

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE FILTROS CROSSOVER DE
TRES VIAS UTILIZANDO EL MODULO DE DESARROLLO DE BAJO
COSTO EZ-KIT LITE ADSP-2181**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

CONSTANTE CASTRO LUIS IVAN

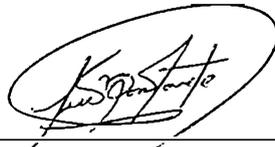
DIRECTOR: ING. RAMIRO MOREJÓN

Quito, mayo de 2003

DECLARACIÓN

Yo, Constante Castro Luis Iván, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal; y que, he consultado las referencias bibliográficas que se incluyen en este documento.

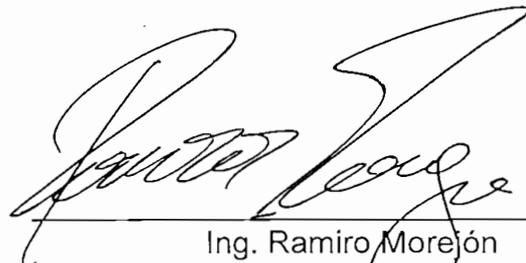
La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley, Reglamento de Propiedad Intelectual y por la normatividad institucional vigente.

A handwritten signature in black ink, enclosed within a hand-drawn oval. The signature is stylized and appears to read 'Constante Castro Luis Iván'.

Constante Castro Luis Iván

CERTIFICACIÓN

Certifico que el siguiente trabajo fue realizado por Constante Castro Luis Iván, bajo mi supervisión.



Ing. Ramiro Morejón
DIRECTOR DE PROYECTO

AGRADECIMIENTO

Un gran reconocimiento debo dar a los maestros de la Escuela Politécnica Nacional y de la carrera en Ingeniería Electrónica y Telecomunicaciones, porque de ellos aprendí: responsabilidad, honestidad y conocimiento

Mi sincera gratitud para los Ings. Ramiro Morejon y Fernando Vásquez que me brindaron un gran soporte moral, científico y técnico en la realización del presente trabajo.

Agradezco a mis compañeros y amigos que me brindaron su franco apoyo y sincera lealtad; en especial a: Wilson, Fernando, Gabriel y William.

DEDICATORIA

A mis padres; que son los artífices de mis aspiraciones y logros, por que de ellos es este pequeño avance en mi vida, lo único que les puedo decir en estos momentos es que DIOS LES BENDIGA.

Una mención especial se merece toda mi familia que estuvo conmigo todos los momentos alegres y tristes de mi vida.

Gracias, mil gracias.

CONTENIDO

DECLARACIÓN	ii
CERTIFICACIÓN.....	iii
AGRADECIMIENTOS	iv
DEDICATORIA	v
CONTENIDO.....	vi
LISTA DE FIGURAS	ix
LISTA DE ABREVIATURAS	xi
RESUMEN	xiv
PRESENTACIÓN.....	xv
CAPÍTULO 1. CONCEPTOS GENERALES	1
1.1. EL SONIDO	2
1.1.1. EL OÍDO HUMANO	2
1.1.2. ALTAVOCES	3
1.1.2.1. Woofer.....	5
1.1.2.2. Squawker.....	6
1.1.2.3. Tweeter.....	6
1.1.3. AUDIO DIGITAL.....	7
1.1.3.1. Parámetros del muestreo	8
1.1.3.2. Espacio necesario para almacenar audio digital	9
1.2. FILTROS	10
1.2.1. FILTROS CROSSOVER.....	11
1.2.1.1. Filtros pasivos.....	13
1.2.1.2. Filtros activos	14
1.2.1.3. Ventajas del filtraje activo.....	14
1.2.2. FILTROS ANALÓGICOS	15
1.2.3. FILTROS DIGITALES	17
1.2.3.1. Ventajas de los filtros digitales sobre los analógicos.....	18

1.3. PROCESADOR DIGITAL DE SEÑALES (DSP)	19
1.3.1. OPERACIONES BÁSICAS	19
1.3.2. PARÁMETROS DE SELECCIÓN DEL DSP ADECUADO	20
1.3.2.1. Formato aritmético.....	22
1.3.2.2. Anchura de datos	23
1.3.2.3. Velocidad.....	23
1.3.2.4. Multiprocesamiento	24
1.3.2.5. Potencia	24
1.3.2.6. Costo	25
CAPÍTULO 2. DISEÑO DE FILTROS CROSSOVER	26
2.1. ESPECIFICACIÓN DE FILTROS DIGITALES	27
2.2. DISEÑO DE FILTROS FIR CON VENTANA DE KAISER	29
2.2.1. SISTEMA DE ALTAVOCES.....	32
2.2.1.1. Retraso relativo de los altavoces	33
2.2.2. PARAMETROS DE DISEÑO.....	35
2.2.2.1. Diseño FIR de woofer	36
2.2.2.2. Diseño FIR de Squawker	39
2.2.2.3. Diseño FIR de Tweeter	41
2.2.2.4. Respuesta combinada FIR.....	43
CAPÍTULO 3. IMPLEMENTACIÓN DE LOS FILTROS CROSSOVER DE TRES VIAS	44
3.1. CARACTERÍSTICAS DE LA RED CROSSOVER IMPLEMENTADA	44
3.2. DESCRIPCIÓN DEL SISTEMA CROSSOVER	45
3.2.1. MÓDULO DE DESARROLLO EZ-KIT LITE.....	46
3.2.1.1. Conversión analógica – digital (A/D) y digital – analógica (D/A).....	47
3.2.1.2. Procesador digital de señales	49
3.2.1.3. Funcionamiento de un filtro de respuesta impulsiva finita	51
3.2.1.4. Comunicación serial entre la PC y la ADSP-2181	54
3.2.2. AMPLIFICADORES Y SISTEMA DE ALTAVOCES	58

3.2.3. MATLAB.....	61
3.3. PROGRAMAS DESARROLLADOS	62
CAPÍTULO 4. RESULTADOS Y APLICACIONES.....	66
4.1. SISTEMA CROSSOVER DIGITAL.....	66
4.2. RESULTADOS EXPERIMENTALES.....	72
4.2.1. FILTRO PASABAJOS.....	73
4.2.2. FILTRO PASABANDA.....	75
4.2.3. FILTRO PASAALTOS	78
4.2.4. ANALISIS DE RESULTADOS	81
4.3. APLICACIONES.....	82
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES	83
5.1. CONCLUSIONES.....	83
5.2. RECOMENDACIONES.....	85
BIBLIOGRAFÍA.....	87
ANEXOS	89
ANEXO 1.....CARACTERISTICAS DE ALTAVOCES COMERCIALES	
ANEXO 2.....CROSSOVER COMERCIALES	
ANEXO 3.....EZ-KIT LITE ADSP-2181	
ANEXO 4.....SOFTWARE UART	
ANEXO 5.....AMPLIFICADOR TA8216H	
ANEXO 6.....HERRAMIENTAS DE MATLAB	

LISTA DE FIGURAS

Figura 1.1.	Respuesta en frecuencia de un altavoz.....	4
Figura 1.2.	El altavoz de graves.....	5
Figura 1.3.	Tweeter.....	6
Figura 1.4.	Conversión del audio.....	7
Figura 1.5.	Muestreo y cuantificación de una señal.....	9
Figura 1.6.	(a) Frecuencia de corte y ancho de banda; (b) Frecuencia de cruce.....	10
Figura 1.7.	Frecuencias reproducidas por dos altavoces.....	11
Figura 1.8.	Respuesta en frecuencia de un filtro crossover de tres vías.....	12
Figura 1.9.	Filtro crossover analógico de 3 vías.....	16
Figura 2.1.	Especificaciones de un filtro.....	28
Figura 2.2.	Sistema de altavoces y respuestas de las vías.....	32
Figura 2.3.	Posición de los altavoces.....	33
Figura 2.4.	Sistema de altavoces coaxial.....	34
Figura 2.5.	Respuesta impulsiva teórica sin cuantificar.....	37
Figura 2.6.	Respuesta de frecuencia con escala logarítmica del módulo.....	38
Figura 2.7.	Respuesta de frecuencia con escala lineal del módulo.....	38
Figura 2.8.	Respuestas de frecuencia filtro pasa banda.....	40
Figura 2.9.	Respuesta de frecuencia filtro pasa alto.....	42
Figura 2.10.	Respuesta de frecuencia del sistema de filtros.....	43
Figura 3.1.	Transferencia de coeficientes entre el PC y los DSP's.....	45
Figura 3.2.	Diagrama de bloques de la red crossover de tres vías.....	46
Figura 3.3.	Módulo de desarrollo del ADSP-2181.....	47
Figura 3.4.	Diagrama de bloques del AD1847.....	48
Figura 3.5.	Arquitectura interna del ADSP-2181.....	50
Figura 3.6.	Estructura de un filtro FIR.....	52
Figura 3.7.	Producto vectorial entre la línea de retardo y los coeficientes del filtro.....	54
Figura 3.8.	Configuración general del sistema de transmisión.....	55
Figura 3.9.	Implementación del hardware para la comunicación serial asincrónica.....	56
Figura 3.10.	Datos recibidos.....	57
Figura 3.11.	Diagrama de bloques del TA8216H.....	59

Figura 3.12.	Altavoces de tres vías del F-886	60
Figura 3.13.	Ventana de comando	61
Figura 3.14.	Diagrama de flujo de los programas implementados	62
Figura 4.1.	Instrumentos de medida y generación	66
Figura 4.2.	Diagrama de bloques del crossover digital	67
Figura 4.3.	Etapas de filtraje con tres ADSP-2181	68
Figura 4.4.	Etapas de amplificación y reproducción	68
Figura 4.5.	Sistema crossover digital.....	69
Figura 4.6.	Pantalla de presentación	70
Figura 4.7.	Ingreso de parámetros del diseño de los filtros	70
Figura 4.8.	Señal de error al ingresar parámetros fuera de rango.....	71
Figura 4.9.	Filtros crossover con ganancias diferentes.....	72
Figura 4.10.	Medición de las respuestas del sistema crossover.....	73
Figura 4.11.	Respuesta ideal y experimental del filtro pasa bajos	74
Figura 4.12.	Respuestas de los filtros pasa bajos analógico y digital	75
Figura 4.13.	Respuesta ideal y experimental del filtro pasabanda.....	77
Figura 4.14.	Respuestas de los filtros analógico y digital pasa banda.....	78
Figura 4.15.	Respuesta ideal y experimental del filtro pasa altos	79
Figura 4.16.	Respuestas de los filtros pasa altos analógico y digital.....	80
Figura 4.17.	Respuesta experimental del sistema.....	80
Figura 4.18.	Distribución de altavoces en una sala de cine.....	82
Figura 4.19.	Altavoces en un automóvil	82

LISTA DE ABREVIATURAS

A	Amplitud
ADC	Conversor analógico - digital (<i>Analog to Digital Converter</i>)
ADSP-2181	Procesador digital de señales fabricado por Analog Devices
ALU	Unidad aritmética lógica (<i>Aritmetic logia Unit</i>)
A/D	Etapas de conversión analógica a digital
CD	Disco compacto (<i>Compact Disc</i>)
CODEC	Codificador-decodificador
DAC	Conversor digital - analógico (<i>Digital to Analog Converter</i>)
dB	Decibelios
DMA	Acceso directo a memoria (<i>Data Memory Access</i>)
DSP	Procesador digital de señales (<i>Digital Signal Processor</i>)
D/A	Etapas de conversión digital a analógica
EPROM	Memoria solo de lectura programable y borrable (<i>Eraser Program Read Only Memory</i>)
E/S	Entrada / Salida
f	Frecuencia
fc	Frecuencia de corte en Hertz
FI	Bandera de entrada (<i>Flag In</i>)
FIR	Respuesta impulsiva finita (<i>Finite Impulse Response</i>)
FL1	Bandera 1 (<i>Flag 1</i>)
fm	Frecuencia de muestreo
FO	Bandera de salida (<i>Flag Out</i>)
fp	Frecuencia de paso en Hertz
fs	Frecuencia de supresión en Hertz
Gs	Ganancia del squawker
Gt	Ganancia del tweeter
Gw	Ganancia del woofer

h[n]	Respuesta impulsiva del filtro
Hz	Hertz
IIR	Respuesta impulsiva infinita (<i>Infinite Impulse Response</i>)
$I_o[]$	Función de Bessel modificada de orden cero y de primer tipo
IRQ2	Interrupción sensitiva a nivel
M	Orden de un filtro
MAC	Multiplicador – acumulador (<i>Multiplier/Accumulator</i>)
MB	Mega bytes
MIPS	Millones de instrucciones por segundo
PC	Computador personal (<i>Personal Computer</i>)
PGA	Encapsulado de tipo <i>Pin Grid Array</i>
PMPO	Salida de poder de picos musicales (<i>Peak Music Power Output</i>)
PQFP	Encapsulado de tipo <i>Plastic Quad Flat Pack</i>
RAM	Memoria de acceso aleatorio (<i>Random Access Memory</i>)
Rx	Recepción de datos
SPORT1	Puerto serial 1 (<i>Serial Port 1</i>)
SQUAWKER	Altavoz reproductor de frecuencias medias
S/N	Relación señal a ruido (<i>Signal/Noise</i>)
T	Período
TQFP	Encapsulado de tipo <i>Thin Quad Flat Pack</i>
TTL	Lógica transistor – transistor (<i>Transistor Transistor Logic</i>)
TWEETER	Altavoz reproductor de agudos
Tx	Transmisión de datos
UART	Recepto/transmisor asincrónico universal (<i>Universal Asynchronous Receiver/Transmitter</i>)
V	Voltios
Wc	Frecuencia de corte normalizada
WOOFER	Altavoz reproductor de graves
Wp	Frecuencia de paso

ω_s	Frecuencia de supresión
$w[n]$	Ventana de Kaiser
α, β	Parámetros de la ventana de Kaiser
δ_1, δ_2	Rizado en la banda de paso y supresión
Δf	Ancho de la banda de transición en Hertz
$\Delta \omega$	Ancho de la banda de transición normalizado
$ We^{j\omega} $	Ganancia ideal del filtro

RESUMEN

El audio digital ha tenido avances vertiginosos, hasta el punto que la reproducción de audio digital es superior a la analógica. Algunas de las ventajas de la utilización de técnicas digitales en proceso de señales son: repetibilidad, elevada estabilidad térmica, reprogramabilidad y adaptabilidad.

Una reproducción acústica de óptima calidad requiere de: filtros, múltiples amplificadores y un buen arreglo de altavoces.

La etapa de filtrado es fundamental en la reproducción de audio, debido a que los altavoces reproducen solo ciertos rangos de frecuencia. El sistema de filtros debe tener fase lineal plana, adecuado rango de frecuencias de corte para cada filtro y una buena atenuación para las bandas de supresión.

Para superar problemas en la inversión de fase se utilizan filtros digitales FIR (Finite Impulsive Response, Respuesta Impulsiva Finita) que tienen fase lineal y como método de diseño de los filtros se utiliza el de enventanado.

La ventana de Kaiser permite tener parámetros ajustables y buena atenuación en la banda de supresión mediante el uso de fórmulas preestablecidas.

El sistema digital implementado permite variar los parámetros de sus filtros desde un PC mediante un programa realizado en Matlab. Con este arreglo de filtros se puede corregir defectos acústicos en salas y/o establecer preferencias del usuario.

PRESENTACIÓN

Los DSP's (Digital Signal Processor, Procesador Digital de Señales) son microprocesadores diseñados para procesamiento digital de señales (manipulación matemática de señales representadas digitalmente). El procesamiento digital de señales es una tecnología cuyas aplicaciones están creciendo rápidamente, como en el caso de comunicaciones inalámbricas, procesamiento de audio, vídeo y control industrial.

El módulo de procesamiento digital ADSP-2181, se basa en un microprocesador de punto fijo de 16 bits que puede realizar 33 millones de instrucciones por segundo. Su arquitectura básica comprende: tres unidades computacionales, generador de direcciones de datos y secuenciador de programa. Para la entrada y salida de señales utiliza dos canales (izquierdo y derecho).

La gama de frecuencias audible (20Hz a 20KHz) es dividida en tres secciones: Woofer, Squawker y Tweeter; éste seccionamiento permite que los altavoces reproduzcan solo las señales para las cuales están diseñados. Cada región es controlada por un filtro FIR implementado sobre una tarjeta DSP.

En el PC se ingresan los parámetros y se visualizan las respuestas de los filtros, el programa además permite informarse del funcionamiento del sistema recuperando los últimos datos transmitidos. En el computador se calculan todos los coeficientes y se transmiten en forma serial asincrónica los datos a los DSP's.

El principal limitante en el EZ-KIT Lite ADSP-2181 para esta aplicación es que posee sólo una salida estereo y se necesitan tres, otra de las restricciones es la cantidad de instrucciones que puede manejar por muestra. Los nuevos desarrollos de la firma Analog Devices solucionan estos inconvenientes con codecs que tienen 2 conversores A/D y seis D/A como el AD1836 y microprocesadores de 100 millones de instrucciones por segundo como el ADSP-21161.

CAPÍTULO I

CONCEPTOS GENERALES

La acústica se encarga del estudio de la generación, transmisión y recepción de energía que se encuentra en forma de perturbaciones u ondas vibratoriales en un medio material de transmisión.

Los avances tecnológicos que tiene el audio digital se producen de una manera vertiginosa llegándose al punto en que la calidad de reproducción de un sistema digital de audio es superior a la del analógico, citándose como ventajas principales las siguientes:

- Permiten ser almacenados en forma inalterable dado que lo que se almacenan son números o símbolos binarios y es mucho más difícil alterar la información guardada contrario al caso en que se guarda un campo magnético proporcional a la señal, como en un cassette.
- La técnica digital tiene bajo costo ya que los chips pueden ser multifuncionales; a diferencia de los circuitos analógicos que están formados por una serie de diferentes componentes tanto en forma como tamaño, por lo que su montaje y ajuste resulta costoso.
- Se puede aprovechar la tecnología de procesamiento digital de señales para introducir efectos, modificaciones o mejoras imposibles o muy difíciles de lograr analógicamente. Por ejemplo, es posible conseguir retardos, filtrado, efectos de reverberación, supresión de ruido, etc.
- La calidad de reproducción de un sistema de audio digital es independiente de los medios de almacenamiento y transporte de la señal, pero depende de los procesos de conversión tanto de sonido analógico a digital y viceversa.

En los sistemas de altavoces digitales que utilizan como entrada señales de audio digital se utilizan filtros digitales FIR (Finite Impulse Response, Respuesta Impulsiva Finita) para seccionar en bandas de frecuencia la señal original, a continuación se convierte dichas bandas a formato analógico para finalmente amplificarlas y dirigir las a los altoparlantes adecuados.

1.1. EL SONIDO

Se considera como sonido a las sensaciones producidas por el movimiento vibratorio de los cuerpos en el oído. Al vibrar los cuerpos producen variaciones de presión en el medio dentro del cual se encuentran, por lo que se requiere que el medio sea elástico para que se propaguen las ondas sonoras y puedan llegar al oído.

El sonido posee como características: la intensidad o amplitud que depende de la fuente que produce el sonido; y el timbre que está determinado por la cantidad de armónicos que tenga la frecuencia del sonido. Producto de estas características es que un sonido difiera de otro aún teniendo la misma frecuencia.

1.1.1. EL OÍDO HUMANO

Las vibraciones producidas por un cuerpo dan como resultado en un determinado punto variaciones de presión con una frecuencia igual a la de oscilación del objeto. Si suponemos que este punto es el oído humano las variaciones de presión se producirán en la membrana timpánica, vibrando ésta por lo tanto a la misma frecuencia que el objeto generador del sonido, las vibraciones de la membrana se propagarán a través del oído medio hasta llegar al órgano de Corti, órgano en el cual se convierten las vibraciones en estímulos nerviosos que cuando llegan al cerebro dan la sensación auditiva. La percepción del sonido difiere en cada individuo, factores como la edad influyen notablemente en la sensibilidad del oído, comprobándose que al aumentar la edad se pierde la percepción de los sonidos de alta frecuencia (agudos). En general se considera que las frecuencias de vibración que puede reproducir el oído están entre 16 ó 20

Hertz hasta los 20000 Hertz; se debe tener presente que la respuesta que tiene el oído a las variaciones de presión sonora no es lineal, sino más bien logarítmica.

El oído humano posee una gran agudeza, ya que puede detectar ínfimas cantidades de distorsión en el sonido. Por lo que se puede considerar como sonido perfecto al que el oído es incapaz de detectar deformaciones. Teniendo en cuenta este criterio, la calidad es puramente subjetiva y puede comprobarse solo con pruebas de audición; de pruebas subjetivas realizadas se obtienen los parámetros de sensibilidad que deben poseer los instrumentos de medición de la señal audible.

1.1.2. ALTAVOCES

Los altavoces son los encargados de producir la señal audible, para lo cual transforman la energía eléctrica recibida en energía acústica, como paso intermedio está la transformación a energía mecánica. De acuerdo a estas propiedades se puede dividir a un altavoz en las siguientes partes constituyentes:

- Sección *electromagnética*: formada por el imán y la bobina móvil. Aquí llega la señal eléctrica a la bobina móvil que se encuentra dentro del campo magnético del imán produciéndose el movimiento de la bobina.
- Sección *mecánica*: constituida por el cono y su suspensión. Sobre el cono se encuentra montada la bobina móvil, la cual al desplazarse hace que el cono vibre.
- Sección *acústica*: es la encargada de transmitir la energía sonora producida por el cono.

La respuesta de frecuencia que tiene un altavoz (Fig. 1.1.), está definida como el rango de frecuencias que puede reproducir éste; y es la relación existente entre la presión sonora y la frecuencia de la señal; es una de las características

más importantes de los altavoces, ya que gracias a ella podemos determinar a qué rango de frecuencia se reproduce mejor el sonido en un altavoz.

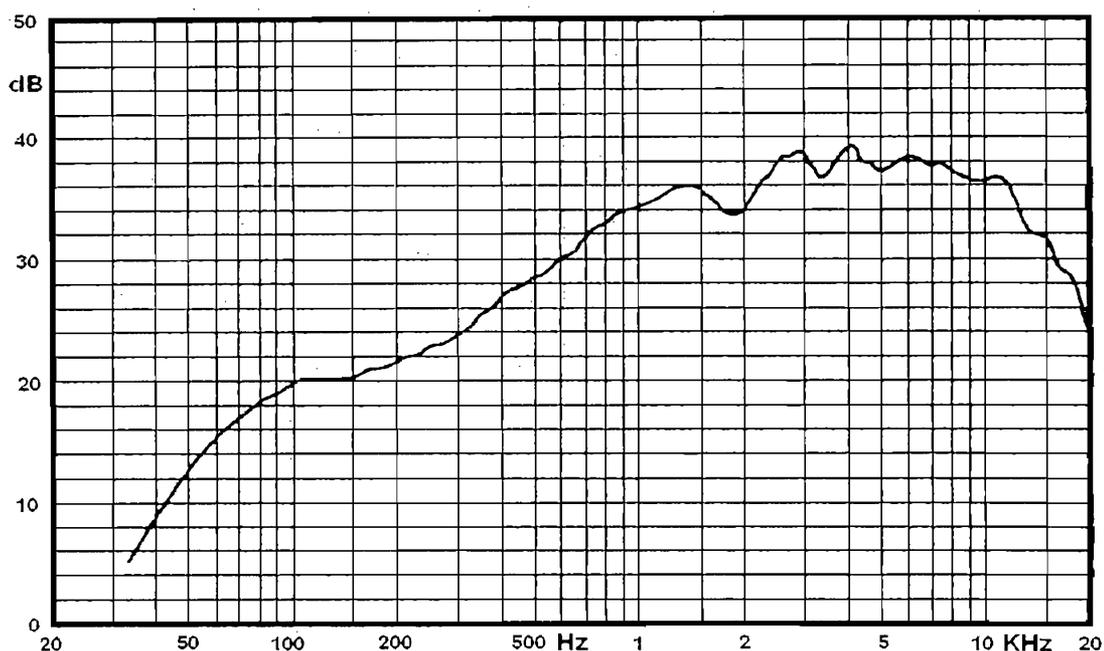


Figura 1.1. Respuesta en frecuencia de un altavoz [12]

Un valor importante a tener en cuenta es la impedancia que se tiene a la entrada de un altavoz y que guarda relación con la frecuencia a la cual está expuesto; la curva de impedancia la proporcionan los fabricantes y en ésta se presentan generalmente un pico el cual corresponde a la frecuencia de resonancia del altavoz seguido de una porción llana de la curva, que es la que determina la impedancia nominal del altavoz. Es muy importante que la curva de impedancia no descienda a valores muy bajos, ya que podría causar problemas al amplificador, sobre todo si se utilizan más de un sistema de altavoces al mismo tiempo.

Para una máxima calidad de reproducción sonora se debe recurrir al empleo de varios y diversos altavoces, ya que las características constructivas de los mismos los hacen adecuados para una gama de frecuencias e inadecuados para otras. En base a estas características tendremos altavoces grandes para frecuencias bajas o graves (*woofer*), altavoces más pequeños para frecuencias

medias (*squawker*) y otros aún más pequeños para radiar las frecuencias más altas o agudas (*tweeter*).

1.1.2.1. Woofers

Los altavoces de tipo woofer se emplean para reproducir tonos graves debido a que tienen una *frecuencia de resonancia*¹ muy baja, esta frecuencia disminuye al aumentar los diafragmas del altavoz; por lo que estos serán los que tengan mayor dimensión, suelen tener alrededor 12 pulgadas.

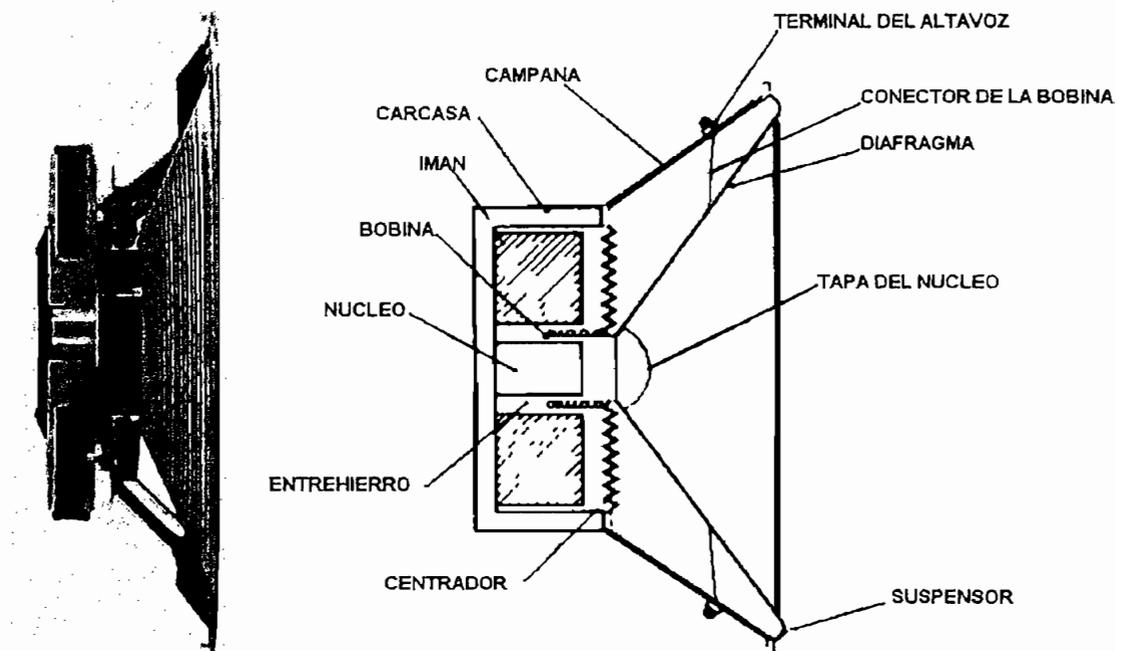


Figura 1.2. El altavoz de graves [4]

Al aplicar frecuencias bajas (menores a 800 Hz.) todo el diafragma se desplaza, proporcionando un rendimiento excelente para dichas notas. En la figura 1.2 se puede apreciar la estructura del altavoz para tonos graves.

¹ La frecuencia de resonancia de un altavoz es la frecuencia material de vibración del diafragma y de la bobina móvil; depende esencialmente de las características constructivas del altavoz, tales como el sistema mecánico de montaje, masa del cono, carga acústica, etc.

1.1.2.2. Squawker

Estos tipos de altavoces poseen una respuesta de frecuencia entre una frecuencia de resonancia no superior a los 200 Hz y una *frecuencia de corte*¹ de entre 6 y 8 KHz. Puede emplearse cualquier altavoz de alta fidelidad cuyo diámetro este comprendido entre 5 y 10 pulgadas.

1.1.2.3. Tweeter

Este tipo de altavoces se encargan de la reproducción de frecuencias altas, hasta 20 KHz, empiezan a responder a su frecuencia de resonancia que está cercana a los 2 KHz. Los altavoces para agudos necesitan estar provistos de trompetas como se aprecia en la figura 1.3.; la trompeta sirve para adaptar su *impedancia acústica*² con la del aire. Tienen el menor diámetro exterior de los altavoces.

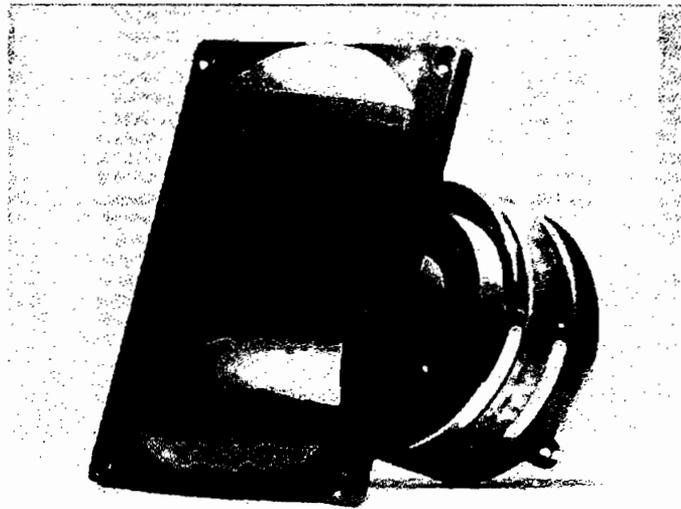


Figura 1.3. Tweeter [4]

¹ La frecuencia de corte es aquella para la que la intensidad sonora proporcionada por el altavoz es aproximadamente de 3 a 5 dB inferior a una recta imaginaria, sustitutiva a la curva de respuesta para el centro de la banda.

² La impedancia acústica de un fluido que actúa sobre una superficie es el cociente de la presión acústica en la superficie dividida por la velocidad en la superficie.

1.1.3. AUDIO DIGITAL

Por audio digital se entiende a la digitalización de sonido real, ya sea procedente de voces, instrumentos musicales acústicos o electrónicos, grabaciones, etc, para ser tratados en los microprocesadores. Dicho de otra forma es la representación de una señal de audible mediante números, en general codificados en forma binaria.

Para convertir una señal eléctrica a una señal digital se utiliza los denominados ADC, o Conversores de Analógico a Digital, circuitos que utilizan un proceso de *muestreo* (discretización en el tiempo) y *digitalización* (discretización en amplitud). El muestreo consiste en tomar valores de la señal a intervalos regulares de tiempo. La digitalización consiste en subdividir el rango útil total de la señal en cierta cantidad de "casilleros" o subintervalos numerados, y asignar a cada muestra el número de subintervalo en el cual se encuentra una determinada frecuencia (figura 1.4).

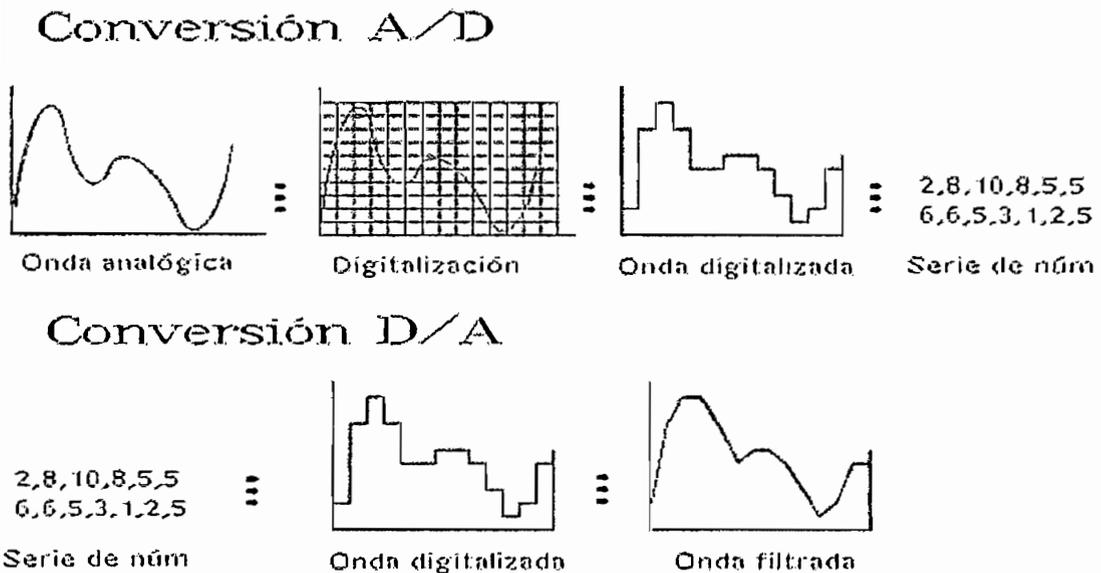


Figura 1.4. Conversión del audio

Pero al hablar de audio digital no hay que olvidar que no sirve de nada tener el sonido digitalizado si no podemos escucharlo. Para ello, necesitamos

hacer el proceso inverso al del muestreo y digitalización: la conversión de digital a analógica, encargada a los circuitos DAC o conversores digital a análogo. Además de convertir los números almacenados en el procesador a una señal eléctrica se debe filtrar ésta para obtener una señal válida. En la calidad de dichos filtros reside, en muchas ocasiones, la calidad de sonido de una tarjeta de muestreo, obteniendo en algunas un nivel de ruido de fondo que las hace inútiles para usuarios exigentes.

Como es de suponer, entre los procesos de digitalización y escucha, tenemos acceso a una variada gama de manipulaciones del sonido, que nos permiten obtener resultados imposibles, o, al menos, muy difíciles de realizar por otros métodos, sin necesidad de usar un caro equipo especializado.

1.1.3.1. Parámetros del muestreo

Para medir la calidad del muestreo, debemos referirnos a dos parámetros: la frecuencia de muestreo y la resolución o cuantización:

La frecuencia de muestreo se refiere al número de mediciones que se realizan por segundo. Cuanto mayor sea esta frecuencia, más parecido será el resultado obtenido al sonido original. Según el teorema de Nyquist, la frecuencia mínima de muestreo debe ser el doble de la frecuencia máxima de la señal original. Se debe acotar que no es suficiente que sea mayor que el doble de la máxima frecuencia útil, ya que si hay ruido por encima de ésta, podría producirse un tipo de distorsión denominado aliasing. Supongamos, por ejemplo, que queremos muestrear una señal audible que contiene además un ruido de 35 KHz. Si utilizamos una frecuencia normalizada de muestreo de 44,1 KHz, a pesar de que ese ruido es originalmente inaudible (por ser mayor que el límite superior de 20 KHz del oído humano), al intentar recuperar la señal aparecerá un ruido de 9,1 KHz ($= 44,1 \text{ KHz} - 35 \text{ KHz}$), que es perfectamente audible. Este tipo de frecuencias que aparecen dentro del espectro útil se denominan frecuencias "alias".

La resolución es la cantidad de bits que se utiliza para representar las muestras de audio, es decir la cantidad de bits que conforman cada palabra. Cuanto mayor sea la resolución, más precisa será la representación. Se mide en bits: si la resolución es de 8 bits tenemos 256 niveles posibles. Si ampliamos a 16 bits, cada medida puede estar en un rango de 0 a 65.535. Como se ve, la precisión en este último caso es mucho mayor. En la figura 1.5 se aprecian los parámetros del muestreo.

La mejor manera de evaluar el ruido de cualquier sistema (incluidos los de audio digital) es a través de la relación señal / ruido (S/N) en decibeles. Para el audio digital, la máxima S/N que puede obtenerse es, aproximadamente, igual a $6 \cdot n$, donde n es la resolución en bits.

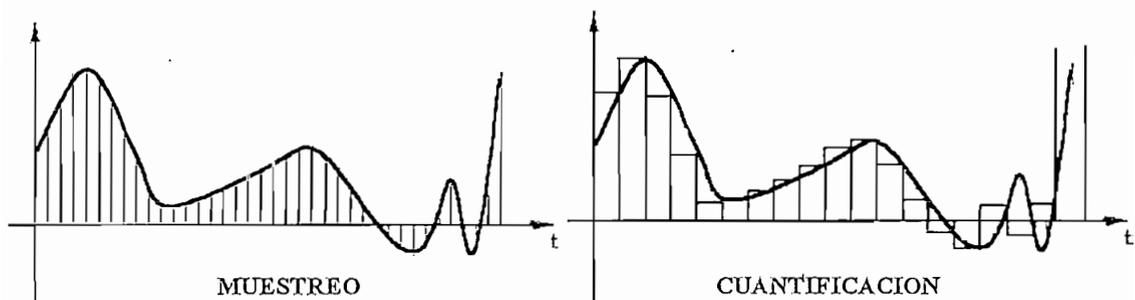


Figura 1.5. Muestreo y cuantificación de una señal

1.1.3.2. Espacio necesario para almacenar audio digital

La cantidad de memoria requerida depende de la calidad de la grabación. Si se quiere calidad CD, son 5 MB por pista y por minuto ($44.100 \text{ muestras/seg} \times 2 \text{ bytes} \times 60 \text{ seg.} = 5.292.000 \text{ bytes}$, es decir, 5,04 MB), lo que suma unos 40 Mb para una canción de 4 minutos. De lo expuesto se recomienda usar frecuencias de muestreo más bajas y resolución de 8 bits para aquellos trabajos que no requieran tanta calidad (juegos, enciclopedias, etc...)

Existen sistemas de compresión que almacenan y leen en tiempo real los ficheros de audio, con la ayuda de chips DSP (procesadores de señales digitales),

consiguiendo ratios de 1:4 o superiores. Si deseamos utilizar un sistema de grabación multipista, debemos multiplicar esos 5 MB por los minutos y por las pistas que vayamos a usar.

1.2. FILTROS

Un filtro es un sistema que permite el paso de las componentes de la señal existentes en un determinado intervalo frecuencial (banda de paso), y no deja pasar al resto (banda atenuada o de rechazo).

El ancho de banda de un filtro comprende todas aquellas frecuencias capaces de atravesar el circuito, la amplitud de la señal obtenida a la salida no debe ser menor al 70% del valor de la señal aplicada; este punto es el de media potencia ó -3 dB. Las frecuencias que se encuentran en los puntos de media potencia se las denomina frecuencia de corte.

La frecuencia de cruce entre dos filtros requiere que la frecuencia superior de corte del filtro pasabajos coincida con la frecuencia inferior de corte del filtro pasaaltos. La frecuencia de corte y de cruce se las puede apreciar en la figura 1.6.

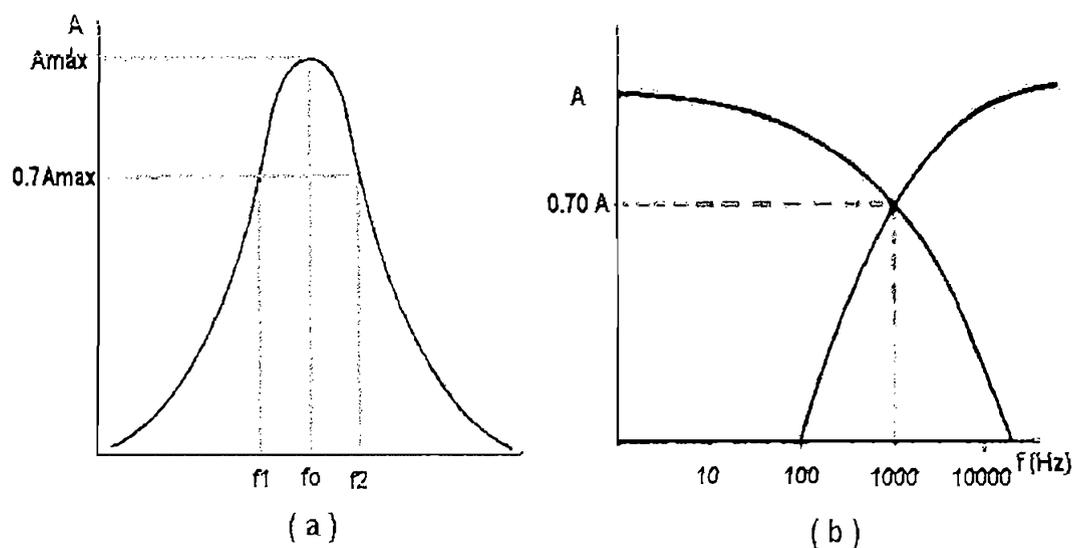


Figura 1.6. a) Frecuencias de corte y ancho de banda ; b) Frecuencia de cruce

1.2.1. FILTROS CROSSOVER

Para implementar un sistema de altavoces hay que tener en cuenta un par de principios. El primero es romper el cortocircuito acústico que se produciría si las moléculas de aire desplazadas por la parte anterior del cono se desplazaran hacia la depresión creada en la parte posterior, que sería más grande a más bajas frecuencias reproducidas; para eliminar ese cortocircuito acústico, teóricamente necesitaríamos un panel de dimensiones infinitas, aunque en la realidad no es necesario un panel de exageradas dimensiones, y se suele encerrar el altavoz en cajas.

El segundo se debe a las limitaciones de reproducción de frecuencias de los altavoces tratadas anteriormente. Al incluir en el mismo recinto acústico o para el mismo sistema de altavoces, un altavoz destinado por ejemplo a frecuencias bajas y otro a frecuencias bajas / medias, el espectro de frecuencias en el cual actúan los dos altavoces se verá favorecido, y esa zona del espectro se escuchará con un mayor nivel.

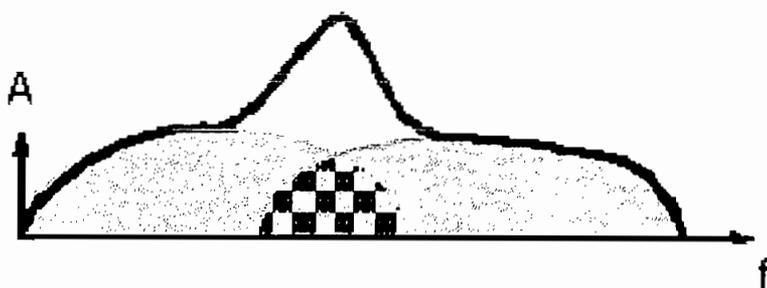


Figura 1.7. Frecuencias reproducidas por dos altavoces

En la figura 1.7. las dos curvas representan las respuestas de frecuencia de los dos altavoces, y la zona a cuadros es la zona del espectro en la cual responden los dos altavoces, y que se ve involuntariamente desfavorecida, lo cual desvirtúa la audición. Para modificar la curva de respuesta del sistema, y evitar en la medida de lo posible el fenómeno que acabamos de mostrar es para lo cual se utilizan los filtros divisores de frecuencia o crossover.

Los filtros divisores de frecuencia tienen por finalidad la de separar las frecuencias de audio, para que puedan alimentar a un altavoz adecuado a la banda. La determinación de las frecuencias de corte en los filtros crossover es un compromiso entre el margen de frecuencia que reproduzca el altavoz y la capacidad de potencia del mismo; deben evitarse frecuencias de corte inferiores a las que recomienda el fabricante ya que entonces la potencia que ha de soportar el altavoz es superior. Cuando tenemos los altavoces dispuestos en cajas acústicas, a cada uno de ellos se le llama vía o canal.

Los filtros crossover de tres vías tienen dos frecuencias de corte como se indica en la figura 1.8.; la primera frecuencia de corte es la que corresponde al del filtro de graves con el de medios y la segunda frecuencia de corte es aquella entre el canal de medios y agudos. Los tres filtros deben ser complementarios, de forma que la respuesta total del conjunto sea plana en toda la gama de audio; es decir, que la amplitud de la señal a la salida del conjunto se mantenga siempre por encima del 70% de la amplitud máxima o lo que es igual o superior a los -3dB .

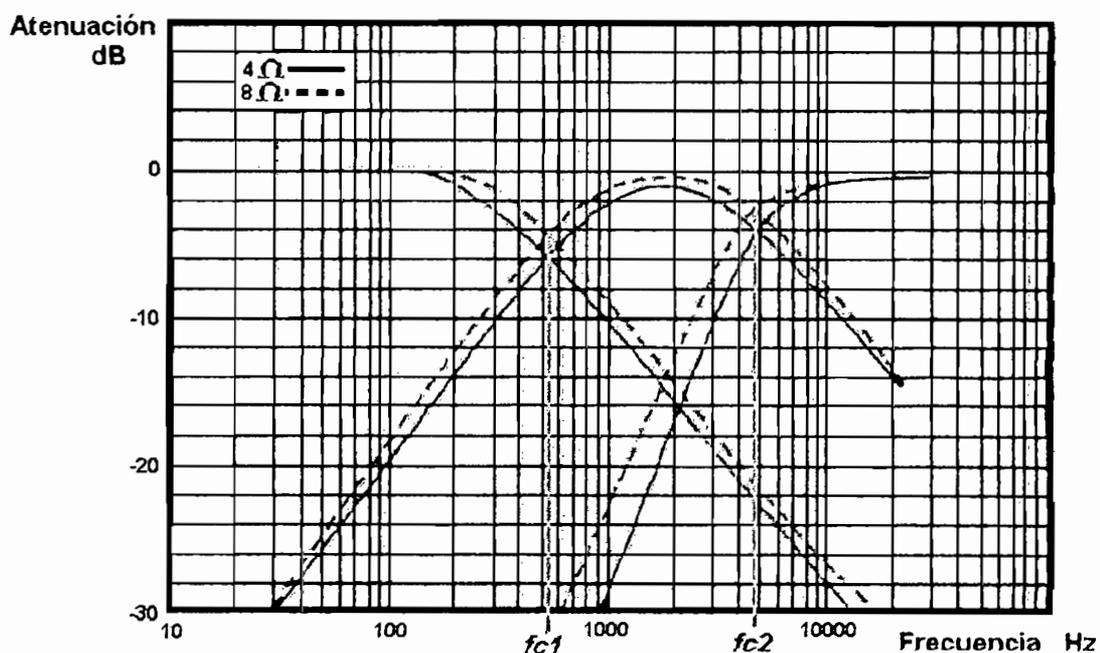


Figura 1.8. Respuesta en frecuencia de un filtro crossover de tres vías. [12]

Los crossover se construyen generalmente de manera específica para una configuración determinada de altavoces, por lo que no suele ser seleccionable el valor de corte de los divisores, ya que es propio de los altavoces. Deben calcularse con bastante cuidado las razones de las pendientes y respuestas de frecuencia de los filtros separadores a fin de garantizar que los altavoces no sean sobreexcitados.

Existen dos clases de filtros que son los pasivos y los activos. Los filtros pasivos tienen como característica principal que estos actúan sobre la señal procedente del amplificador. Los filtros activos por su parte actúan sobre la señal que proviene de la fuente, por lo que la señal que se amplifica tiene un determinado rango de frecuencias para ser enviada al altavoz que le corresponda.

1.2.1.1. Filtros pasivos

Es el que se usa habitualmente en las cajas acústicas convencionales en el audio doméstico. Consiste en un filtro divisor de frecuencias, a base de componentes pasivos (resistencias, condensadores, bobinas e incluso transformadores, etc), este tipo de filtro recibe la señal ya amplificada por la etapa de potencia y la envía de forma seleccionada por frecuencias a cada uno de los transductores, de manera que los altavoces reciban la señal en función de la respuesta en frecuencia que se les ha asignado, y de modo también, que la respuesta conjunta de todos ellos presente una correcta linealidad tanto en coherencia tímbrica como en nivel. Estos filtros no son manipulables por el usuario.

Algunos fabricantes de cajas acústicas, sin embargo, conscientes de que la captación musical es siempre y absolutamente subjetiva, ya previeron hace unos 40 años y siguen en la actualidad instalando potenciómetros de "ajuste fino" en la zona de las altas y medias frecuencias, manteniendo intocable la zona de los graves.

1.2.1.2. Filtros activos

Es usado habitualmente en el audio profesional, consiste en tratar la señal antes de ser amplificada por la etapa de potencia, de forma que una vez recibida de una fuente o del pre-amplificador pasa a un sistema de filtraje electrónico o digital (con alimentación externa, por tanto) que permitirá realizar una serie de funciones y al final del recorrido cada una de las "fracciones" de la señal original será entregada a cada una de las etapas de potencia y de ahí a su correspondiente transductor sin intermediación de ningún componente pasivo. Es decir, que si se ha previsto un sistema de tres vías, será necesario aprovisionarse del filtro digital, más tres etapas de potencia estéreo (o seis mono) y de las cajas en que se anclarán los seis altavoces, (tres por canal).

1.2.1.3. Ventajas del filtraje activo

✚ Al usar una etapa de potencia de forma individual para cada transductor se reduce muchísimo la complejidad de la carga asociada al amplificador cuando éste debe atacar varias vías.

✚ Por el mismo hecho se reduce notablemente la distorsión por intermodulación, quedando suprimida la que se genera en la interfaz amplificador / filtro pasivo.

✚ El filtraje activo permite diseñar "a medida" el escalonamiento de los diversos amplificadores en función de sus respectivos altavoces, lo que permite alcanzar niveles de presión sonora más elevados, sin distorsión.

✚ Por el hecho de prescindir de los componentes pasivos hay un notable aumento en la capacidad dinámica del sistema, o, dicho de otra forma, los detalles salen a relucir con mucha más facilidad.

✚ En los sistemas de filtraje pasivo un amplificador puede llegar a su punto de saturación produciendo una distorsión de banda pasante muy amplia,

afectando la sobrecarga no sólo a las zonas medias y graves, sino recortando brutalmente los armónicos de orden superior que debería entregar el altavoz de agudos. En un sistema con filtraje activo y multiamplificación este hecho no se produce, pues cada amplificador trabaja de forma independiente respecto al resto.

✚ Las diferencias entre las sensibilidades individuales de los altavoces seleccionados, incluso las imputables a pequeños desajustes en las tolerancias de fabricación, pueden ser reajustadas con ayuda del filtro activo gracias a los controles de ganancia para cada amplificador.

✚ El filtro activo también puede ser un eficaz factor de equilibrio en aquellas salas que, por sus características, presentan notables desequilibrios en orden a una restitución musical coherente y focalizada.

✚ En los filtros activos digitales no hay rotación de fase.

Existen más argumentos relacionados con aspectos relativos a impedancias, resistencia de los altavoces, control de las resonancias de los altavoces, ausencia de coloraciones a causa del factor de amortiguamiento, técnicas de realimentación negativa, entre otras.

1.2.2. FILTROS ANALÓGICOS

Para filtrar la señal analógica se emplea circuitos compuestos por resistencias, inductores, capacitores, amplificadores operacionales, etc; su principio está dado en el hecho de que la reactancia capacitiva de un condensador es mayor en cuanto sea mayor la frecuencia de la señal aplicada a sus terminales, contrariamente la reactancia inductiva es mayor cuando disminuye la frecuencia de la señal aplicada a sus terminales.

En la figura 1.9. se representa uno de los tantos circuitos divisores de frecuencias que se puede encontrar en el mercado. Este circuito está constituido por un inductor en serie con el woofer, éste inductor se opone al paso de frecuencias medias y altas hacia dicho altavoz; un condensador y un inductor en serie con el squawker, la inductancia se opone al paso de las frecuencias altas y el condensador al paso de las bajas frecuencias; un condensador en serie con el tweeter, el cual se opone al paso de las frecuencias medias y bajas hacia dicho altavoz. Para obtener un mejor resultado del filtro se conecta una inductancia en paralelo con el tweeter, la cual cortocircuita las señales de media frecuencia que pueden llegar al altavoz, es decir atenúan más a las señales de *media* frecuencia.

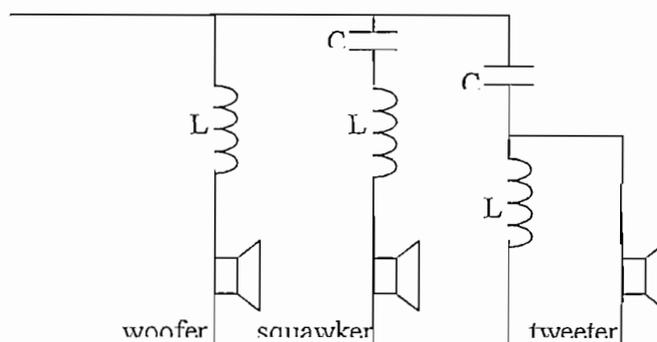


Figura 1.9. Filtro crossover analógico de 3 vías

En el cálculo del filtro divisor de frecuencias se debe fijar el valor de la frecuencia de cruce y en función de ésta calcular los valores de inductancias y capacitancias a colocarse en el mencionado filtro. Los valores que se obtienen no son de fabricación estándar por lo que al utilizar estos componentes no se obtendrá la respuesta prevista del filtro. Generalmente los capacitores varían entre unos pocos microfaradios hasta unos 50 microfaradios, según permitan el paso de altas y medias frecuencias respectivamente. En cuanto a inductores se refiere el valor de estos oscila entre 3 mH y algunas décimas de milihenrios de acuerdo permitan pasar frecuencias bajas o medias respectivamente.

Es preciso anotar que dada las elevadas capacidades de los condensadores empleados estos no pueden ser electrolíticos pues funcionan con corriente alterna, por lo que el volumen de los capacitores es grande y de costo

elevado. Otro inconveniente se encuentra en el devanado de las inductancias que debe ser grande para resistir la corriente que circulará por ellas, y por estar formadas por muchas vueltas también tendrán un tamaño relativamente grande.

1.2.3. FILTROS DIGITALES

Se considera como filtro digital a un sistema diseñado para filtrar señales muestreadas, la cantidad de muestras que se toman por segundo se denomina frecuencia de muestreo y ésta según el criterio de Nyquist de ser mayor que dos veces la frecuencia de la señal, las muestras se representan en forma numérica generalmente en sistema binario y sobre estas se realizan operaciones aritméticas reales especificadas por las ecuaciones que caracterizan a los filtros analógicos.

La respuesta en frecuencia de un filtro se suele expresar con su función de transferencia en módulo y en fase:

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\phi}$$

Un filtro ideal presenta módulo constante y fase lineal con la frecuencia en la banda de paso, mientras que en la banda eliminada el módulo toma un valor nulo. Expresado matemáticamente:

$$|H(e^{j\omega})| = \begin{cases} 1 & \text{en bandas de paso} \\ 0 & \text{en las bandas atenuadas} \end{cases}$$

La respuesta impulsional del filtro ideal, en cambio, es no causal, por tanto no realizable. Se debe entonces tomar unas tolerancias a la hora de definir nuestro sistema razón por la que se define entonces una banda de paso, una de transición, y una banda atenuada o eliminada.

Los filtros lineales e invariantes, así como causales y estables están caracterizados por una ecuación en diferencias finitas de coeficientes reales y constantes.

$$y[n] = \sum_{k=0}^Q b_k x[n-k] - \sum_{k=1}^P a_k y[n-k]$$

Las muestras de salida se componen de $Q+1$ muestras de la señal de entrada y $P+1$ muestras de la propia señal de salida en instantes anteriores. La función de transferencia queda entonces:

$$H(z) = \frac{\sum_{k=0}^Q b_k z^{-k}}{1 + \sum_{k=0}^P a_k z^{-k}}$$

A partir de la definición de la función de transferencia se definen los dos filtrados digitales: FIR (Finite Impulse Response, Respuesta Impulsiva Finita) e IIR (Infinite Impulse Response, Respuesta Impulsiva infinita).

1.2.3.1. Ventajas de los filtros digitales sobre los analógicos

- *Intervalo dinámico:* El límite superior es fijado por el tamaño del número que el hardware utilizado pueda representar, y el límite inferior es fijado por el ruido de cuantización y errores de redondeo. El intervalo entre estos límites depende únicamente de las longitudes de las palabras utilizadas (bits utilizados en la representación binaria de la señal).
- *Ausencia de problemas de componentes:* No existen problemas de tolerancias de las resistencias, capacitores, inductores o amplificadores; así como tampoco existen problemas de impedancia de entrada o salida de carga entre etapas. Al aumentar la longitud de la palabra es posible lograr cualquier grado de exactitud deseado.
- *Conmutabilidad:* Si los parámetros de los que depende un filtro se conservan en un registro se los puede cambiar a voluntad y en tiempo real.

- *Adaptabilidad:* Los filtros digitales implementados en hardware se adaptan sin dificultad para el control en procesadores de sus parámetros.

1.3. PROCESADOR DIGITAL DE SEÑALES (DSP)

Los DSP son microprocesadores diseñados para procesamiento digital de señales (manipulación matemática de señales representadas digitalmente). El procesamiento digital de señales es una tecnología cuyas aplicaciones están creciendo rápidamente, como en el caso de comunicaciones sin hilo, procesamiento de audio, vídeo y control industrial. A la vez que aumenta la popularidad de las aplicaciones DSP la variedad de procesadores DSP ha aumentado espectacularmente desde la introducción de los primeros chips comerciales a principios de los ochenta. Los procesadores de hoy son dispositivos sofisticados con características impresionantes.

1.3.1. OPERACIONES BÁSICAS

Los DSP son microprocesadores que comparten una arquitectura base optimizada. El procesamiento digital de la señal se basa en la manipulación matemática de las señales representadas en un formato digital, este análisis matemático conlleva un volumen de operaciones numéricas muy elevado. Por esto, la arquitectura de los DSP debe responder a las necesidades de las operaciones que soportan, fundamentalmente: filtrado, convolución, análisis de espectros, correlación, etc. Para filtrado tenemos que:

$$y[n] = \sum_{k=0}^{N-1} a[k]x[n-k] \quad \text{Para filtros FIR.}$$

$$y[n] = \sum_{k=0}^M b_k x[n-k] + \sum_{k=0}^N a_k y[n-k] \quad \text{Para filtros IIR}$$

En estos cálculos la operación básica a realizar es multiplicar-acumular, es decir, sumas acumulativas de productos en los que los factores varían en cada operación. Analíticamente operaciones del tipo:

$$y(n) = \sum_{k,n} a(n)b(k)$$

Siendo $a(n)$ y $b(k)$ dos secuencias cuyas muestras varían en cada producto. Para optimizar esta operación la mayoría de los DSP comparten las siguientes características:

- Permiten realizar la operación multiplicar-acumular en un solo ciclo.
- Permiten realizar varios accesos a memoria en un solo ciclo de instrucción, de esta manera el procesador puede buscar una instrucción mientras a la vez está, realizando la búsqueda de operandos y/o almacenando el resultado de una instrucción anterior.
- Para permitir múltiples accesos a memoria, los DSP incluyen memorias multipuerto e incluso bancos de memoria independientes.
- Poseen una o más unidades generadoras de direcciones independientes. Dichas unidades operan en paralelo con la ejecución de instrucciones aritméticas.
- La mayoría de los DSP disponen de un set de instrucciones que soportan la ejecución de bucles, debido a que los algoritmos DSP implican cálculos repetitivos.
- Incorporan varios puertos de entrada / salida serie o paralelo para manejar interrupciones y accesos directos a memoria sin intervención del procesador.

1.3.2. PARÁMETROS DE SELECCIÓN DEL DSP ADECUADO

Como hemos visto hasta ahora, el DSP adecuado para cada tarea depende enormemente de esa tarea. Por ejemplo, un procesador que desarrolla

bien ciertas tareas puede ser una pobre elección para otras. Basándonos en esto podemos considerar ciertas características que varían de un DSP a otro a la hora de elegir un procesador.

Una forma de clasificar los dispositivos DSP y sus aplicaciones es por su rango dinámico. Se denomina rango dinámico al conjunto de valores, entre el menor y el mayor, que puede ser procesado en el curso de una operación. Esto nos ha de proporcionar un conjunto de valores para describir por completo una forma de onda señalada, desde el mínimo más profundo hasta el pico más alto. El rango ha de ser más amplio que el requerido para los cálculos, ya que se irán generando valores mayores y menores a partir de las multiplicaciones y divisiones. El dispositivo DSP ha de tener la capacidad de manipular los valores que se generen, si no pudiera se produciría un desbordamiento.

La capacidad del procesador en función de su anchura de datos (el número de bits que manipula) y del tipo de aritmética que permite (punto fijo o flotante), nos permite asignarle un tipo correspondiente de aplicaciones, así como determinar cual dispone de un mayor rango dinámico. Un DSP de 32 bits tiene un rango dinámico mayor que uno de 24 bits, y este a su vez, mayor que uno de 16 bits. Los chips de punto flotante tienen rangos dinámicos más amplios que los dispositivos de punto fijo. Cada tipo de procesador es ideal para un rango específico de aplicaciones. Los DSP's de 16 bits de punto fijo son buenos para sistemas de voz, como teléfonos, ya que estos DSP trabajan con el rango relativamente estrecho de las frecuencias del sonido. Las aplicaciones estéreo de alta fidelidad tienen un rango de frecuencias más amplio, usando un ADC de 16 bits y un DSP de 24 bits de punto fijo; el ADC es de 16 bits ya que es suficientemente amplio para obtener la señal de alta fidelidad completa, el DSP ha de ser de 24 bits para poder manipular los valores que se obtienen al procesar la señal. El procesamiento de imágenes, gráficas en 3D y simulaciones científicas tiene un rango dinámico mucho más amplio y necesitan DSP's de 32 bits con aritmética de punto flotante.

1.3.2.1. Formato aritmético

Una de las características más fundamentales de los procesadores digitales programables es el tipo de aritmética utilizada por el procesador. La mayor parte de los DSP usan aritmética de punto fijo, donde los números se representan como enteros o como fracciones entre -1.0 y $+1.0$. Otros procesadores usan aritmética de punto flotante, donde los valores se representan por una mantisa y un exponente como $\text{mantisa} \times 2^{\text{exp}}$. La mantisa generalmente es una fracción con rango entre -1.0 y $+1.0$, mientras el exponente es un entero que representa en binario el número de lugares a partir del punto que se debe desplazar a izquierda o derecha para obtener el valor representado.

La aritmética en punto flotante es mucho más flexible que la de punto fijo. En punto flotante, los diseñadores de sistemas tienen acceso a un rango dinámico más amplio (la distancia entre mayor y el menor valor que puede representar). Como resultado, los DSP de punto flotante son generalmente más fáciles de programar que sus correspondientes de punto fijo, pero normalmente más caros.

El incremento del costo se debe a la compleja circuitería necesaria para realizar los procesos de punto flotante. En punto flotante el programador no necesita conocer en muchos casos ni el rango dinámico ni la precisión, mientras que, en punto fijo, los programadores han de ser cuidadosos asegurándose de que sus señales no excedan el rango dinámico.

Los procesadores de punto fijo, se usan en muchas aplicaciones debido a su bajo costo. En estas aplicaciones son necesarios programas y algoritmos diseñados para determinar el rango dinámico y la precisión. En las aplicaciones en las que el costo es poco importante o bien es necesario un amplio rango dinámico o gran precisión, se utilizarán los procesadores de punto flotante.

En los procesadores de propósito general, la aritmética de punto flotante se suele simular mediante software, es decir, generando rutinas que emulen el elemento de aritmética en punto flotante. Estas rutinas tienen un alto costo en

términos de ciclos de procesador. Una técnica más eficiente para incrementar el rango numérico de procesadores de punto fijo es el punto flotante de bloque, donde un grupo de números de mantisas diferentes y exponente común se procesan como un bloque de datos. Estos bloques se suelen manejar por software, aunque algunos procesadores tienen circuitería hardware para lograr su implementación.

1.3.2.2. Anchura de datos

Todos los DSP comunes de punto flotante usan una palabra de datos de 32 bits. Para DSP de punto fijo, el tamaño de palabra más común es de 16 bits. El tamaño del dato tiene una mayor repercusión en el costo, ya que influye notablemente en el tamaño del chip y el número de pines que requiere, así como el tamaño de los dispositivos externos conectados al DSP. Por todo esto, los diseñadores tratan de emplear anchuras de palabras lo mas pequeñas posibles de acuerdo a sus necesidades.

Como en el caso de la elección entre punto fijo y punto flotante, hay que buscar el equilibrio entre el tamaño de palabra y la complejidad de desarrollo. Por ejemplo, un procesador de 16 bits puede llevar a cabo operaciones aritméticas de 32 bits en doble precisión a través de encadenar juntas una combinación adecuada de instrucciones. Por supuesto la aritmética de doble precisión es mucho más lenta que la precisión simple. Si el volumen de una aplicación puede manejarse con aritmética de precisión simple, pero la aplicación necesita mas precisión para una sección pequeña del código, valdría la pena emplear selectivamente la aritmética de doble precisión. La familia ADSP-21XX que trabaja con una palabra de datos de 16 bits y una de instrucciones de 24 bits.

1.3.2.3. Velocidad

Se puede obtener una medida de cuan adecuado es un procesador para una determinada tarea a través de su velocidad de ejecución. Hay muchas formas de medir la velocidad de ejecución de un procesador. Quizá la fundamental sea el

tiempo de ciclo de instrucción: la cantidad de tiempo empleado en ejecutar la instrucción más rápida del procesador. El inverso de este tiempo dividido por un millón es la velocidad de ejecución del procesador en millones de instrucciones por segundo o MIPS.

1.3.2.4. Multiprocesamiento

Aplicaciones con una gran carga aritmética (como radar o sonar) a menudo requieren varios DSP's. En esos casos, facilidad de interconexión (en términos de tiempo de diseño de la circuitería de comunicaciones y el costo de unir los procesadores) y su funcionamiento (en términos de velocidad de comunicación y latencia) son los factores importantes. Algunas familias de DSP's (especialmente Texas Instruments y Analog Devices ADSP-2106x) proporcionan hardware especial para el diseño de un sistema multiprocesador.

1.3.2.5. Potencia

Los DSP's cada vez se usan más en aplicaciones portátiles como teléfonos móviles donde el consumo se convierte en una característica importante. Por esto, la mayoría de los fabricantes de DSP's han reducido los voltajes de alimentación de los procesadores e incluyen algunas características que permiten al programador reducir el consumo, algunas de estas características son las siguientes:

- Reducción del voltaje: los fabricantes han introducido versiones de baja potencia (3'3 ó 3'0 V) de sus DSP's. Estos procesadores consumen aproximadamente un 40% que sus equivalentes de 5 V a la misma frecuencia de reloj.
- Modos "sleep" o "idle": estos modos permiten desconectar el reloj del procesador, excepto de ciertas secciones del procesador, reduciendo el consumo. En algunos casos se sale de este estado mediante una

interrupción no enmascarable, en otros sólo se sale a través de alguna línea externa de interrupción.

- Divisores de reloj programables: algunos DSP's actuales permiten variar la frecuencia del reloj por software para usar la mínima frecuencia de reloj para cada tarea.
- Control de periféricos: algunos DSP's permiten al programador deshabilitar los periféricos que no se usan.
- A pesar de estas características es difícil obtener el consumo de los DSPs porque este consumo varía, hasta por un factor de 3, dependiendo de la instrucción que está ejecutando. Los vendedores suelen publicar el consumo típico o máximo sin especificar lo que constituye un programa típico.

1.3.2.6. Costo

Los diseñadores procuran usar el DSP de menor costo que cumple los requerimientos de la aplicación, aunque sean menos flexibles y más difíciles de programar.

Un factor que influye en el precio es el tipo de encapsulado. Encapsulados del tipo "plastic quad flat pack" (PQFP) y "thin quad flat pack" (TQFP) son más baratos que un encapsulado del tipo "pin grid array" (PGA).

CAPITULO II

DISEÑO DE FILTROS CROSSOVER

En el diseño de un sistema digital crossover se deberá considerar las siguientes condiciones:

1. Fase lineal combinada plana.
2. Adecuado rango de frecuencias de corte para cada filtro.
3. Buena atenuación para las bandas de supresión de cada filtro, previniendo señales fuera de la banda que puedan saturar y posiblemente dañar los altavoces.

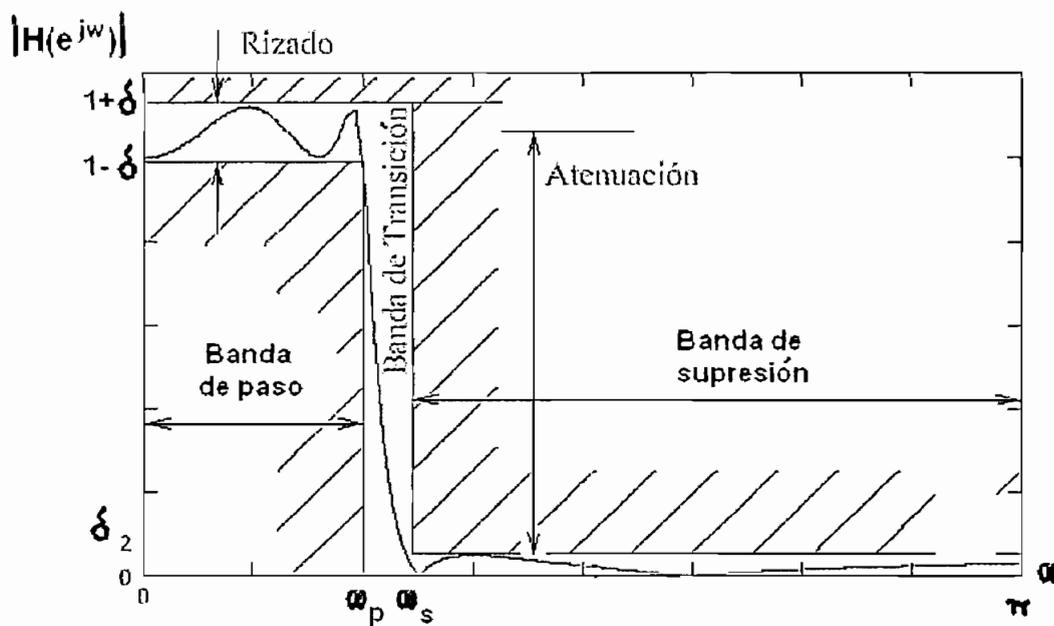
Adicionalmente es muy deseable que el sistema altavoces/crossover cerca del oyente produzca un nivel de presión plano en todo el espectro de audio, es decir sin distorsión en la amplitud y fase; desgraciadamente los altavoces son dispositivos electromecánicos pasivos que introducen errores. Tradicionalmente se compensaban esos errores con circuitería analógica, los diseños analógicos podían reducir estos errores parcialmente debido a que los filtros introducen no linealidades. En la actualidad para evitar estos hechos se utilizan procesadores digitales de señales, las redes crossover digitales son capaces de fraccionar la señal en múltiples bandas de frecuencias sin introducir amplificación o atenuación.

El requisito de una respuesta lineal de fase en el sistema crossover se debe a que la señal proveniente de los distintos altavoces lleguen simultáneamente al oyente evitándose distorsiones en la señal resultante; los filtros digitales FIR mantienen la linealidad de fase haciéndolos ideales para los requerimientos del sistema.

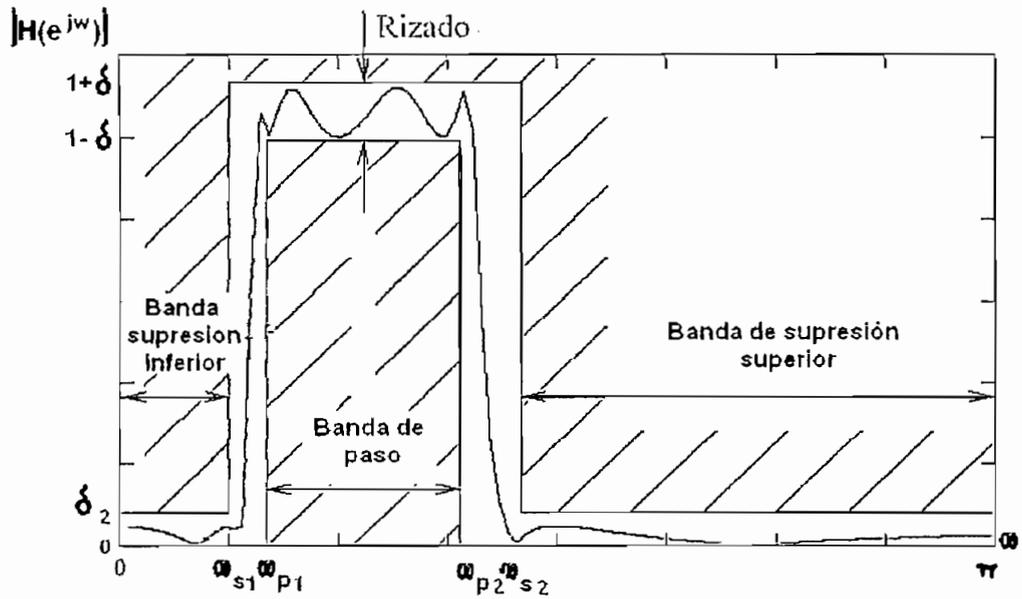
Existen diversos tipos de filtros FIR así como también métodos para crearlos, entre los métodos está el de enventanado que disminuye el número de oscilaciones multiplicando la respuesta del pulso infinito original con una función ventana; algunas ventanas brindan buena atenuación en la banda de supresión, mientras otras tienen parámetros ajustables; para el presente sistema la ventana de Kaiser es la apropiada por tener parámetros ajustables y buena atenuación en la banda de supresión.

2.1. ESPECIFICACIONES DE FILTROS DIGITALES

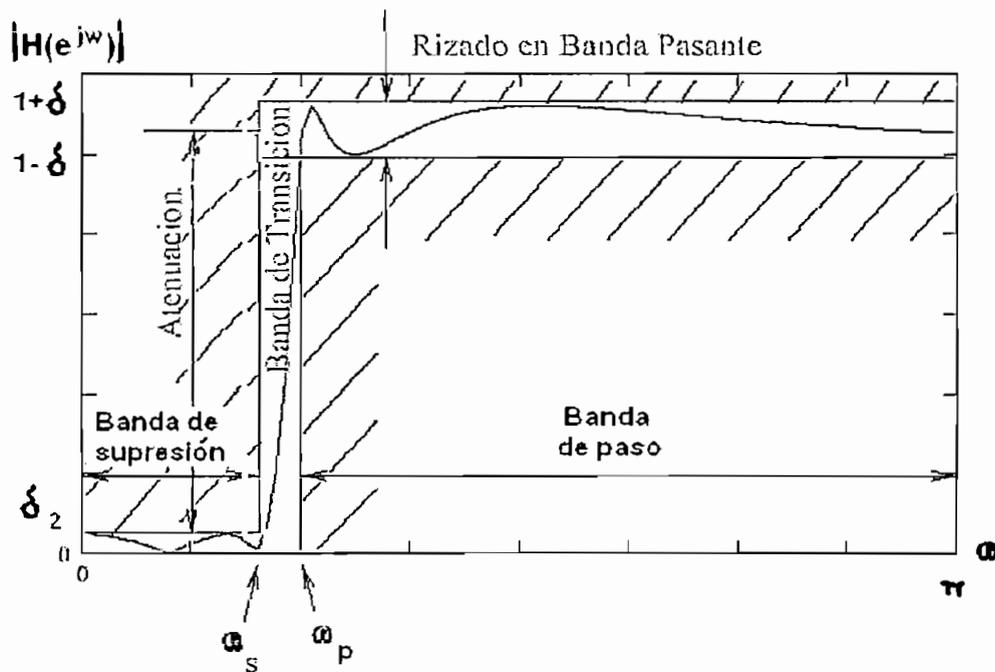
Para poder implementar un filtro se debe tomar consideraciones como que para pasar de la banda de paso a la de supresión se debe pasar primero por una zona de amortiguamiento que se la denomina banda de transición, permitiéndose de esta manera desviaciones de la respuesta ideal. En la figura 2.1. se observan los tres tipos de filtros a utilizarse en la presente aplicación con sus respectivas características.



a) Filtro pasa bajo



b) Filtro pasa banda



c) Filtro pasa alto

Figura 2.1. Especificaciones de un filtro

Se definen dos frecuencias: de paso (ω_p) y supresión (ω_s); adicionalmente se pueden observar que existen δ_1 y δ_2 que corresponden a los rizados de la banda de

paso y supresión respectivamente, estos rizados son los límites de los errores de aproximación a un filtro ideal.

El ancho de la banda de transición por lo tanto será igual a: $\Delta\omega = \omega_p - \omega_s$; justo en la mitad de este ancho de banda se encuentra la frecuencia de cruce de los filtros (ω_c), esta frecuencia de cruce adicionalmente está en el punto de media potencia ó de -3 dB haciendo que los filtros crossover sean complementarios.

2.2. DISEÑO DE FILTROS FIR CON VENTANA DE KAISER

Los filtros FIR tienen ventajas muy interesantes que les hacen ser ampliamente utilizados en múltiples aplicaciones, la característica más destacable es su facilidad de diseño para conseguir una respuesta en frecuencias de fase lineal, esto es, la señal que pase a través de él no será distorsionada. Los FIR son por su propia constitución estables, no habiendo problemas en su diseño o fase de implementación.

Aunque el diseño de los FIR requiera de una gran cantidad de operaciones de sumas y multiplicaciones, tanto su estructura de programación como su implementación en soporte físico resultan fáciles y escalables. Su mayor desventaja está en que para iguales requisitos de especificaciones del filtro resulta con menor orden los filtros IIR que los FIR, implicando programas más largos.

Una ventana de Kaiser de longitud M está dada por la siguiente fórmula: ^[8]

$$w[n] = \begin{cases} \frac{I_0 \left[\beta \sqrt{1 - \left[\frac{(n-\alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)}, & 0 \leq n \leq M \\ 0, & \text{en los demás casos} \end{cases}$$

Donde $\alpha = M/2$ e $I_0(.)$ es la función de Bessel modificada de orden cero y del primer tipo; que está definida por:

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left(\frac{1}{kM(2)} \left(\frac{x}{M} \right)^k \right)^2$$

Para la aproximación a un filtro ideal se tiene que $A = -20 \log_{10} \delta$; tomándose el menor de los rizados para el cálculo de la amplitud en dB, adicionalmente de una forma experimental Kaiser determina que:

$$\beta = \begin{cases} 0.1102(A - 8.7), & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & 21 \leq A \leq 50 \\ 0.0, & A < 21 \end{cases}$$

En el cálculo del orden del filtro con una precisión de ± 2 se tiene:

$$M = \frac{A - 8}{2.285 * \Delta\omega}$$

En general los filtros pueden ser diseñados de acuerdo a la tabla 2.1 que se muestra a continuación. ^[8]

PASA BAJO	PASA BANDA	PASA ALTO
$\omega_p = \frac{f_p}{f_m} * 2\pi$	$\omega_{s1} = \frac{f_s}{f_m} * 2\pi$	$\omega_s = \frac{f_s}{f_m} * 2\pi$
$\omega_s = \frac{f_s}{f_m} * 2\pi$	$\omega_{p1} = \frac{f_p}{f_m} * 2\pi$	$\omega_p = \frac{f_p}{f_m} * 2\pi$
	$\omega_{p2} = \frac{f_p}{f_m} * 2\pi$	

	$\omega_{s2} = \frac{f_s}{f_m} * 2\pi$	
$\Delta\omega = \omega_s - \omega_p$	$\Delta\omega_1 = \omega_{p1} - \omega_{s1}$	$\Delta\omega = \omega_p - \omega_s$
	$\Delta\omega_2 = \omega_{p2} - \omega_{s2}$	
	$\min(\Delta\omega_1, \Delta\omega_2)$	
$\omega_c = \frac{\omega_p + \omega_s}{2}$	$\omega_c = \frac{\omega_p + \omega_s}{2}$	$\omega_c = \frac{\omega_p + \omega_s}{2}$

<i>COMUN PARA LOS TRES TIPOS DE FILTROS</i>
$A = -20 \log(\delta)$
$M = \frac{A - 8}{2.285 * \Delta\omega}$
$\alpha = \frac{M}{2}$
$w[n] = \frac{I_0 \left[\beta \sqrt{1 - \left[\frac{(n - \alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)}$

Tabla 2.1. Fórmulas para el diseño de filtros digitales [9]

Para encontrar la respuesta de frecuencia, es decir los coeficientes del filtro tenemos que multiplicar la respuesta al impulso ideal del filtro por la ventana, así:

PASA BAJA

$$h[n] = \frac{\text{sen}[\omega_c * (n - \alpha)]}{\pi * (n - \alpha)} * w[n]$$

PASA BANDA

$$h[n] = \frac{\text{sen}[\omega_{c2} * (n - \alpha)] - \text{sen}[\omega_{c1} * (n - \alpha)]}{\pi * (n - \alpha)} * w[n]$$

PASA ALTO

$$h[n] = \delta[n - \alpha] - \frac{\text{sen}[\omega_c * (n - \alpha)]}{\pi * (n - \alpha)} * w[n]$$

2.2.1. SISTEMA DE ALTAVOCES

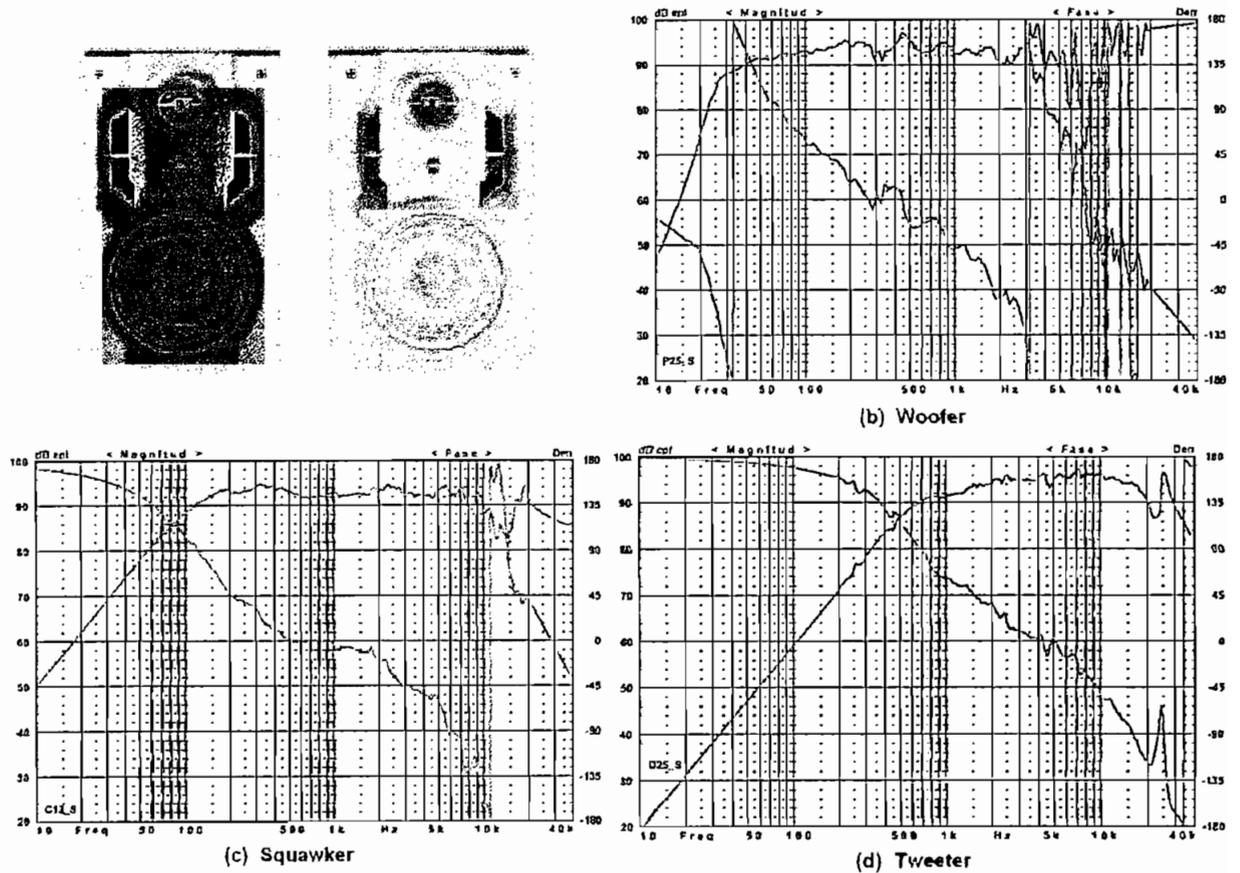


Figura 2.2. Sistema de altavoces y respuestas de las vías. [22]

El sistema de altavoces de tres vías de un equipo de sonido marca LG modelo F-886 se muestra en la gráfica 2.2 (a), los tres gráficos siguientes de la figura 2.2 indican la magnitud del nivel de presión y la fase para cada uno de los tres transductores usando un rango de frecuencias de 10Hz a 40 KHz, la fase medida no contiene retrasos.

2.2.1.1. Retraso relativo de los altavoces

Para determinar la posición de cada uno de los altavoces se debe seleccionar un punto de referencia en el espacio como se indica en la figura 2.3, a altas frecuencias el punto de origen acústico está cerca de las bobinas del transductor, aquí es donde se hace la conversión de señal eléctrica a acústica y empieza la propagación de ondas. Por esta razón el tweeter se usa como referencia ya que su bobina está en el panel de los altavoces, en cuestión de diseño se considera que el punto de referencia está en el mismo eje del tweeter, la bobina de altavoz de medio rango está a 0.033 metros del panel de altavoces y la del woofer está a 0.058 metros.

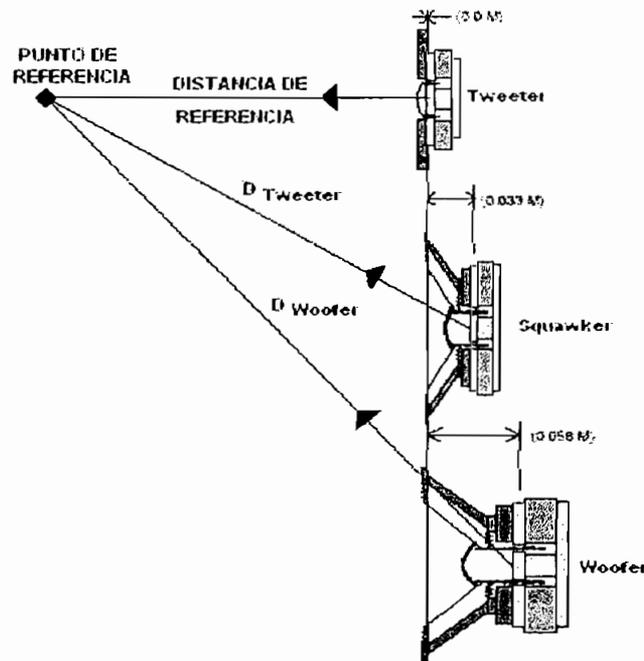


Figura 2.3. Posición de los altavoces

Sin embargo cabe señalar que la diferencia del camino total entre las bobinas y el punto de referencia en el espacio depende de la distancia de adelanto escogida con respecto al panel y al espacio vertical entre altavoces. A distancias muy lejanas la distancia entre altavoces es relativamente sin importancia y la diferencia de caminos entre los altavoces se pone igual a las bobinas del exterior del sistema, a

distancias pequeñas el espaciado entre los altavoces debe ser incluido en el cálculo de caminos.

Normalmente se dan distancia de referencia típicos entre 1 o 2 metros para la mayoría de altavoces, para el análisis se utilizará 1 metro como distancia de referencia y que el espacio entre altavoces es de aproximadamente 0.15 metros.

Usando geometría básica encontramos las distancias acústicas:

$$D_{\text{TWEETER}} = 1 \text{ m.}$$

$$D_{\text{SQUAWKER}} = [(1+0.033)^2 + 0.15^2]^{1/2} = 1.044 \text{ m.}$$

$$D_{\text{WOOFER}} = [(1+0.058)^2 + 0.3^2]^{1/2} = 1.100 \text{ m.}$$

Por consiguiente la diferencia de retrasos en los woofer y los squawker considerando una velocidad del sonido de 350 m/seg. son de:

$$\Delta D_{\text{SQUAWKER}} = 0.044 / 350 \text{ seg.} = 126 \text{ useg.}$$

$$\Delta D_{\text{WOOFER}} = 0.100 / 350 \text{ seg.} = 286 \text{ useg.}$$

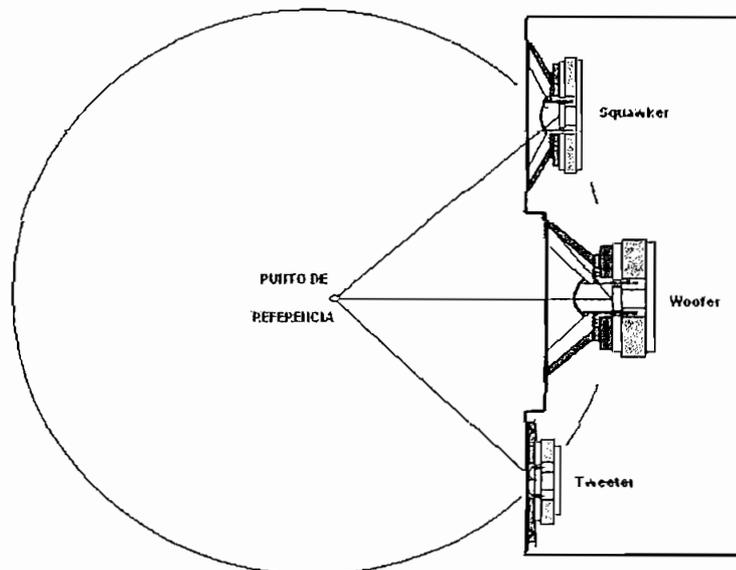


Figura 2.4. Sistema de altavoces coaxial

Para evitar estos retrasos se propone un sistema multivía coaxial como el de la figura 2.4 que por tener la misma distancia al punto de referencia no producirían los citados retardos.

2.2.2. PARAMETROS DE DISEÑO

Para elegir las frecuencias de cruce de los filtros crossover se deben tomar en cuenta el rango de frecuencias para las cuales el altavoz tiene una respuesta lo más plana posible, de acuerdo a las características del gráfico 2.2 así como de otros altavoces comerciales que se encuentran en el Anexo 1 se puede deducir que estos rangos de frecuencia están entre:

$$100 \text{ Hz.} \leq f_{c1} \leq 800 \text{ Hz.}$$

$$2500 \text{ Hz.} \leq f_{c2} \leq 8000 \text{ Hz.}$$

El oído humano escucha hasta frecuencias de 20000 Hz , lo cual indica que la frecuencia de muestreo debe ser mayor o igual a 40000 Hz, se escoge una frecuencia de muestreo estandarizada de 44100 Hz.

Para visualizar un ejemplo de diseño del sistema de filtros se toman como frecuencias de cruce entre el filtro pasa bajo y el pasa banda el valor de 350 Hz y entre el filtro pasa banda con el pasa alto se escoge 3500 Hz. La atenuación a utilizarse para los filtros es de 40 dB la cual es mucho menor a la que ofrecen los filtros crossover comerciales que se pueden observar en el Anexo 2.

El siguiente problema es el de escoger un ancho de banda conveniente para la zona de transición, se debe considerar que en el diseño del filtro pasa banda se utiliza el menor ancho de banda de las dos zonas de transición (tabla 2.1), adicionalmente se debe recordar que todos los filtros deben ser complementarios, razón por la cual se debe escoger el mismo valor para todas las bandas de transición, además al tener el mismo ancho de banda para todos los filtros éstos

tendrán el mismo orden evitando desfases de tiempo en las señales. Para el diseño se elige 350 Hz como ancho de banda de transición.

2.2.2.1. Diseño FIR de woofer

En los diseños de filtros FIR se omiten algunos cálculos, debido a que se dan como datos algunos parámetros que generalmente son calculados a partir de otros.

DATOS

FRECUENCIA DE CRUCE 1	$f_{c1} := 350$	Hz
ATENUACION	$A := 40$	dB
BANDA DE TRANSICION	$\Delta f := 350$	Hz
FRECUENCIA DE MUESTREO	$f_m := 44100$	Hz

CALCULOS

Normalizacion de Frecuencias

$$W_{c1} := \frac{f_{c1}}{f_m} \cdot 2 \cdot \pi \quad W_{c1} = 0.05$$

$$\Delta W := \frac{\Delta f}{f_m} \cdot 2 \cdot \pi \quad \Delta W = 0.05$$

Calculo del orden del filtro

$$\beta := 0.1102 \cdot (A - 8.7) \quad \beta = 3.449$$

$$M := \frac{A - 8}{2.285 \cdot \Delta W} \quad M = 280.837$$

$$\alpha := \frac{M - 1}{2} \quad \alpha := 140$$

El filtro de nivel 280 localiza su eje de simetría en $n=140$. A continuación se calcula la ventana y por ultimo la respuesta impulsiva.

$$w[n] = \frac{I_0 \left[\beta \sqrt{1 - \left[\frac{(n-\alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)}$$

$$h[n] = \frac{\text{sen}[\omega_c * (n-\alpha)] * w[n]}{\pi * (n-\alpha)}$$

Al graficar la respuesta impulsiva del filtro de orden 280 (figura 2.5) con ayuda del programa Mathcad 5.0 podemos observar la simetría que existe con respecto a $n=140$.

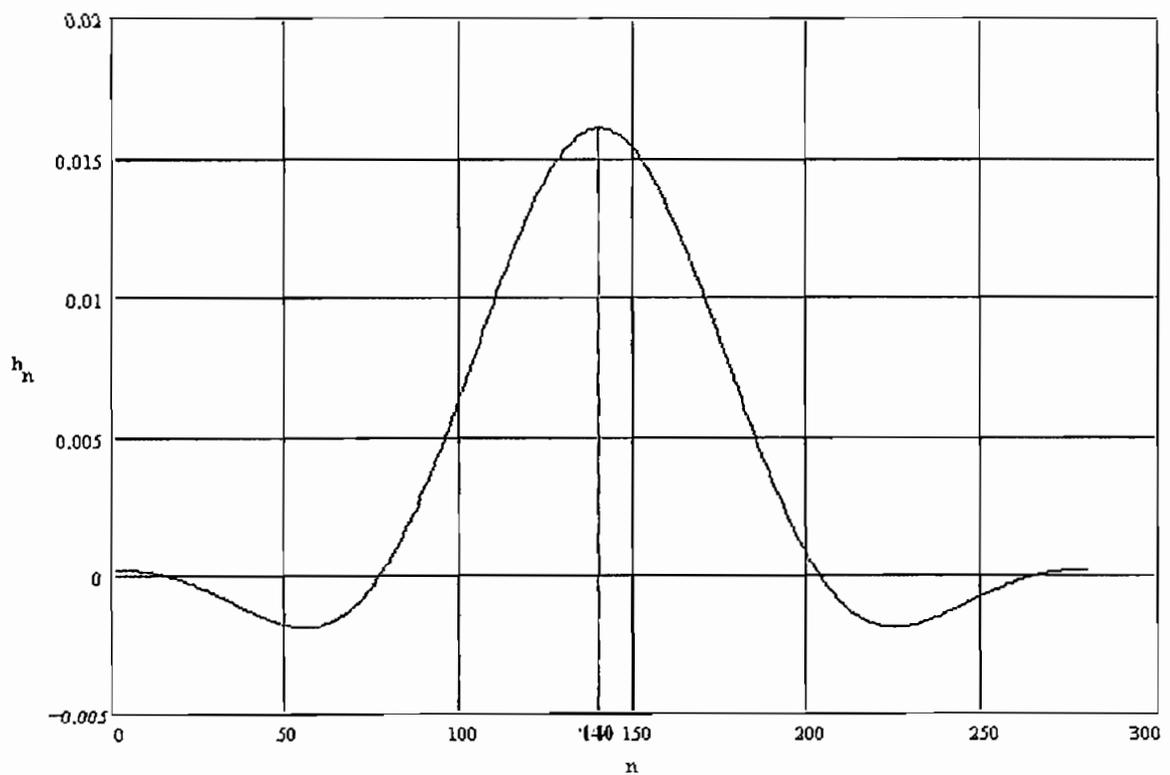


Figura 2.5. Respuesta impulsiva teórica sin cuantificar

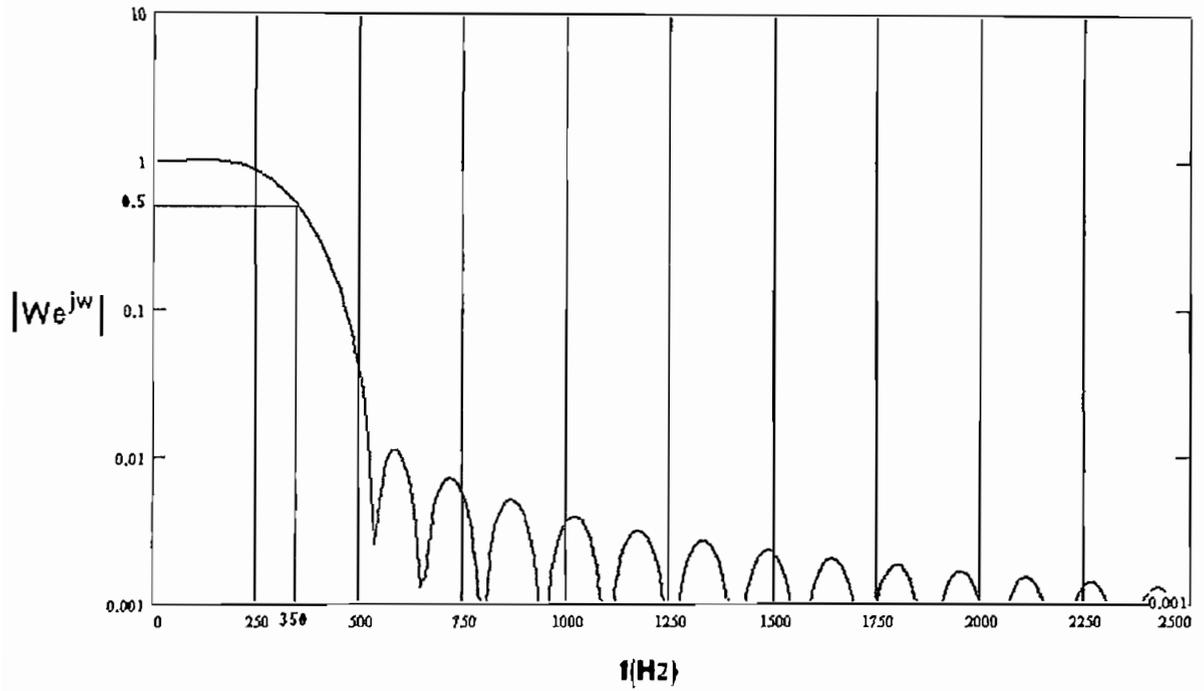


Figura 2.6. Respuesta de frecuencia con escala logaritmo del módulo

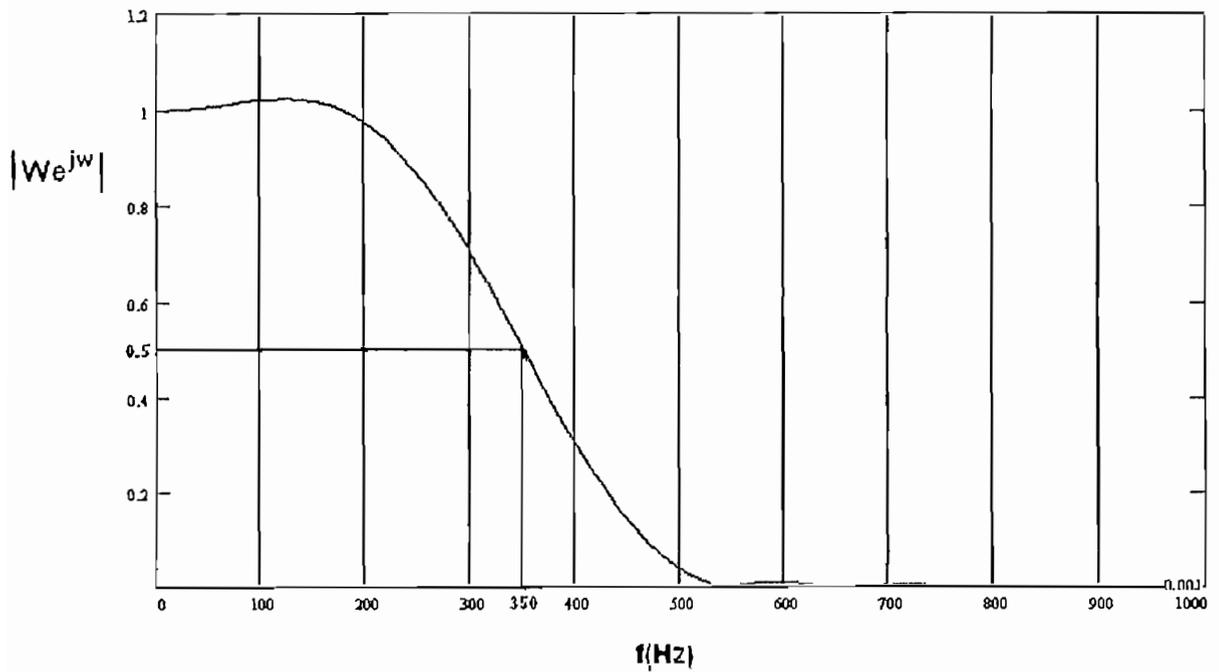


Figura 2.7. Respuesta de frecuencia con escala lineal del módulo.

2.2.2.2. Diseño FIR de Squawker

DATOS

FRECUENCIA DE CRUCE 1	$f_{c1} := 350$	Hz
FRECUENCIA DE CRUCE 2	$f_{c2} := 3500$	Hz
ATENUACION	$A := 40$	dB
BANDA DE TRANSICION	$\Delta f := 350$	Hz
FRECUENCIA DE MUESTREO	$f_m := 44100$	Hz

CALCULOS

Normalizacion de Frecuencias

$$W_{c1} := \frac{f_{c1}}{f_m} \cdot 2 \cdot \pi \quad W_{c1} = 0.05$$

$$W_{c2} := \frac{f_{c2}}{f_m} \cdot 2 \cdot \pi \quad W_{c2} = 0.499$$

$$\Delta W := \frac{\Delta f}{f_m} \cdot 2 \cdot \pi \quad \Delta W = 0.05$$

Calculo del orden del filtro

$$\beta := 0.1102 \cdot (A - 8.7) \quad \beta = 3.449$$

$$M := \frac{A - 8}{2.285 \cdot \Delta W} \quad M = 280.837$$

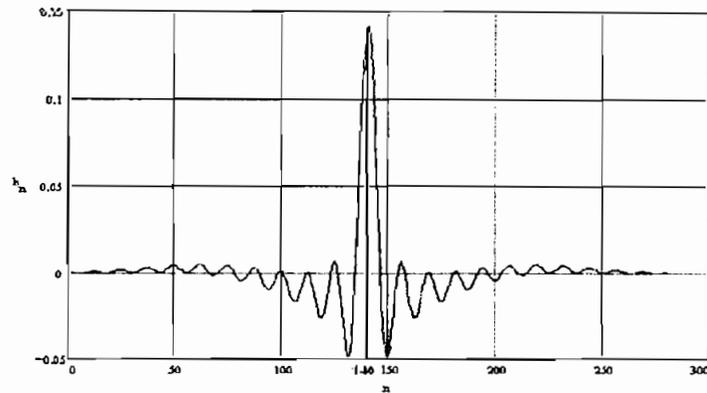
$$\alpha := \frac{M - 1}{2} \quad \alpha := 140$$

La respuesta impulsiva calculada a partir de la ventana de Kaiser es:

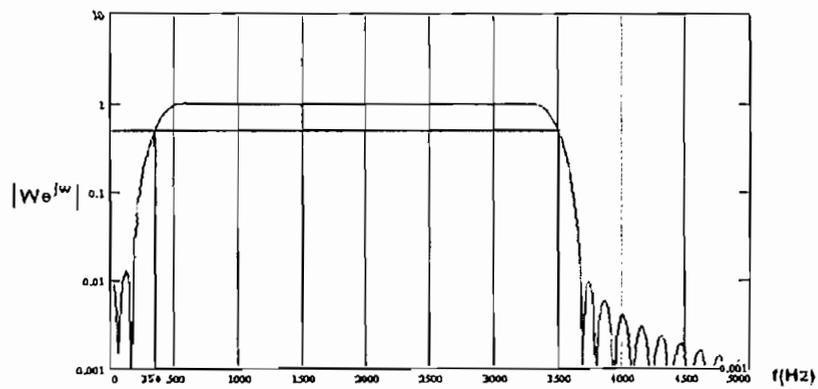
$$w[n] = \frac{I_0 \left[\beta \sqrt{1 - \left[\frac{(n - \alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)}$$

$$h[n] = \frac{\sin[\omega_{c2} * (n - \alpha)] - \sin[\omega_{c1} * (n - \alpha)]}{\pi * (n - \alpha)} * w[n]$$

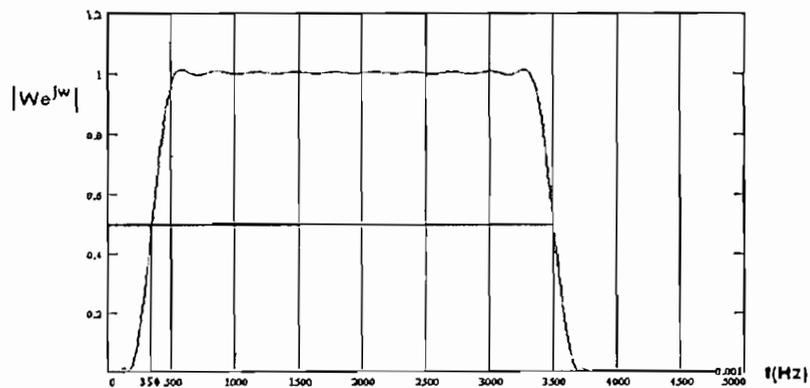
Las gráficas (figura 2.8) que se obtienen de la respuesta impulsiva del filtro pasa banda se presentan a continuación:



(a) Respuesta impulsiva sin cuantificar



(b) Respuesta de frecuencia con escala logarítmica



(c) Respuesta de frecuencia escala lineal

Figura 2.8. Respuesta de frecuencia filtro pasa banda.

2.2.2.3. Diseño FIR de Tweeter

DATOS

FRECUENCIA DE CRUCE 2	$f_{c2} := 3500$	Hz
ATENUACION	$A := 40$	dB
BANDA DE TRANSICION	$\Delta f := 350$	Hz
FRECUENCIA DE MUESTREO	$f_m := 44100$	Hz

CALCULOS

Normalizacion de Frecuencias

$$\omega_{c2} := \frac{f_{c2}}{f_m} \cdot 2 \cdot \pi \qquad \omega_{c2} = 0.499$$

$$\Delta \omega := \frac{\Delta f}{f_m} \cdot 2 \cdot \pi \qquad \Delta \omega = 0.05$$

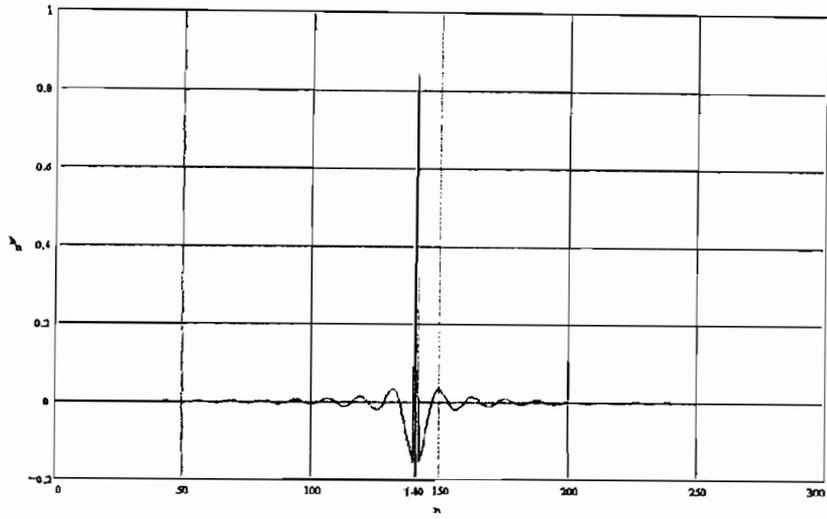
Calculo del orden del filtro

$$\beta := 0.1102 \cdot (A - 8.7) \qquad \beta = 3.449$$

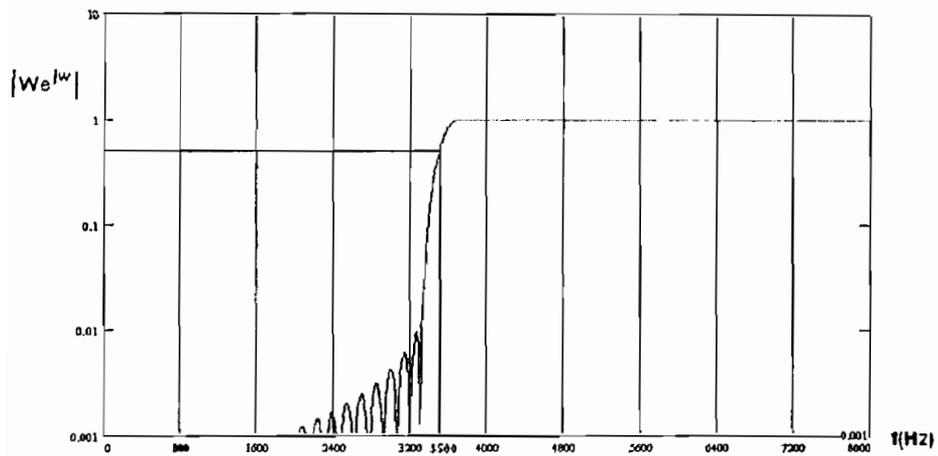
$$M := \frac{A - 8}{2.285 \cdot \Delta \omega} \qquad M = 280.837$$

$$\alpha := \frac{M - 1}{2} \qquad \alpha := 140$$

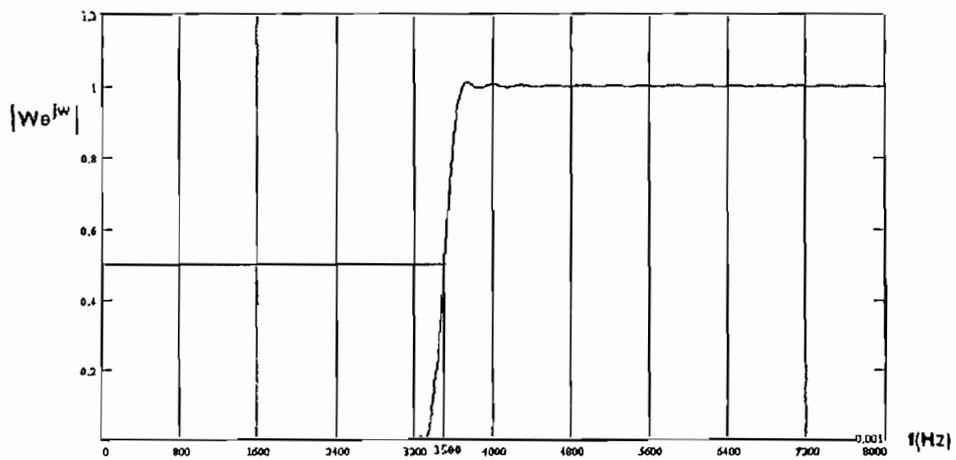
Las gráficas (figura 2.9) que se obtienen de la respuesta impulsiva del filtro pasa banda se presentan a continuación:



(a) Respuesta impulsiva sin cuantificar



(b) Respuesta de frecuencia con escala logarítmica

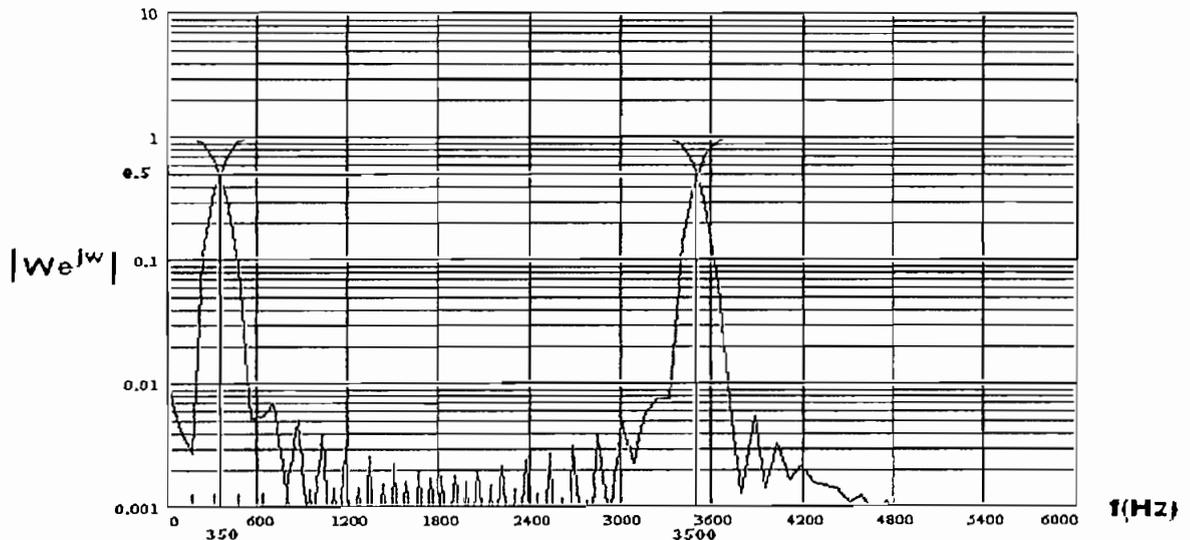


(c) Respuesta de frecuencia con escala lineal

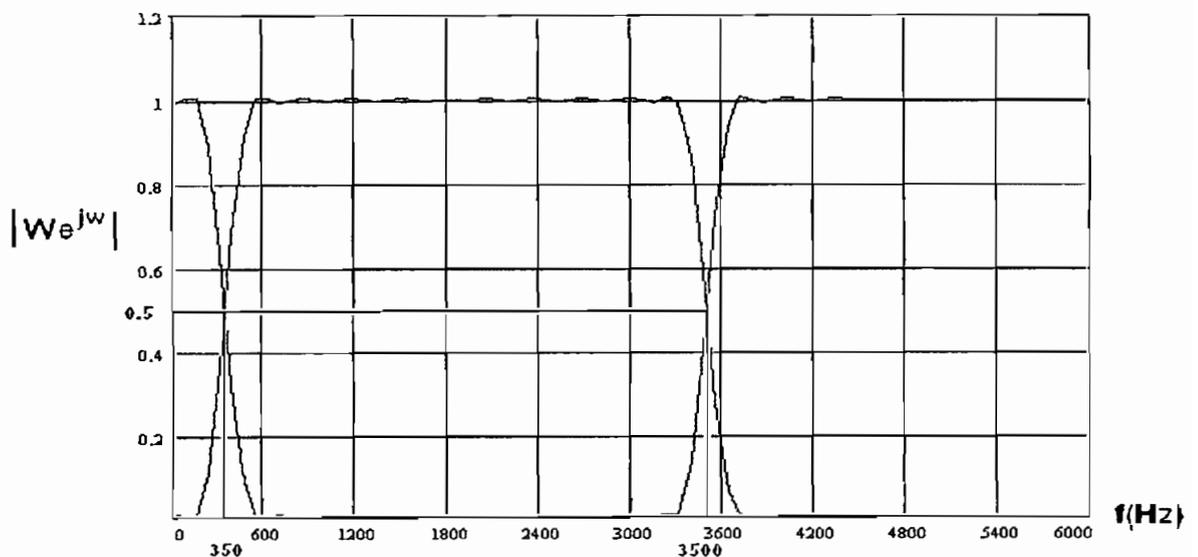
Figura 2.9. Respuesta de frecuencia filtro pasa alto.

2.2.2.4. Respuesta combinada FIR

Para obtener la respuesta total del sistema de filtros crossover se trazan las tres respuestas de los filtros como se indica en la figura 2.10., se puede observar que todos los filtros son complementarios cumpliendo los requerimientos propuestos para esta aplicación y asegurando además tener una respuesta plana sin distorsiones.



(a) Respuesta de frecuencia del sistema con escala logarítmica



(b) Respuesta de frecuencia del sistema con escala lineal

Figura 2.10. Respuesta de frecuencia del sistema de filtros

CAPITULO III

IMPLEMENTACION DE LOS FILTROS CROSSOVER DE TRES VIAS

La presente red crossover es un sistema de tecnología moderna, fiable y versátil que ofrece soluciones eficaces a problemas reales de carácter objetivo (propiedades acústicas de las salas) y subjetivo (gustos personales); está destinada a encontrar los parámetros que ayuden a una reproducción fiel. Se pretende alejar al usuario de la idea clásica de respetar la señal original propuesta por los sistemas "lector + amplificador + filtro pasivo + altavoces", ya que realmente lo que se respeta es el trabajo de seres totalmente subjetivos.

3.1. CARACTERISTICAS DE LA RED CROSSOVER IMPLEMENTADA

Las principales características de la red crossover son las siguientes:

- ✦ Entrada y salidas analógicas estereo
- ✦ Frecuencia de muestreo de 44.1 KHz
- ✦ Convertidores A/D y D/A de 16 bits
- ✦ Crossover con frecuencias de corte variables
- ✦ Filtros digitales con bandas de transición mayores a 350 Hz y atenuación de 40 dBs
- ✦ Control de ganancia y muting para cada una de las vías
- ✦ Fase lineal para el arreglo de altavoces
- ✦ Control del sistema crossover desde un PC a través de un interfaz serial

3.2. DESCRIPCION DEL SISTEMA CROSSOVER

En la implementación del sistema de filtros con procesamiento en tiempo real se requiere de:

- Un computador que calcule los coeficientes de cada uno de los filtros de acuerdo a las especificaciones que el usuario ingrese.
- Un interfaz serial RS-232 que permita la comunicación entre la PC y los DSPs de acuerdo a la figura 3.1.
- Tres módulos de desarrollo EZ-KIT Lite ADSP-2181, que discriminan los coeficientes que le corresponden a cada uno de sus filtros y a continuación empiecen a procesar la señal de audio.

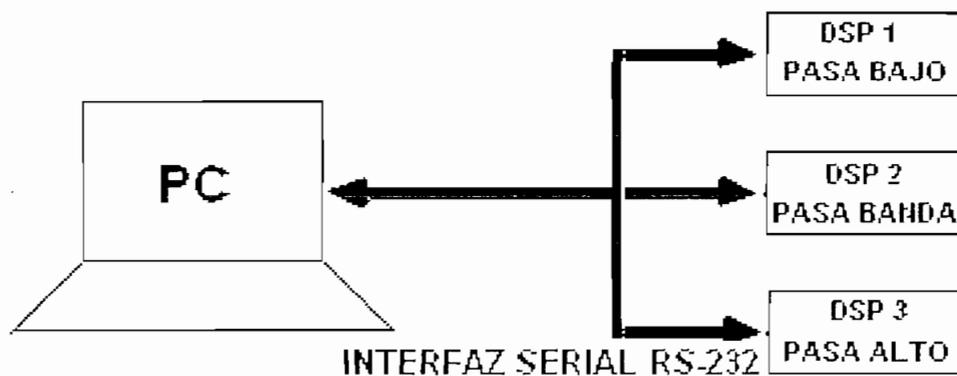


Figura 3.1. Transferencia de coeficientes entre el PC y los DSP's

Para filtrar la señal analógica de entrada, ésta debe ser digitalizada en el CODEC (Codificador-Decodificador), a continuación se trata la señal digital en el procesador DSP y luego se decodifica, la señal filtrada debe ser amplificada para poder ser enviada al sistema de altavoces como se indica en el diagrama de bloques de la figura 3.2.

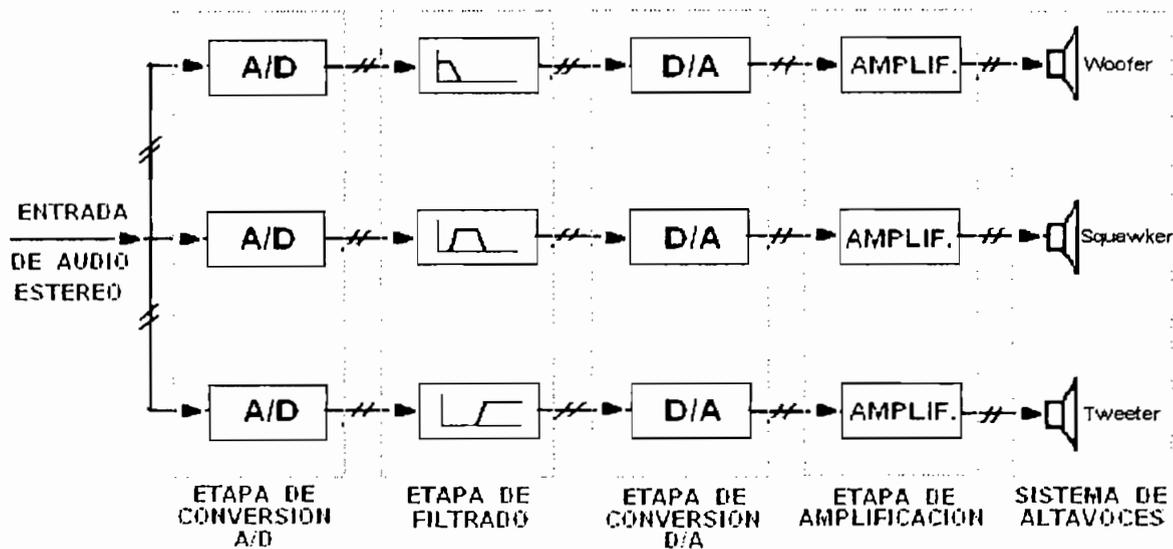


Figura 3.2. Diagrama de bloques de la red crossover de tres vías

3.2.1. MODULO DE DESARROLLO EZ-KIT LITE

Las tres primeras etapas de la red crossover se realizan dentro de la tarjeta EZ-Kit Lite fabricada por Analog Devices que se muestra en el gráfico 3.3. y tiene como características principales:

- © Sistema basado en el procesador digital de señales ADSP-2181 de 33 MIPS¹.
- © CODEC AD1847 estereo de 16 bits que se comunica con el DSP a través del puerto serial 0. Se puede programar la frecuencia de muestreo entre 5.5125 Khz y 48 Khz.
- © Botones de Reset e Interrupción que actúan sobre el DSP.
- © Leds que indican la situación de FL1 en el DSP y de alimentación del módulo.
- © El equipo puede trabajar de forma independiente como conectado a un PC a través del puerto serial 1, que es un interfaz serial RS-232;

¹ MIPS = Millones de instrucciones por segundo

adicionalmente posee diversas líneas de entrada y salida de uso general.

- © Conectores que permiten escoger tamaños de la memoria EPROM, niveles de la señal de entrada (de línea o micrófono)¹.
- © Incluye los programas utilitarios Assembler, Linker, Prom Splitter así como también un simulador.

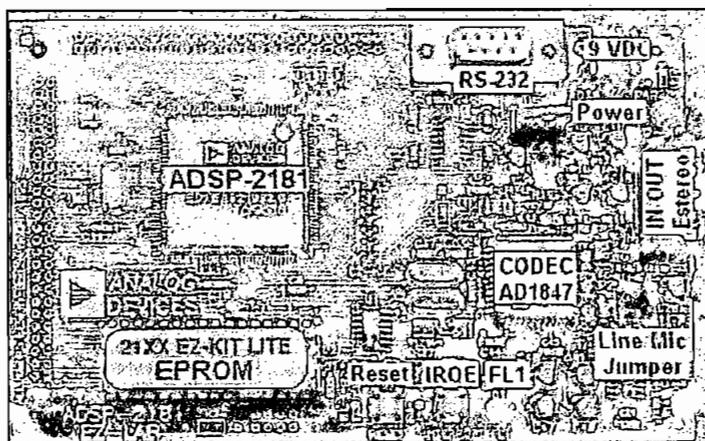


Figura 3.3. Módulo de desarrollo del ADSP-2181

Más características y detalles acerca de este módulo de desarrollo se pueden apreciar en el Anexo 3.

3.2.1.1. Conversión analógica – digital (A/D) y digital – analógica (D/A)

El convertidor Analógico / Digital es aquel que presenta en su salida una señal digital a partir de una señal analógica de entrada realizando las funciones de cuantificación y codificación.

La cuantificación implica la división del rango continuo de entrada en una serie de pasos, de modo que para infinitos valores de la entrada la salida sólo puede presentar una serie determinada de valores, por tanto la cuantificación implica una

¹ El nivel de línea es de hasta 2 V RMS, mientras que el de micrófono es de hasta 20 mV RMS.

pérdida de información que no podemos evitar. En otras palabras se discretiza a la señal en el tiempo mediante una frecuencia de muestreo que por lo menos es el doble de la frecuencia máxima de la señal a digitalizar.

La codificación es el paso por el cual a la señal digital se le asigna un determinado código binario, de modo que las etapas posteriores al convertidor puedan leer estos datos adecuadamente, dicho de otra manera se discretiza la amplitud

El codec AD1847 se basa en una técnica llamada Sigma Delta que le permite una resolución de 16 bits. La característica principal de estos tipos de convertidores es la de trabajar a una frecuencia de muestreo mucho mayor que la de Nyquist pudiendo intercambiar resolución en tiempo por resolución en amplitud; para este codec en particular la frecuencia de sobremuestreo es de 64 veces la frecuencia de muestreo que se está utilizando.

La señal procedente del codec es enviada por el interfaz serial al procesador y tiene un esquema TDM (*Time Division Multiplex*, Multiplexación por división de tiempo).

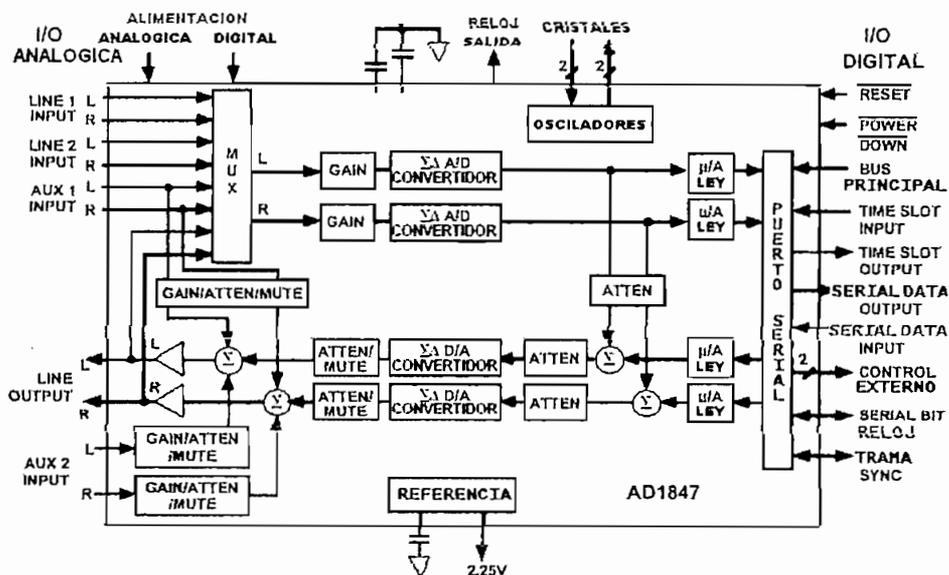


Figura 3.4. Diagrama de bloques del AD1847 [17]

Al convertir una señal analógica a digital se introducen errores de cuantización y de codificación, el efecto del ruido de cuantización en el sistema es limitar la relación de señal a ruido (S/N) obtenible. Sin embargo este efecto puede ser reducido usando mayor cantidad de bits para representación de los datos. En el caso de la representación binaria, la S/N está dada por la siguiente expresión:

$$(S/N) \text{ dB} = 4.8 + 6 \times n$$

Donde n es el número de bits usados para la representación del dato. En el caso de este codec $n = 16$ por lo que nos da una S/N máxima permitida de 100.8 dB. A manera de comparación se puede anotar que en audio de calidad musical se utilizan valores de S/R alrededor de 72dB los cuales pueden lograrse con 12 bits.

El error producido por la codificación resulta de la representación de los coeficientes del filtro con un número limitado de bits. Esta característica tiene el efecto adverso de modificar la respuesta de frecuencia deseada. En un filtro pasabanda por ejemplo, se limita la máxima atenuación posible, de tal forma que se permite el paso de señal en bandas no deseadas. Una solución a este problema consiste en usar más bits para representar los coeficientes del filtro.

3.2.1.2. Procesador digital de señales

El ADSP-2181 es un microprocesador de punto fijo de 16 bits optimizado para el procesamiento digital de señales y otras aplicaciones numéricas, puede realizar 33 MIPS. Este procesador combina la arquitectura básica de la familia ADSP-2100 (tres unidades computacionales, generador de direcciones de datos y secuenciador de programa) con dos puertos DMA, uno interno de 16 bits y otro externo de 8 bits; un temporizador programable, líneas de E/S para señalización y mecanismos avanzados de control de interrupciones. El procesador posee 80 Kb de memoria interna RAM, distribuidas en 16K palabras de 24 bits para instrucciones de programas y 16k palabras de 16 bits.

Las unidades computacionales independientes que tiene el procesador son: la unidad aritmética lógica (ALU), el multiplicador/acumulador (MAC) y la unidad de desplazamiento (Shifter). Cada unidad es capaz de realizar una instrucción en un solo ciclo de máquina, pero solo una de estas unidades puede estar activa por ciclo.

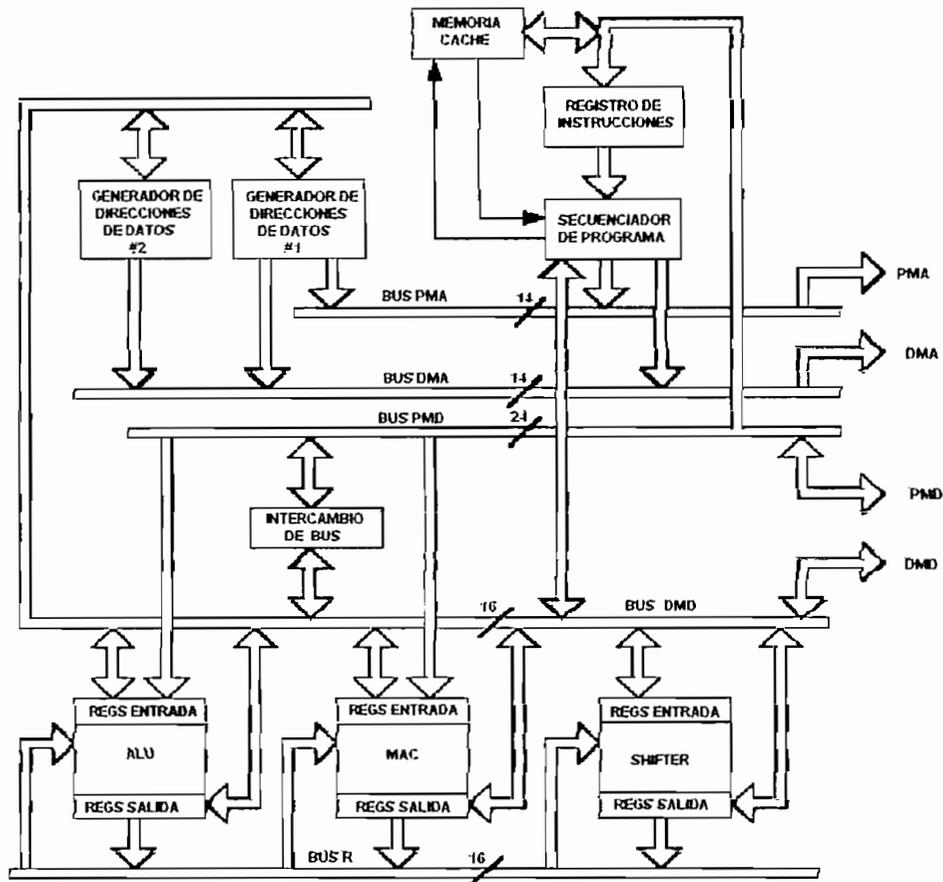


Figura 3.5. Arquitectura Interna del ADSP-2181 [17]

La ALU utiliza operandos de 16 bits y realiza operaciones básicas aritméticas y lógicas, incremento/decremento y valor absoluto. Para lograr el funcionamiento del MAC en un solo ciclo de máquina se integra el multiplicador con el acumulador en un bus principal de datos, el acumulador tiene una palabra de datos de mayor precisión que de los operandos, proporcionando bits extras para evitar el desborde aritmético¹ en el resultado; La MAC incluye un multiplicador con 2 puertos

¹ El desborde aritmético es la generación de un valor más grande que el número representable

de entrada de 16 bits y un puerto de salida de 32 bits para el producto, posee 4 registros de entrada (dos para memoria de datos y dos para memorias de programa), un sumador/restador de 40 bits y un registro de acumulación de 40 bits con 8 bits de desbordamiento.

La unidad de desplazamiento acepta como operando de entrada un registro específico o un registro que contenga el resultado de una operación de desplazamiento previa ALU o MAC; la entrada de 16 bits es desplazada y transferida para un registro de resultados de 32 bits; soporta desplazamientos lógicos o aritméticos de 32 bits para la izquierda o derecha, además de aritmética en punto flotante.

La capacidad de completar varios accesos a la memoria en un solo ciclo de máquina se debe a la incorporación de dos generadores de datos independientes. Este múltiplo acceso permite simultáneamente al procesador mirar las instrucciones de programa y la memoria de datos; además permiten el direccionamiento indirecto y modificación automática de direcciones utilizado normalmente para la aceleración del proceso aritmético.

El secuenciador de programas permite el control de flujo del programa a través de generar la dirección de la siguiente instrucción; además ejecutan saltos condicionales e incondicionales, procesamiento automático de interrupciones, llamados y retornos de subrutinas.

3.2.1.3. Funcionamiento de un filtro de respuesta impulsiva finita

Un filtro FIR es un sistema discreto cuya salida es la suma de un número finito de muestras; son no recursivos porque carecen de lazos de realimentación, por lo que su respuesta contiene solo ceros.

El mecanismo básico de funcionamiento de un filtro FIR se ilustra en la figura 3.6.; los bloques nominados como Z^{-1} representan operadores de retraso unitario, siendo su salida idéntica a la entrada pero desfasada en un período de muestra. Un vector de memoria de datos es utilizado para almacenar los elementos de retraso llamados generalmente "líneas de retardo". En cualquier instante se considera a N el número total de muestras de entrada utilizadas para el cálculo de cada salida, y a $N-1$ las muestras más recientes que se almacenan en las líneas de retardo. Las muestras de entrada están designadas por $x[n-k]$, siendo $x[n]$ el valor actual de la señal de entrada.

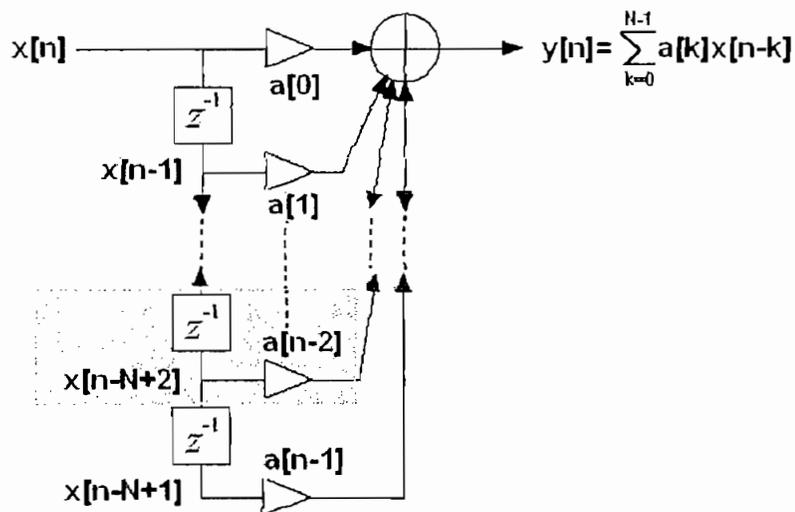


Figura 3.6 Estructura de un filtro FIR

A cada nueva muestra de entrada recibida las muestras almacenadas en la línea de retardo son retrasadas en una posición, descartándose a la más antigua y posicionando la más reciente al inicio de la línea; a continuación la muestra de salida es calculada a través del producto vectorial entre la línea de retraso y sus

coeficientes conforme a la ecuación $y[n] = \sum_{k=0}^{N-1} a[k]x[n-k]$, donde $a[k]$ representa al

vector de coeficientes almacenados en la memoria de programa, $y[n]$ representa la salida del sistema en el instante de tiempo n . La combinación de N secciones como la sombreada en la figura 3.6 especifica un orden $N-1$ del filtro, cada sección

sombreada está compuesta por elementos de retraso, operaciones de multiplicación y suma que junto con los valores escogidos de coeficientes definen las características del filtro.

Para la implementación del sistema cada tarjeta debe ejecutar el filtrado de forma independiente a cada uno de los canales de entrada (izquierdo y derecho), esta independencia es posible por la utilización de 2 buffer circulares de retardo de línea en la memoria de datos y un buffer común de coeficientes del filtro en la memoria de programa. El límite en el orden de un filtro para la implementación en tiempo real es determinado por el tiempo de instrucción del procesador, frecuencia de muestreo y cantidad de otras operaciones necesarias, para el procesador ADSP-2181 cada ciclo de máquina se realiza en 30 nseg y la frecuencia de muestreo utilizada es de 44.1 KHz por lo que $1/(30\text{nseg} \times 44.1\text{KHz})$ da un número máximo de instrucciones de 756; cabe señalar que este número de instrucciones son para los dos canales por lo que cada canal podrá manejar 378 instrucciones, con éste límite de instrucciones se restringe el ancho de la banda de transición a no menos de 370 Hz.

El algoritmo de procesamiento de muestras puede simplificarse en 3 operaciones:

- * Leer las muestras de entrada en la línea de retardo.
- * Calcular el producto punto de salida.
- * Actualizar la línea de retardo.

La figura 3.7 ilustra el producto vectorial entre la línea de retardo y los coeficientes del filtro, se mira que i4 está apuntando a los coeficientes del filtro, originalmente apuntando al inicio del buffer; i2 señala los estados del filtro, al principio apuntaba al estado cero.

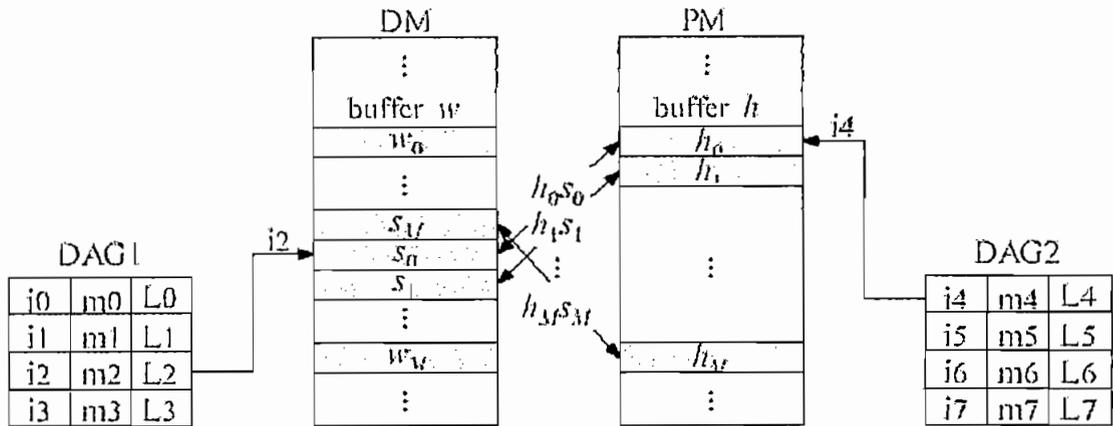


Figura 3.7. Producto vectorial entre la línea de retardo y los coeficientes del filtro

3.2.1.4. Comunicación serial entre la PC y la ADSP-2181

Una de las grandes limitaciones de la ADSP-2181 es la falta de un puerto de comunicaciones serial asíncrono pero es posible el uso de un puerto serial sincrónico de la ADSP-2181 en forma bidireccional a través un programa de simulación UART (*Universal Asynchronous Receiver/Transmitter*, Receptor/Transmisor Asíncronico Universal) para un conector RS-232. El software es desarrollado para proveer las siguientes características:

- * Operación full duplex (independencia en recepción y transmisión).
- * Doble localidad de memoria para transmisión y recepción.
- * Operación asíncronica
- * Programabilidad (posibilidad de escoger entre diferentes velocidades de transmisión y formatos de los datos; 7 u 8 bits de datos y 1 o 2 bits de parada)

En general la configuración del sistema es presentado en la figura 3.8., el procesador ADSP-2181 está conectada a un chip interfaz entre niveles de voltaje

TTL (5 V.) y RS-232 (9 V.), y de este chip se conecta a cualquier dispositivo compatible RS-232.

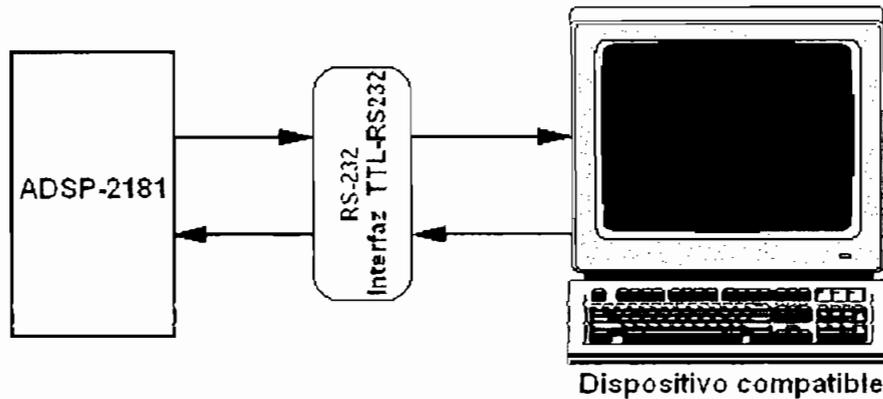


Figura 3.8. Configuración general del sistema de transmisión.

La implementación del hardware para el estándar UART se lo realiza con un chip AD233 que hace de interfaz. Los pines de bandera de entrada (FI, Flag In) y bandera de salida (FO, Flag Out) de el ADSP-2181 son utilizados como líneas independientes de recepción y transmisión, El integrado AD233 es el controlador ideal debido a que no necesita capacitores externos y es alimentado con una fuente de cinco voltios, está incorporado dentro del EZ-KIT Lite y su conexión se muestra en la figura 3.9.^[1]

En la implementación del programa UART se utilizan dos conjuntos separados de memoria para los registros de recepción y transmisión, una localidad es usada por el UART como reloj para los datos de entrada y salida; la otra localidad es usada para leer desde (y escribir a) el UART, de esta manera provee acceso a el UART mientras éste está en operación. Esto permite un flujo de datos continuo.

El programa UART puede manejar una variedad de velocidad de transmisión y formato de datos modificando las características del timer y del shifter en la ADSP-2181. El timer genera interrupciones de una duración igual a tres tiempos de la velocidad de transmisión, suministrando la suficiente resolución del reloj para

manejar asincrónicamente el flujo de datos transmitidos y recibidos. La velocidad de las interrupciones del timer puede ser incrementada para suministrar una resolución adicional del reloj.

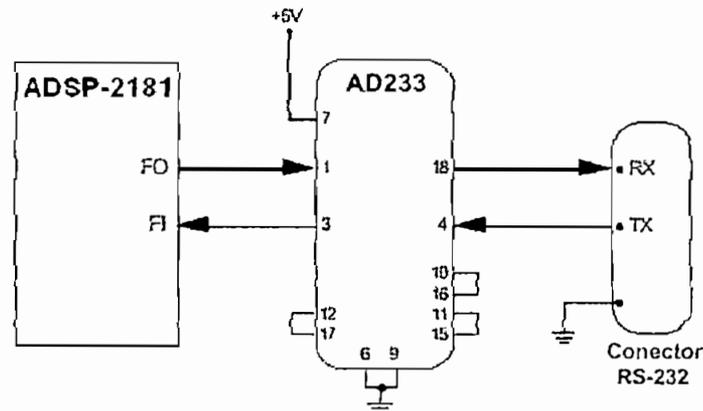


Figura 3.9. Implementación del hardware para la comunicación serial asincrónica¹

Las seis subrutinas que utiliza el programa son las siguientes, y se pueden apreciar en detalle en el anexo 4.

- Rutina de inicialización
- Rutina de interrupción del timer
- Rutina de transmisión de caracteres
- Rutina de recepción de caracteres
- Rutina de habilitar recepción
- Rutina de deshabilitar recepción

La rutina de inicialización debe ser llamada luego de que el sistema se reestableció; la rutina de interrupción del timer es el alma del programa, transmite y recibe datos cuando es necesario. La parte de recepción de la subrutina puede ser deshabilitada por la ejecución de la rutina de deshabilitar recepción, la rutina de interrupción del timer prepara al UART para el uso por:

¹ Los pines 2, 5, 8, 13, 14, 19 y 20 en el AD233 no tienen conexión.

Para la operación en modo automático de velocidad de transmisión se debe conectar en la ADSP-2181 el pin IRQ2 al pin FI.

1. Poner al timer a generar una interrupción para una velocidad apropiada,
2. Configurar los pines FI/FO de SPORT1 (Puerto Serial 1),
3. Fijar banderas para indicar que el UART no está ocupado; y
4. Limpiar cualquier interrupción pendiente y habilitar el timer.

La rutina de transmisión de caracteres espera por cualquier carácter transmitido anteriormente hasta ser completamente enviado, y envía los próximos caracteres. Por su parte la rutina de recepción de caracteres espera hasta recibir completamente los caracteres, entonces adquiere los caracteres y retorna al programa principal. Las rutinas de habilitado y deshabilitado del UART reciben partes de la rutina de interrupción del timer para cumplir con su tarea.

Para la transmisión de datos la temporización empieza al inicio del carácter y termina al final, se añaden dos elementos de señal a cada carácter para indicar al receptor el inicio y fin de este. La rutina de la inicialización (`init_uart`) primero fija los timers para generar las interrupciones en tres veces la velocidad de transmisión, esto proporciona la resolución del reloj suficiente para manejar los datos asíncronos. Inicialmente el contador del timer se fija a cuatro, esto hace saltar al bit de inicio y alinear la bandera de entrada en la mitad de los bits de datos recibidos, como se presenta en el esquema 3.10.

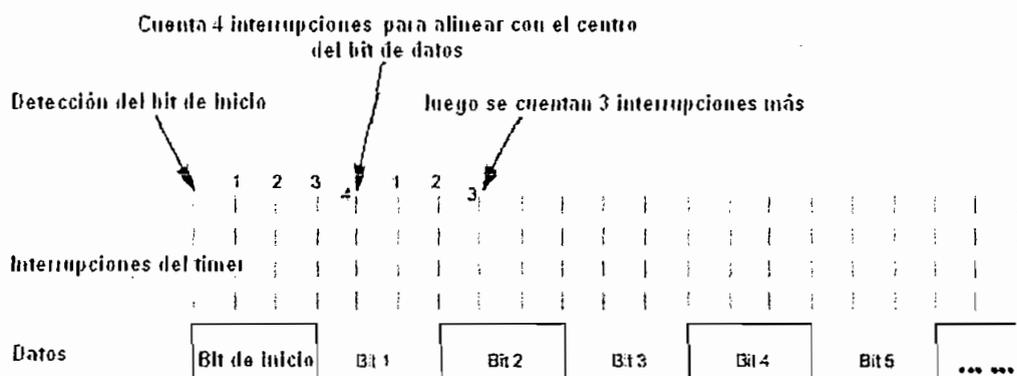


Figura 3.10. Datos recibidos

3.2.2. AMPLIFICADORES Y SISTEMA DE ALTAVOCES

Un amplificador es un circuito con una ganancia de voltaje o corriente. Recibe una señal de algún transductor de capacitación u otra fuente de entrada, y proporciona una versión más grande la señal para algún dispositivo de salida u otra etapa de amplificación. Las características principales de un amplificador de gran señal son la eficiencia de potencia del circuito, la cantidad máxima de potencia de que es capaz de manejar el circuito, y el acoplamiento de impedancias con el dispositivo de salida.

La eficiencia de potencia de un amplificador, definida como la relación de potencia de salida entre la potencia de entrada, mejora entre los amplificadores de clase A¹, hasta los amplificadores clase D.^[6]

El TA8216H es un amplificador de audio dual de clase A que tiene una potencia de salida estereo de 13 vatios por canal, la resistencia de carga que maneja es de 8 ohms y el voltaje de polarización que requiere es de 28 voltios; otras propiedades de este integrado son:

- o Alta potencia de salida
- o Bajo ruido
- o Requiere muy pocos componentes externos
- o Resistencia térmica baja

¹ Etapas de potencia que consumen corrientes altas y continuas de su fuente de alimentación, independientemente de si existe señal de audio o no. Esta amplificación presenta el inconveniente de generar una fuerte y constante emisión de calor. No obstante, los transistores de salida están siempre a una temperatura fija y sin alteraciones. En general, podemos afirmar que esta clase de amplificación es frecuente en circuitos de audio y en los equipos domésticos de gama alta, ya que proporcionan una calidad de sonido potente, a la vez que "fina".

El diagrama de bloques de este circuito integrado se muestra en la figura 3.11; mayores detalles se exponen en el anexo 5.

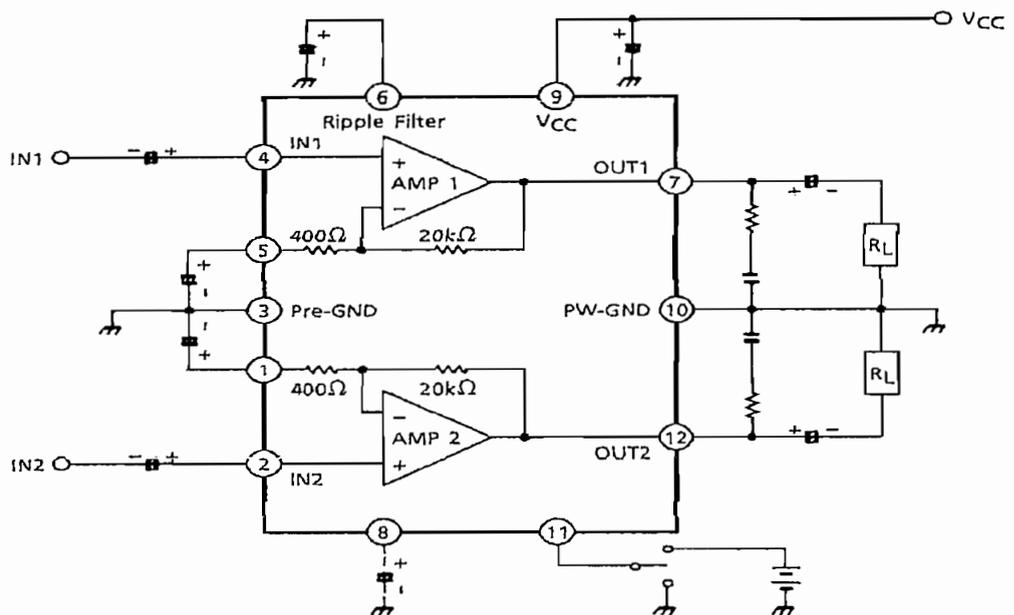


Figura 3.11. Diagrama de bloques del TA8216H

Las cajas acústicas son las que definen el sonido final del equipo aunque la relación entre cada uno de los componentes cumple un papel muy importante, existen dos tipos de caja: las de estante que son relativamente pequeñas y las del tipo torre para colocar sobre el piso.

El volumen de la caja define la capacidad de respuesta a las bajas frecuencias, además la misión es la de cancelar la onda posterior ya que un altavoz al aire libre provoca una onda anterior y un posterior en oposición de fase. Adicionalmente se deben tomar consideraciones como sensibilidad relativamente alta para que los amplificadores trabajen mejor al no tener que dar tanto volumen, que sea bien construida para evitar que vibre y así eliminar posibles ruidos, que siempre perjudican al sonido producido por el transductor. Sin embargo la mejor manera de elegir y adquirir unas buenas cajas, es escuchar y comparar.

Hoy en día lo normal es usar cajas con una abertura (*Bass reflex*, reflector de bajos) que permite aprovechar algo de la onda posterior producida por los altavoces, potenciando en gran medida los graves. Para la ejecución del proyecto se eligió el sistema de altavoces del equipo de sonido LG modelo F-886 mostrado en la figura 3.12. que tiene como características:

- ◆ Altavoces de tres vías con reflectores de bajos
- ◆ Impedancia de 6 ohms
- ◆ Respuesta de frecuencia: 50 Hz a 20000 Hz.
- ◆ Potencia máxima de 150 vatios por canal
- ◆ Salida de 4000 PMPO¹
- ◆ Dimensiones (ancho x alto x profundidad) de 240 x 370 x 313 mm
- ◆ Altavoz de altos: 3 cm.
- ◆ Altavoz de medios: 10 cm.
- ◆ Altavoz de bajos: 20 cm.

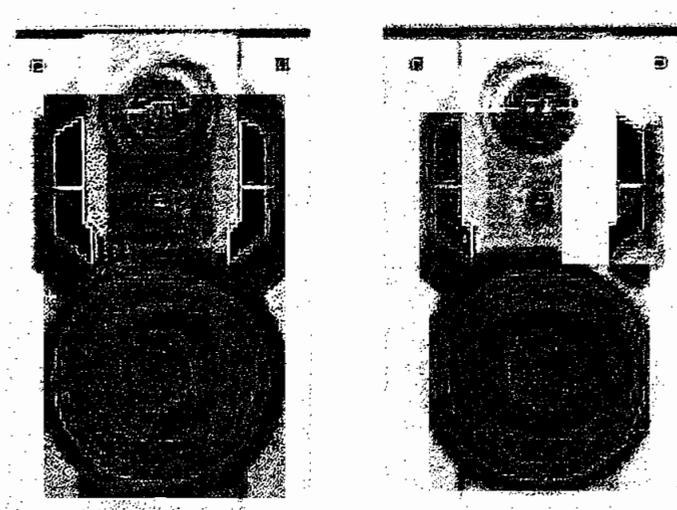


Figura 3.12. Altavoces de tres vías del F-886

¹ *Peak Music Power Output*, Salida de Poder de Picos Musicales y se refiere a la salida de potencia de todos los canales de amplificación sumados, utilizando una señal de prueba de 5000Hz con una impedancia de carga cercana a 0 Ohm un instante antes de que el equipo se queme.

3.2.3. MATLAB

El sistema MATLAB 5.3 (*MATRIX LABORATORY*; The Mathworks, Natick, Massachusetts), es un sistema basado en el cálculo matricial para desarrollar aplicaciones matemáticas y de ingeniería. Posee un único tipo de datos, una matriz o arreglo rectangular de números. Además dispone de un cierto lenguaje de órdenes, que permite realizar los más variados tipos de cálculos, e incluso escribir programas para automatizar tales cálculos (efectuar simulaciones).

Entre otras cosas dispone de un amplio conjunto de funciones para graficación en dos y tres dimensiones; como lenguaje de programación permite construir herramientas re-usables, denominadas archivos M (*M-files*). La Figura 3.13 ilustra la ventana de comando que aparece al ejecutar el programa, mediante esta ventana, es posible impartir las instrucciones al programa y realizar todas las operaciones, incluyendo el llamado a los M-files o a las funciones.

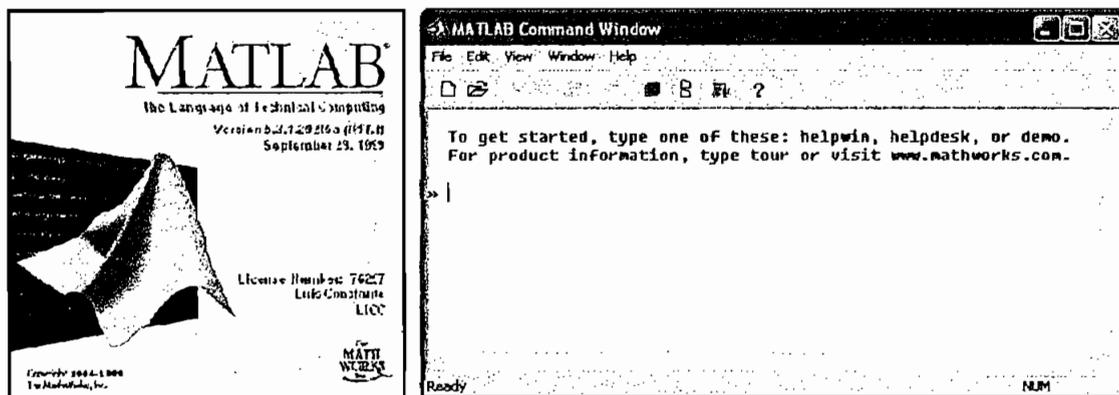


Figura 3.13. Ventana de comando

MATLAB integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían racionalmente, sin necesidad de hacer uso de la programación tradicional. Como caso particular puede también trabajar con números escalares, tanto reales como complejos.

El software dispone de gran capacidad gráfica y viene acompañado de una variedad de herramientas (*Toolbox*) para aplicaciones específicas (i.e., procesamiento de imágenes, control no lineal, análisis espectral, etc.), para el diseño de filtros se presentan en el anexo 6 las funciones que se utilizan.

3.3. PROGRAMAS DESARROLLADOS

Para la implementación del sistema crossover de tres vías se elaboraron programas¹ en MATLAB 5.3 y lenguaje ensamblador para el procesador ADSP-2181. Los diagramas de flujo de dichos programas se presentan de forma detallada en la figura 3.14.

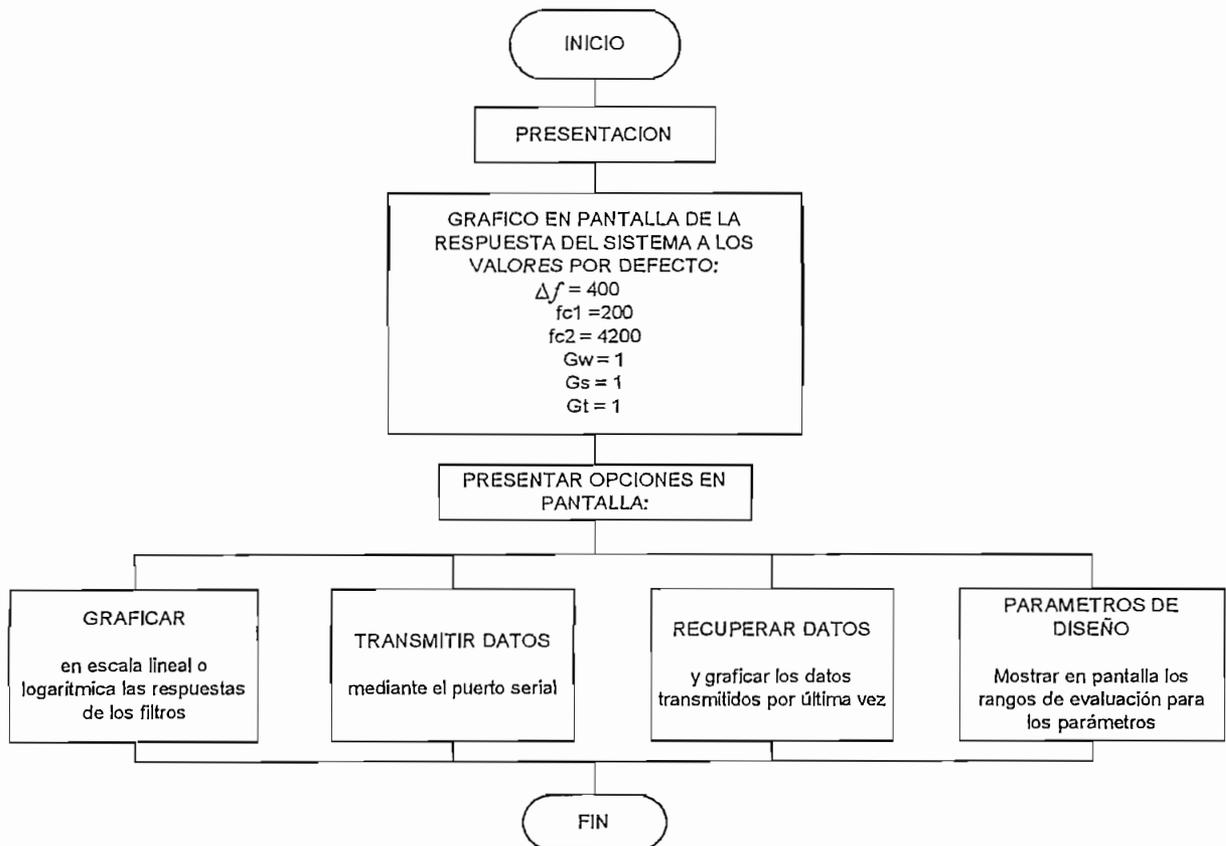


Figura 3.14 (a) Diagrama de flujo. Programa principal implementado en Matlab 5.3

¹ Los programas desarrollados tanto en Matlab 5.3 como en lenguaje ensamblador se adjuntan en el *Manual de usuario*

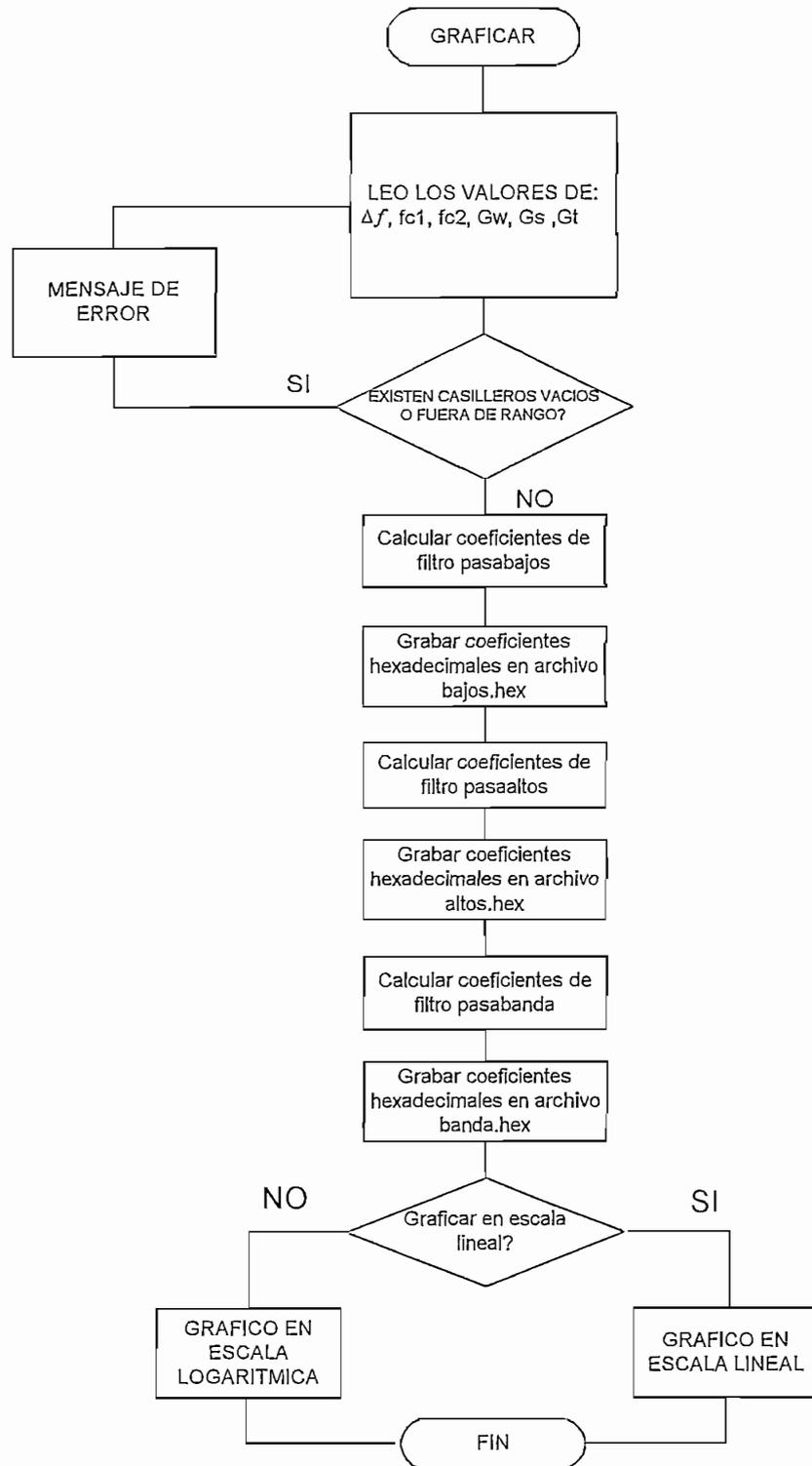


Figura 3.14 (b) Diagrama de flujo. Gráfico de la respuesta de los filtros en Matlab 5.3

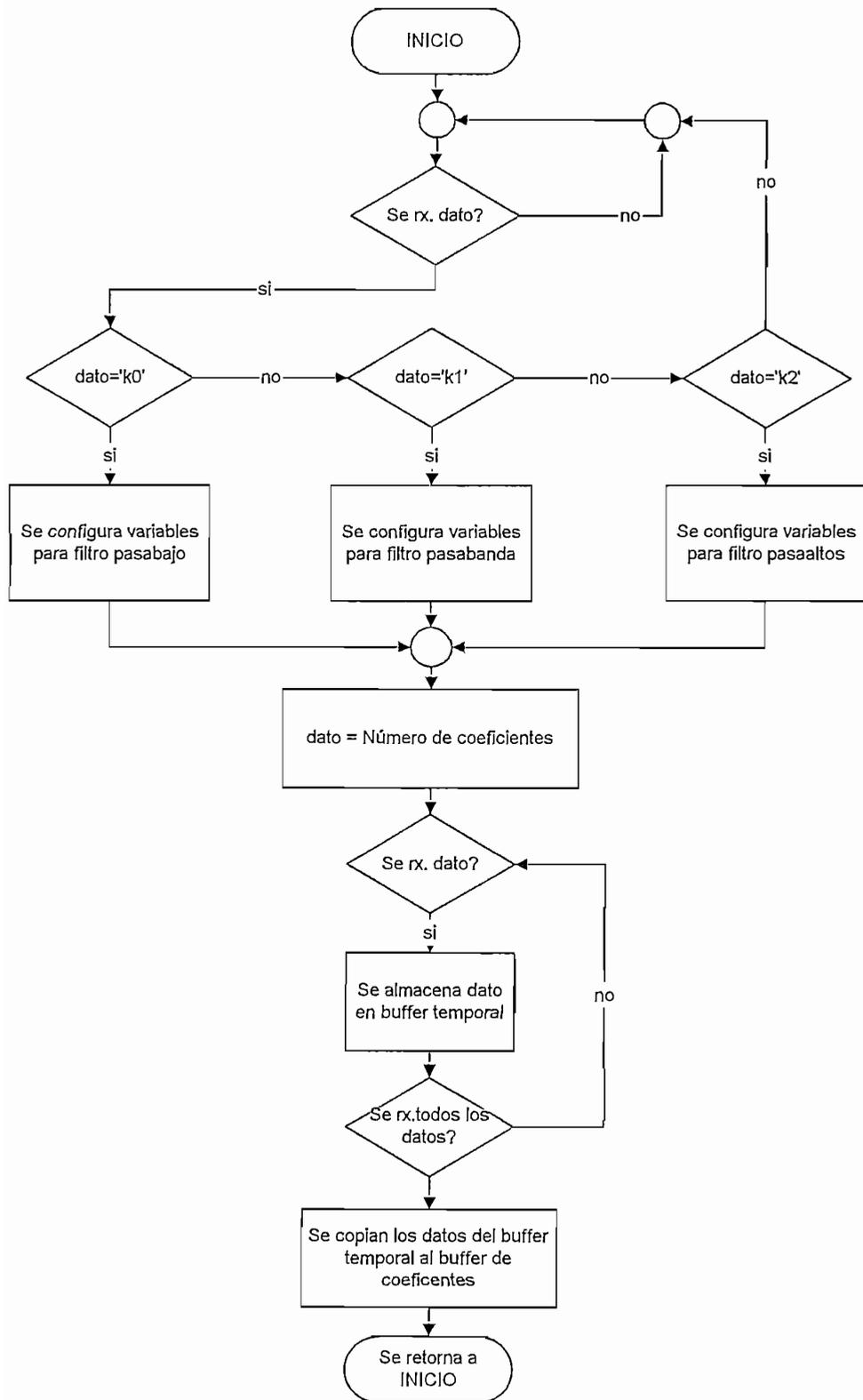


Figura 3.14 (c) Diagrama de flujo. Recepción de datos en la ADSP-2181.

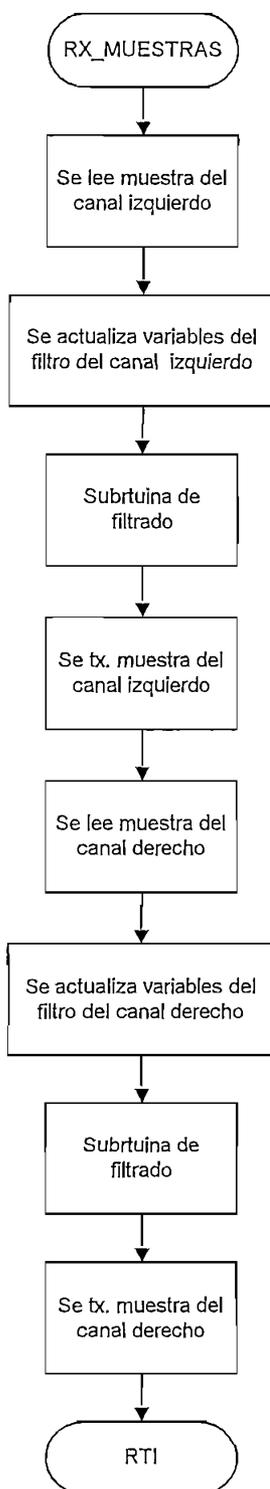


Figura 3.14 (d) Diagrama de flujo. Filtrado en las ADSP-2181

CAPITULO IV

RESULTADOS Y APLICACIONES

Para medir la respuesta del sistema crossover se utilizaron los instrumentos que se muestran en la figura 4.1 y que se detallan a continuación:

- ◆ Generador de audio marca BK PRECISION modelo 3011B
- ◆ Osciloscopio digital marca TEKTRONIX modelo TDS-210
- ◆ Multímetro digital marca FLUKE modelo 112 True RMS

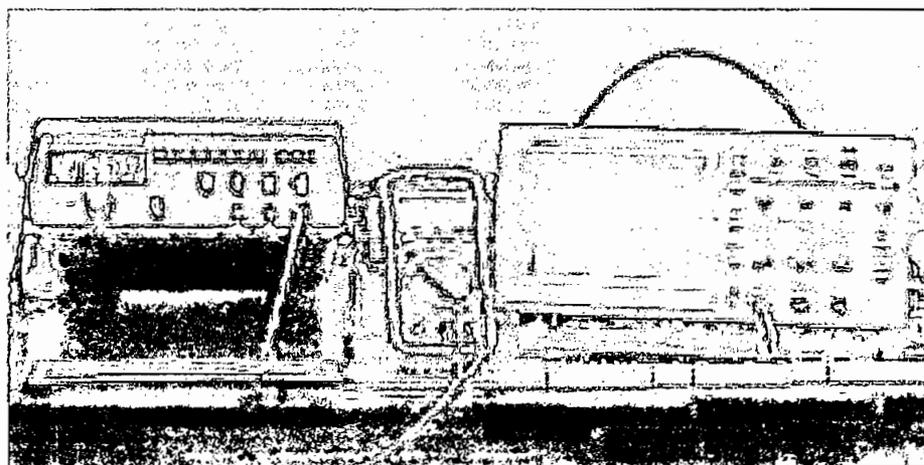


Figura 4.1. Instrumentos de medida y generación

4.1. SISTEMA CROSSOVER DIGITAL

El sistema digital crossover ilustrado mediante el diagrama de bloques (figura 4.2) está constituido por las siguientes etapas:

- ◆ Módulo de procesamiento digital
- ◆ Sistemas de amplificación y altavoces
- ◆ Interfaz serial y software

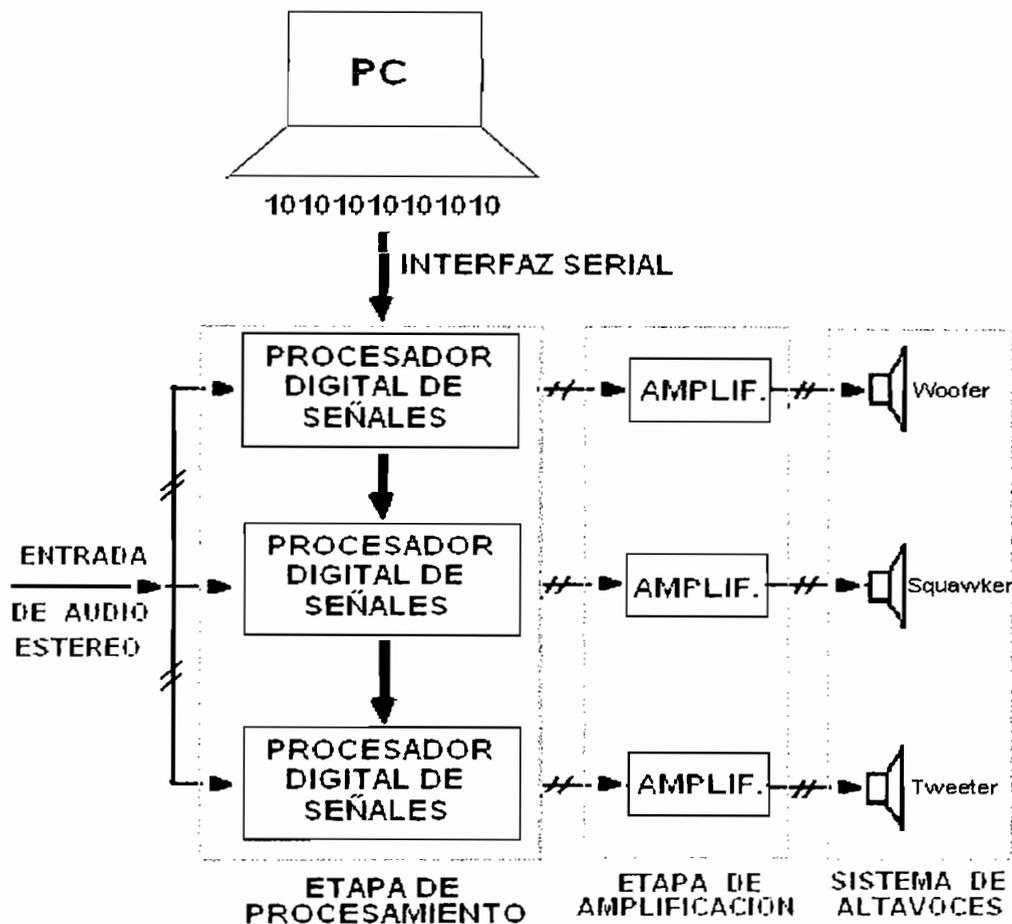


Figura 4.2. Diagrama de bloques del crossover digital

El filtrado se ejecuta en las tres tarjetas ADSP-2181 que se encuentran instaladas dentro de un cofre como se indica en la figura 4.3; la entrada de audio se realiza por dos conectores RCA hembra (canal izquierdo y derecho), las salidas se obtienen a través de seis conectores RCA (woofer, squawker, tweeter para cada canal). Adicionalmente posee una entrada para la comunicación serial con la PC y otra para alimentación.

La señal obtenida del sistema de filtros se envía a los amplificadores que se encuentran dentro de la caja izquierda del arreglo de altavoces, los amplificadores a su vez alimentan a los altoparlantes como se indica en la figura 4.4.

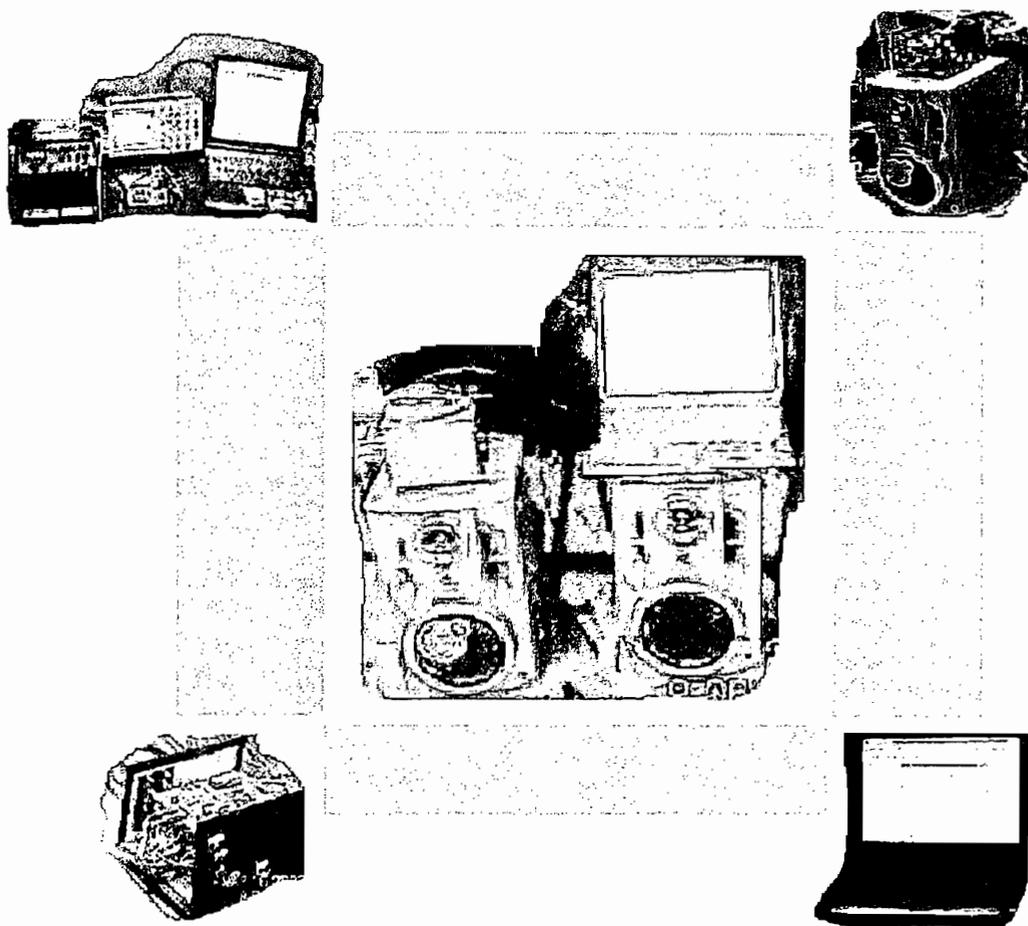


Figura 4.5. Sistema crossover digital

La pantalla de presentación de la aplicación desarrollada en MATLAB 5.3 se indica en la figura 4.6, el programa permite variar parámetros de los filtros, graficación, transmisión y recuperación de datos (figura 4.7).

El valor Δf permite variar el ancho de banda de la zona de transición y debe ser mayor a 350 para no superar el número máximo de instrucciones de 756 para los dos canales a una frecuencia de muestreo de 44.1 KHz como se explicó en la sección 3.2.1.2.

Para escoger el rango entre las cuales deben estar las frecuencias de corte de los filtros (f_{c1} y f_{c2}) se basó en las características de los altavoces comerciales del anexo 1.

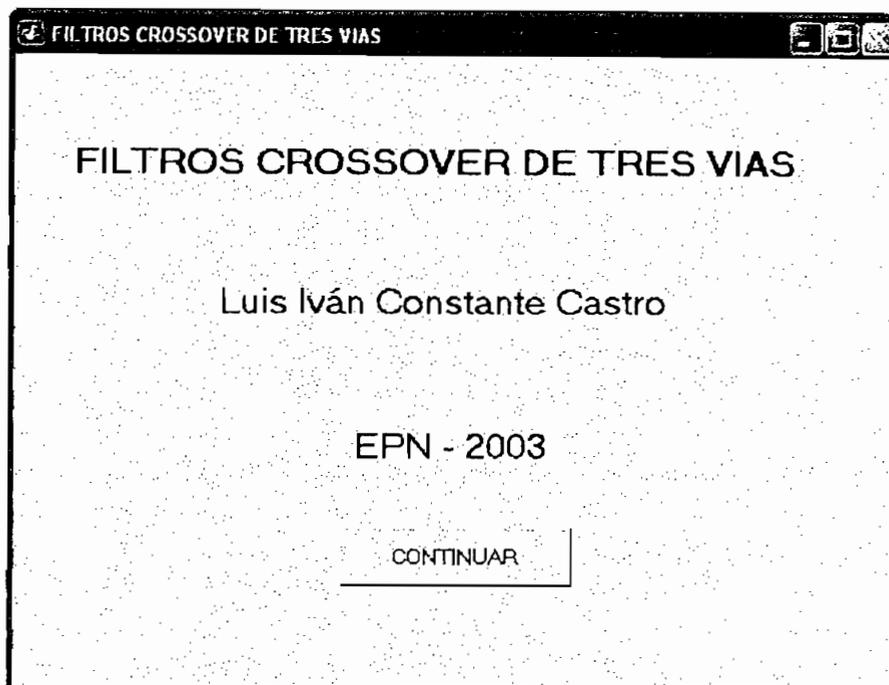


Figura 4.6. Pantalla de presentación

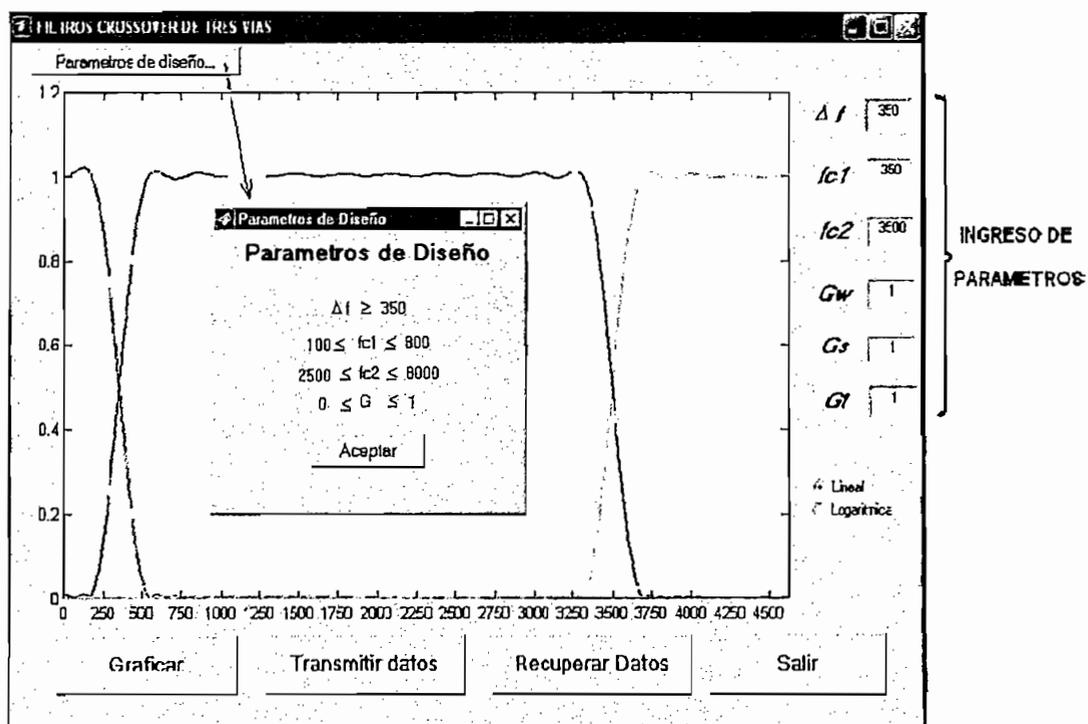


Figura 4.7. Ingreso de parámetros del diseño de los filtros

En el trazo de los filtros se asignan los colores; rojo para los filtro pasa bajos, azul para pasa bandas y verde para pasa altos. Respectivamente G_w , G_t y G_s controlan las ganancias de los filtros y van desde cero hasta uno.

Si el usuario asigna un valor que esté fuera de los rangos aparecerá una señal de error como la de la figura 4.8.

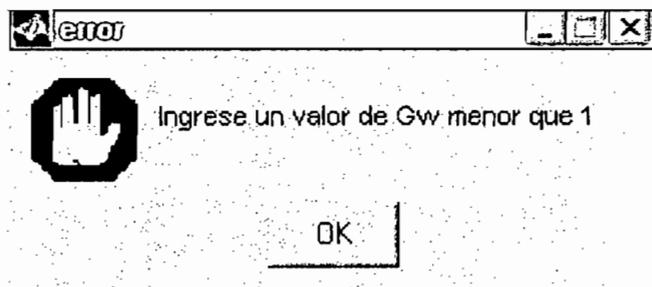


Figura 4.8. Señal de error al ingresar parámetros fuera de rango

Al inicializar el sistema por defecto se toma los valores de:

$$\Delta f = 400 \text{ Hz}$$

$$f_{c1} = 200 \text{ Hz}$$

$$f_{c2} = 4200 \text{ Hz}$$

$$G_w, G_s \text{ y } G_t = 1$$

Estos parámetros se asignan con el fin de que los altavoces no se saturen y como consecuencia de ello se quemen o deterioren. Cuando el usuario ingrese los valores deberá proceder a graficarlos ya sea en escala lineal o logarítmica para acto seguido transmitirlos a los DSPs, un ejemplo de valores tomados es el de la figura 4.9.

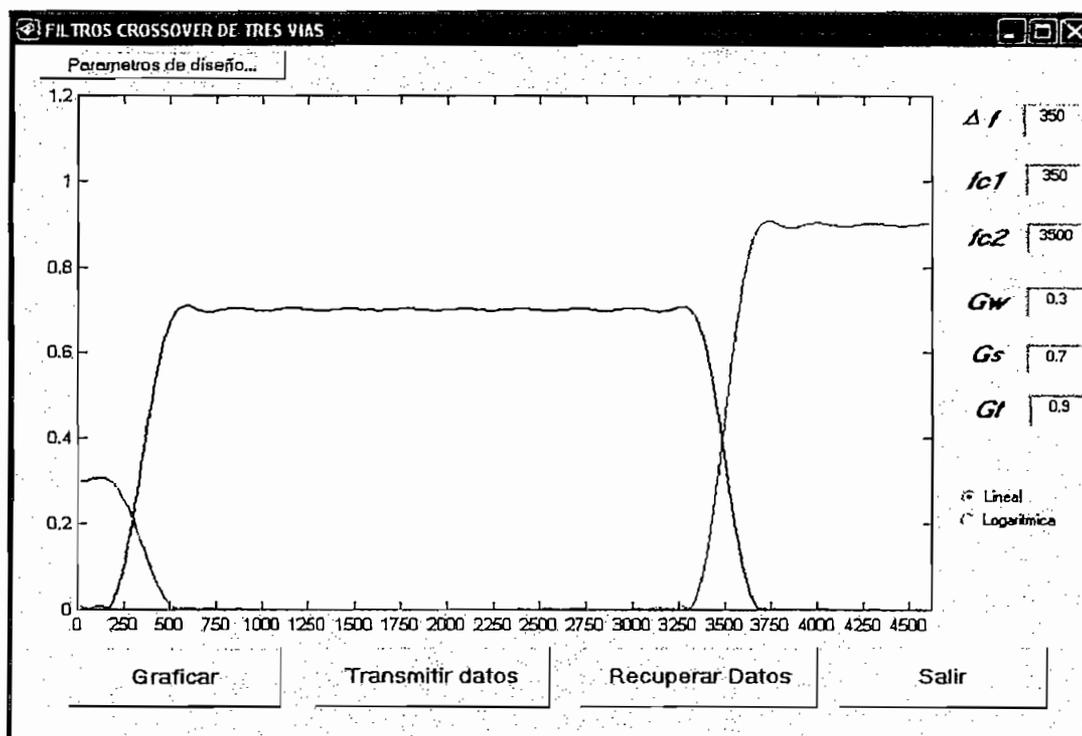


Figura 4.9. Filtros crossover con ganancias diferentes

Si se transmiten los datos graficados, éstos se guardarán en la memoria de la PC de tal forma que el programador siga trazando nuevos sistemas sin transmitirlos; no obstante si en algún momento deseara saber con que arreglo están trabajando las DSP's tiene la opción de recobrar los valores que se transmitieron por ultima vez mediante el botón "Recuperar Datos".

4.2 RESULTADOS EXPERIMENTALES

Para obtener las respuestas en frecuencia de los diferentes filtros se utiliza de referencia una señal senoidal de 1040 mV. pico, frecuencias de corte de 350 Hz y 3500 Hz con un ancho de banda en la zona de transición de 350 Hz. Los demás parámetros son los mismos que se utilizaron para el diseño de los filtros de la sección 2.2.2. Todas las mediciones se las tomó del canal izquierdo de cada uno de los tres filtros (figura 4.10); para comparar los resultados obtenidos entre

un sistema crossover digital y un analógico se recurre a los datos conseguidos en una tesis anterior [7] la cual utiliza filtros de tercer orden de Butterworth. A continuación se presentan los valores.

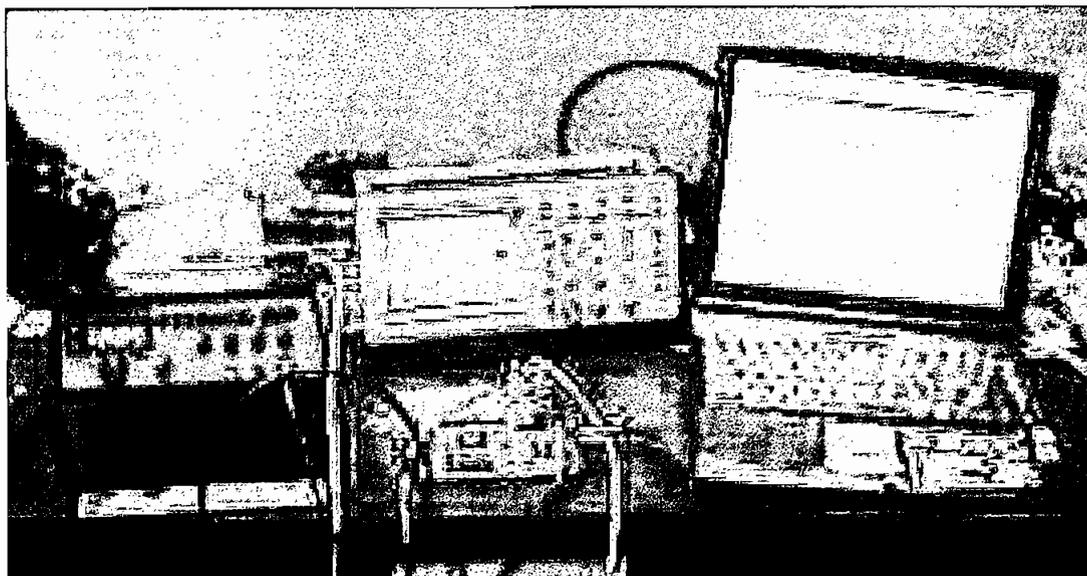


Figura 4.10. Medición de las respuestas del sistema crossover

4.2.1. FILTRO PASABAJOS

<i>DIGITAL</i>	
f (Hz)	mV
28	1040
55	1040
83	1060
110	1060
138	1060
165	1040
193	1020
221	980
248	920
276	820

<i>ANALOGICO</i>	
f (Hz)	mV
20	998
40	998
80	995
160	973
250	890
300	794
325	733
350	659
375	595
400	527

303	740
331	620
358	500
386	400
413	280
441	190
469	110
496	70
524	20
551	10
579	20
606	10
634	0
662	0

450	403
650	149
900	57
1350	17
1700	8
2100	4
2500	2

Primero se trazará la respuesta ideal obtenida en la sección 2.2.2.1 (figura 4.11) conjuntamente con los datos medidos experimentalmente, luego se dibujará las respuestas digital y analógica (figura 4.12).

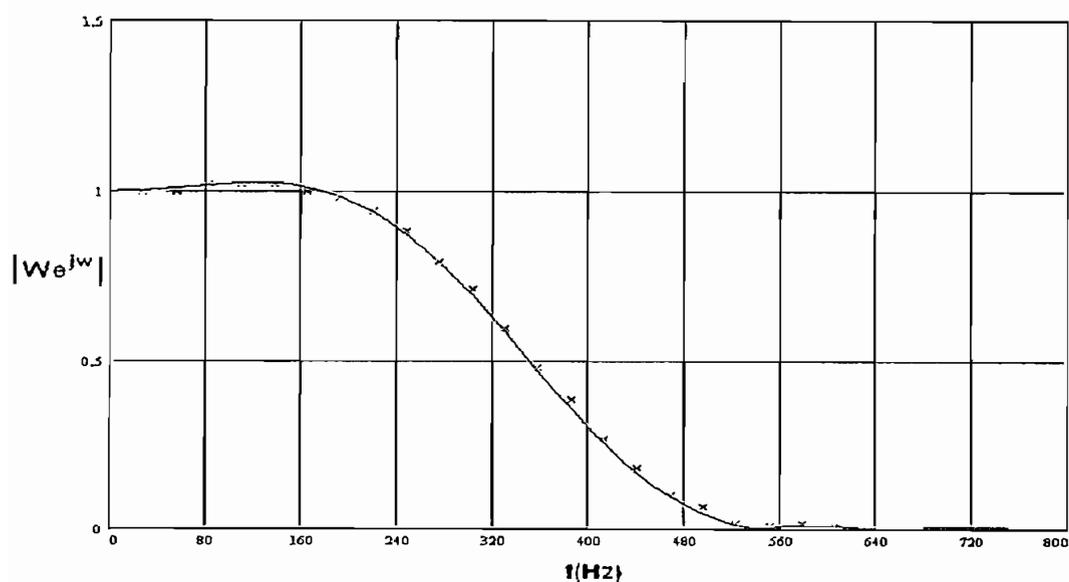


Figura 4.11. Respuesta ideal y experimental del filtro pasa bajos¹

¹ El trazo continuo corresponde a la respuesta de frecuencia teórica y las marcas a los resultados experimentales

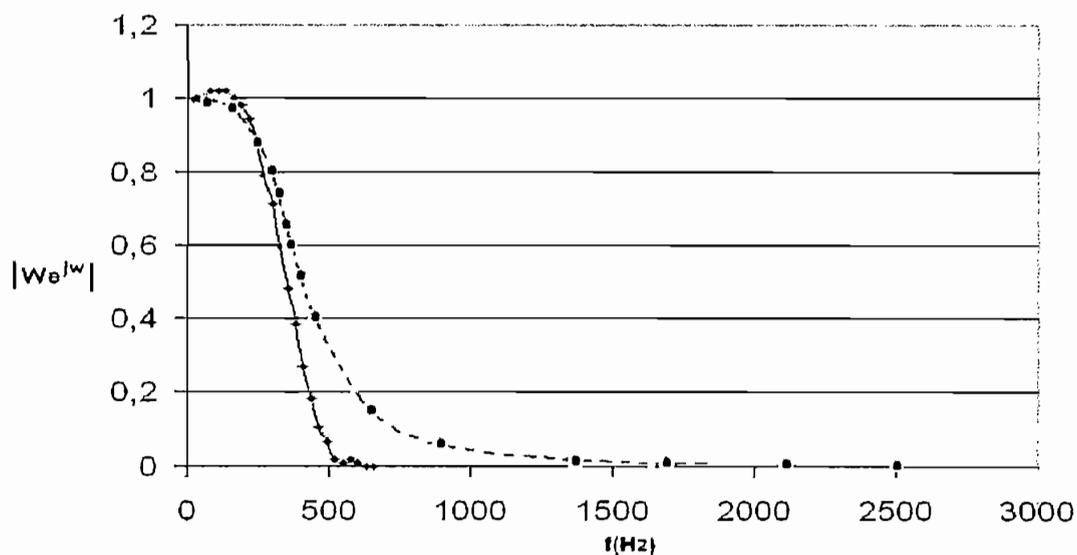


Figura 4.12. Respuestas de los filtros pasa bajos analógico y digital²

4.2.2. FILTRO PASABANDA

<i>DIGITAL</i>	
f (Hz)	mV
28	0
55	0
83	0
110	0
138	0
165	0
193	20
221	40
248	120
276	240
303	320
331	440
358	540

<i>ANALOGICO</i>	
f (Hz)	mV
20	1
40	2
80	13
160	93
250	390
300	498
325	581
350	662
375	722
400	776
450	854
650	948
900	964

² El trazo continuo representa la respuesta experimental del filtro digital y la línea entrecortada al filtro analógico

386	680
413	800
441	880
469	920
496	1000
524	1060
551	1060
579	1060
606	1040
634	1040
662	1040
689	1040
854	1040
1020	1040
1185	1040
1351	1040
1516	1040
1681	1040
1847	1040
2012	1040
2177	1040
2343	1040
2508	1040
2674	1040
2839	1040
3004	1060
3170	1040
3445	880
3473	780
3500	600
3528	440
3556	380

1350	968
1700	965
2100	950
2500	910
3000	814
3250	748
3500	676
3750	603
4000	531
4500	407
5000	314
6000	191
7500	101
10000	43

3583	300
3611	160
3638	80
3666	50
3693	0
3721	0
3749	0
3776	0
3804	0
3831	0
3859	0
3886	0
3914	0
3941	0
3969	0
3997	0

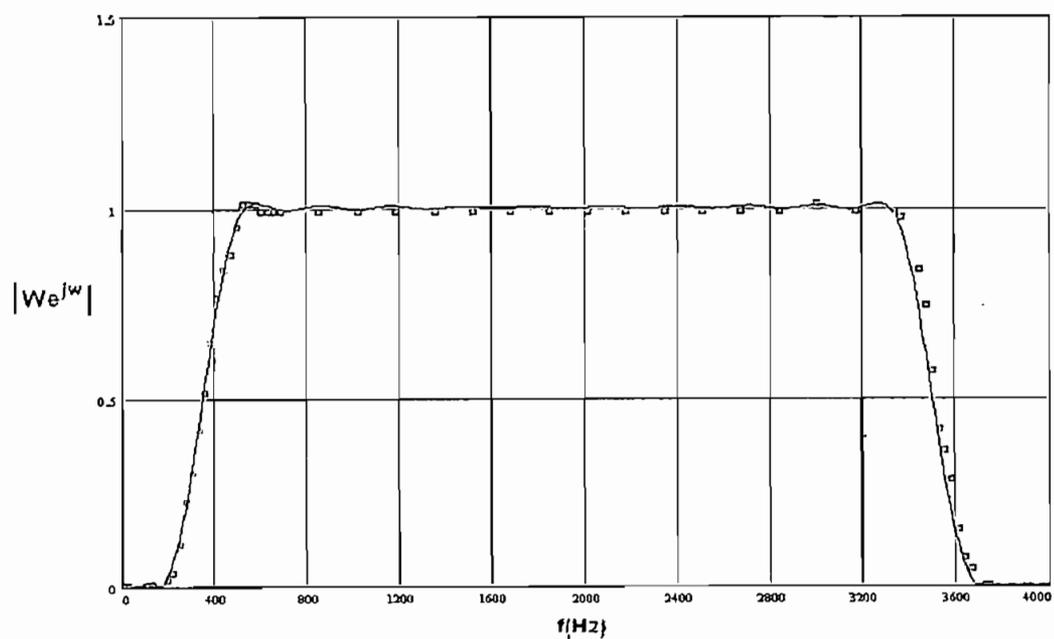


Figura 4.13 Respuesta ideal y experimental del filtro pasabanda

Las graficas 4.13 y 4.14 representan la respuesta ideal y experimental del filtro pasabanda así como la comparación de resultados entre sistema analógico y digital.

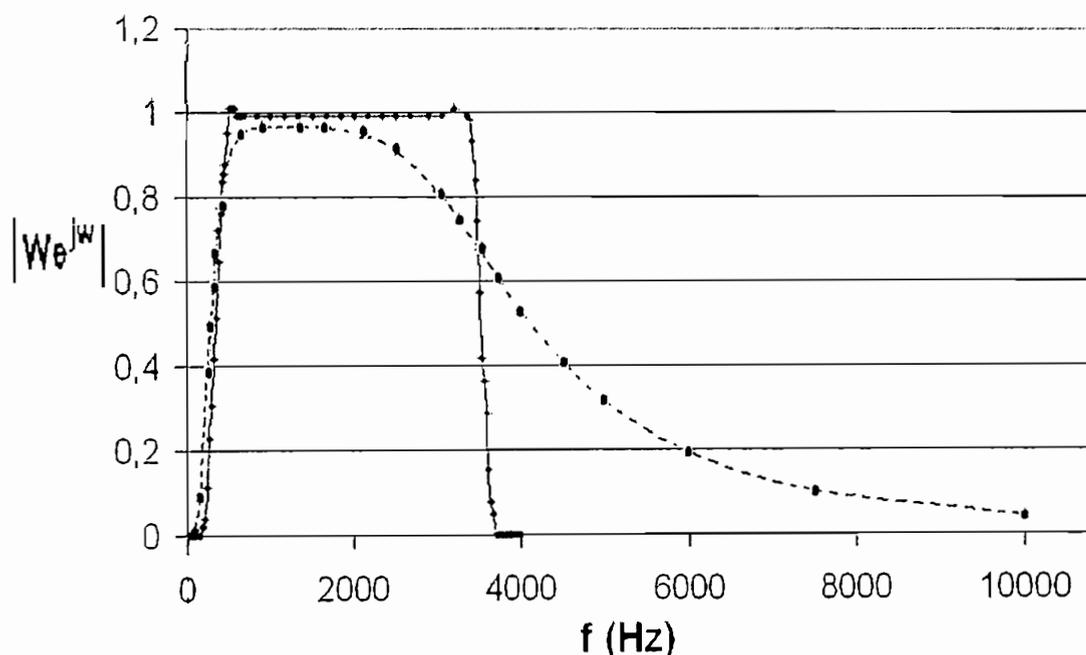


Figura 4.14. Respuestas de los filtros analógico y digital pasa banda

4.2.3. FILTRO PASAALTOS

<i>DIGITAL</i>	
f (Hz)	mV
2756	0
2811	0
2867	0
2922	0
2977	10
3032	10
3087	10
3142	10
3197	10
3252	20

<i>ANALOGICO</i>	
f (Hz)	mV
350	1
375	1.5
400	2
450	3
650	7
900	17
1350	57
1700	110
2100	204
2500	329

3308	20
3363	40
3418	160
3473	380
3528	680
3583	820
3638	980
3693	1040
3749	1060
3804	1040
3859	1040
3914	1040
3969	1040
4024	1040

3000	505
3250	590
3500	660
3750	729
4000	780
4500	851
5000	893
6000	934
7500	959
10000	975

Las graficas 4.15 y 4.16 representan la respuesta ideal y experimental del filtro pasabanda así como la comparación de resultados entre sistema analógico y digital.

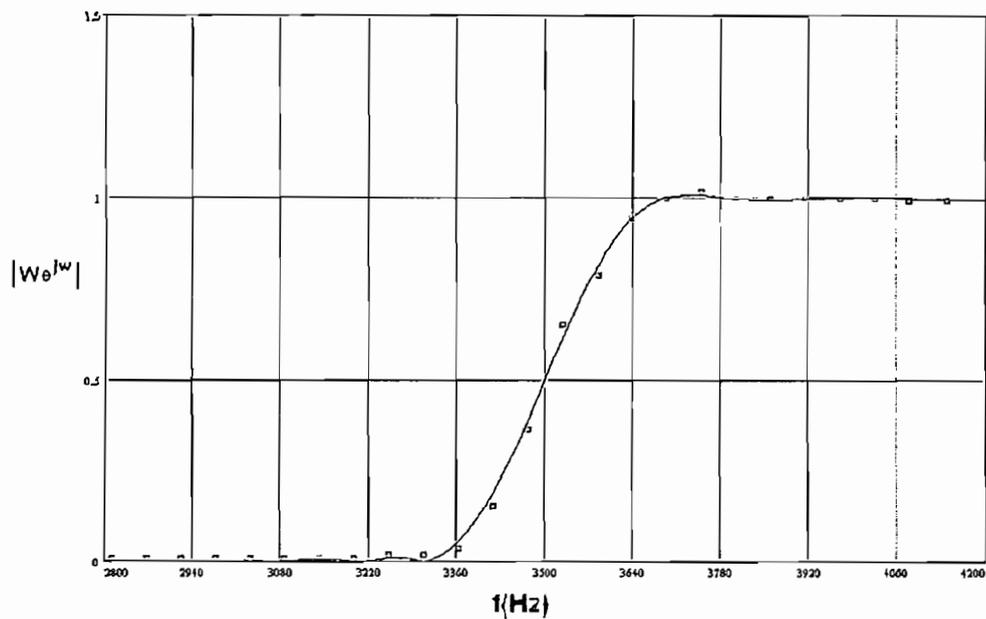


Figura 4.15. Respuesta ideal y experimental del filtro pasa altos

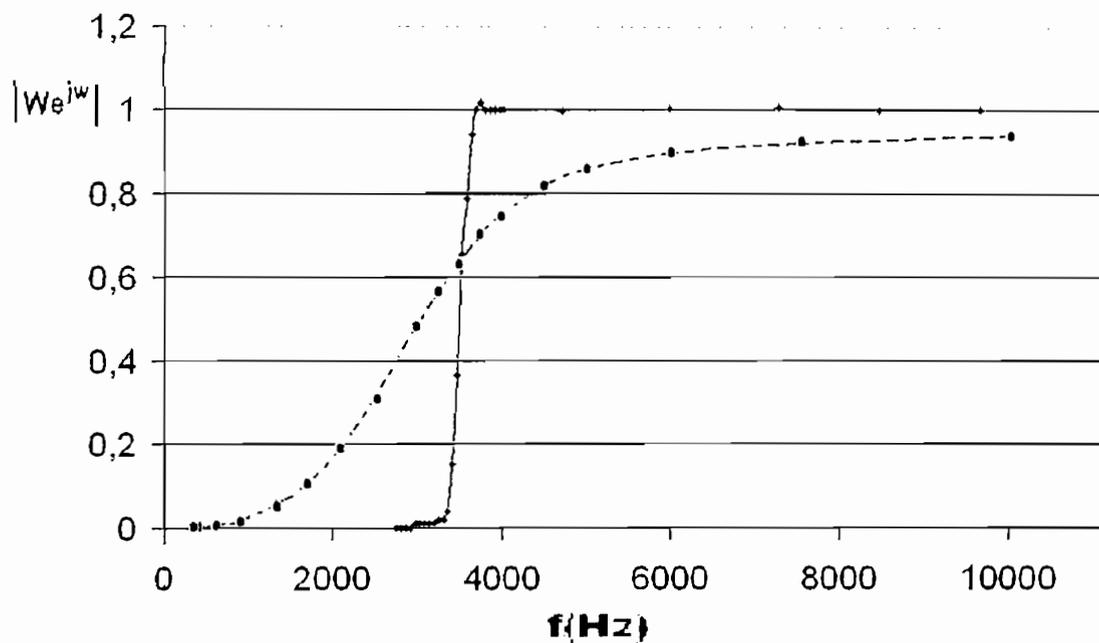


Figura 4.16. Respuestas de los filtros pasa altos analógico y digital

Por último se presenta la respuesta total experimental del sistema.

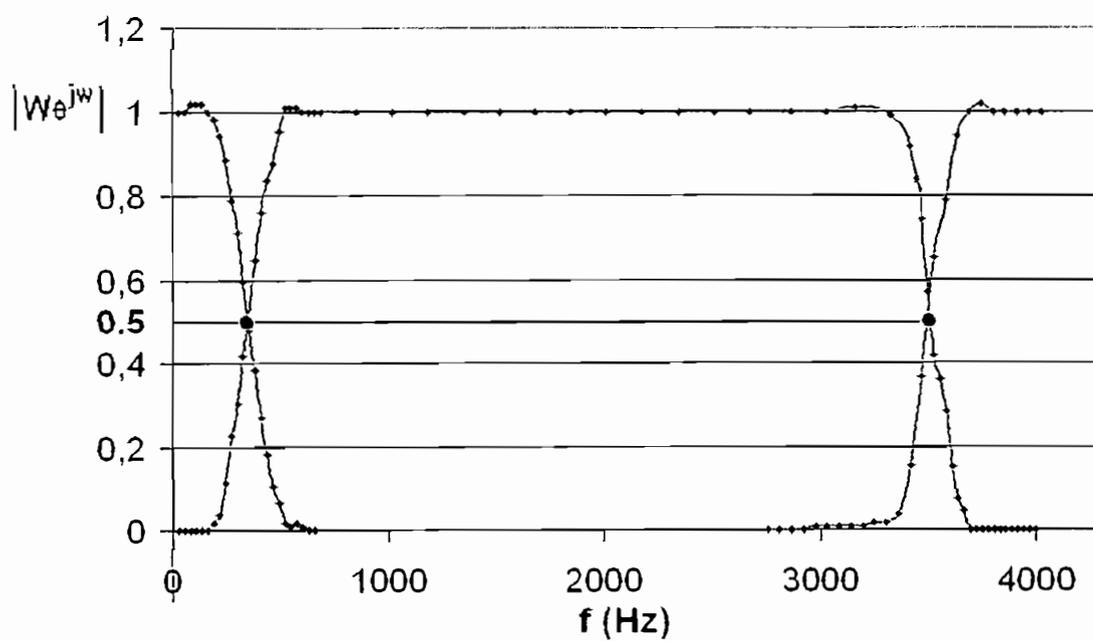


Figura 4.17. Respuesta experimental del sistema

4.2.4. ANALISIS DE RESULTADOS

Al comparar la respuesta ideal de un filtro y la experimental se observa una operación del sistema cercana a la esperada; la diferencia radica básicamente por la falta de precisión de los instrumentos de medición, ya que estos deberían tener más bits que los 16 bits con los que trabaja la ADSP-2181.

Otros aspectos a considerar son los errores producidos por el efecto de utilizar un número limitado de bits (16) para representar los datos. Los ruidos en el sistema básicamente son:

- Cuantización de la señal; este ruido se presenta en la conversión análoga a digital.
- Cuantización de los coeficientes; éste resulta de la representación de los coeficientes de los filtros con un número limitado de bits.
- Redondeo y truncamiento; se producen al descartar los bits menos significativos y el número debe ser cambiado por el más cercano.

En el sistema implementado se ha forzado el tamaño de los coeficientes a 16 bits, ya que son calculados con una precisión de 32 bits con aritmética de punto flotante en el computador.

Los filtros digitales son muy superiores a los analógicos en la rapidez de transición entre la banda de paso y supresión.

En el sistema digital se tiene una respuesta extremadamente plana y uniforme; las frecuencias de corte son exactamente en el punto de -3 dB, mientras que con los filtros analógicos el punto de corte está en -3.5 dB [7]; este problema radica en las limitaciones de los componentes electrónicos.

Existe mayor facilidad y variedad en el cambio de parámetros en los filtros digitales, así como de graficación y recuperación de datos.

4.3 APLICACIONES

El sistema crossover tiene aplicación en todo lugar en el cual se requiera amplificar un sonido, mejorar la acústica de una sala, enfatizar o anular ciertos rangos de frecuencias.

La reproducción en los cines tiene gran importancia ya que según los críticos el audio es el 50% de la película y si no se cuenta con un buen sistema de sonido se pierde dramáticamente gran parte de la película, un arreglo típico para una sala de cine se muestra en la figura 4.18. En este ámbito se debe tener presente el auge que tienen los karaokes y teatros en casa.



Figura 4.18. Distribución de altavoces en una sala de cine

No menos importante es la calidad del sonido en los autos, ya que incluso al dividir el sonido en varias etapas, se puede aprovechar las características físicas de éstos haciendo del portaequipaje una caja de resonancia para mejorar los graves (figura 4.19).

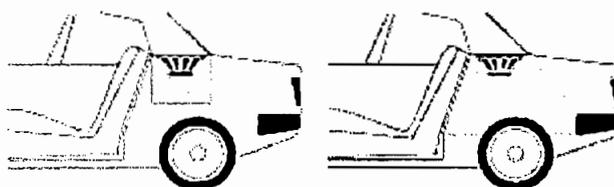


Figura 4.19. Altavoces en un automóvil

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Un altavoz es adecuado para reproducir cierta gama de frecuencias e inadecuado para otras, por lo que para alimentarlo con una banda adecuada se debe incluir un filtro divisor de frecuencia que le permite alcanzar niveles de presión sonora más elevados sin distorsión
- El usar múltiples amplificadores en el filtraje activo presenta una ventaja ante el sistema estándar; porque evita que los fallos, ruidos y distorsiones de una vía afecten a las demás.
- La superioridad de los filtros digitales sobre los analógicos, se debe principalmente a que son fáciles de diseñar e implementar, no les afecta la humedad y temperatura; poseen alta precisión, estabilidad, conmutabilidad, adaptabilidad e inmunidad al ruido.
- El DSP es una herramienta computacional de gran integración que debido a su arquitectura permite realizar en un solo ciclo de máquina operaciones como multiplicación, acumulación y accesos a memoria.
- Un filtro es definido por los coeficientes de la respuesta al impulso. Para obtener los coeficientes de un sistema crossover se deben ingresar las características en frecuencia de los filtros deseados en la aplicación desarrollada en Matlab 5.3. A continuación deben ser transmitidos a la DSP para que puedan ser convolucionados con la señal de entrada digitalizada.
- Los filtros crossover implementados son complementarios, tienen fase lineal combinada plana, además de una buena atenuación en las bandas pasantes

(40 dB); la frecuencia de muestreo que se utiliza es de 44.1 KHz para que abarque toda la zona audible (20 Hz – 20 KHz).

- Los rangos de variación para las frecuencias de corte (f_{c1} , f_{c2}) son los adecuados para utilizar altavoces que se encuentran en el mercado. La elección del ancho de banda para la zona de transición depende de la cantidad de instrucciones que puede manipular la ADSP-2181 por muestra (756 para los dos canales)
- Para transmitir los coeficientes entre el PC y las DSP se simula una comunicación serial asincrónica mediante software. Los datos que salen de la PC llegan simultáneamente a las tres tarjetas ADSP-2181, éstas discriminan los valores que le corresponden al filtro grabado en su memoria para empezar a procesar la señal de entrada.
- El volumen físico de la caja de altavoces es un parámetro importante, si se usa una caja pequeña los graves serán menos potentes que si se utilizase una más grande. Para incrementar los graves se invierte la fase de las ondas reflejadas utilizando cajas con reflectores de bajos.
- Las diferencias entre las sensibilidades individuales de los altavoces seleccionados, incluso las imputables a pequeños desajustes en las tolerancias de fabricación, pueden ser reajustadas gracias a los controles de ganancia del sistema de filtraje activo.
- Los controles de ganancia permiten re-equalizar la señal para adaptar el sistema a las características de la sala, satisfaciendo los diferentes gustos musicales.
- No se mide la respuesta de frecuencia de la caja de altavoces debido a la carencia de: un micrófono omnidireccional que tenga respuesta plana para toda la gama de frecuencias y un lugar adecuado que elimine los efectos de difracción audibles. Adicionalmente se alejaría del objetivo principal del

presente trabajo que es el de mostrar las ventajas del filtraje digital en tiempo real para un sistema crossover de tres vías.

- Se debe recordar que no existe un sistema ideal para todas las circunstancias, sino que depende de las innumerables aplicaciones y de los gustos de cada individuo.

5.2 RECOMENDACIONES

- Para mejorar la calidad del sonido es necesario distribuir los altavoces de acuerdo al diseño y características de la sala, se debe colocar el sistema de altavoces en una posición lo más equilibrada del oyente y a una altura similar a la del oído ya que los altavoces situados cerca del suelo o en las esquinas producen más graves.
- Los altavoces jamás deben colocarse a los dos lados o en las cuatro esquinas de una sala ya que el oyente escuchara varios altavoces a la vez con muchos retrasos de tiempo, si los retrasos superan a los 65 mili segundos se escuchan como ecos.
- Como norma general y por seguridad, se suele considerar que los altavoces deben admitir potencias un 50% mayor que la salida RMS del amplificador, o lo que es igual, dada la potencia máxima admitida por un altavoz, el amplificador asociado debe entregar máximo una potencia RMS del 33% menos.
- Los cables influyen en el sonido, la diferencia es sutil pero determinante en el sonido final. Puede tener efectos muy variados, como disminuir la potencia de agudos, los graves, o restar claridad y definición al sonido. Por lo que se recomienda el uso de cables robustos para bajar la resistencia y para bajar la inductancia lo mejor es trenzarlos.

- Para mejorar la calidad de los bajos se puede pensar en una cuarta vía con la utilización de subwoofer para reproducir frecuencias menores a 120 Hz.

- Si se deseara utilizar un solo módulo de desarrollo en la etapa de filtrado se recomienda una de las últimas versiones producidas por la Analog Devices, la ADSP-21161N que soporta 100 MIPS y frecuencias de muestreo de hasta 96 KHz. Uno de los codecs ideal para ésta aplicación es el AD1836 de la misma fábrica, que tiene 2 conversores analógico a digital y 6 conversores digital a análogo, soportando palabras de 16, 18, 20 y 24 bits.

BIBLIOGRAFÍA

- [1] ANALOG DEVICES; *ADSP-2100 Family EZ-KIT Lite Reference Manual*, 1ª edición, Canadá 1995.
- [2] ANALOG DEVICES; *ADSP-2100 Family User's Manual*, 3ª edición, Canadá 1995.
- [3] BELOVE, Charles; *Enciclopedia de la electrónica ingeniería y técnica*, Grupo editorial Océano, España 1990.
- [4] CUENCA, David, GOMEZ, Eduardo; *Tecnología Básica del Sonido II*, editorial Paraninfo, 2ª edición, España 1999.
- [5] LAWRENCE, Kinsler y otros; *Fundamentos de Acústica*, editorial Limusa, 1ª edición, México 1991.
- [6] MOMPIN, José; *Manual de alta fidelidad y sonido profesional*, editorial Marcombo, España 1991.
- [7] MUÑOZ, Fernando; *Análisis del filtro de tercer orden de Butterworth y aplicación en un crossover electrónico de tres vías de dos canales con frecuencias de corte variables*, EPN 1991.
- [8] OLEAS, Fausto, QUINTANA, Wilson; *Diseño e implementación de filtros FIR con parámetros variables en tiempo real utilizando la ventana de Kaiser*, EPN 2002.
- [9] OPPENHEIM, Alan, SCHAFER, Ronald W.; *Tratamiento de señales en tiempo discreto*, 2ª edición, editorial Prentice Hall, España 2000.

- [10] ORFANIDIS, Sophocles J.; *Digital Signal Processing Laboratory*, Rutgers University, USA 2001.
- [11] PROAKIS, John G., MANOLAKIS, Dimitri; *Tratamiento digital de señales*, 1998.
- [12] RUIZ, Vasallo Francisco, *Manual de baffles y altavoces*, editorial CEAC, España 1987.
- [13] SOLIMAN, Samir S., SRINATH, Mandyann; *Señales y sistemas continuos y discretos*, 2ª edición, 1999.
- [14] TRIBALDOS, Clemente; *Sonido profesional*, editorial Paraninfo, España 1999.
- [15] VASQUEZ, Fernando; *Arquitectura del ADSP-2181*, EPN 1997.
- [16] WATKINSON, John; *Audio digital*, Editorial Paraninfo, 1ª edición, España 1996.
- [17] www.analog.com
- [18] www.fceia.unr.edu.ar
- [19] www.hacermúsica.com
- [20] www.ii.uam.es
- [21] www.redeya.com
- [22] www.speakercity.com
- [23] www.mathworks.com

ANEXO 1

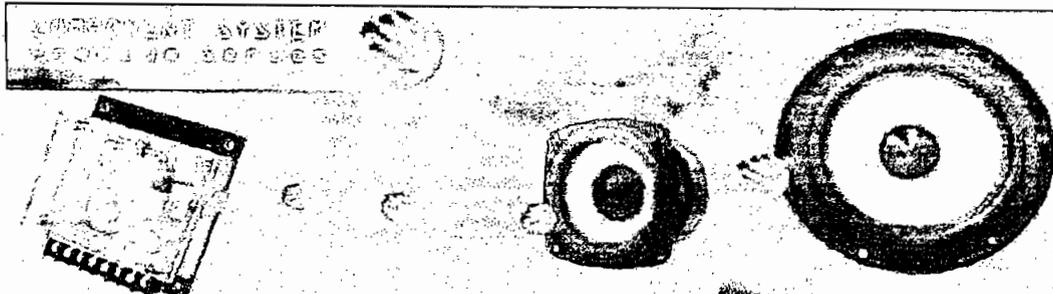
CARACTERISTICAS DE ALTAVOCES COMERCIALES

Pioneer
BETTER
CAR AUDIO



POLE POSITION

KENWOOD



MODELO		PMH-34	PMH-35	PMH-36	T	PMH-53
WOOFER Ø	cm	10	13	16.5	H	16.5
MEDIO Ø	cm	—	—	—	R	10
TWEETER	cm	2.5	2.5	2.5	E	2.5
IMPEDANCIA	ohm	4	4	4	E	4
POTENCIA RMS	W	50	60	75		75
POTENCIA PICO	W	150	180	200	M	200
SENSIBILIDAD	dB	88	89	90	O	90
RESP. FRECUENCIA	Hz	70-22k	60-22k	55-22k	D	55-22k
FREC. DE CORTE	Hz	4500	4200	4200	E	

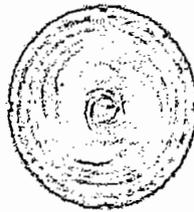


KFC-WS253

Imán de estroncio de gran tamaño (5-11/16 x 5-11/16 x 13/16 pulgadas)

• Potencia máxima 600 W • Potencia nominal 150W • Altavoz de graves de cono P.P. moldeado con inyección de margarita • Respuesta de frecuencia: 28-800 Hz • Diseño para envoltura tipo caja pequeña sellada

600W MAX



KFC-W3003

Altavoz de sub-graves componente de 12"

• Potencia máxima 600W • Potencia nominal 160W • Altavoz de graves de cono P.P. moldeado con inyección de margarita • Respuesta de frecuencia: 25-800 Hz

600W MAX



KFC-W2503

Altavoz de sub-graves componente de 10"

• Potencia máxima 500W • Potencia nominal 130W • Altavoz de graves de cono P.P. moldeado con inyección de margarita • Respuesta de frecuencia: 28-800 Hz

500W MAX

seas

Tweeters serie Excel

Ref	Modelo	D mm	Z	VAS	FS	Qts	Sens.	Rango freq.	Material	Precio
<u>E0006</u>	T25CF001	25	6	-	750	-	90 db	2000-25000	Tela	87,50€
<u>E0011</u>	T25CF002	25	6	-	500	-	88 db	2000-25000	Tela	197,00€
<u>E0036</u>	T25C003	25	6	-	750	-	92 db	2000-25000	Tela	87,50€

Woofers serie Excel

Ref.	Modelo	D cm	Z	VAS	FS	Qts	Sens.	Rango freq.	Material	Precio
<u>E0021</u>	W12CY001	12	8	5,80	46	0,32	85 db	60-3500	Magnesio	152,00€
<u>E0015</u>	W15CY001	15	8	13,5	37	0,34	86 db	50-3000	Magnesio	154,00€
<u>E0016</u>	M15CH001	15	8	10,9	54	0,30	90 db	150-3000	Polipropileno	202,00€
<u>E0002</u>	W17CY-001	17	8	39,20	35	0,42	89 db	38-4000	Fibra Vidrio	126,00€
<u>E0005</u>	W17E-001	17	8	31,80	35	0,35	87,5db	40-3000	Fibra Vidrio	136,00€
<u>E0003</u>	W17EX-001	17	8	27,40	37	0,32	89 db	40-3000	Fibra Vidrio	139,00€
<u>E0018</u>	W18E001	18	8	37,0	31	0,34	86,5db	40-2500	Magnesio	166,00€
<u>E0017</u>	W18EX001	18	8	37,0	31	0,24	88 db	40-2500	Magnesio	171,00€
<u>E0004</u>	W21EX-001	21	8	63,80	31	0,41	91 db	30-3000	Papel	155,00€
<u>E0022</u>	W22EX-001	22	8	75,00	27	0,34	90,5db	30-2000	Magnesio	185,00€
<u>E0026</u>	W26FX-001	26	8	161,00	20	0,35	87db	20-1000	Magn.-Alum.	288,00€

Tweeters serie estandar

Ref.	Modelo	D mm	Z	VAS	FS	Qts	Sens.	Rango freq.	Material	Precio
<u>H 414</u>	19TAF/G	19	8	-	1700	-	88 db	4000-20000	Aluminio	30,00€
<u>H 561</u>	19TAF/D	19	8	-	1700	-	88 db	4000-20000	Aluminio	30,00€
<u>H 532</u>	19TAFD/G	19	8	-	1700	-	89 db	4000-20000	Aluminio	30,00€
<u>H 737</u>	19TFF1	19	8	-	1700	-	87 db	4000-20000	Tela	27,00€
<u>H 398</u>	25TAF/G	25	6	-	1400	-	90 db	3000-25000	Aluminio	35,00€
<u>H 534</u>	25TAF/D	25	6	-	1400	-	90 db	3000-25000	Aluminio	35,00€
<u>H 569</u>	25TAF/DTV	25	6	-	1600	-	88,5db	3000-25000	Aluminio	35,00€
<u>H 400</u>	25TAC/G	25	6	-	660	-	91 db	2000-25000	Aluminio	41,00€
<u>H 417</u>	25TAC/GW	25	6	-	660	-	91 db	2000-25000	Aluminio	45,00€
<u>H 535</u>	25TAC/D	25	6	-	660	-	91 db	2000-25000	Aluminio	40,00€
<u>H 537</u>	25TAFCD	25	6	-	1200	-	90 db	2000-25000	Aluminio	38,00€
<u>H 457</u>	25TFF	25	6	-	1400	-	91 db	3000-25000	Tela	38,00€
<u>H 519</u>	25TFFC	25	6	-	1200	-	90 db	2000-25000	Tela	38,00€
<u>H 614</u>	25TAFNG	25	6	-	1800	-	89 db	3000-25000	Aluminio	33,00€
<u>H 623</u>	25TAFNQG	25	6	-	1800	-	89 db	3000-25000	Aluminio	33,00€



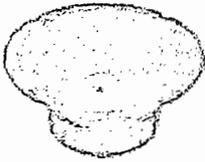
Ph 25

Potencia admisible: 60 w
 Diámetro nominal: 25 mm
 Rango de Frecuencia: 2-22 KHz
 Sensibilidad 1w 1m: 95 dB



PI 500

Potencia admisible: 150 w
 Diámetro nominal: 125 mm 5.25"
 Rango de Frecuencia: 60-7.000 Hz
 Sensibilidad 1w 1m: 90 dB



PI 650

Potencia admisible: 200 w
 Diámetro nominal: 165 mm 6.5"
 Rango de Frecuencia: 50-6.000 Hz
 Sensibilidad 1w 1m: 90 dB



PI 690

Potencia admisible: 240 w
 Diámetro nominal: 6x9"
 Rango de Frecuencia: 45-6.000 Hz
 Sensibilidad 1w 1m: 90 dB



PI 502

Potencia admisible: 150 w
 Diámetro nominal: 125 mm 5.25"
 Rango de Frecuencia: 60-20.000 Hz
 Sensibilidad 1w 1m: 90 dB



Ast 05

Potencia admisible: 100 w
 Diámetro nominal: 42 mm 1.5"
 Rango de Frecuencia: 3-20 kHz
 Sensibilidad 1w 1m: 105 dB



Ast 09

Potencia admisible: 150 w
 Diámetro nominal: 90 mm 3.5"
 Rango de Frecuencia: 2.5-22 kHz
 Sensibilidad 1w 1m: 107 dB

ANEXO 2

CROSSOVER COMERCIALES



FDS-310

Filtro divisor 2 vías estéreo o 3 vías mono

El FDS-310 es un sistema divisor de frecuencias variables de dos vías estéreo o tres vías mono construido sobre nuestros filtros Linkwitz-Riley de alta calidad de 24dB/octava. Cada banda de frecuencia tiene su propio control de nivel, control de frecuencia, cambio de polaridad, botón de mute, LED de presencia de señal y aviso de picos. Las posibilidades a través de puentes internos incluyen la ecualización para bocinas de directividad constante y enlace de unidades mono.

La gama de frecuencia en el modo estéreo de dos vías es de 180Hz-2kHz, y en el modo mono de tres vías de 180Hz-2kHz en medios-graves y de 800Hz-9kHz en medios-agudos. Las entradas y salidas están electrónicamente balanceadas con conectores TRS de 1/4" y XLR.

FDS-318

Filtro divisor 2/3/4 vías estéreo con cortes variables

El FDS-318 es un filtro divisor variable multimodo, conmutable entre operación estéreo de 3 y 4 vías, con modos de superposición seleccionable pasa-altos y pasa-bajos más unos modos de biamplificación de cuatro canales. Las funciones incluyen baja frecuencia mono y corrección de ecualización de trompetas de directividad constante, indicadores LED de presencia de

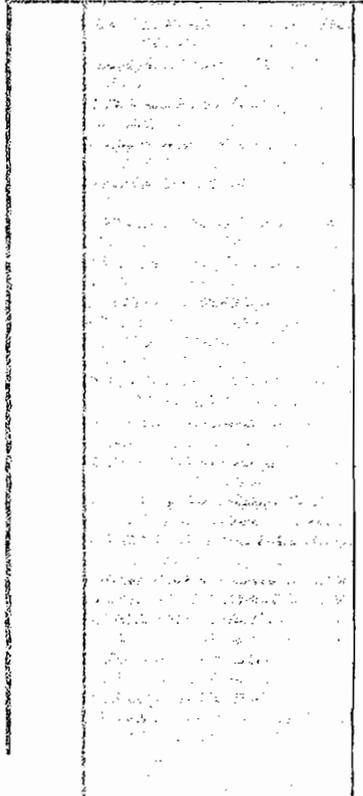
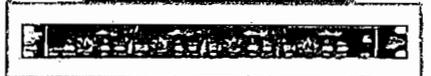
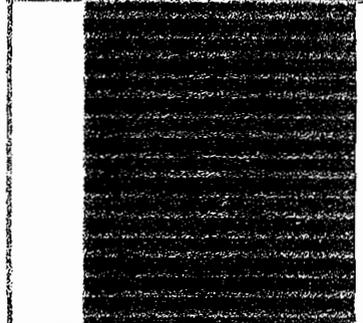
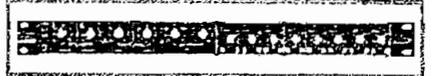
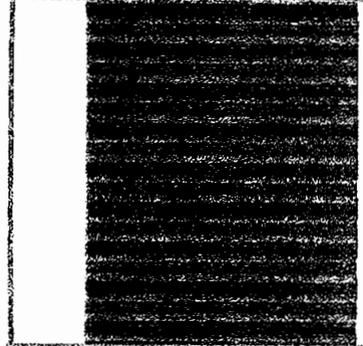
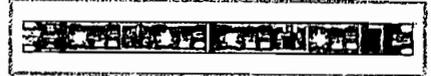
señal y picos (+10dB), controles de ganancia de salida, mute y polaridad. La conmutación de escala es automática bajo selección multimodo basada en PROM. Los filtros son de 24dB/octava Linkwitz-Riley, y las entradas y salidas están electrónicamente balanceadas.

FDS-360

Filtro divisor de 3/4 vías mono ó 2 vías estéreo

El FDS-360 es un filtro divisor electrónico de cuarto orden y frecuencia fija estéreo de 2 vías o mono de 3 ó 4 vías. Sus funciones incluyen la limitación independiente de la banda de medios, el cambio de polaridad, hasta 360° de corrección de fase, circuito de auto-mute, enlace de dos unidades mono, indicación LED para limitación, presencia de señal,

mutes y modo de funcionamiento, y tarjetas de frecuencia intercambiables. Dispone de una regleta en la parte trasera para insertar procesadores, como líneas de retardo para alineación de tiempo o ecualizadores gráficos para la ecualización de cada banda del sistema. Se suministra con 3 tarjetas de corte.





FDS-334/336



El Minidrive de BSS es una unidad de bajo coste que permite a un amplio mercado de aplicaciones la potencia de procesamiento, prestaciones y la calidad del sistema Omnidrive.

Proporciona las cuatro funciones principales de la gama Omnidrive: crossovers (Linkwitz-Riley, Bessel or Butterworth), limitadores de medios, ecualización libremente asignable a las entradas y las salidas, y entradas y salidas con retardo. No obstante, dejando fuera las prestaciones más avanzadas del Omnidrive, BSS ha conseguido unos precios realmente bajos, asegurando que el Minidrive sea asequible a una gama de aplicaciones para las cuales el Omnidrive no es rentable.

El Minidrive está disponible en 2 modelos, el FDS-334 (con 2 entradas y 4 salidas) y el FDS-336 (con 2 entradas y 6 salidas). El FDS-336 es capaz de manejar sistemas estéreo de 3 vías y estar presente en el rack FOH de una PA, mientras que el FDS-334 se adecua perfectamente en un rack de monitores. Cada unidad dispone de hasta 60 bandas de filtro.

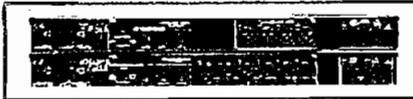
Como un valor añadido, el Minidrive puede ser controlado desde el programa de análisis SIASoft's Smaart Live, permitiendo al instalador o técnico de sonido crear sistemas con procesamiento y medición integrados.

Las salidas individuales del Minidrive pueden funcionar en toda la gama de frecuencias. Con el bajo precio de las unidades Minidrive, estas son altamente aconsejables para su utilización como controladores de zona en centros de convenciones, salas de baile, rellenos con ecualización y retardo, o para reemplazar unidades analógicas tradicionales en un sistema de sonido ya instalado en una sola unidad de rack.

BSS le ha proporcionado al Minidrive un interface de usuario similar al de su modelo superior FDS-366 Omnidrive Compact Plus.

El sistema de navegación de cuatro flechas Navipad permite el acceso a todos los menús y prestaciones, mientras que con los pulsadores de control de nivel de salida se asegura un rápido acceso a la edición de los canales. Los datos son mostrados en una brillante pantalla LCD azul, con la entrada de datos a través de una rueda de parámetros. Los medidores de nivel de entrada y salida dan al técnico de sonido una confortable inmediatez a su trabajo, al igual que las teclas de mute individuales por banda. Todo esto convierte al Minidrive en el DSP más conveniente y rápido de ajustar que cualquier unidad de su clase.

- Rápido acceso a la edición de cada entrada y salida
- Cada salida es asignable a cualquier entrada, o la suma de entradas
- Pantalla azul brillante
- Crossover, ecualización asignable, Limitadores de medios, polaridad y retardo en cada salida
- Ecualización paramétrica y retardo asignable en cada entrada
- Medidores LED en cada entrada y salida
- Crossover de 6, 12, 18, 24, ó 48dB por octava Butterworth, Bessel, o Linkwitz-Riley
- 60 Programas de usuario
- Niveles y mutes bloqueables en el panel frontal
- Control MIDI





FDS-366

Sistema de gestión de altavoces Omnirive Compact Plus

Una unidad FDS-366 puede controlar un sistema de 3 vías estéreo real, o 3 salidas bi-amp para monitores. Añadiendo más unidades y enlaces MIDI esclavos se pueden conseguir sistemas de 4, 5 o 6 vías.

Se pueden crear zonas para 6 áreas separadas utilizando toda la gama o salidas de banda restringida. Como la suma mono puede aplicarse a todas las entradas, el FDS-366 puede utilizarse en sistemas de monitoraje para producciones LCR con salida sub mono. Su entrada digital estéreo AES/EBU soporta frecuencias de muestreo de hasta 96kHz.

El Asistente de Alineamiento del FDS-366 calcula los programas de retardos de altavoces de forma rápida y automática. Una entrada de micrófono mide y corrige al momento cualquier desplazamiento de altavoces y pantallas, y tiene en cuenta cualquier cambio de fase presente en el sistema.

El FDS-366 se suministra con el software de control para PC Soundbench V3.

- 3 Entradas y 6 salidas
- Frecuencia de muestreo de 96kHz.
- Convertidores de 24-bit de última generación que proporcionan una gama dinámica que excede los 112dB.
- Algoritmos DSP para obtener una fase correcta.
- El Alignment Assistant se ocupa de los programas de retardo de altavoces.
- Entrada digital estéreo AES/EBU.
- Ecualización dinámica en cada entrada y salida.
- Nuevo diseño del limitador con parámetros ajustables.
- Filtros de EQ libremente asignables a entradas y salidas.
- Control remoto para PC Soundbench V3 para controlar varias unidades.
- Controladores RS-232, RS-485 y MIDI.

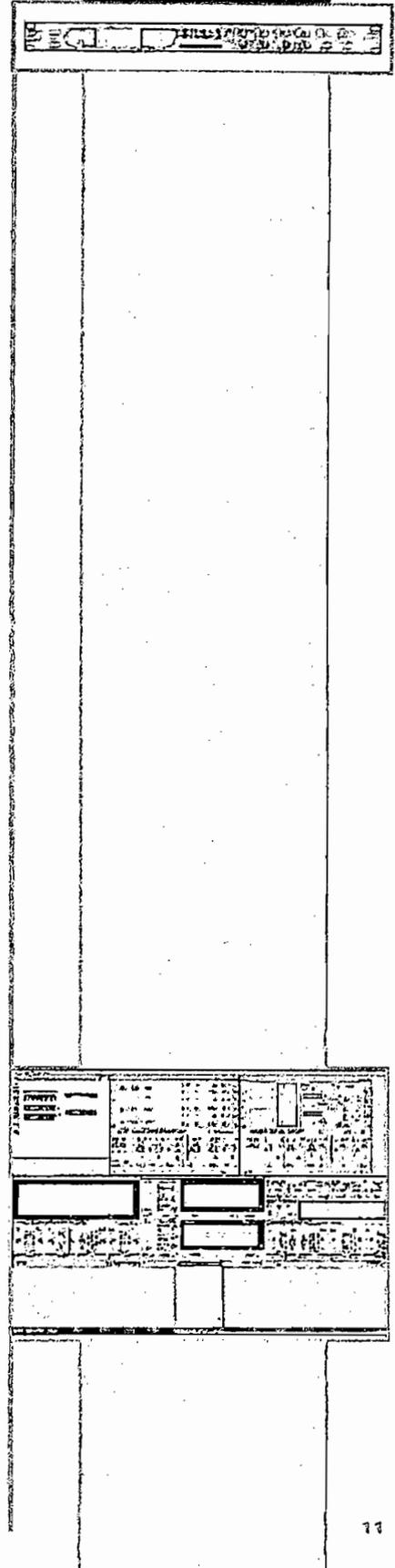
SOUNDBENCH

Software de control PC para Varicurve y Omnirive

Para asistir a los diseñadores de sistemas en el uso de las unidades VARICURVE y OMNIRIVE en estas instalaciones, BSS Audio ha desarrollado el programa SOUNDBENCH, que puede usarse en cualquier PC (por ejemplo, un portátil con Windows) para controlar estos dispositivos. Esto ofrece un interfaz de usuario gráfico elegante, portátil y fácil de usar. Los usuarios pueden ajustar parámetros como la frecuencia, la ganancia, Q, puntos de corte, retardos, etc. introduciendo valores numéricos, usando los botones o simplemente desplazando las curvas directamente sobre la pantalla usando el ratón. Dispone también de otras funciones como el almacenamiento y recuperación en memorias.

Ofrece posibilidades de archivo en la memoria del ordenador, pero SOUNDBENCH también tiene las posibilidades típicas de CUT & PASTE para ofrecer un modo de copiar datos no sólo entre unidades diferentes, sino también en documentos de proceso de textos como referencia de ingeniería del sistema.

También es posible usar SOUNDBENCH sin conexión, y configurar curvas de ecualización y programas OMNIRIVE para recuperarlos más tarde cuando lo conectamos.



ANEXO 3

EZ-KIT LITE ADSP-2181

2 GETTING STARTED

2.1 Overview

This chapter provides you with the information you need to install your software and the ADSP-2181 evaluation board. It is important that you install your software and hardware in the order presented for correct operation.

2.2 Contents of Your EZ-KIT Lite Package

The EZ-KIT Lite evaluation board contains ESD (electrostatic discharge) sensitive devices. Electrostatic charges readily accumulate on the human body and equipment and can discharge without detection. Permanent damage may occur on devices subjected to high energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Unused EZ-KIT Lite should be stored in the protective shipping package.



Your ADSP-2181 EZ-KIT Lite evaluation package contains the following items. If any item is missing, contact the vendor where you purchased your EZ-KIT Lite or Analog Devices.

- ADSP-2181 EZ-KIT Lite board
- Power cable with 8-10V DC power supply
- RS-232 serial port 9-pin cable
- One CD-ROM containing the VisualDSP++ software and EZ-KIT Lite software with examples.

2.3 Installation Procedures

The following procedures are provided for the safe and effective use of the ADSP-2181 evaluation board. It is important that you follow these instructions in the order presented to ensure correct operation of your software and hardware. After you have completed the physical set up of your board, you can load and run the demonstration programs contained on the distribution media. For more information, see Chapter 4 Demonstration Programs.

2.3.1 Installing the EZ-KIT Lite Board

The ADSP-2181 EZ-KIT Lite board is designed to run outside your personal computer as a stand-alone unit. You do not have to remove the chassis from your computer. Use the following steps to connect the EZ-KIT Lite board:

1. Remove the EZ-KIT Lite board from the package—be careful when handling the board to avoid the discharge of static electricity, which may damage some components.
2. Connect the RS-232 cable to an available Comm Port on the PC and to J3 on the ADSP-2181 evaluation board.
3. Plug the provided cord into a 120-Volt AC receptacle and plug the connector at the other end of the cable into J4 on the evaluation board.

All of the LEDs light up briefly. The power (green) LED remains on and FL1 blinks. If the LED does not light up, check the power connections.

To configure your board to take advantage of the audio capabilities of the demos, use the following procedure:

1. Plug a set of self-powered computer speakers into jack J1 on the board. Turn on the speakers and set the volume to an adequate level.
2. Connect the line out of an electronic audio device to jack J2 on the board. Set jumper JP2 to LINE.
3. Open Jumper JP2 to GND to enable the AD1847 codec. (This is the board default).

This completes the hardware installation. For complete information on the EZ-KIT Lite board, see “Board Layout”.

2.3.2 Installing VisualDSP++

Your EZ-KIT Lite comes with the latest version of VisualDSP++ for the ADSP-21xx DSPs. You must install this software prior to installing the EZ-KIT Lite software.

2.3.2.1 System Requirements

Verify your PC has the minimum requirements.

Table 2-1. System Requirements

Windows® 98, ME, 2000, XP	Windows NT™
Windows 98, ME, 2000, XP	Windows NT 4.0, Service Pack 3 or later
Pentium processor 166MHz or faster	Pentium processor 166MHz or faster
100 MB available space	100 MB available space
16 MB RAM	16 MB RAM
VGA Monitor and color video card	VGA Monitor and color video card
CD-ROM	CD-ROM

2.3.2.2 To Install the Software

1. From the initial screen, select VisualDSP++ Installation.
2. Respond to the installation dialog screens.

2.3.2.3 To Install the License

You need to supply the serial number provided on the sticker affixed to the CD case.

1. From the initial screen, select Install License.
2. Select single user license.
3. Respond to the installation dialogs.

This installs a permanent limited license.

This limited license, lets you run EZ-KIT Lite sessions only. Simulation and emulation are not supported. For a full license, contact your local Analog Devices sales representative.

2.3.2.4 To Register the Product

You can fax your registration card to (603) 882-2655 or mail to:

Attn: Registration
 Analog Devices, Inc.
 20 Cotton Rd
 Nashua NH, 03063

Please note that the VisualDSP++ software that comes with your EZ-KIT Lite is a demo version that limits executable file size to 8K bytes. The EZ-KIT Lite board is also the only target you can attach to. You may upgrade your license by contacting your local Analog Devices sales representative.

2.3.2.5 To View the Documentation

To view documentation on-line, you must have a .pdf reader installed. The Adobe® Acrobat® installation kit has been included on the CD-ROM for your convenience. To install Adobe Acrobat Reader, click Adobe Acrobat Reader in the initial screen and respond to the dialog boxes as prompted.

2.3.3 Installing the EZ-KIT Lite Software

The EZ-KIT Lite utility software is supplied the same CD as VisualDSP++, please follow these steps:

1. Close VisualDSP++ and all Windows applications.

You cannot install any of the EZ-KIT Lite software if any VisualDSP++ applications are running. You should close all Windows applications also.

2. Install the EZ-KIT Lite software by responding to the installation dialog screens.

2.3.3.1 Default Settings

After you have installed the board and utility software, your PC and EZ-KIT Lite have the default settings shown in Table 2-2. You can change these settings in the dialog box that pops up the first time you start VisualDSP++.

Table 2-2 User Configurable EZ-KIT Lite Settings

Selection	Default Setting
Comm Port	Comm 1
Baud Rate	9600

3 USING EZ-KIT LITE SOFTWARE

3.1 Overview

The combination of the EZ-KIT Lite board and the monitor software operate as a target for the VisualDSP++ debugger. The debugger lets you view processor registers and memory and perform several debugging activities, such as setting breakpoints, stepping through code, and plotting a range of memory.

3.2 Standard Operation

This section covers the standard operation of the EZ-KIT Lite board. It describes the I/O capabilities of the on-board components, board power-up, and the on-board monitor program.

3.2.1 I/O Devices

3.2.1.1 Flags

The ADSP-2181 has one asynchronous FLAG I/O pin. The FL1 Pin is connected to the red FL1 LED. This lets you visually inspect states of your program.

3.2.1.2 Interrupts

The ADSP-2181 EZ-KIT Lite has one external interrupt connected through pushbutton switch S2. This corresponds to external interrupt IRQE.

The external interrupts are controlled through the ICNTL and IMASK registers and are configured by modifying the interrupt vector table or through instructions in user code. The ICNTL register also controls the interrupt sensitivity between level and edge. To prevent an interrupt from being masked, write to the IMASK register.

The monitor program running on the ADSP-2181 uses one interrupt (the timer) for normal operation. When downloading your own code through the monitor program, the timer interrupt vector is protected and cannot be overwritten. If these vectors are overwritten, or the timer interrupt is masked in any way, the debugger will not be able to communicate with the host program. The following rules and restrictions should be followed when using interrupts:

- Do not step into an interrupt.
- Interrupts are disabled when the user program is halted.

The EPROM test consists of verifying a number in memory. If the monitor code is corrupted, the monitor may crash before reaching the actual program code.

3.2.2.2 UART Check/Initialization

The software UART check is performed when it attempts to connect to the EZ-KIT Lite through a Transmitted Loop Back routine. This UART test is performed by the host after the POST is complete. In this test, the host sends the UART test protocol. This protocol specifies the number of bytes that are transmitted to the EZ-KIT Lite board, and instructs the board to echo the byte stream back to the host. This test determines whether the EZ-KIT Lite board is set to the correct baud rate, and verifies the external connections between the board and the host.

On power up, the EZ-KIT Lite board defaults to a baud rate of 9600 with 8 data bits, 1 stop bit, and no parity. To change this rate, wait for the POST routine to complete and then use the Settings Baud Rate command in the debugger. Note that setting the baud rate to a lower number can significantly slow the board's response to all debug activities.

Different baud rates should be selected based on the type of code you are working with. For real-time interrupt driven programs, a lower baud rate setting slows performance but the timer interrupt occurs less frequently. This gives your program a larger share of the processors resources.

3.2.2.3 AD1847 Check/Initialization

On reset, the AD1847 is inactive. An initialization routine initializes the codec by sending a series of command words through the SPORT0 TX interrupt. Once the commands have been sent and the AD1847 is initialized, it begins transmitting the clock which synchronizes data transfers to and from the DSP. Once this bit goes high, the AD1847 is ready for standard communication over SPORT0.

3.2.3 Monitor Program Operation

The monitor program runs on the EZ-KIT Lite board as part of the DSP executable, and provides the ability to download, debug, and run user programs. The VisualDSP++ debugger is the interface to the monitor. Using the EZ-KIT Lite as a target with the debugger lets you operate the board remotely.

There are three main components of the monitor program:

- Halt loop
- UART ISR (Timer ISR)
- Command Processing Kernel

The monitor program idles in the halt loop when it is not running user code. While there, you can read/write memory, read/write registers, download programs, set breakpoints, change the UART's baud rate, and single-step through code. To enter the halt loop from your code, you must suspend or stop user code—either with a breakpoint or a halt instruction. At this point, the halt loop polls the UART. With every character received from the UART, the command-processing kernel

verifies whether a full command has been received. If a command has been received, the kernel processes the command; otherwise, control is returned to the halt loop to wait for more characters. The only method of executing your code once the halt loop has been entered is to send a Run or Single-step command from the debugger.

The UART ISR is entered when your code is running, but the host is still interacting with the board. As the host sends bytes, the UART ISR takes the data stream from the UART, and builds the command. As with the halt loop, each character received is passed to the command-processing kernel. Unlike the halt loop, the monitor returns to your code immediately after the interrupt is serviced.

The following restrictions should be followed to ensure correct board operation.

- ⊗ The host loses contact with the monitor while the user program is running if the user program disables the Timer interrupt or changes the Timer interrupt vector.
- ⊗ The host loses contact with the monitor while the program is running, and it enters an Interrupt Service Routine when nesting is turned on.
- ⓘ The host cannot halt with the debugger's Debug Halt command if global IRQ enable is disabled. However, breakpoints will work.
- ⓘ The debugger will have trouble halting at a baud rate over 9600 while using the monitor program.

Command processing, initiated from either the UART ISR or the halt Loop, is done in the command-processing kernel. This kernel parses the commands and executes the instructions. If the instruction requires data to be sent back to the host, the kernel initiates the response.

3.2.3.1 Breakpoints

The ability to stop the execution of code and examine processor registers and memory is extremely helpful when debugging code. Note that the debugger automatically inserts breakpoints at the function Main(), when the Settings Run To Main command is selected, and at _exit instruction.

3.2.4 AD1847 Transmissions

After initialization, the AD1847 generates the clock used to transfer data across SPORT0. The ADSP-2181 initiates all transmissions with the AD1847 by sending a synchronization pulse. Even though the AD1847 transmits the data clock, it may not be ready for normal operation.

Initialization of the AD1847 is performed by sending 13 control words contained in a circular buffer to the AD1847. This is usually done via the SPORT0 TX interrupt routine. Once the codec is initialized, autobuffering is used to fill up the TX and RX buffers, which use circular buffering. Once the circular buffer wraps around, then either a TX or RX interrupt occurs. Then the DSP will process the interrupt request.

3.3 Running Your Own Programs

This section provides the basic information you need to run your own programs on the ADSP-2181 EZ-KIT Lite. You build these programs using the 16-bit tools. This information includes rules for using processor memory and a simple program generation procedure.

Although there are many ways to go about developing programs in the VisualDSP++ environment, all program development within the environment should include the following steps:

- Step 1: Create a New Project File
- Step 2: Set Target Processor Project Options
- Step 3: Add and Edit Project Source Files
- Step 4: Customize Project Build Options
- Step 5: Build a Debug Version of the Project
- Step 6: Debug the Project
- Step 7: Build a Release Version of the Project

By following these steps, your DSP projects build consistently and accurately with minimal project management. Note the following restrictions of this system:

- ⊘ The size of the DSP executable that you can build using the EZ-KIT Lite tools is limited to 8K.
- ⊘ Do not run more than one ADSP-2181 EZ-KIT Lite session in the debugger at any one time. You may run an EZ-KIT Lite session and a simulator or ICE session at the same time, or you can open two debugger interfaces to run more than one EZ-KIT Lite session.

5 WORKING WITH EZ-KIT LITE HARDWARE

5.1 Overview

This chapter discusses hardware design issues on the ADSP-2181 EZ-KIT Lite board. The EZ-KIT Lite board schematics are available as an insert at the end of this manual.

5.2 EZ-KIT Lite Specifications

Processor:	ADSP-2181KS-133 operating at an instruction rate of 33 MHz (16.667 external clock)
Analog interface:	AD1847 stereo codec
Analog inputs:	One stereo pair of 2V RMS AC-coupled line level inputs One stereo pair of 20mV RMS AC-coupled microphone inputs
Analog outputs:	One stereo pair of 1V RMS AC-coupled line level outputs
Power source:	8 to 10V DC at 300mA
Environment:	0 to 70° Centigrade; 10 to 90 percent relative humidity (non-condensing)

5.3 System Architecture

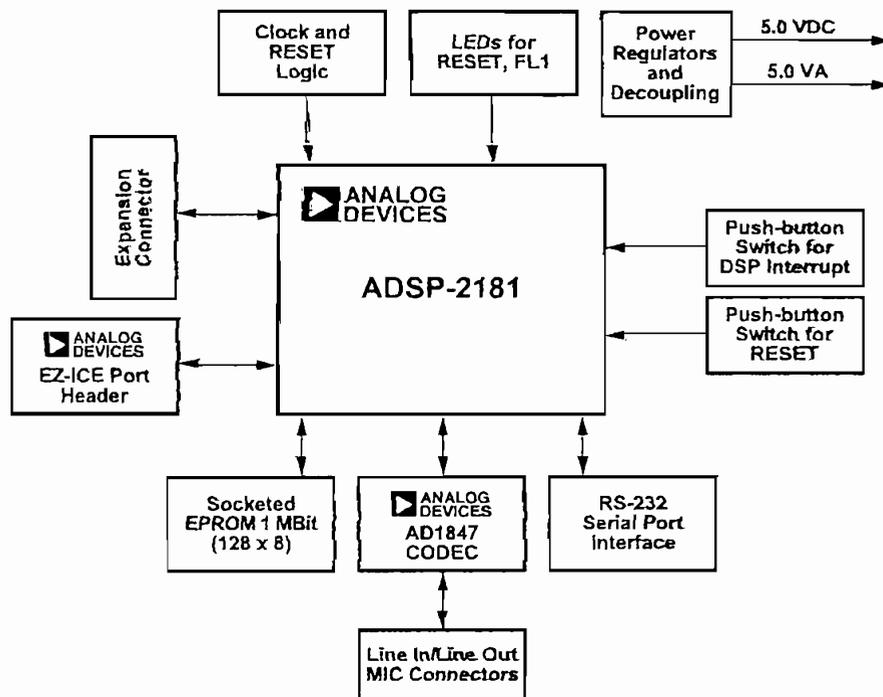


Figure 5-1 EZ-KIT Lite System Block Diagram

5.4 Board Layout

Figure 5-2 shows the layout of the EZ-KIT Lite board. This figure highlights the locations of the major components and connectors. Each of these major components is described in the following sections.

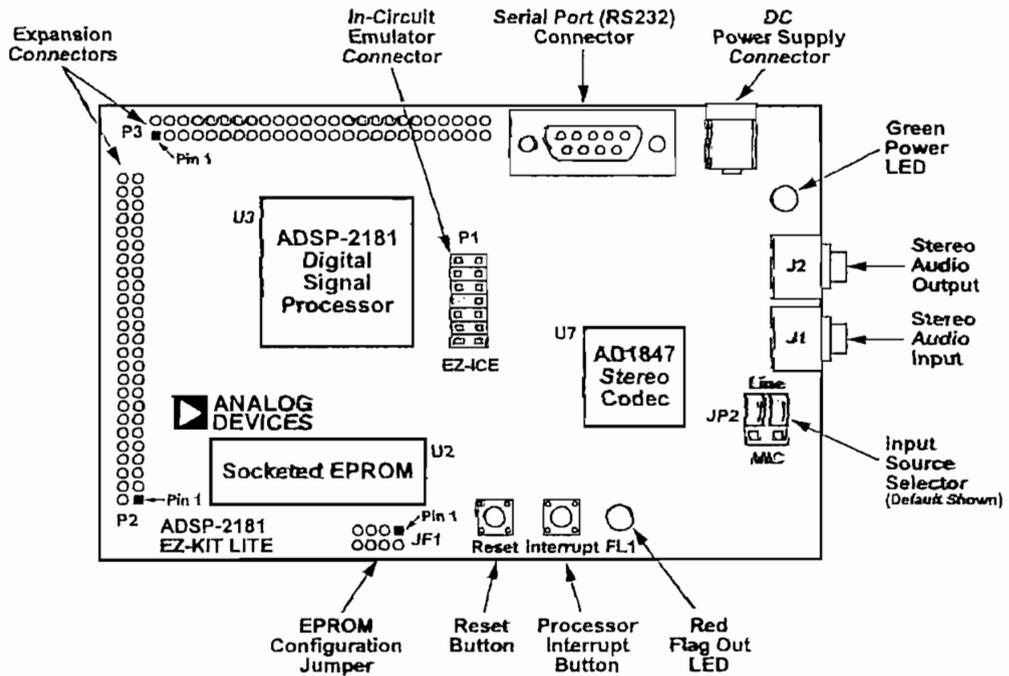


Figure 5-2 EZ-KIT Lite Board Layout

5.4.1 Socketed Memory

The socketed EPROM provides up to 128K x 8 bits of program storage that can be loaded by the ADSP-2181 when the DSP is programmed to boot from the socketed EPROM. After the ADSP-2181 is reset, the BDMA feature is used to load the first 32 words of program memory from byte memory space. Program execution is held off until all 32 words are loaded. Refer to the appropriate processor's Family User's Manual and the ADSP-2181 data sheet for more information on program booting and processor modes.

5.4.2 User LEDs

D1 is a red light emitting diode, which is controlled by the FL1 output of the ADSP-2181 processor. Software can control the state of this indicator by writing to an internal register.

D2 is a green light emitting diode, which is on whenever the board has power.

5.4.3 Switches

S1 is the reset pushbutton switch. Pushing this button causes the ADSP-2181 processor and the AD1847 codec to enter the hardware reset state and remain there until it is released. The switch outputs are de-bounced electronically to prevent multiple transitions due to mechanical contact bounce.

S2 is the interrupt pushbutton switch. Pushing this button causes the ADSP-2181 to receive an $\overline{\text{IRQE}}$ interrupt input. The processor then executes the current $\overline{\text{IRQE}}$ interrupt handler software if the interrupt is enabled and the $\overline{\text{IRQE}}$ interrupt vector is in place. The interrupt switch output is de-bounced electronically to prevent multiple interrupts due to mechanical contact bounce.

5.4.4 Power Connector

The power connector supplies DC voltages to the EZ-KIT Lite board. Table 5-1 shows the power connector pin-out. If you do not use the power supply provided with your EZ-KIT Lite board, replace it with one that has the connections shown in Table 5-1.

Table 5-1 Power Connection

Terminal	Connection
Center pin	8 – 10v DC @ 300mA
Outer Ring	positive

5.4.5 European Power Supply Specifications

Table 5-2 European Power Supply Specifications

DC VOLTAGE:	8 to 10V DC
CURRENT:	300mA
DC CONNECTOR:	
Type:	Switchcraft 760 style, FEMALE
Plug Size:	5.5 (OD) X 2.1 (ID) X 12 (length) millimeters
Polarity:	Center is Negative (inside terminal)

5.4.6 AD1847 Connections

When the AD1847 is enabled, you can access the audio input and output jacks on the board. Each of the audio connectors are stereo mini jacks and accept standard commercially available stereo mini plugs.

The Microphone/Line_in input jack connects to the LINE_IN_L (left) and LINE_IN_R (right) pins or the MIC1 and MIC2 of the AD1847 SoundPort Stereo codec, depending on the setting of jumpers JP2. For more information, see the connections descriptions.

The LINE Output jack connects to the left (L) LINE_OUT and right (R) LINE_OUT pins of the codec.

5.4.7 Expansion Port Connectors

The two expansion port connectors provide access to the bus signals of the ADSP-2181. One possibility for the use of these connectors, beyond debugging, is host control. All interrupts, bus signals, and PWM event signals are available through this port. For more information, see “Expansion Connectors”.

⊗ **WARNING:** External port loading can effect external bus speed and performance.

5.4.8 Connectors and Headers

J1 is a 1/8 inch (3.5 mm) stereo jack. This jack is used to bring line level or microphone audio signals into the board.

J2 is a 1/8 inch (3.5 mm) stereo jack. This jack is used to bring out line level audio signals from the board.

J3 is a female 9-pin D-Sub connector. It is used to communicate with a host computer using RS-232 signal levels and asynchronous serial protocols.

J4 is a jack for a 5.5 mm cylindrical plug. It is used to supply power to the board. The center pin of the jack is 2 mm diameter and should connect to the negative side of the power source. The outer sleeve of the mating plug must be positive.

JP1 is a site for an eight-pin header. It can be used to configure the board for EPROM sizes other than the 1 Mbit (128K byte) EPROM (27C010) shipped with the board. Most users will not need this feature. For more information see section on EPROM Jumper Settings.

JP2, shown in figure 5-3, is a six-pin header. It is used to configure input jack J1 for line level or microphone input. The center pin in each group of three is connected to one of the AD1847 codec's input pins. Jumpers (also known as shunts or shorting links) can be used to connect these pins to the output of the microphone amplifier or to the output of the line level input filter.

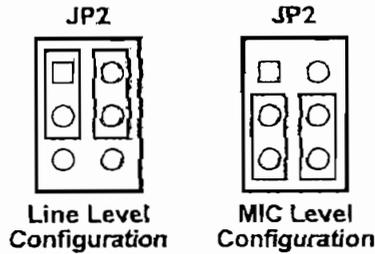


Figure 5-3 JP2 Jumper Settings

P1 is a 14-pin header connector used to connect to an ADSP-218x EZ-ICE[®] in-circuit emulator. Pin 7 should be removed for keying purposes. For more information, see “EZ-ICE Connector”.

P2 and P3 are sites for 50-pin header connectors. These connectors can be used to access the ADSP-2181 signals for expansion or test purposes. For more information, see “Expansion Connectors”.

U2 is a socket for an EPROM in a DIP package. As built the board will accept a 27C512 (64K byte) or 27C010 (128K byte) EPROM. Changing connections at JP1 allows the board to accept a 27C256 (32K byte), 27C020 (256K byte), 27C040 (512K byte), or 27C080 (1 Mbyte) EPROM. This socket is connected to the ADSP-2181’s byte-wide memory interface.

R28 is a site for a zero ohm resistor. If this resistor is installed the ADSP-2181 processor can reset the board under software control. The software would assert reset by configuring PF0 as an output and then setting it low.

R29 is another site for a zero ohm resistor. If this resistor is installed and X3 and C37 are removed the codec can operate off of the ADSP-2181’s CLKOUT signal instead of its own 24.576 MHz clock. It will also be necessary to change X1 to a lower frequency value to stay within the codec’s ratings.

5.4.9 EPROM Jumper Settings

JP1 allows the ADSP-2181 EZ-KIT Lite board to be configured for any one of six different EPROM sizes. As shipped the board can accommodate either a 27C512 or 27C010. If some other size EPROM is installed in the socket at U2 it will be necessary to change the connections at JP1 shown in figure 5-4.

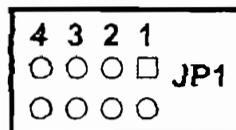


Figure 5-4 JP1 Jumper Settings

Connections are made vertically between pads. The pair of pads below each number constitutes the jumper position associated with that number. Connections can be made in several ways. If an eight-pin header is installed and the etch connections on the back are cut, EPROM size changes can be accommodated easily by installing and removing shunts. If frequent size changes are not contemplated, it may be sufficient to solder wires between the pads and so make the connections permanent.

Connections for 27C256 EPROM should look like figure 5-5.

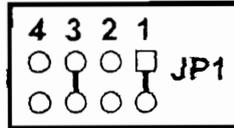


Figure 5-5 JP1 Jumper Setting for 27C256 EPROM

Note: This involves cutting the etch on the back of the board at jumper position 2 and adding a connection at jumper position 1.

Connections for a 21C512 or 27C010 EPROM should look like figure 5-6.

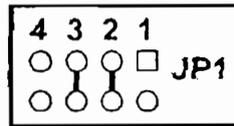


Figure 5-6 JP1 Jumper Settings for 21C512/27C010 EPROM

Note: This is how the connections are arranged when the board is manufactured.

The connections for a 27C020, 27C040, or 27C080 EPROM should look like figure 5-7.

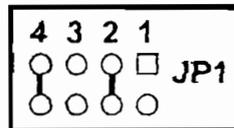


Figure 5-7 JP1 Jumper Settings for 27C020/27C040/ 27C080 EPROM

Note: This involves cutting the etch on the back of the board at jumper position 3 and adding a connection at jumper position 4.

5.4.10 Hardware Operation

When power is applied to the board, the reset circuit holds the processor in reset for approximately 30 ms. Reset is then deasserted and the processor begins the boot process. The BMODE and MMAP pins on the ADSP-2181 are grounded so the processor boots from the byte memory interface, which is connected to the EPROM socket. If the EPROM supplied with the board is installed in the socket, the operation of the board will proceed as documented in the software section of this manual.

5.4.11 EZ-ICE Connector

The ADSP-218x EZ-ICE® Emulator aids in the hardware debugging of an ADSP-2181 system. The emulator consists of hardware, host computer resident software, and the target board connector. The ADSP-2181 integrates on-chip emulation support with a 14-pin ICE-Port interface (Figure 5-3). This interface provides a simpler target board connection that requires fewer mechanical clearance considerations than other ADSP-2100 family EZ-ICEs. The ADSP-2181 device need not be removed from the target system when using the EZ-ICE, nor are any adapters needed. Due to the small footprint of the EZ-ICE connector, emulation can be supported in final board designs.

The EZ-ICE performs a full range of functions, including:

- In-target operation
- Up to 30 breakpoints
- Single-step or full-speed operation
- Registers and memory values can be examined and altered
- PC upload and download functions
- Instruction-level emulation of program booting and execution
- Complete assembly and disassembly of instructions
- C source-level debugging

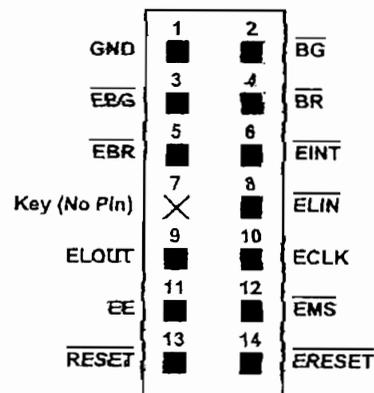


Figure 5-8 EZ-ICE 14-Pin Header (P6)

For more information on connecting to an ICE, see the ADSP-2181 data sheet and Application Note EE-34.

5.5 Designing an EZ-ICE Compatible Target

This section describes the ADSP-218x family EZ-ICE theory of operation to aid you in your design of a compatible target system.

The hardware consists of a printed circuit board measuring 3.5 inches by 5.5 inches. Assembled onto the printed circuit board are an ADSP-2181 digital signal processor, a socketed EPROM, an AD1847 codec, and various support circuits and connectors. The board is a complete signal processing system designed to demonstrate the capabilities of the ADSP-2181 digital signal processor. It can also be used as a platform to develop new applications for the ADSP-2181.

The EZ-KIT Lite board is an example of a minimum implementation of an ADSP-2181 processor. The socketed EPROM is connected to the processor via the Byte DMA port. This interface uses only eight of the twenty-four data lines to carry data (D8 through D15). Eight of the spare data lines (D16 through D23) are used to provide additional address bits. This allows the ADSP-2181 to address up to 32 Mbits (4 Mbytes) of memory. The DSP is configured to boot from the socketed EPROM when RESET is de-asserted or if power is applied to the board.

The AD1847 codec is connected to the DSP via SPORT0. This high-speed synchronous serial port carries all of the data, control, and status information between the DSP and the codec. It is possible to disable the codec if the serial port is to be used for another purpose. The CODECDIS signal available on connector P3 can be used to disable the codec. When this signal is brought low, the codec is disabled and its signals are put in a high-impedance state.

The SPORT1 pins are used to communicate with the host PC via the RS-232 interface (J3). The Flag In and Flag Out pins carry the receive and transmit data. Software running on the DSP emulates a UART to provide the proper protocol for asynchronous serial communications up to a data rate of 115K bits per second.

5.5.1 Hardware Debugging

If the green LED fails to light, check your power connections. Verify that your power supply has the proper size connector and that the polarity is correct. The power supply voltage measured at the connector to the board should be 8V to 10V DC. Also, ensure that there are no objects beneath or on top of the board that may cause a short circuit. Press the reset button (S1) if the board appears to be operating improperly.

5.6 Expansion Connectors

The two expansion connectors provide access to the ADSP-2181's interface pins. These pins let you watch data transmissions. In addition, the host interface, interrupt, and pwm_event pins are also available on this connector.

P2 and P3 are sites for 50-pin header connectors. These connectors can be used to access the ADSP-2181 signals for expansion or test purposes. The pin numbers on these connectors are arranged as follows.

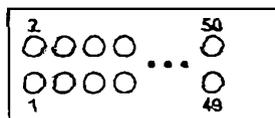
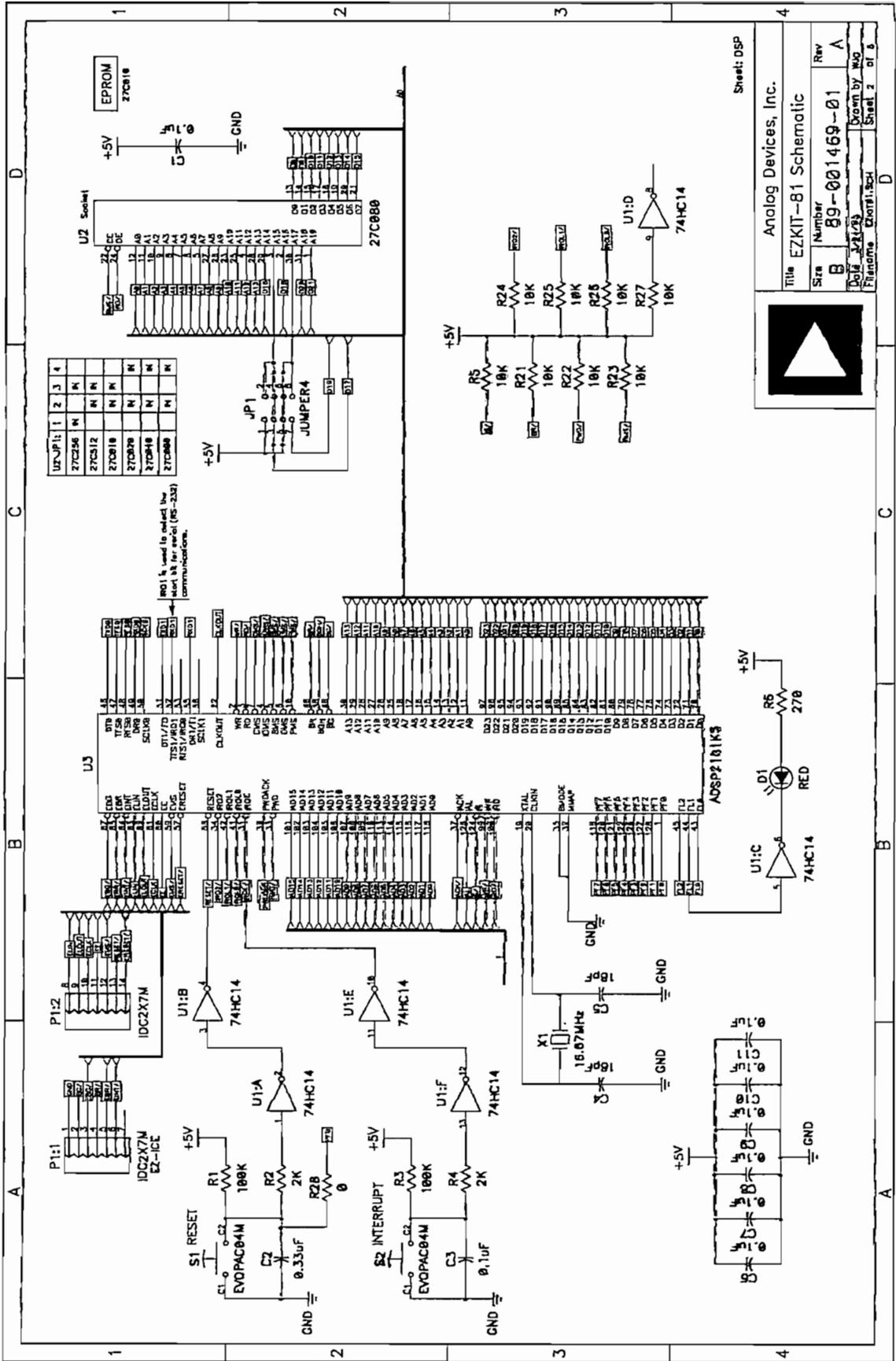


Figure 5-9 Expansion Connector

The signals available on these pins are shown in Table 5-3.

Table 5-3 ADSP-2181 Pin Names

P2				P3			
Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
1	A0	2	A1	1	GND	2	IAD0
3	A2	4	A3	3	IAD1	4	IAD2
5	A4	6	A5	5	IAD3	6	IAD4
7	A6	8	A7	7	IAD5	8	IAD6
9	A8	10	A9	9	IAD7	10	IAD8
11	A10	12	A11	11	IAD9	12	IAD10
13	A12	14	A13	13	IAD11	14	IAD12
15	D0	16	D1	15	IAD13	15	IAD14
17	D2	18	D3	17	IAD15	18	GND
19	D4	20	D5	19	$\overline{\text{LACK}}$	20	IAL
21	D6	22	D7	21	$\overline{\text{IS}}$	22	$\overline{\text{IWR}}$
23	D8	24	D9	23	$\overline{\text{IRD}}$	24	GND
25	D10	26	D11	25	PF0	26	PF1
27	D12	28	D13	27	PF2	28	PF3
29	D14	30	D15	29	PF4	30	PF5
31	D16	32	D17	31	PF6	32	PF7
33	D18	34	D19	33	FL0	34	FL1
35	D20	36	D21	35	FL2	36	CLKOUT
37	D22	38	D23	37	$\overline{\text{RESET}}$	38	$\overline{\text{IRQ0}}$
39	$\overline{\text{WR}}$	40	$\overline{\text{RD}}$	39	IRQL1	40	$\overline{\text{IRQ2}}$
41	$\overline{\text{IOMS}}$	42	$\overline{\text{BMS}}$	41	$\overline{\text{PWD}}$	42	PWDACK
43	$\overline{\text{DMS}}$	44	$\overline{\text{CMS}}$	43	$\overline{\text{CODECDIS}}$	44	TXD0
45	$\overline{\text{PMS}}$	46	$\overline{\text{BR}}$	45	TFS0	46	RFS0
47	$\overline{\text{BGH}}$	48	BG	47	RXD0	48	SCK0
49	VCC	50	GND	49	VCC	50	GND

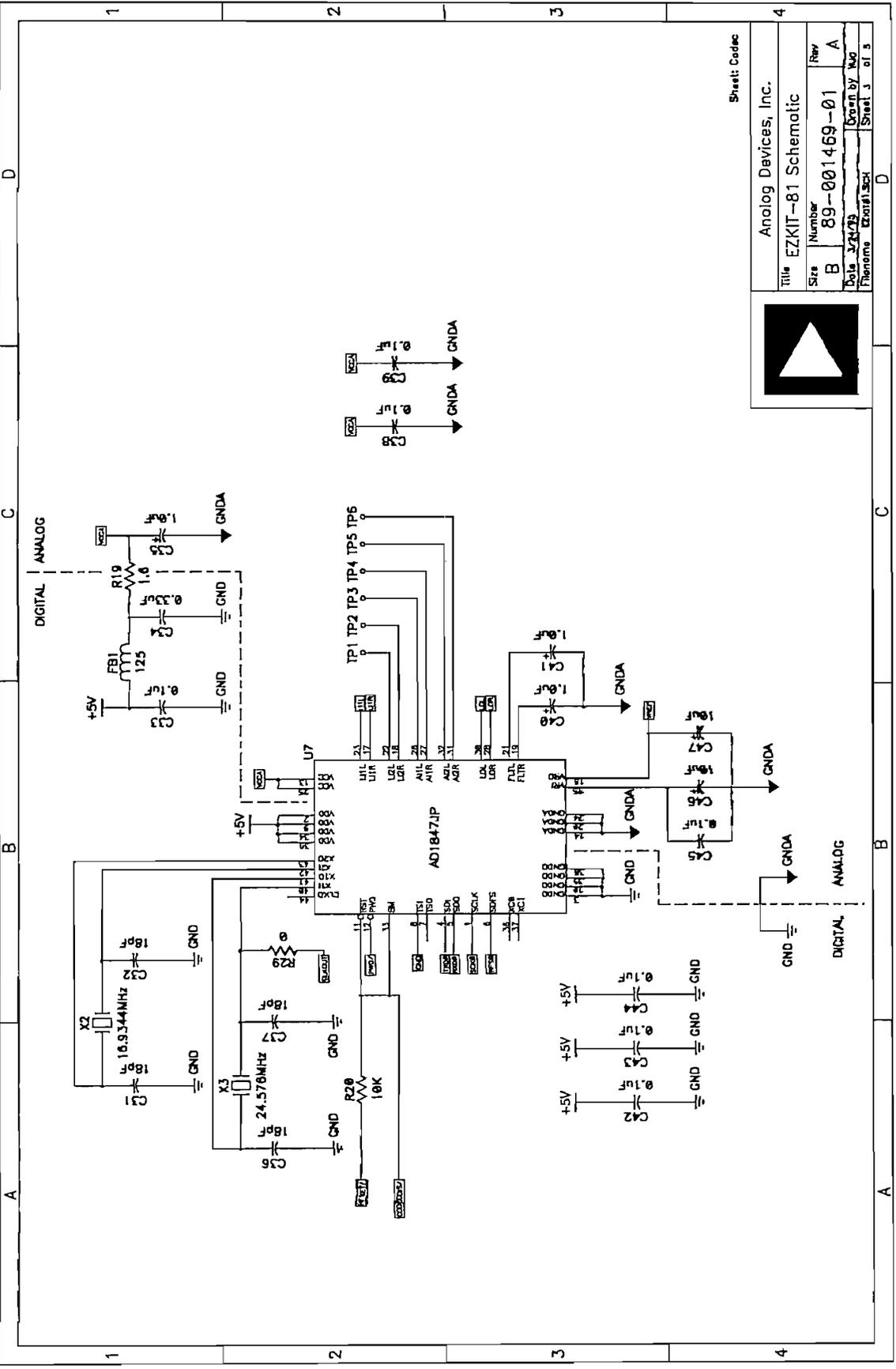


U2 Jumper	1	2	3	4
27C256	IN	IN	IN	IN
27C512	IN	IN	IN	IN
27C818	IN	IN	IN	IN
27C819	IN	IN	IN	IN
27C869	IN	IN	IN	IN

RO1 is used to protect the board in for serial (RS-232) communications.

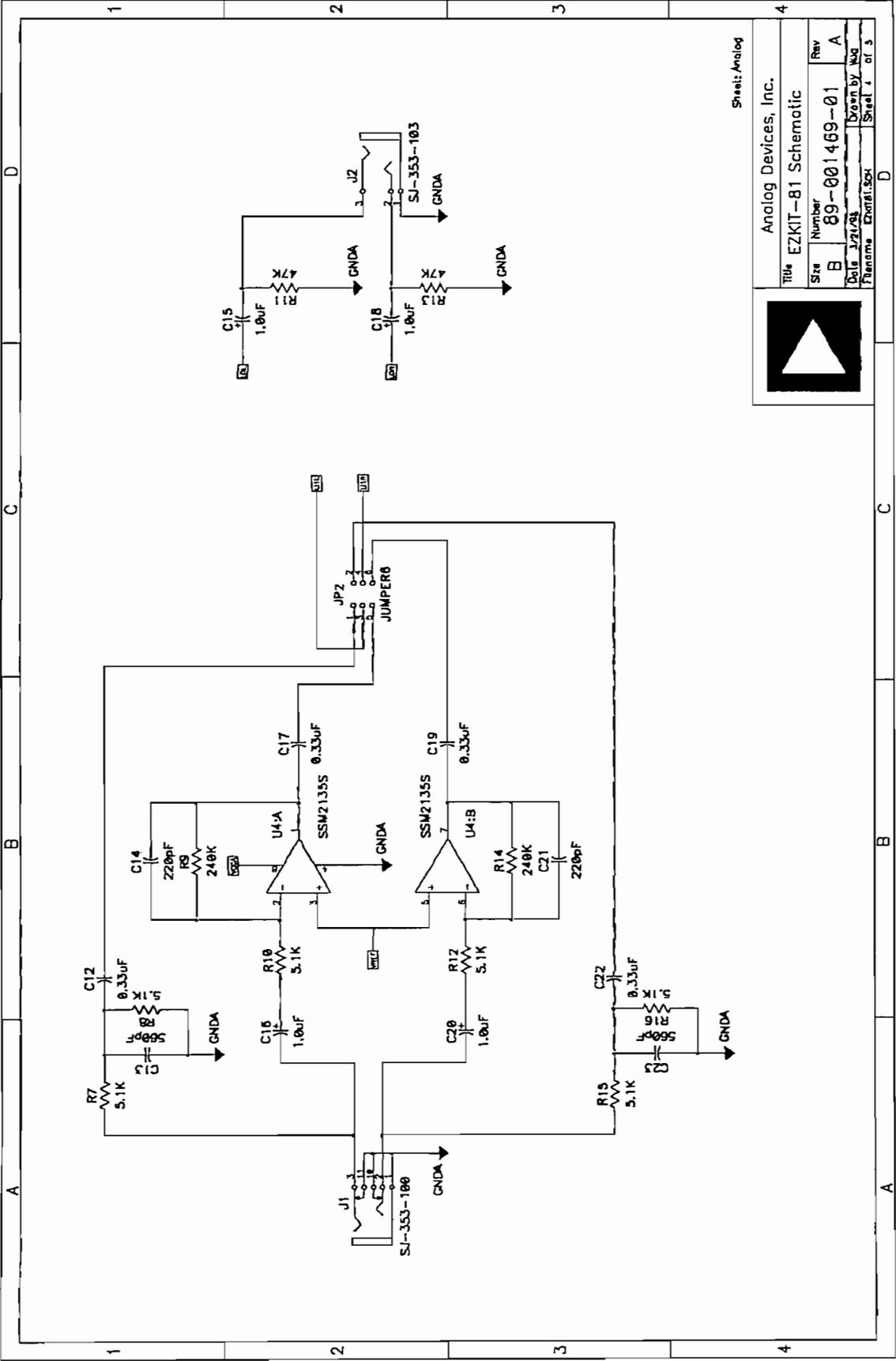
Sheet: DSP
 Title: EZKIT-81 Schematic
 Size: B
 Number: 89-001469-01
 Rev: A
 Date: 12/2/89
 Drawn by: MJO
 Checked by: JCS
 Sheet 2 of 3





Sheet Code: Analog Devices, Inc.

Title	EZKIT-81 Schematic		
Size	Number	Rev	Row
B	89-001469-01		A
Doc. No.	374/73	Drawn by	Map
Filename	Ezkit1.sch	Sheet	5 of 5



Sheet: Analog



Analog Devices, Inc.
Title: EZKIT-81 Schematic

Size	Number	Rev
B	89-001469-01	A
Date	12/1/93	Drawn by
Filename	ezkit81.sch	Sheet # of 3

ANEXO 4

SOFTWARE UART

```
{*****}
```

ADSP-2101 Software UART UART.DSP

This program uses FLAG_IN, FLAG_OUT and the TIMER of the ADSP-2101 to interface to an RS-232 asynchronous serial device such as a VT100 terminal.

for example:

```
ADSP-2101 FLAG_OUT ----> AD233 ----> RS232 RX
ADSP-2101 FLAG_IN <---- AD233 <---- RS232 TX
```

(TIMER maintains baudrate)

Parameters bits/word, baudrate, stopbits & parity are user-programmable. An RS-232 line driver chip (such as the AD233) can be used to electrically interface +5 VDC to the RS-232 line voltage levels.

The operation of the transmitter setup routine is completely independent on the receiver setup routine operation. Although both tx and rx use the same timer as a master clock source, the transmitted bits need not be in sync with the received bits. The default state of the receiver is OFF, so the "turn_rx_on" subroutine must be used to enable RX.

Calling Argument:

For autobaud load the baud constant:

```
dm(baud_period)=(Proc_frequency/(3*Baudrate))-1
```

Useful Subroutines:

init_uart Must be called after system reset.

get_char_axl Waits for RX input and returns with it in AX1.

out_char_axl Waits for last TX output and transmits data from AX1.

turn_rx_on Must be called to enable the receipt of RX data.

turn_rx_off Can be used to ignore input RX data.

Useful Flag:

DM(flag_rx_ready) If this DM location is all ones it indicates that the UART is ready to receive a new word. If it is zero then data is being received. Can be used for xon/xoff flow control.

```
*****}
```

```
.module/boot=1 UART;
```

{The constants below must be changed to modify the UART parameters}

```
.const tx_num_of_bits = 10; {start bits + tx data bits + stop bits}
```

```
.const rx_num_of_bits = 8; {rx data bits (start&stop bits not counted)}
```

```
.const RX_BIT_ADD = 0x0100; {= 1<<rx_num_of_bits }
```

```
.const TX_BIT_ADD = 0xfe00; {= 0xffff<<(tx data bits+1)}
```

{__These constants can be used if autobaud is not needed__}

```
.const PERIOD=74; {13 & 57600} {PERIOD=(Proc_freq/(3*Baudrate))-1}
```

```
{.const PERIOD=112;} {13 & 38400} {PERIOD=(Proc_freq/(3*Baudrate))-1}
```

```
{.const PERIOD=225;} {13 & 19200} {PERIOD=(Proc_freq/(3*Baudrate))-1}
```

```

{.const PERIOD=450;} {13 & 9600} {PERIOD=(Proc_freq/(3*Baudrate))-1}

{_____Definitions for memory-mapped control registers_____}

.const TSCALE = 0x3ffb;
.const TCOUNT = 0x3ffc;
.const TPERIOD = 0x3ffd;
.const System_Control_Reg = 0x3fff;

{_____}

.entry init_uart;           {UART initialize baudrate etc.}
.entry out_char_axl;       {UART output a character}
.entry get_char_axl;       {UART wait & get input character}
.entry turn_rx_on;         {UART enable the rx section}
.entry turn_rx_off;        {UART disable the rx section}
.entry process_a_bit;      {UART timer interrupt routine for RX and
                           TX}

.global flag_rx_ready;
.global baud_period;

.var flag_tx_ready;        {flag indicating UART is ready for new tx word}
.var flag_rx_ready;        {flag indicating UART is ready to rx new word}
.var flag_rx_stop_yet;    {flag tells that a rx stop bit is not pending}
.var flag_rx_no_word;     {indicates a word is not in the user_rx_buffer}
.var flag_rx_off;         {indicates a that the receiver is turned off}
.var timer_tx_ctr;        {divide by 3 ctr, timer is running @ 3x
                           baudrate}
.var timer_rx_ctr;        {divide by 3 ctr, timer is running @ 3x
                           baudrate}
.var user_tx_buffer;       {UART tx reg loaded by user before UART xmit}
.var user_rx_buffer;       {UART rx reg read by user after word is rcvd}
.var internal_tx_buffer;   {formatted for serial word, adds start&stop
                           bits}
                           {'user_tx_buffer' is copied here before
                           xmission}

.var internal_rx_buffer;
.var bits_left_in_tx;     {number of bits left in tx buffer (not yet
                           clkd out) }
.var bits_left_in_rx;     {number of bits left to be rcvd (not yet clkd
                           in) }
.var baud_period;         {loaded by autobaud routine}

{_____Initializing subroutine_____}

init_uart:

    ax0=0;
    dm(TSCALE)=ax0;        {decrement TCOUNT every instruction
                           cycle}
    ax0=dm(baud_period);   {from autobaud or use constant:
                           ax0=PERIOD;}
                           {...and comment in the appropriate
                           constant}

    dm(TCOUNT)=ax0;
    dm(TPERIOD)=ax0;       {interrupts generated at 3x baudrate}
    ax0=0;
    dm(System_Control_Reg)=ax0; {no bmbwait,pmwait states,
                           SPORT1=FI/FO}

    ax0=1;

```

```

dm(flag_tx_ready)=ax0; {set the flags showing that UART is not
                        busy}
dm(flag_rx_ready)=ax0;
dm(flag_rx_stop_yet)=ax0;
dm(flag_rx_no_word)=ax0;
dm(flag_rx_off)=ax0;    {rx section off}
set flag_out;           {UART tx output is initialized to high}
ifc=0x003f;             {clear all pending interrupts}
nop; {wait for ifc latency }
imask=b#000001;        {enable TIMER interrupt handling}
ena timer;              {start timer now}
rts;

{ _____ process_a_bit _____ }

```

(TIMER interrupt routine)

This routine is the heart of the UART. It is called every timer interrupt (i.e. 3x baudrate). This routine will xmit one bit at a time by setting/clearing the FLAG_OUT pin of the ADSP-2101. This routine will then test if the UART is already receiving. If not it will test flagin (rx) for a start bit and place the UART in receive mode if true.

If already in receive mode it will shift in one bit at a time by reading the FLAG_IN pin. Since the internal timer is running at 3x baudrate, bits need only be transmitted/received once every 3 timer interrupts.

```

}

process_a_bit:
ena sec_reg;           {Switch to background register set}
ax0=dm(flag_tx_ready); {if not in "transmit", go right to
                        "receive"}

ar=pass ax0;
if ne jump receiver;

{ _____ Transmitter Section _____ }

ay0=dm(timer_tx_ctr); {test timer ctr to see if a bit}
ar=ay0-1;              {is to be sent this time around}
dm(timer_tx_ctr)=ar;  {if no bit is to be sent}
if ne jump receiver;  {then decrement ctr and return}

srl=dm(internal_tx_buffer); {shift out LSB of internal_tx_buffer}
sr=lshift srl by -1 (hi); {into SRL. Test the sign of this bit}
dm(internal_tx_buffer)=srl; {set or reset FLAG_OUT accordingly}
ar=pass sr0;           {this effectively clocks out the}
if ge reset flag_out; {word being xmitted one bit at a time}
if lt set flag_out;   {LSB out first at FLAG_OUT.}

ay0=3;                {reset timer ctr to 3, i.e. next bit}
dm(timer_tx_ctr)=ay0; {will be sent after 3 timer interrupts}

ay0=dm(bits_left_in_tx); {number of bits left to be xmitted}
ar=ay0-1;              {is now decremented by one,}
dm(bits_left_in_tx)=ar; {indicating that one is now xmitted}
if gt jump receiver;   {if no more bits left, then ready}

ax0=1;                {flag is set to true indicating}

```

```

dm(flag_tx_ready)=ax0;          {a new word can now be xmitted}

{ _____ Receiver Section _____ }

receiver:

ax0=dm(flag_rx_off);           {Test if receiver is turned on}
ar=pass ax0;
if ne rti;

ax0=dm(flag_rx_stop_yet);      {Test if finished with stop bit of}
ar=pass ax0;                   {last word or not. if finished then}
if ne jump rx_test_busy;       {continue with check for receive.}

ay0=dm(timer_rx_ctr);          {decrement timer ctr and test to see}
ar=ay0-1;                     {if stop bit period has been reached}
dm(timer_rx_ctr)=ar;           {if not return and wait}
if ne rti;

ax0=1;                          {if stop bit is reached then reset}
dm(flag_rx_stop_yet)=ax0;      {to wait for next word}
dm(flag_rx_ready)=ax0;

ax0=dm(internal_rx_buffer);    {copy internal rx buffer}
dm(user_rx_buffer)=ax0;        {to the user_rx_buffer}

ax0=0;                          {indicated that a word is ready in}
dm(flag_rx_no_word)=ax0;       {the user_rx_buffer}
rti;

rx_test_busy:

ax0=dm(flag_rx_ready);         {test rx flag, if rcvr is not busy}
ar=pass ax0;                   {receiving bits then test for start.If}
if eq jump rx_busy;            {it}
                                {is busy, then clk in one bit at a time}

if flag_in jump rx_exit;       {Test for start bit and return if none}

ax0=0;
dm(flag_rx_ready)=ax0;         {otherwise, indicate rcvr is now busy}
dm(internal_rx_buffer)=ax0;    {clear out rcv register}
ax0=4;                         {Timer runs @ 3x baud rate, so rcvr}
dm(timer_rx_ctr)=ax0;         {will only rcv on every 3rd interrupt.}
                                {Initially this ctr is set to 4.}
                                {This will skip the start bit and will}
ax0=rx_num_of_bits;           {allow us to check FLAG_IN at the center}
dm(bits_left_in_rx)=ax0;      {of the received data bit.}

rx_exit:

rti;

rx_busy:

ay0=dm(timer_rx_ctr);          {decrement timer ctr and test to see}
ar=ay0-1;                     {if bit is to be rcvd this time around}
dm(timer_rx_ctr)=ar;           {if not return, else receive a bit}
if ne rti;

rcv:                            {Shift in rx bit}

```

```

ax0=3;                {reset the timer ctr to 3 indicating}
dm(timer_rx_ctr)=ax0; {next bit is 3 timer interrupts later}

ay0=RX_BIT_ADD;
ar=dm(internal_rx_buffer);
if not flag_in_jump pad_zero; {Test RX input bit and}
ar=ar+ay0;             {add in a 1 if hi}

pad_zero:

sr=lshift ar by -1 (lo); {Shift down to ready for next bit}
dm(internal_rx_buffer)=sr0;

ay0=dm(bits_left_in_rx); {if there are more bits left to be rcvd}
ar=ay0-1;                {then keep UART in rcv mode}
dm(bits_left_in_rx)=ar;  {and return}
if gt rti;               {if there are no more bits then ...}
                        {...that was the last bit }

ax0=3;                {set timer to wait for middle of the}
dm(timer_rx_ctr)=ax0;  {stop bit}
ax0=0;                {flag indicated that uart is waiting}
dm(flag_rx_stop_yet)=ax0; {for the stop bit to arrive}
rti;

{ _____ invoke_UART_transmit subroutine _____ }

```

This is the first step in the transmit process. The user has now loaded 'user_tx_buffer' with the ascii code and has also invoked this routine.

```

}

invoke_UART_transmit:

ax0=3;                {initialize the timer decimator ctr}
dm(timer_tx_ctr)=ax0; {this divide by three ctr is needed}
                        {since timer runs @ 3x baud rate}
ax0=tx_num_of_bits;  {this constant is defined by the}
dm(bits_left_in_tx)=ax0; {user and represents total number of}
                        {bits including stop and parity}
                        {ctr is initialized here indicating}
                        {none of the bits have been xmitted}

sr1=0;
sr0=TX_BIT_ADD;      {upper bits are hi to end txmit with hi}
ar=dm(user_tx_buffer); {transmit register is copied into }
sr=sr or lshift ar by 1 (lo); {the internal tx reg & left justified}
dm(internal_tx_buffer)=sr0; {before it gets xmitted}
ax0=0;                {indicate that the UART is busy}
dm(flag_tx_ready)=ax0;
rts;

{ _____ get an input character _____ }

output: ax1
modifies: ax0

}

get_char_ax1:
ax0=dm(flag_rx_no_word);
ar=pass ax0;

```

```

if ne jump get_char_ax1;      {if no rx word input, then wait}

ax1=dm(user_rx_buffer);      {get received ascii character}
ax0=1;
dm(flag_rx_no_word)=ax0;     {word was read}
rts;

{ _____ output a character _____

input: ax1
modifies: ax0, srl, sr0, ar
}

out_char_ax1:
ax0=dm(flag_tx_ready);
ar=pass ax0;
if eq jump out_char_ax1;     {if tx word out still pending, then wait}
dm(user_tx_buffer)=ax1;
call invoke_UART_transmit;   {send it out}
rts;

{ _____ enable the RX section _____

modifies: ax0
}

turn_rx_on:
ax0=0;
dm(flag_rx_off)=ax0;
rts;

{ _____ disable the RX section _____

modifies: ax0
}

turn_rx_off:
ax0=1;
dm(flag_rx_off)=ax0;
rts;

.endmod;

```

ANEXO 5

AMPLIFICADOR TA8216H

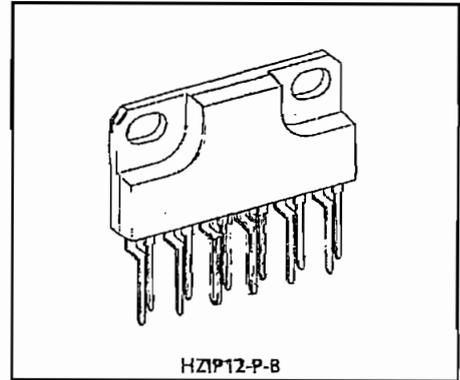
DUAL AUDIO POWER AMPLIFIER

The TA8216H is dual audio power amplifier for consumer applications.

This IC provides an output power of 13 watts per channel (at $V_{CC}=28V$, $f=1kHz$, THD = 10%, $R_L=8\Omega$). It is suitable for power amplifier of music center.

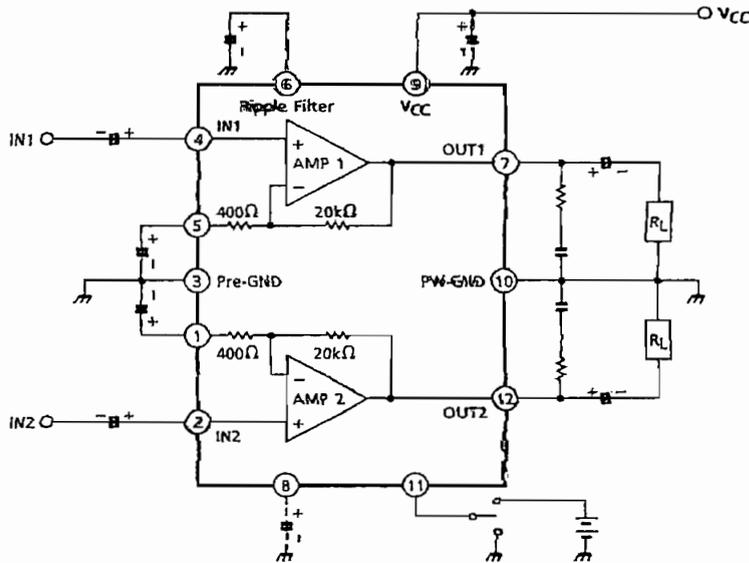
FEATURES

- High Output Power : $P_{out}=13W$ /channel (Typ.)
($V_{CC}=28V$, $R_L=8\Omega$, $f=1kHz$, THD = 10%)
- Low Noise : $V_{no}=0.14mV_{rms}$ (Typ.)
($V_{CC}=28V$, $R_L=8\Omega$, $G_v=34dB$, $R_g=10k\Omega$,
 $BW=20Hz\sim 20kHz$)
- Very Few External Parts.
- Built in Audio Muting Circuit.
- Built In Thermal Shut Down Protector Circuit.
- Operation Supply Voltage Range
: $V_{CC}(opr)=10\sim 37V$ (at $R_L=8\Omega$, $T_a=25^\circ C$)
: $V_{CC}(opr)=10\sim 24V$ (at $R_L=4\Omega$, $T_a=25^\circ C$)



Weight : 4.04g (Typ.)

BLOCK DIAGRAM



© The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.

② These TOSHIBA products are intended for use in general commercial applications (office equipment, communication equipment, measuring equipment, domestic appliances, etc.), please make sure that you consult with us before you use these TOSHIBA products in equipment which requires extraordinarily high quality and/or reliability, and in equipment which may involve life threatening or critical applications, including but not limited to such uses as atomic energy control, airplane or spaceship instrumentation, traffic signals, medical instrumentation, combustion control, all types of safety devices, etc. TOSHIBA cannot accept and hereby declines liability for any damage which may occur in case the TOSHIBA products are used in such equipment or applications without prior consultation with TOSHIBA.

TA8216H - 1

1996 - 6 - 17

TOSHIBA CORPORATION

APPLICATION INFORMATION

(1) Voltage gain

The closed loop voltage gain is determined by R_1 , R_2 .

$$G_V = 20 \log \frac{R_1 + R_2}{R_2} \text{ (dB)}$$

$$= 20 \log \frac{20\text{k}\Omega + 400\Omega}{400}$$

$$= 34 \text{ (dB)}$$

(2) Amplifier with gain < 34dB

$$G_V = 20 \log \frac{R_1 + R_2 + R_3}{R_2 + R_3} \text{ (dB)}$$

When $R_3 = 220\Omega$

$$G_V \approx 30 \text{ (dB)}$$

is given.

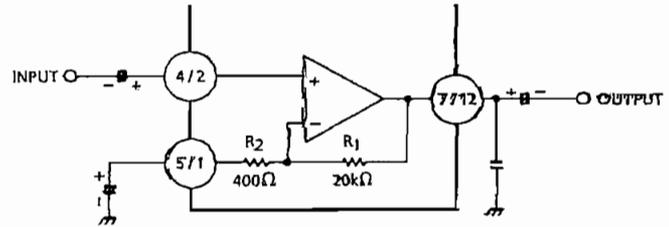


Fig.1

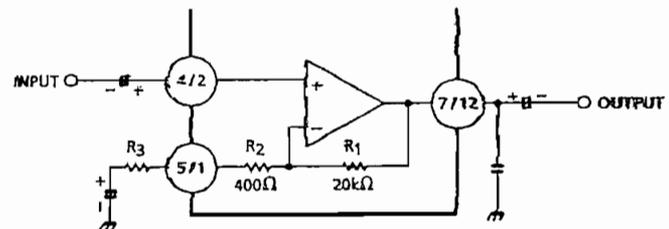
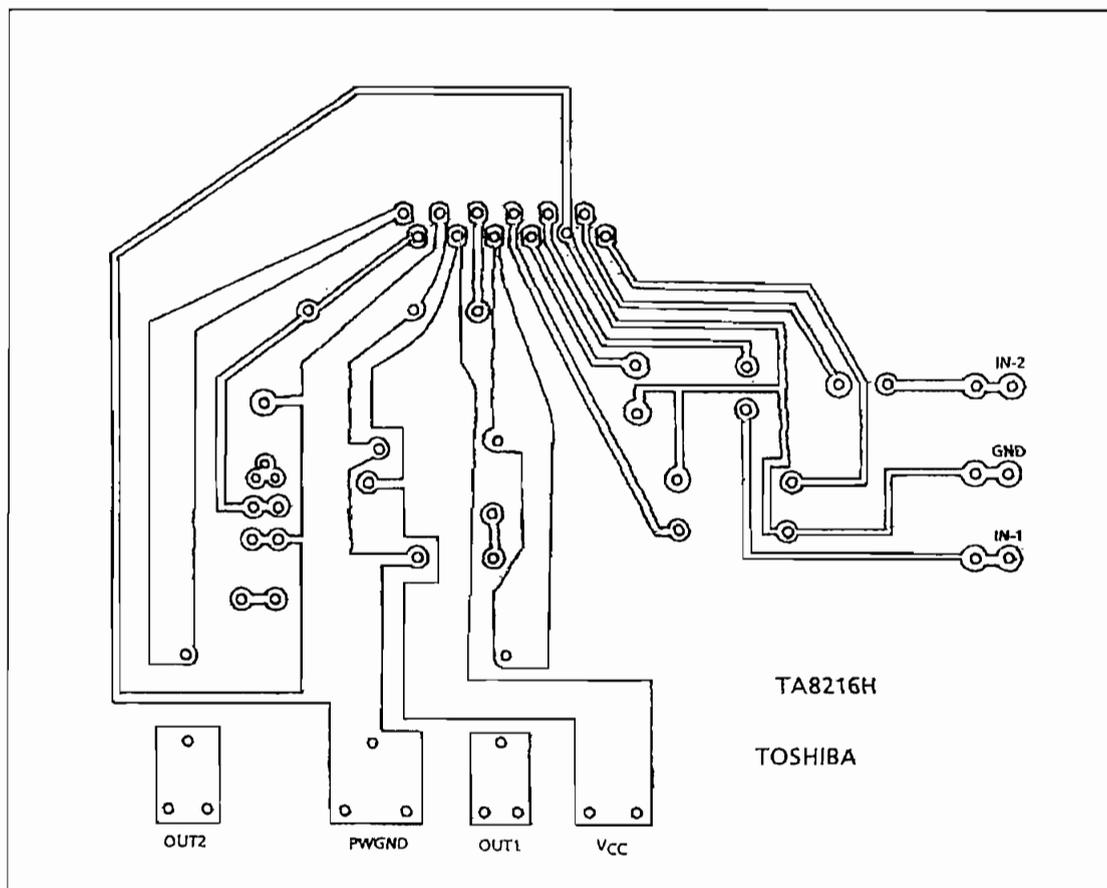


Fig.2

(3) Standard PCB TA8216H



(BOTTOM VIEW)

4. Precaution for 4Ω load resistance use

Internal output current detection and protection circuit protect the IC from the influence of unusual excess current. And this function causes the interrupted sound in case of excess input voltage with V_{CC} higher than recommended supply voltage (24V). Therefore, the power supply regulation must be fully investigated so as not to make the V_{CC} be high than recommendation supply voltage (24V).

TA8216H-4
1996-6-17
TOSHIBA CORPORATION

MAXIMUM RATINGS (Ta = 25°C)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Supply Voltage	V _{CC}	40	V
Output Current (Peak / Ch)	I _O (peak)	3.0	A
Power Dissipation	P _D (Note)	25	W
Operating Temperature	T _{opr}	-20~75	°C
Storage Temperature	T _{stg}	-55~150	°C

(Note) Derated above Ta=25°C in the proportion of 200mW/°C.

ELECTRICAL CHARACTERISTICS

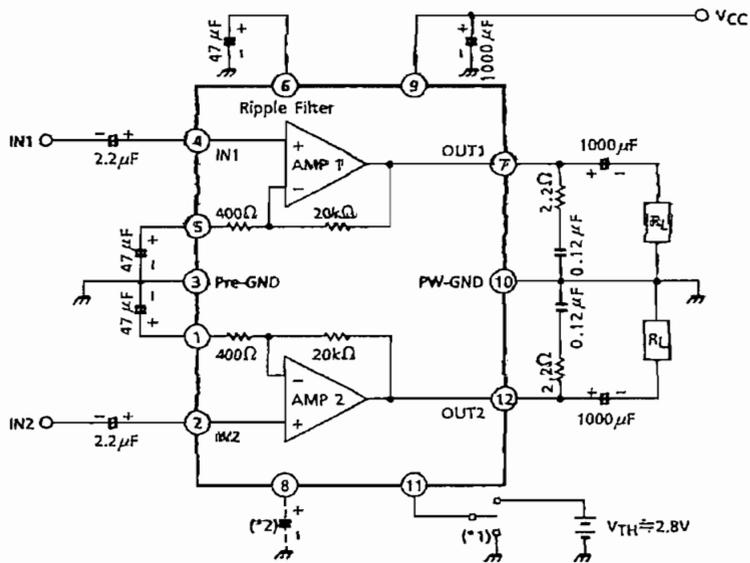
(Unless otherwise specified, V_{CC}=28V, R_L=8Ω, R_g=600Ω, f=1kHz, Ta=25°C)

CHARACTERISTIC	SYMBOL	TEST CIR-CUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Quiescent Current	I _{CCQ}	—	V _{in} = 0	—	50	105	mA
Output Power	P _{out} (1)	—	THD = 10%	10	13	—	W
	P _{out} (2)	—	THD = 1%	—	10	—	
	P _{out} (3)	—	THD = 10%, V _{CC} = 24V, R _L = 4Ω	—	13	—	
Total Harmonic Distortion	THD	—	P _{out} = 2W	—	0.04	0.2	%
Voltage Gain	G _v	—	V _{out} = 0.775V _{rms} (0dBm)	32.5	34.0	35.5	dB
Input Resistance	R _{IN}	—	—	—	30	—	kΩ
Ripple Rejection Ratio	R.R.	—	R _g = 0, f _{ripple} = 100Hz V _{ripple} = 0.775V _{rms} (0dBm)	-40	-50	—	dB
Output Noise Voltage	V _{no}	—	R _g = 10kΩ, BW = 20Hz~20kHz	—	0.14	0.3	mV _{rms}
Cross Talk	C.T.	—	R _g = 10kΩ, V _{out} = 0.775V _{rms} (0dBm)	—	-70	—	dB
Muting Threshold Voltage	V _{th} Ⓜ	—	—	2.6	2.8	—	V

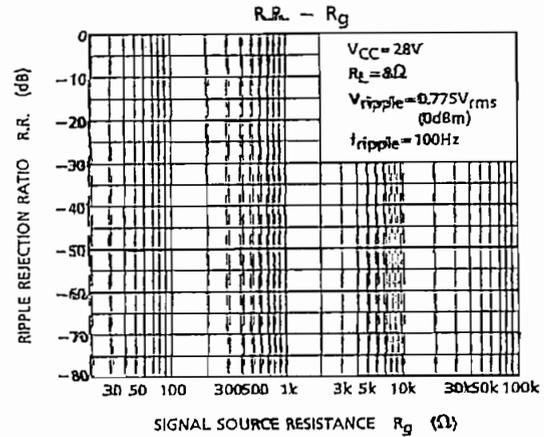
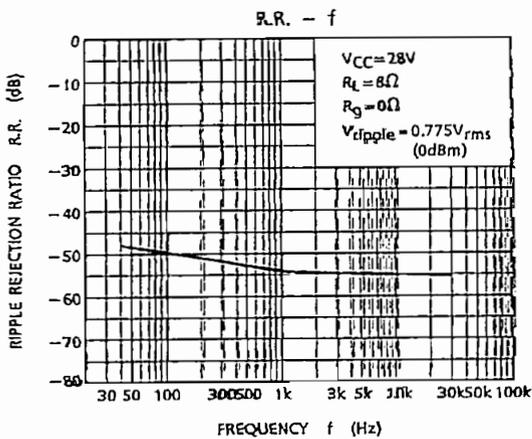
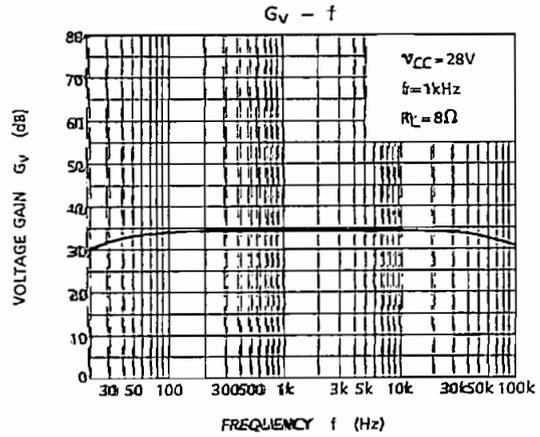
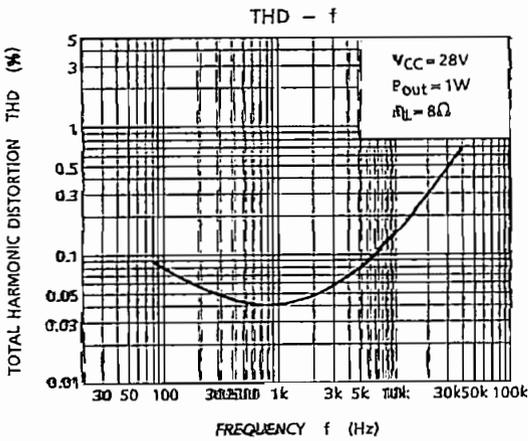
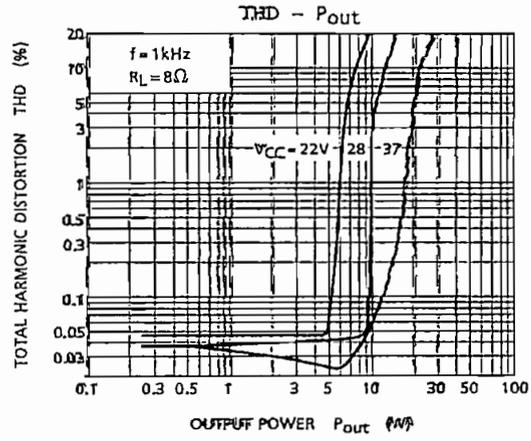
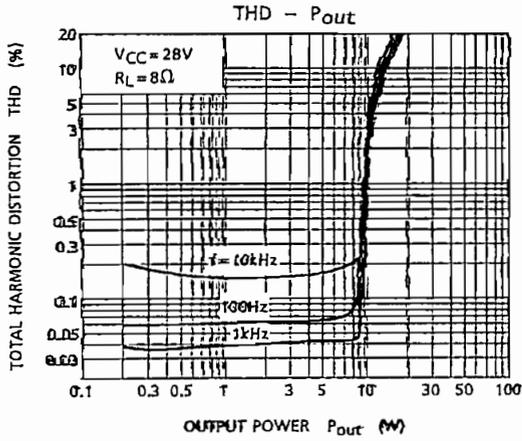
TYP. DC VOLTAGE OF EACH TERMINAL (V_{CC} = 28V, Ta = 25°C)

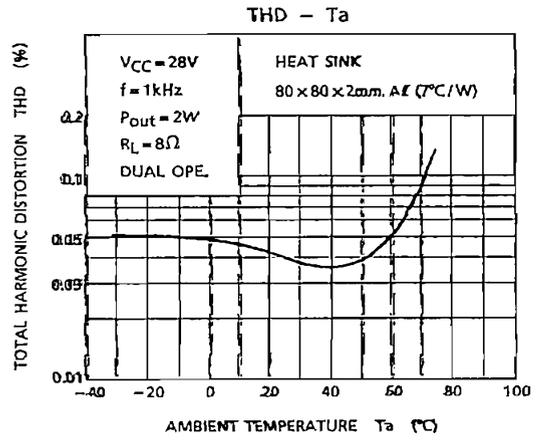
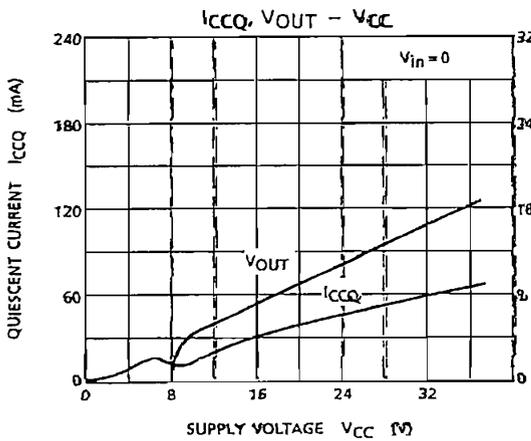
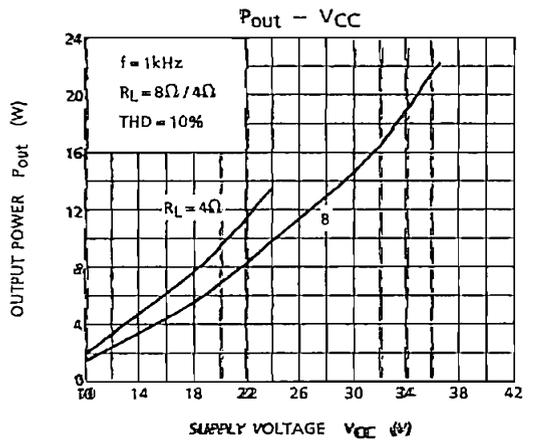
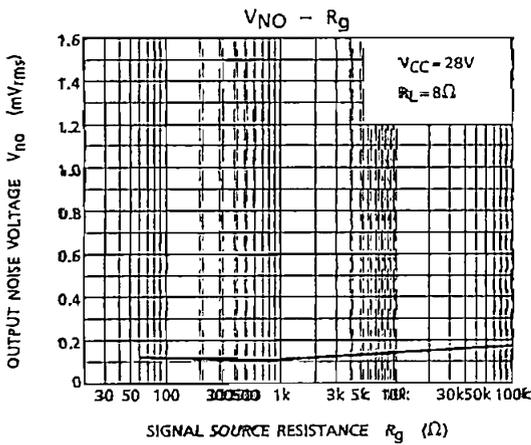
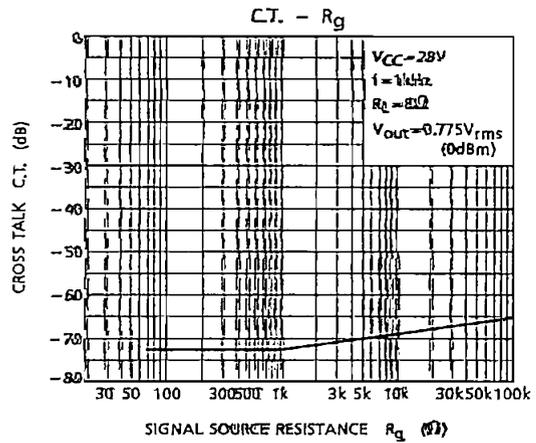
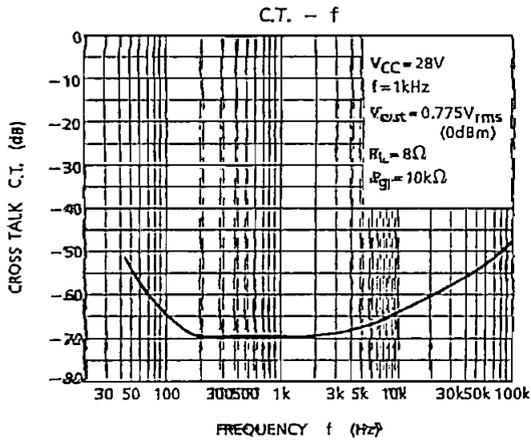
TERMINAL No.	1	2	3	4	5	6	7	8	9	10	11	12
DC Voltage (V)	1.6	20m	GND	20m	1.6	9.4	13.0	5.0	V _{CC}	GND	2.8	13.0

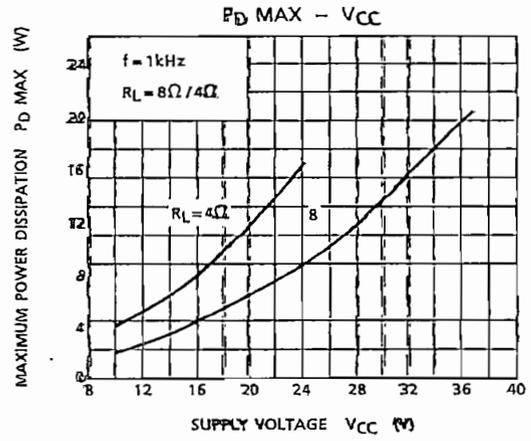
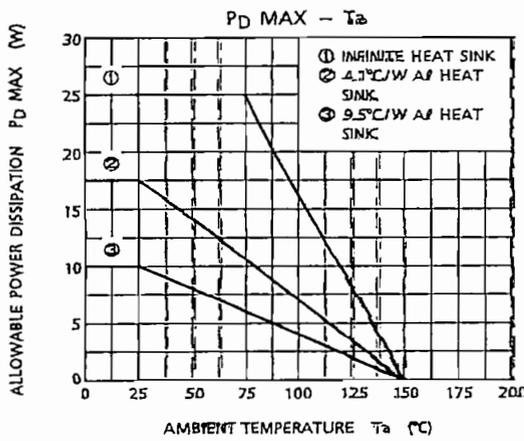
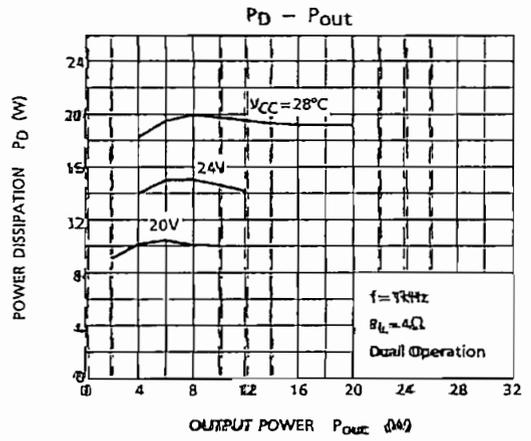
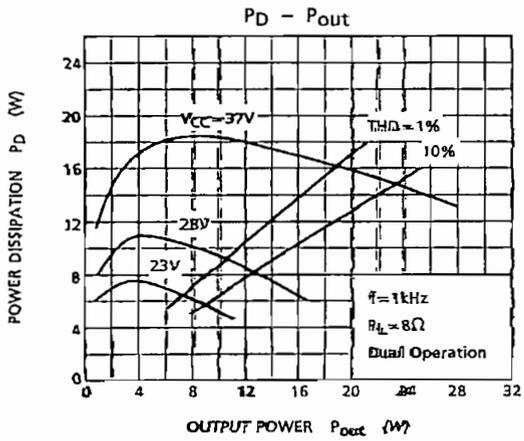
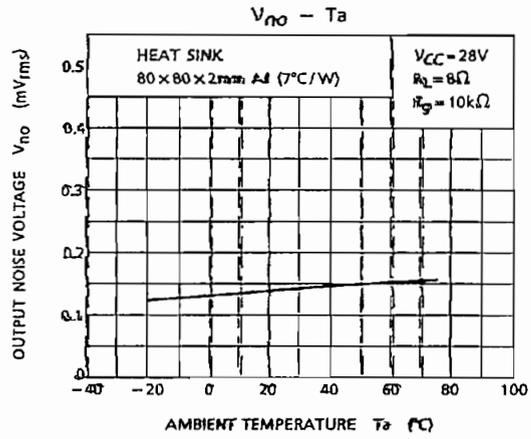
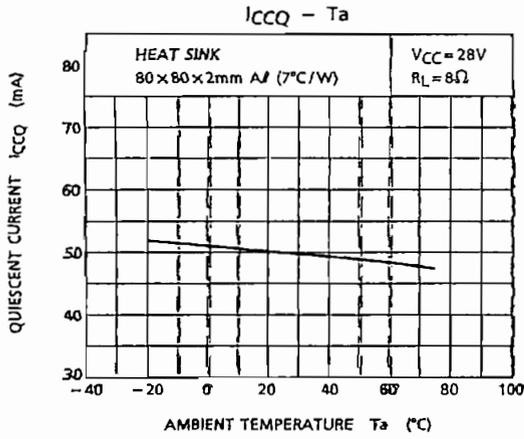
TEST CIRCUIT



- (*1) MUTE ON at ① pin LOW
 $V_{TH} = 2.8V$ (Typ.), $V_{CC} = 28V$, $T_a = 25^\circ C$
- (*2) The capacitor for reducing POP noise at mute ON.







ANEXO 6

HERRAMIENTAS DE MATLAB

fix(x)	Redondea los componentes del vector x al entero más cercano.
for	Repite declaraciones un número específico de tiempo, ver while.
format	Formato numerico (digitos significativos, exponente).
freqz(b,a,N)	Calcula la transformada Z de la respuesta de frecuencia de un filtro digital, ver filter.
function	Crea una función en un archivo de instrucciones.
grid	Dibuja una cuadrícula en el gráfico actual., ver axes.
gtext	Añade un texto actual al gráfico presente con el mouse, ver text.
hold	Mantiene el gráfico actual, ver figure.
if	Sentencia de ejecución condicional, ver for.
ifft(x,N)	Implementa un algoritmo FFT inverso base 2 de N puntos, ver fft.
imag(x)	Devuelve la parte imaginaria de un número complejo, ver real.
impz	Respuesta impulsiva de un sistema lineal de tiempo continuo, ver step.
input	Solicita una entrada de usuario, ver keyboard
inv(X)	Halla el inverso de la matriz X, ver pinv.
kaiser(M)	Genera una ventana Kaiser de M puntos.
laplace(x)	Genera variables al azar de una distribución de Laplace.
legend	Genera un gráfico con leyenda, ver plot.
length(x)	Longitud de un vector, ver size.
linspace	Devuelve un vector espaciado linealmente.
log(x)	Calcula el logaritmo natural de cada uno de los elementos de x.
log10(x)	Calcula el logaritmo en base 10 de cada uno de los elementos de x.
logspace	Devuelve un vector espaciado logarímicamente, ver linspace.
mean(x)	Calcula el valor medio del vector x, ver median.
margin	Devuelve el margen de ganancia, margen de fase y frecuencias de corte, ver bode.
norm(x)	Normaliza un vector, ver cond.
nyquist(sys)	Traza un diagrama de Nyquist de un sistema LTI, ver Bode
plot	Traza un gráfico, ver figure.
plot	Traza un gráfico, ver figure.
poly(r)	Calcula los coeficientes del polinomio p con raíces r.
print	Imprime el gráfico actual
zmap(sys)	Mapa de polos y ceros de un sistema lineal, ver rlocus.

Filter Design Toolbox 2

for designing and analyzing advanced floating-point and fixed-point filters

The Filter Design Toolbox is a collection of tools that provides advanced techniques for designing, simulating, and analyzing digital filters. It extends the capabilities of the Signal Processing Toolbox by adding filter architectures and design methods for complex real-time DSP applications. The Filter Design Toolbox also provides functions that simplify the design of fixed-point filters and the analysis of quantization effects.

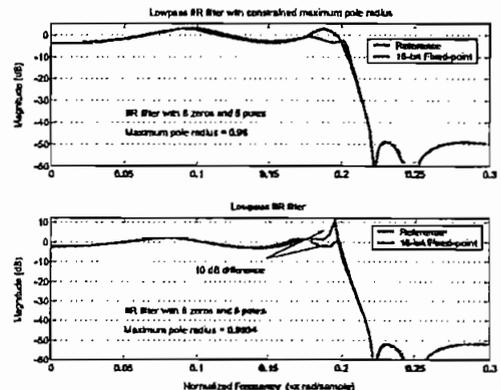
Working with the Filter Design Toolbox Advanced Filter Design

Many real-world systems require digital filters with stringent specifications. The toolbox provides advanced filter designs for these applications.

- The advanced equiripple FIR design automatically determines the minimum filter order required. It also provides constrained ripple, minimum phase, extra ripple, and maximal ripple designs.
- The least P-th norm FIR design optimizes your filter for any norm from 2 to infinity. It allows you to adjust the trade-off between minimum stopband energy and minimum order equiripple characteristics.
- Allpass IIR filter design with arbitrary group delay enables the equalization of nonlinear group delays of other IIR filters to obtain an overall approximate linear phase passband response.
- Least P-th norm IIR design creates optimal IIR filters with arbitrary magnitude. It lets you specify the numerator and denominator orders independently. It lets you optimize the design for any norm from 2 to infinity.
- Constrained least P-th norm IIR design constrains the maximum radius of the filter poles to improve the robustness of the quantization.

KEY FEATURES

- Advanced FIR filter design methods
- Advanced IIR filter design methods
- Filter conversions, including transfer function to coupled allpass and transfer function to coupled allpass lattice
- Quantization functions for signals, filters, and FFTs
- Quantization of filter coefficients (including coefficients created using the Signal Processing Toolbox)
- Analysis of quantized filters, including frequency response, pole-zero response, impulse response, group delay, step response, and phase response
- Noise loading methods for nonlinear frequency response analysis
- Limit cycle analysis for fixed-point IIR filters
- Design and analysis of filters from the Filter Design and Analysis Tool or from the command line



Comparison of quantization effects for two IIR filters. The top filter is designed with a constraint on the maximum pole radius of 0.98. The bottom filter has a maximum pole radius of 0.9934. The poles closest to the unit circle can severely affect the frequency response of the filter when it is quantized.

Sample Functions

Filter design

<code>firlpnorm</code>	Least P-norm optimal FIR filter design
<code>gremez</code>	Generalized REMEZ FIR filter design
<code>iirgrpdelay</code>	Allpass filter design given a group delay
<code>iirlpnorm</code>	Least P-norm optimal IIR filter design
<code>iirlpnormc</code>	Constrained least P-norm IIR filter design

Filter conversions

<code>ca2tf</code>	Coupled allpass-to-transfer function
<code>cl2tf</code>	Coupled allpass-lattice-to-transfer function
<code>iirpowcomp</code>	IIR power complementary filter
<code>tf2ca</code>	Transfer function to coupled allpass
<code>tf2cl</code>	Transfer function to coupled allpass lattice

Quantized object construction

<code>qfft</code>	Construct a quantized FFT object
<code>qfilt</code>	Construct a quantized filter object
<code>quantizer</code>	Construct a quantizer object
<code>unitquantizer</code>	Construct a unit quantizer object

Bit-true simulation

<code>fft</code>	Quantized fast Fourier transform (FFT)
<code>filter</code>	Quantized filtering
<code>ifft</code>	Quantized inverse FFT

Quantized filter analysis

<code>freqz</code>	Quantized filter Z-transform frequency response
<code>impz</code>	Quantized filter impulse response
<code>isallpass</code>	True for allpass filter
<code>isfir</code>	True for FIR filter
<code>islinphase</code>	True for linear phase filter
<code>ismaxphase</code>	True for maximum phase filter
<code>isminphase</code>	True for minimum phase filter
<code>issos</code>	True for filter with second-order sections
<code>isstable</code>	True for stable filter
<code>limitcycle</code>	Detect limit cycles in quantized filter
<code>nlim</code>	Noise-loading method estimate of frequency response
<code>zplane</code>	Quantized filter Z-plane pole-zero plot

Quantized filter conversion

<code>convert</code>	Convert from one filter structure to another
<code>sos</code>	Convert to second-order section form and scale

Quantized statistics

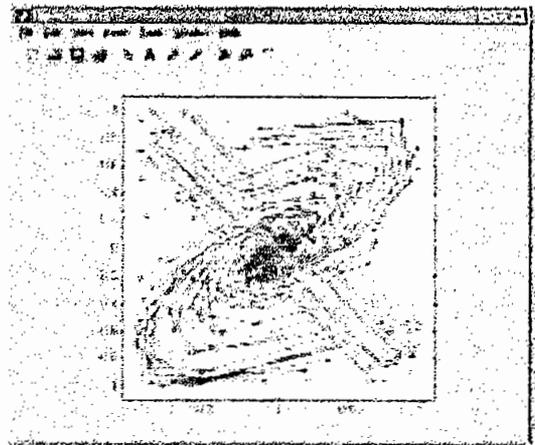
<code>max</code>	Maximum value attained during quantized operation
<code>min</code>	Minimum value attained during quantized operation
<code>noperations</code>	Number of quantization operations
<code>noverflows</code>	Number of overflows
<code>nunderflows</code>	Number of underflows
<code>qreport</code>	Display quantization report

Common quantized object properties

<code>coefficientformat</code>	Coefficient format
<code>inputformat</code>	Input format
<code>multiplicandformat</code>	Multiplicand format
<code>numberofsections</code>	Number of sections
<code>outputformat</code>	Output format
<code>productformat</code>	Product format
<code>scalevalues</code>	Scale values at input to sections
<code>sumformat</code>	Sum format

Quantized property manipulation

<code>get</code>	Get object properties
<code>reset</code>	Reset quantizer states
<code>set</code>	Set object properties
<code>setbits</code>	Set all quantized data property values



Progression of states in the limit cycle analysis of a second-order, fixed-point filter with random initial conditions and zero input. Some initial states end in large-scale limit cycles oscillating between (0.8, -0.8) and (-0.8, 0.8).

Quantized FFT-specific properties

length Length of FFT
 radix Radix of FFT

Quantized FFT methods

tostring QFFT object to string
 twiddle Return twiddle factors associated with QFFT

Quantized filter-specific properties

filterstructure Filter structure
 quantizedcoefficients Quantized coefficients
 referencecoefficients Reference coefficients
 statespersection States per section

Quantized filter methods

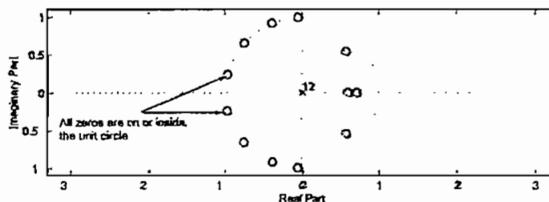
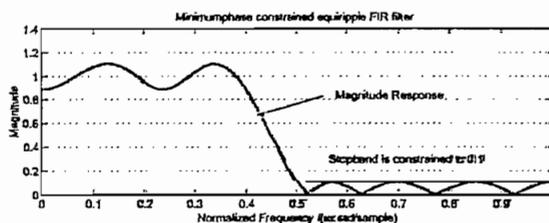
normalize Normalize filter coefficients
 num2bin Quantized coefficients to binary string
 num2hex Quantized coefficients to hex string
 num2int Quantized coefficients to decimal integers
 order Filter order
 qfilt2tf Quantized filter-to-transfer function

Quantizer object properties

format Format of quantizer
 mode Mode of quantizer
 overflowmode Overflow mode of quantizer
 roundmode Round mode of quantizer

Quantizer attributes derived from properties

denormalmax Largest denormalized quantized number
 denormalmin Smallest denormalized quantized number
 eps Quantized relative accuracy
 exponentbias Exponent bias of quantizer
 exponentlength Exponent length of quantizer
 exponentmax Maximum exponent
 exponentmin Minimum exponent
 fractionlength Fraction length
 range Numerical range of quantizer
 realmax Largest positive quantized number
 realmin Smallest positive, normalized quantized number
 wordlength Word length



This plot shows the magnitude response and pole-zero plot of a minimum phase equiripple FIR filter whose stopband ripple has been constrained so as not to exceed 0.1.

Quantizer numeric conversions

bin2num Binary string to numeric
 hex2num Hex to numeric
 num2bin Numeric to binary
 num2hex Numeric to hex
 num2int Numeric to decimal integer

Quantizer methods

quantize Quantize numeric data
 randquant Uniformly distributed quantized random number
 round Round with quantizer but do not overflow
 tostring Quantizer object to string
 unitquantize Quantize except numbers within eps of 1