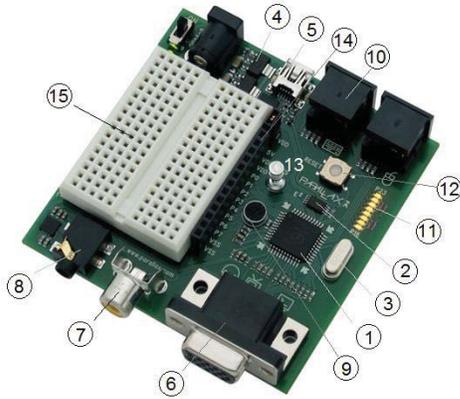


Fig. 2. Tarjeta Demo Propeller

Características de la Tarjeta Demo Propeller

1. Chip Propeller P8X32A-Q44.
2. EEPROM 24LC256-I/ST para almacenamiento de programas.
3. Cristal de 5.000 MHz reemplazable.
4. Reguladores de 3.3V y 5V con switch de prendido/apagado.
5. Interface USB-a-serial para cargar y comunicación.
6. Salida VGA.
7. Salida TV.
8. Salida estéreo con amplificador de audífonos de 16-ohm.
9. Entrada de micrófono Electret.
10. Conectores PS/2 para mouse y teclado de entrada/salida.
11. 8 LEDs (comparten los pines VGA).
12. Botón push para reseteo.
13. Gran post de tierra para enganche de largo alcance.
14. Los pines de entrada/salida P0-P7 están libres para conexión con cabezales.
15. Protoboard para circuitos propios.

Para el prototipo portátil se construyó una caja de material acrílico, la cual constará la Tarjeta Demo Board, el Transmisor de Audio Bluetooth BTT005 y la batería recargable.

Este prototipo será llevado por la persona con deficiencia auditiva en el pecho para recibir la señal de audio, y establecerá comunicación inalámbrica con los Audífonos Bluetooth Motorola S10-HD para poder escuchar la señal ecualizada en tiempo real.



Fig.3. Prototipo ubicación Pecho



Fig.4. Tarjeta Demo Propeller y Audífonos Bluetooth

III. MÉTODOS

Implementación de Software

FILTROS DIGITALES FIR

Los filtros digitales de Respuesta Finita Impulsiva o filtros FIR por sus siglas en ingles Finite Impulse Response, se trata de un tipo de filtros digitales en el que, como su nombre indica, si la entrada es una señal impulso la salida tendrá un número finito de términos no nulos. La estructura de señal a la salida del filtro se basa solamente en la combinación lineal de las entradas actuales y anteriores, esto es:

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k] = \sum_{k=0}^{N-1} h[k] x[n-k] \text{ con } h[k] = \{h_0, h_1, \dots, h_{N-1}\}$$

Secuencia de salida del filtro¹

En donde N es el orden del filtro, que también coincide con el número de términos no nulos y con el número de coeficientes b_k del filtro. Observe que la expresión

¹ www.utn.edu.ar/download.aspx?idFile=4655

corresponde a la convolución de la señal de entrada $x[n]$ con la respuesta impulso del filtro FIR $h[n]$.

Aplicando la transformada Z a la respuesta impulso del filtro FIR $h[n]$, se tiene:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)}$$

Función de transferencia del filtro²

En la Figura 5, se muestra el diagrama en bloques de la estructura básica del filtro FIR.

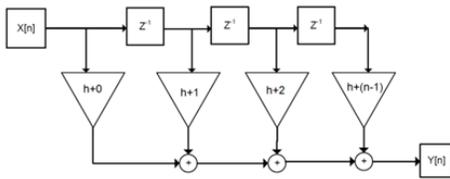


Fig. 5. Filtro FIR

Algoritmo Filtro FIR

MULTIPLICADORES DÍGITO CANÓNICO SIGNADO (CSD)

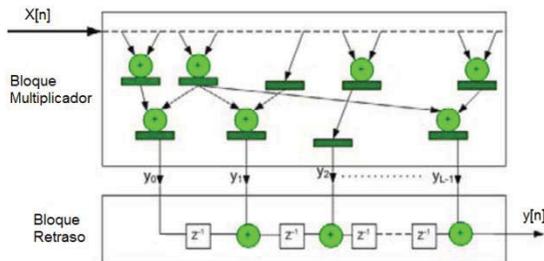


Fig. 6. Bloque Multiplicador usando sumas y desplazamientos

Para el caso donde uno de los multiplicandos es constante, los multiplicadores CSD (Canonic-Signed Digit) son la mejor opción para diseños de bajo consumo, ya que son de menor tamaño y más reciente en el uso de energía que los multiplicadores de propósito general. Para realizar una multiplicación con CSD se sustituye el término constante por una serie de desplazamientos y sumas del multiplicando variable. Los desplazamientos necesarios para obtener el producto son definidos por la

codificación CSD del operando constante, como se observa en la Fig. 6.

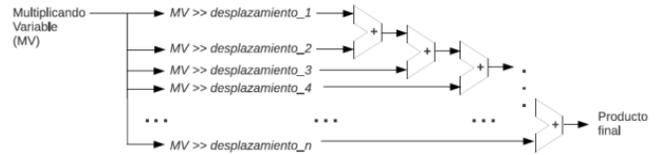


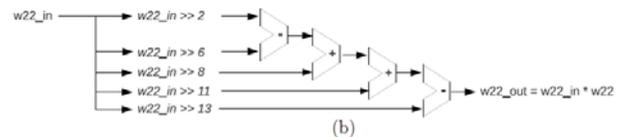
Fig. 7. Ejemplo de una multiplicación usando la técnica CSD

Un número en CSD tiene menos dígitos distintos de cero que su equivalente en codificación binaria, lo que reduce la cantidad de sumas necesarias para realizar una multiplicación.

Al ser una representación canónica, no hay dígitos distintos de cero adyacentes entre sí, por lo que dicha reducción puede alcanzar en el peor de los casos el 50% en comparación con un multiplicador de propósito general.

Esto implica que el sistema debe hacer menos cálculos, tendrá menos conmutación, ocupa menos área y tiene menor consumo de potencia.

$$w_{22} = 0,2386699307646236_{10} = 0.0010000\bar{1}010010\bar{1}00_{CSD} = 2^{-2} - 2^{-6} + 2^{-8} + 2^{-11} - 2^{-13}$$



Ejemplo de una multiplicación usando la técnica CSD³

Fig. 8. Multiplicador CSD

En la Fig. 8(a) se obtiene a la codificación CSD del multiplicando constante correspondiente al coeficiente **w22**. En Fig. 8(b) se observa la cantidad de desplazamientos a la derecha que se deben hacer a la señal de entrada **w22_in** y las sumas y restas necesarias para obtener el producto de dicha señal por la constante **w22**.

FIR USANDO ARITMÉTICA DISTRIBUIDA

Aritmética Distribuida (DA) es un enfoque diferente para la implementación de filtros digitales. La idea básica

² www.utn.edu.ar/download.aspx?idFile=4655

³ bibliodigital.itcr.ac.cr:8080/xmlui/bitstream/handle/.../Tesis.pdf?...1

es sustituir todas las multiplicaciones y adiciones por una tabla y un desplazador acumulador. DA se basa en el hecho de que los coeficientes de filtro son conocidos, por lo que multiplicar $c[n] \cdot x[n]$ se convierte en una multiplicación por una constante. Esta es una diferencia importante y un requisito previo para un diseño de DA.

El diagrama de bloques para la aplicación de DA filtro FIR se muestra en la Fig. 5, bien la aplicación de DA es necesario para almacenar las entradas como igual a la longitud coeficiente en la etapa buffer.

Una vez hecho esto, el bit menos significativo (LSB) de todos los coeficientes se toma como la dirección a la Tabla de Consulta (Look up Table [LUT]). Esto es, una LUT palabra 2^n está preprogramado para aceptar una dirección de N bits, donde N es el número de coeficientes. Asignaciones individuales se ponderan por la potencia apropiada de dos factores y el acumulador. La acumulación utiliza de manera eficaz usando un cambio sumador como se muestra en la Fig. 9. Para la aplicación de hardware, en lugar de cambiar cada valor intermedio por factor de potencia que requiere un desplazador costoso, cambiar el contenido del acumulador sí mismo en cada dirección de un bit a la derecha.

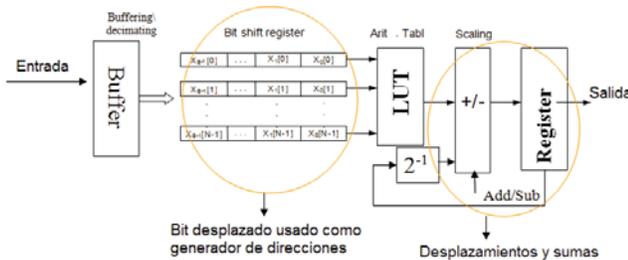


Fig. 9. Diagrama de Bloque Aritmética Distribuida

EJEMPLO SIGNADO DE ARITMÉTICA DISTRIBUIDA.

Supongamos que los datos se dan en (N = 5bit) dos de codificación cumplido y los coeficientes son $c[0] = 2$, $c[1] = 3$, $c[2] = 1$, $c[3] = 4$, la LUT correspondiente se muestra en la Tabla 1.

Tabla 1. Tabla Aritmética Distribuida

Step	$x_c[3]$	$x_c[2]$	$x_c[1]$	$x_c[0]$	$f(c[k], x[n]) =$	Sum
0	0	0	0	0	$4x0+1x0+3x0+2x0$	0
1	0	0	0	1	$4x0+1x0+3x0+2x1$	2
2	0	0	1	0	$4x0+1x0+3x1+2x0$	3
3	0	0	1	1	$4x0+1x0+3x1+2x1$	5
4	0	1	0	0	$4x0+1x1+3x0+2x0$	1
5	0	1	0	1	$4x0+1x1+3x0+2x1$	3
6	0	1	1	0	$4x0+1x1+3x1+2x0$	4
7	0	1	1	1	$4x0+1x1+3x1+2x1$	6
8	1	0	0	0	$4x1+1x0+3x0+2x0$	4
9	1	0	0	1	$4x1+1x0+3x0+2x1$	6
10	1	0	1	0	$4x1+1x0+3x1+2x0$	7
11	1	0	1	1	$4x1+1x0+3x1+2x1$	9
12	1	1	0	0	$4x1+1x1+3x0+2x0$	5
13	1	1	0	1	$4x1+1x1+3x0+2x1$	7
14	1	1	1	0	$4x1+1x1+3x1+2x0$	8
15	1	1	1	1	$4x1+1x1+3x1+2x1$	10

Los valores de $x[n]$ como, $x[0] = 1$, $x[1] = 3$, $x[2] = 7$, $x[3] = 15$. La salida muestra en el índice k, es decir, se define como se muestra en la Tabla 2.

Tabla 2. Cálculos Aritmética Distribuida

step(t)	$x_t[3]$	$x_t[2]$	$x_t[1]$	$x_t[0]$	$f(t) \times 2t + Y[t-1] =$	$Y[t]$
0	1	1	1	1	$10x2^0 + 0$	10
1	1	1	1	0	$8x2^1 + 10$	26
2	1	1	0	0	$5x2^2 + 26$	46
3	1	0	0	0	$4x2^2 + 46$	78
	$x_t[3]$	$x_t[2]$	$x_t[1]$	$x_t[0]$	$-f(t) \times 2t + Y[t-1]$	
4	1	1	1	0	$-8x2^2 + 78$	-50

Los resultados de la comprobación numérica

$$c[0] \times x[0] + c[1] \times x[1] + c[2] \times x[2] + c[3] \times x[3] + c[4] \times x[4] = x[2] + (-13) \times x[3] + (-9) \times x[1] + (-1) \times x[4] = -50$$

DIAGRAMA DE FLUJO FILTRO PRINCIPAL

Este proceso comienza con la inicialización, donde se arranca el filtro y se define la fuente de la señal de entrada, que pueden ser los datos que han sido previamente guardados en un espacio de la Memoria Flash Serie del dispositivo (Filtrado fuera de línea), o las muestras que se obtienen desde un conversor A/D (filtrado en línea). Es necesario definir también el destino de la Señal Filtrada.

Se realiza la configuración ADC/DAC y se lee la señal de entrada (micrófono).

Luego se procede a enviar los datos ADC al Programa Filtro FIR en donde se realiza el proceso de filtrado, una vez concluido lo reenvía al programa Principal.

Por último se tiene la Señal Filtrada lista para enviar al amplificador y luego hacia los audífonos.

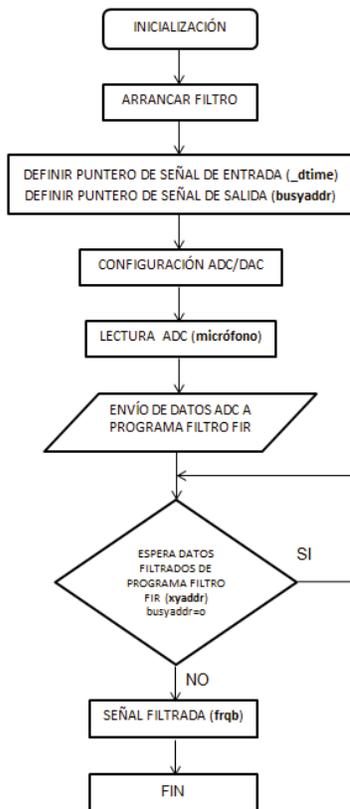


Fig. 10. Diagrama de flujo Filtro Principal

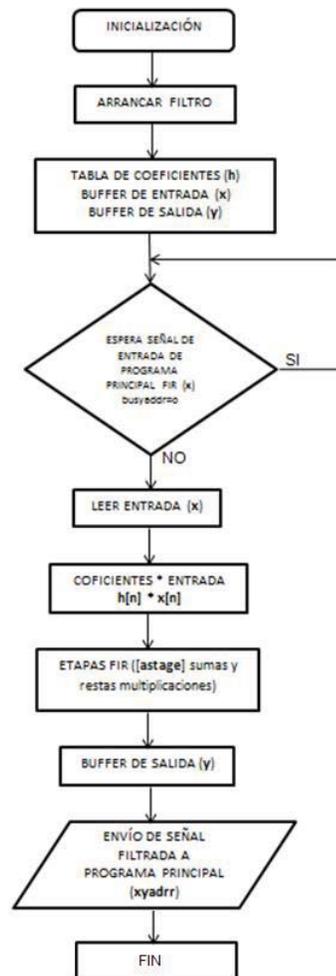


Fig. 11. Diagrama de flujo FILTRO FIR

DIAGRAMA DE FLUJO FILTRO FIR

El proceso de filtrado comienza con la inicialización, donde se arranca el filtro. Se definen las direcciones Tabla de Coeficientes, de los buffers de entrada y de salida. El buffer de entrada se llena según el número de coeficientes del filtro, para así garantizar igual número de retardos que de coeficientes.

Se lee el buffer de entrada es decir los datos que se envió de Programa Principal.

Luego se procede a realizar la multiplicación entre la tabla de coeficientes y las muestras que se encuentran en el buffer de entrada. Cada muestra multiplicada y sumada se envía al buffer de salida y luego se lo envía al Programa Principal.

GENERACIÓN DE COEFICIENTES EN MATLAB

```
% Diseñe un filtro pasa bajo 1KHz.
% Asuma una frecuencia de muestreo de 20000 Hz.
% Orden del filtro = 90
%
clear all; close all;
fs = 20000; % frecuencia de muestreo
Fmax = fs/2;
order = 90; % Orden del filtro
wn = [1500]/ Fmax; % vector de frecuencias
b = fir1(order,wn,'high') % diseño del filtro FIR,
b'
% Respuesta en frecuencia:
% Diagrama de Bode
freqz(b,1,512,20000);
% Plot módulo de h:
[h,freq] = freqz(b,1,512,20000);
figure
plot(freq,abs(h),'k');
xlabel('Frecuencia (Hz)'); ylabel('H(f)');
```

figure
 plot(freq,20*log10(abs(h)), 'k');
 xlabel('Frecuencia (Hz)'); ylabel('H(f) en dB');

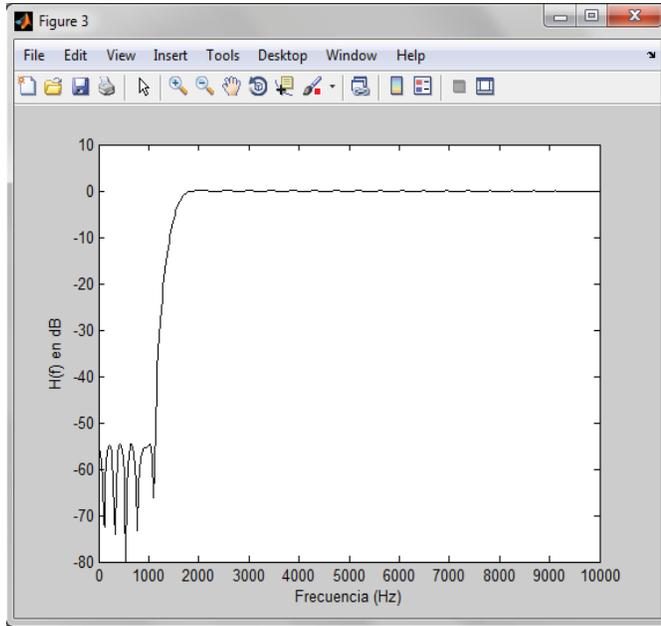


Fig. 11. Filtro pasa bajos 1.5KHz

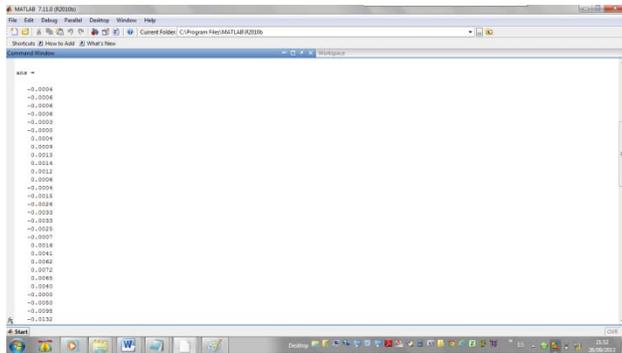


Fig. 12. Coeficientes Generados

Se copia estos coeficientes a la interfaz gráfica Generador Asembler Propeller SPIN y se genera el programa FILTRO FIR que será utilizado por el PROGRAMA PRINCIPAL para realizar la ecualización. Ver Fig. 13.

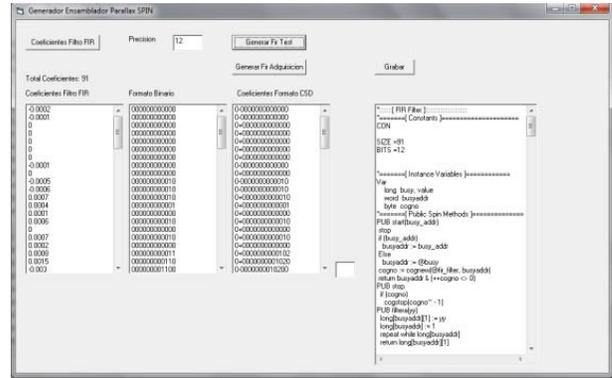


Fig. 13. Generación de programa FILTRO FIR

A continuación se carga el PROGRAMA PRINCIPAL en el software Propeller Tools de Parallax,

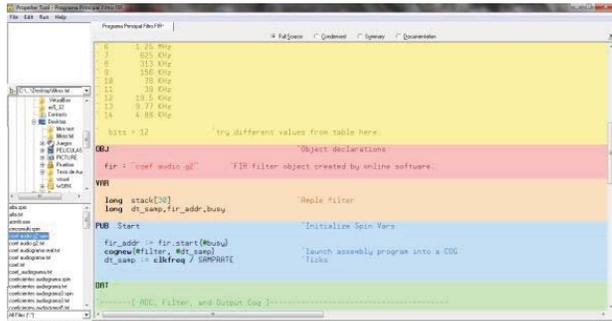


Fig. 14. Cargar PROGRAMA PRINCIPAL en Software Propeller Tools

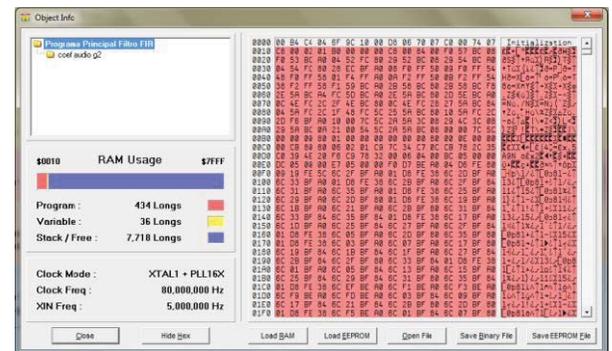


Fig. 15. Compilación de programa

Finalmente se cargará el programa en la tarjeta Demo Propeller.



Fig. 16. Programación de Prototipo de Audio

PRUEBAS FILTROS

Para realizar las pruebas se usa la señal de Ruido Blanco generado anteriormente Fmuestreo=20KHz ver Fig.18

Filtro pasa bajos



Fig.19. Filtro pasa bajos 1.5KHz.

IV. RESULTADOS

GENERACIÓN DE RUIDO BLANCO PARA PRUEBAS

Mediante el software **Impuse Lite Audio Analyzer** se genera ruido blanco.



Fig.17. Software Impulse Lite Audio Analyzer



Fig. 20: Filtro pasa bajos 3KHz

Filtro pasa altos



Fig. 18. Analizador de Espectro de la señal de Ruido Blanco



Fig.21. Filtro pasa altos 1.5KHz



Fig.22. Filtro pasa altos 3KHz

Filtro pasa banda



Fig.23. Filtro pasa banda 1 y 3 KHz

Filtro rechaza banda



Fig.25. Filtro rechaza banda 4 y 5 KHz.



Fig.26. Filtro rechaza 1-2 KHz y 5-6 KHz.

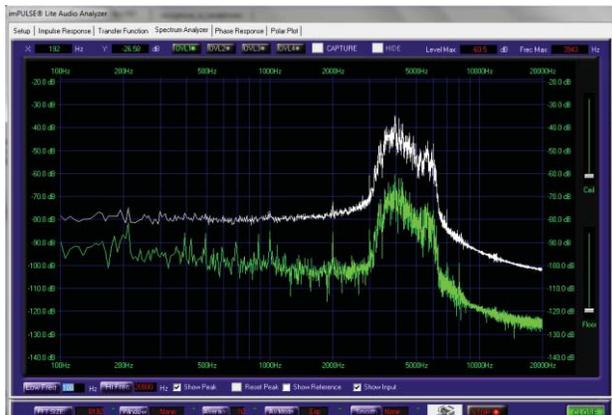


Fig.24. Filtro pasa banda 3 y 6 KHz

PRUEBAS REALIZADAS A PERSONAS CON DEFICIENCIA AUDITIVA.

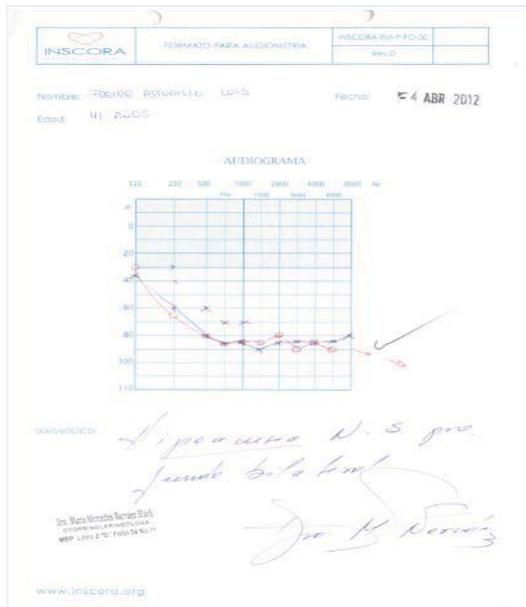


Fig.27. Audiograma de persona discapacitada del CONADIS

Para poder resolver la deficiencia auditiva tendremos que generar una señal ecualizada en Matlab en las frecuencias donde tenemos problemas según el audiograma.

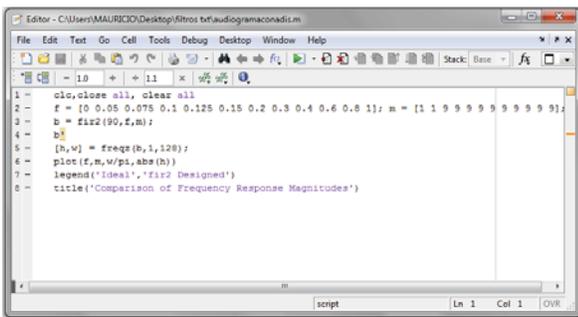


Fig.28. Generación de coeficientes de señal ecualizada en MATLAB

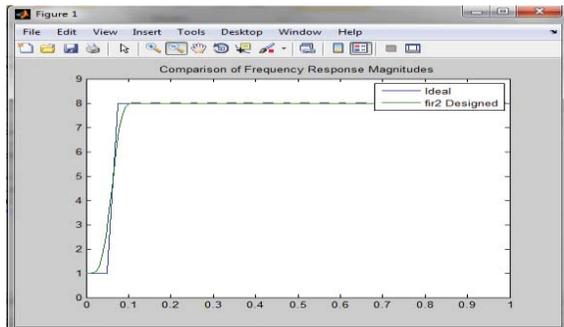


Fig.29. Señal ecualizada en MATLAB de acuerdo a audiograma.



Fig. 30. Pruebas con personas con deficiencia auditiva

OBSERVACIONES:

- Tomando en consideración que el grado de la pérdida auditiva puede ser de leve a profundo, que la pérdida de audición puede ser mayor o menor en cada oído; el Prototipo puede ser útil para Personas con el siguiente perfil: Pérdida auditiva leve (Una persona con pérdida auditiva leve puede escuchar algunos sonidos del habla, pero no oye claramente los susurros), que haya utilizado permanentemente audífonos desde que se diagnosticó la sordera.
- El volumen del prototipo es bajo, no abarca sonidos del entorno inmediato.
- El equipo debe ser calibrado y configurado para cada persona dependiendo de su deficiencia auditiva, que debe ser leve, si es profunda las personas con este tipo de deficiencia auditiva no podrán escuchar totalmente, es decir no habrá utilidad en el uso del equipo.
- Se debería hacer un molde (para cada usuario) y éste conectado al prototipo.
- La presentación del prototipo portátil debería ser más pequeña.

V. CONCLUSIONES

El Prototipo de ecualizador de audio se realizó en la Tarjeta Demo Propeller de la compañía Parallax con el chip Propeller P8x32A-44 ya que además de ser rápido, bajo costo y tamaño reducido, proporciona flexibilidad y

potencia a través de sus ocho procesadores, llamados Cogs, que pueden realizar simultáneamente tareas independientes. No tiene limitaciones sobre edad y género de la persona, además tiene la versatilidad para ajustar la amplificación o atenuación en cualquier frecuencia, permitiendo una precisa adaptación según la deficiencia auditiva

RECONOCIMIENTO

Es necesario destacar la participación de jóvenes hipoacúsicos con un nivel de pérdida auditiva leve con carnet del CONADIS para las pruebas del prototipo de ecualizador de audio, así como también el interés de la ESCUELA POLITÉCNICA NACIONAL, por aportar para la investigación y creación de ayudas técnicas que faciliten la inclusión social de las personas con discapacidad.

REFERENCIAS

- [1] Martin, Jeff, Manual del Microcontrolador Propeller v1.1, Parallax inc., 2009.
- [2] Lindsay, Andy, Kit Educativo Propeller Fundamentos v1.1.1, parallax inc., 2009.
- [3] Gracey, Chip, Programming and Customizing the Mmulticore Microcontroller, McGraw-Hill, 2010.
- [4] Sandhu Singh, Harprit, Programming the Propeller with spin: A Beginner's Guide to Parallel Processing, McGraw-Hill, 2010.
- [5] Shoab Ahmed, Khan, Digital Design of Signal Processing Systems, Edition first, John Wiley & Sons, Ltd, England, 2011.
- [6] Halvorson, Michael, Visual Basic 2008 (paso a paso), Anaya Multimedia, Madrid, 2008.
- [7] Antoniu, Andreas, Digital Filters. Analysis, Design and Applications, 2^o edition, McGraw-Hill, New York, 1993.
- [8] <http://www.parallax.com/Portals/0/Downloads/docs/books/edu/M anualPropeller-v1.1-Espanol-v1.0.pdf>
- [9] <http://www.parallax.com/Portals/0/Downloads/sw/propeller/Setup -Propeller-Tool-v1.3.2.zip>
- [10] <http://www.parallax.com/Portals/0/Downloads/docs/prod/prop/PropellerDemoBd-RevG-Schem.pdf>

Con las mejoras que la tecnología y la investigación, puedan aportar en el prototipo portátil de ecualizador de audio, éste puede ser útil como ayuda técnica para la inclusión de niños y jóvenes con pérdida auditiva leve a la educación, a la capacitación, a la participación en conferencias, actividades culturales, de ocio, etc.

BIOGRAFÍAS

Pablo Cevallos



Nacido en Quito, Ecuador, en 1974. Egresado de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, carrera de Electrónica y Telecomunicaciones.

Este trabajo es su proyecto de titulación.

Mauricio Quitama



Nacido en Cotacachi, Ecuador, en 1978. Egresado de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, carrera de Electrónica y Control. Este trabajo es su proyecto de titulación.

Director: Robin Alvarez Rueda



Nacido en Cayambe, Ecuador, en 1969. Ingeniero en Telecomunicaciones, graduado de la Escuela Politécnica Nacional, Quito - Ecuador, 1995; M.Sc. en Telecomunicaciones por la Universidad de Cantabria, Santander - España, 2001; Ph.D en Telecomunicaciones por la Universidad Politécnica de Madrid-España, Enero-2006. Su interés desde hace seis años está en el campo de la bioingeniería.