GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGIA

TRABAJO DE FIN DE GRADO

DISEÑO DE UN ECUALIZADOR DE CUATRO BANDAS

AUTOR: SUNNY MOOLCHAND MORYANI MORYANI

TUTOR: FRANCISCO JAVIER LLOPIS CÁNOVAS

Agradecimientos

A mi familia, amigos, compañeros de clase y docentes que han aportado su pequeño granito de arena a lo largo de todo este tiempo, lo que me ha ayudado a crecer no solo como persona, sino también como profesional.

Quiero hacer una mención especial a Francisco Javier Llopis, tutor del presente proyecto por su gran ayuda, comprensión y colaboración en el desarrollo del mismo.

Resumen

El presente proyecto tiene como objetivo llevar a cabo el diseño de un ecualizador de 4 bandas. Para ello se han estudiado las distintas posibilidades y se han analizado soluciones analógicas además de implementarse un banco de filtros digitales.

La función principal del ecualizador es recibir una señal de audio y tratarla en el dominio de la frecuencia con arreglo a los requisitos que impone el usuario obteniendo así a la salida una señal de audio personalizada.

En primer lugar, se analiza la respuesta de dos tipos de ecualizadores analógicos. El funcionamiento del filtro biquad analógico se verifica además en el laboratorio. Por otra parte, se implementa un banco de filtros biquad digitales empleando un software de diseño digital cuya respuesta también se ha verificado experimentalmente.

Abstract

The aim of this Project is the design of a four band equalizer. For this, different possibilities have been studied and analog solutions have been analysed as well as implementing a bank of digital filter.

The main function of the equalizer is to receive an audio signal and process it in the frequency domain according to the users requirements, obtaining as a result a customized audio signal at the output.

First, the response of two types of analog equalizers is analysed. The performance of the analog biquad filter is also verified in the laboratory. On the other hand, a digital biquad filter bank is implemented using digital design software whose response has also been verified experimentally.

INDICE

1. INTRODUCCIÓN

El oído es uno de los órganos más importantes del cuerpo humano, un órgano sensible y bastante avanzado el cual transmite los sonidos al cerebro a través de sus distintas partes. La tarea principal del oído es detectar, transmitir y convertir los sonidos del mundo exterior en impulsos eléctricos que posteriormente serán percibidos por el cerebro.

Los sonidos percibidos por el oído se reparten en un espectro de frecuencias comprendidos aproximadamente entre los 19 Hz y los 20 kHz, siendo este último el tono más alto y la frecuencia de 19 Hz el tono más bajo.

Existen ciertos tonos a los cuales es más sensible el oído, que pueden provocar placer auditivo y otros los pueden resultar bastante molestos e incluso llegar a dañar el oído.

En la actualidad existen distintas herramientas que ayudan a aumentar la presencia de dichos tonos, reducirla o incluso eliminarlos en el caso de que sean molestos.

Una de las herramientas más usada en la actualidad es el ecualizador, el cual gracias a sus controles nos facilita la modificación de la señal de entrada para obtener a la salida una señal adaptada a las necesidades del usuario. Dicha herramienta ha sido bastante útil en el campo del sonido profesional, facilitando el trabajo a los ingenieros de sonido durante mucho tiempo para adaptar las señales de audio grabadas a un público bastante amplio.

Algunos de los ecualizadores más destacados a lo largo de la historia de la música como el API 550a de la famosa marca API Audio, Puilteq EQP-1A de la marca Manley Labs, el SSL G EQ de la marca Solid State Logic han sido empleados a lo largo de la historia para procesar algunas de las canciones más destacadas a partir de la década de los 70 del siglo XX y durante lo que llevamos del siglo XXI.

Figura 1.1 Ecualizador API 550A

Figura 1.2 Ecualizador Puilteq EQP-1

En este proyecto se estudiarán los distintos sistemas de ecualización del sonido, así como el diseño de un sistema de ecualización básico de 4 bandas que se podría implementar en una mesa de mezclas analógica. Dicho ecualizador será estudiado y diseñado en primer lugar de forma teórica y tras conseguir un diseño optimo se llevará a cabo un diseño real del mismo.

El circuito incorporará cuatro controles a través de los cuales se podrá atenuar o aumentar la ganancia en una banda determinada, variando de forma independiente la intensidad de los distintos tonos percibidos por el oído humano.

2. MOTIVACIÓN

En la actualidad, la instrumentación en el campo del procesado del sonido está bastante desarrollada respecto a la desarrollada durante los años sesenta, si tenemos en cuenta las limitaciones técnicas a las que se enfrentaban los productores e ingenieros por aquel entonces. Por poner un ejemplo, el sintetizador de Robert Moog, se implementó empleando exclusivamente bloques analógicos, ocupando mucho espacio y con grandes dificultades en el proceso de mezcla y masterización. Con la llegada de las nuevas tecnologías y los ordenadores, con gran capacidad a la hora de procesar datos, se han ido arrinconando aquellos clásicos instrumentos analógicos. Sin embargo, hay quienes piensan (en el campo del procesado al igual que en el de la producción) que se ha ido degradando el nivel de creatividad al tener que realizar todo el procesamiento *in the box* (*en la caja,* traducción literal al castellano), lo que hace referencia a llevar a cabo todo el proceso con un mismo ordenador.

Este proyecto nace de la idea de satisfacer la necesidad básica de cualquier músico o banda que desee obtener unos resultados adecuados a las necesidades sin perder la esencia del sector, que es la creatividad.

3. OBJETO

El objetivo de este proyecto es diseñar un sistema de ecualización de audio de 4 bandas para conseguir, modificando los distintos tonos del espectro auditivo, un sonido más adaptado a las necesidades del usuario. Dichas modificaciones consisten en la atenuación o el aumento de la ganancia de los tonos según la necesidad del usuario.

El sistema de ecualización diseñado se basa en el ecualizador gráfico, en el que cada una de las distintas bandas está controlada por un filtro que permite al usuario la atenuación o aumento de la ganancia de la frecuencia central, eliminando así las frecuencias que no nos interesan y aumentando la presencia de la frecuencia que nos interesa, obteniendo como resultado un sonido con mayor calidad.

El software usado para el diseño del ecualizador digital ha sido el Teensyduino y el lenguaje de programación usado en este caso ha sido el C++.

4. CONCEPTOS BÁSICOS

4.2 Frecuencia

La frecuencia, en términos de sonido, es la cantidad de vibraciones por unidad de tiempo que transmite una fuente. Se mide en ciclos por segundo o, lo que es lo mismo en Hercios. A mayor cantidad de ciclos por segundo de la vibración el oído percibe, el sonido es más agudo; cuanto menor sea el número de ciclos por segundo percibe el oído, es más grave [*1*].

4.3 Frecuencia central

Es la frecuencia a la cual la ganancia es mínima si se atenúa las frecuencias de una banda y máxima si se amplifican las mismas. Conforme se alejan frecuencias de dicha frecuencia central, menor es el efecto de atenuación o amplificación *[2].*

4.4 Respuesta en frecuencia

La respuesta en frecuencia nos indica como varia la amplitud y la fase de la salida al aplicar un estímulo a la entrada e ir variando la frecuencia. Cuanto más amplia sea la respuesta en frecuencia de un equipo, mayor calidad tendrá el sonido a la salida ya que pierde menos información. En los sistemas con respuesta plana se trata por igual a todo el sonido entrante, lo que hace que la salida tenga las mismas características *[3].*

4.5 Función de transferencia

La función de transferencia de un sistema es un modelo matemático que relaciona la respuesta del sistema con una señal de entrada o excitación. Dicha relación se da a través de un cociente. En el numerador se encuentra la transformada de Laplace de la salida y en el denominador la transformada de Laplace de la entrada. *[4]:*

$$H(s) = \frac{Y(s)}{X(s)}$$

5. RANGO AUDITIVO HUMANO

El rango auditivo del ser humano es el espectro de frecuencias audibles por los seres humanos, a diferencia de otros seres vivos, encontrándose por encima de este los ultrasonidos y por debajo los infrasonidos. Los seres humanos a diferencia de otros seres vivos disponemos de un rango de auditivo limitado, dicho rango varía según las personas y la edad, comprendiendo de forma general desde los 20 Hz hasta los 20 kHz, lo que equivale a 10 octavas completas.

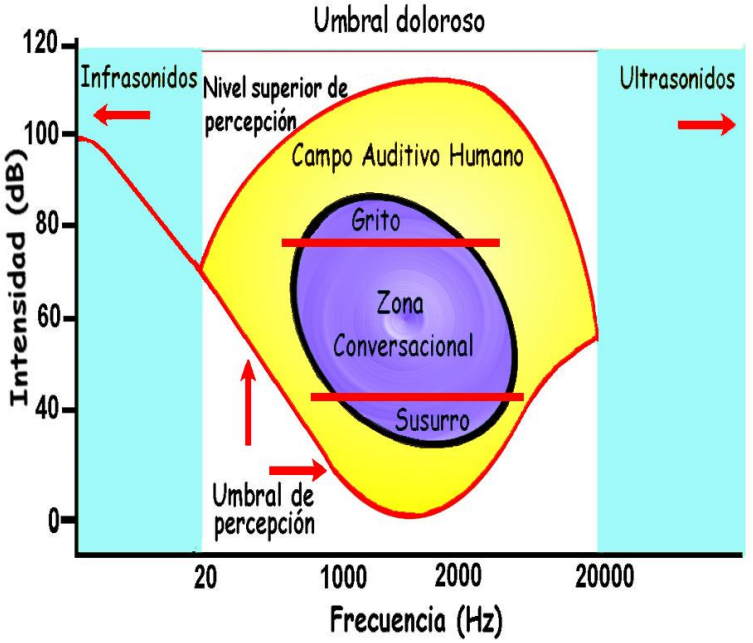


Figura 5.1 Rango auditivo del ser humano

Dentro de dicho rango de frecuencias, el oído humano es más sensible a los sonidos con frecuencias comprendidas entre 1 kHz y 5 kHz.

6. INTRODUCCIÓN A LOS SISTEMAS DE ECUALIZACIÓN DE AUDIO.

Los sistemas de ecualización de audio son sistemas se diseñan con la única finalidad de modificar ciertos parámetros de un determinado rango de frecuencias o frecuencias individuales, para obtener a la salida un espectro de audio personalizado, dependiendo de las necesidades de los usuarios.

Dichas frecuencias son modificadas a través de filtros, que se encargan de atenuar, reforzar o eliminar frecuencias dependiendo con arreglo a las preferencias del usuario. Los filtros básicos más usados a la hora de ecualizar son los presentados a continuación.

6.2 Filtros activos

Son filtros que se caracterizan por usar como mínimo un componente activo que normalmente suele ser un amplificador operacional. La eliminación del inductor y el añadir un amplificador operacional hace que sea posible obtener prácticamente cualquier tipo de respuesta deseada, ya que el inductor es el elemento de circuitería menos ideal además de ser un componente que destaca por su elevado coste. Los tipos de filtros activos más útiles en la actualidad son los estudiados en el apartado 6.4 de este proyecto *[5]*.

6.3 Filtros pasivos

Son filtros analógicos que se caracterizan por incorporar únicamente componentes pasivos como pueden ser las resistencias, los condensadores y las bobinas. Para su funcionamiento no es necesario emplear fuentes de alimentación como ocurre con los filtros activos *[6]*.

6.4 Tipos de sistemas de ecualización de audio.

En la actualidad existen distintos tipos de sistemas de ecualización de audio que se pueden implementar dependiendo de sus aplicaciones. Los sistemas de ecualización más usados son los estudiados a continuación.

6.4.1 Filtro pasa baja *(Low Pass Filter)*

El filtro pasa bajo es un filtro que dispone de dos parejas de terminales, una a la entrada y otro a la salida. Dicho filtro se caracteriza por dejar pasar las frecuencias que se encuentran por debajo de la frecuencia de corte del rango de frecuencias con el que estemos trabajando y atenúa las frecuencias que se encuentren por encima de la frecuencia de corte de la misma.

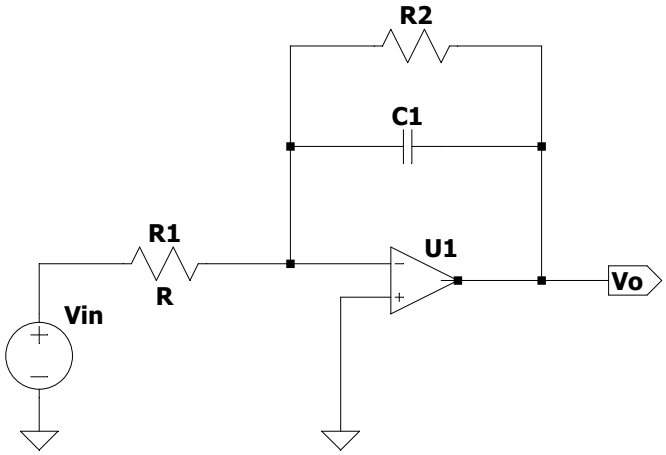


Figura 6.4.1º Circuito del filtro pasa bajas de 1º orden

La función de transferencia del filtro activo de primer orden tiene la siguiente forma,

$$H(s) = -\frac{R_2}{R_1}\frac{1}{R_2Cs+1} \qquad (6.4.1a)$$

Expresión que también se podría poner de la siguiente forma normalizada:

$$H(j\omega) = H_0\frac{1}{(1+j\omega/\omega_0)} \qquad (6.4.1b)$$

Donde

$$H_0 = -\frac{R_2}{R_1}$$

$$\omega_0 = \frac{1}{R_2C}$$

En la siguiente imagen se puede obtener una interpretación gráfica de la respuesta en frecuencia de dicho tipo de filtro.

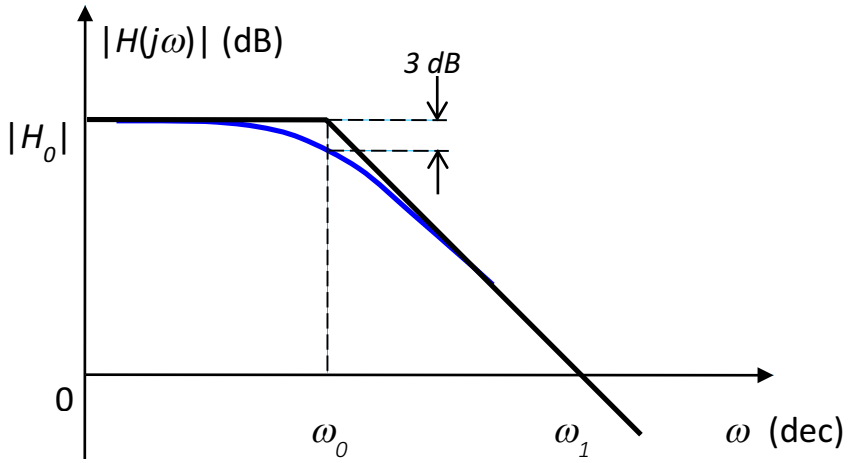


Figura 6.4.1b Respuesta en frecuencia Filtro Pasa Baja (LPF)

Este tipo de ecualización es bastante usado en el campo del audio, eliminando las frecuencias altas y obteniendo a la salida una señal de audio solo con frecuencias medias y bajas, lo que quiere decir que obtendremos a la salida tonos medios y bajos.

6.4.2 Filtro pasa alta *(High Pass Filter)*

El filtro pasa alta es un filtro que al igual que el filtro anterior dispone 2 terminales. En este caso este filtro se caracteriza por dejar pasar las frecuencias que se encuentran por encima de la frecuencia de corte y atenuar aquellas frecuencias que se encuentran por debajo de la frecuencia de corte del rango de frecuencias con el que estemos trabajando.

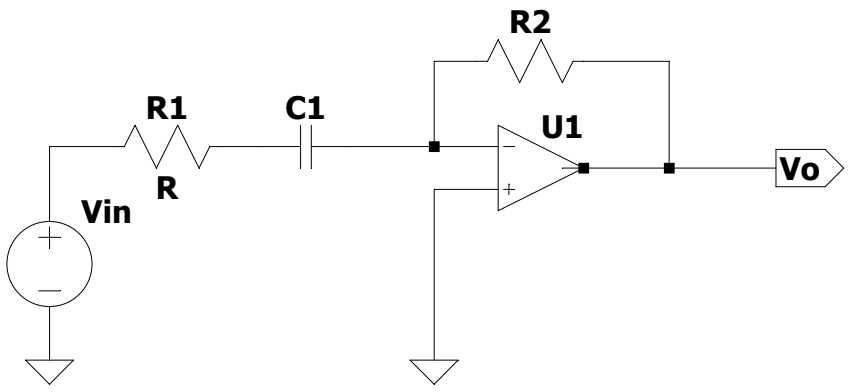


Figura 6.4.2a Circuito del filtro pasa altas de 1º orden

La función de transferencia del filtro pasa alta de primer orden tiene la siguiente forma:

$$H(s) = -\frac{R_2}{R_1}\frac{R_1 C_1 s}{R_1 C_1 s+1} \qquad (6.4.2a)$$

Dicha expresión también se podría poner de la siguiente forma normalizada:

$$H(j\omega) = H_0 \frac{j\omega/\omega_0}{(1+j\omega/\omega_0)} \qquad (6.4.2b)$$

donde,

$$H_0 = -\frac{R_2}{R_1}$$

$$\omega_0 = \frac{1}{R_1 C}$$

En la *figura 6.4.2b* se puede obtener una interpretación gráfica de la respuesta en frecuencia del filtro pasa alta.

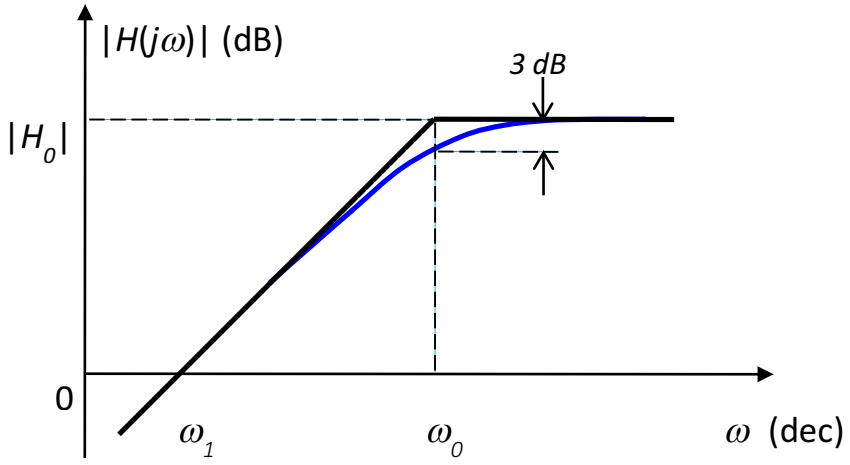


Figura 6.4.2b Respuesta en frecuencia Filtro Pasa Altas (HPF)

Este tipo de ecualización, al igual que el filtro pasa baja es bastante usado en el campo del audio, eliminando los tonos bajos y obteniendo a la salida una señal de audio solo con tonos medios y altos.

6.4.3 Filtro pasa banda *(Band Pass Filter)*

El filtro pasa banda es un filtro que al igual que los filtros anteriores tiene dos terminales, uno de entrada y otro de salida. Dicho filtro se caracteriza por dejar pasar un rango de frecuencia y atenuar el resto de las frecuencias del rango con el que se esté trabajando. A diferencia de los filtros anteriores, el filtro pasa banda (figura 6.4.3a) cuenta con 2 frecuencias de corte distintas, la frecuencia de corte superior (a partir de la cual se atenúan las señales) y la frecuencia de cote inferior (Se atenúan las señales que se encuentren por debajo de dicha frecuencia).

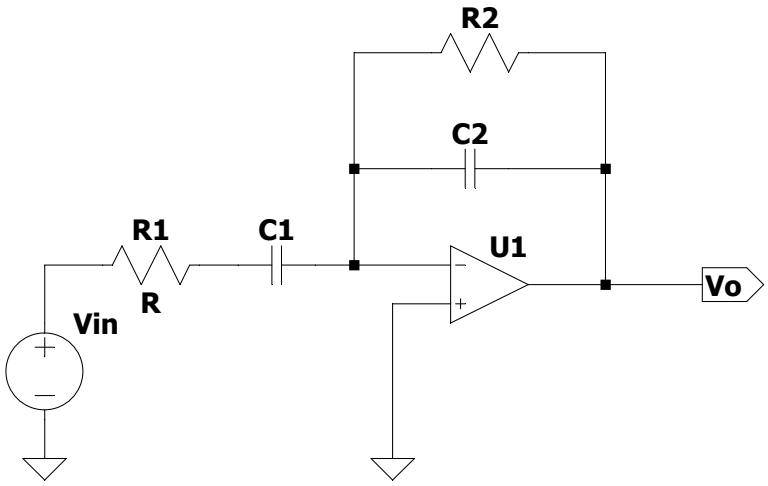


Figura 6.4.3a Respuesta en frecuencia Filtro Pasa Banda (BPF)

La función de transferencia de dicho tipo de filtro pasa banda de primer orden tiene la siguiente forma:

$$H(s) = -\frac{R_2}{R_1}\frac{R_1C_1s}{R_1C_1s+1}\frac{1}{R_2C_2s+1} \qquad (6.4.3a)$$

Dicha expresión también se podría poner de la siguiente forma normalizada:

$$H(j\omega) = H_0\frac{j\omega/\omega_L}{(1+j\omega/\omega_L)(1+j\omega/\omega_H)} \qquad (6.4.3b)$$

Donde

$$H_0 = -\frac{R_2}{R_1}$$

$$\omega_L = \frac{1}{R_1 C_1}$$

$$\omega_H = \frac{1}{R_2 C_2}$$

H_0 representa la ganancia en la banda de paso.

En la figura 6.4.3 se puede obtener una interpretación gráfica de la respuesta en frecuencia del filtro pasa banda de primer orden. El circuito que hemos presentado se puede considerar como un filtro pasa banda de banda ancha.

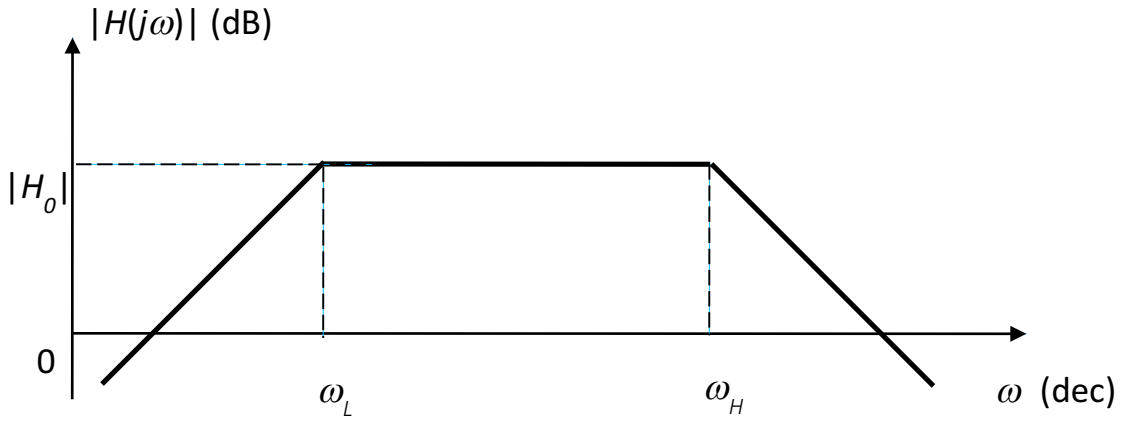


Figura 6..4.3b Respuesta en frecuencia Filtro Pasa Banda (BPF)

Este tipo de ecualización es bastante usado en el campo del audio, eliminando los tonos bajos y altos, obteniendo a la salida una señal de audio solo con tonos medios.

6.4.4 Filtro elimina banda *(Notch Filter)*

El filtro elimina banda, como bien indica su nombre se caracteriza por eliminar el rango de frecuencias que se encuentran ente la frecuencia de corte superior y la frecuencia de corte inferior, dejando intacta el resto de las frecuencias del rango de frecuencias con el que nos encontremos trabajando. Al igual que el filtro pasa band dispone 2 terminales, uno de entrada y otro de salida, así como 2 frecuencias de corte.

En la figura 6.4.4 se puede obtener una interpretación gráfica de la respuesta en frecuencia de dicho tipo de filtro.

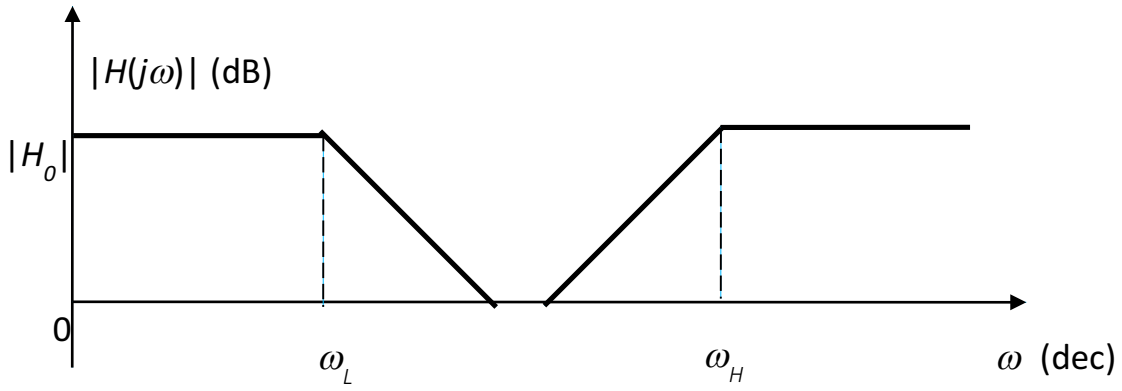


Figura 6.4.4 Respuesta en frecuencia Filtro Elimina Banda (NPF)

Este tipo de ecualización es bastante usado en el campo del audio, eliminando los tonos medios, obteniendo a la salida una señal de audio solo con tonos bajos y altos. Los filtros *notch* (de muesca o ranura), que se pueden considerar como un caso particular de los filtros elimina-banda, presentan una respuesta más selectiva *[5].* Se pueden emplear para atenuar señales indeseadas que aparecen debido a interferencias (por ejemplo, señales de 50 Hz provenientes de la red).

6.4.5 Control activo de tonos *(Active Tone Control)*

El un sistema de ecualización básico que permite el ajuste independiente de la ganancia de un rango de frecuencias determinado. La forma más común de encontrar este sistema de ecualización es el control activo de tonos bajos y altos. Dicho circuito tiene la siguiente forma:

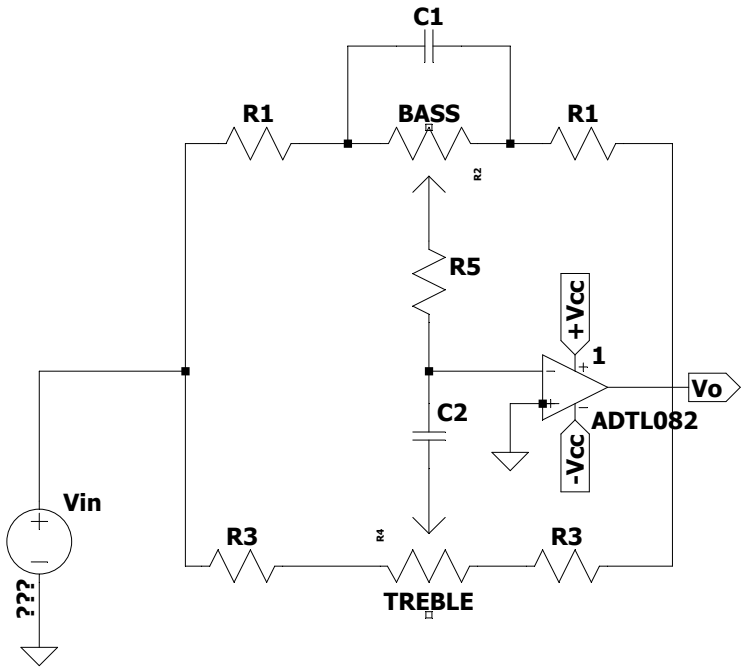


Figura 6.4.5a Circuito Active Tone Control

Para las frecuencias bajas, el capacitor actúa como un circuito abierto, lo que provoca que la única realimentación que es efectiva en este caso es la formada por las resistencias R_1 y R_2. Para las frecuencias altas C_1 anula la respuesta de R_2 hasta que forma un corto hasta que no tiene efecto en la respuesta *[7]*.

La respuesta en frecuencia de dicho sistema de ecualización es la representada es la siguiente:

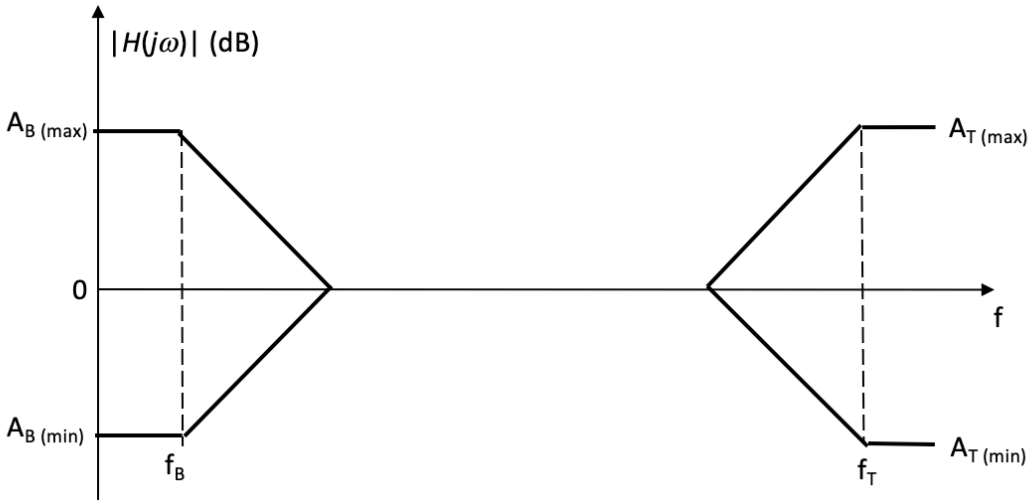


Figura 6.4.5b Respuesta en frecuencia de ecualizador Active Tone Control

La función de transferencia de dicho tipo de filtro tomará la forma de un filtro pasa bajas cuando actúa sobre las frecuencias altas mientras que cuando actúe sobre las frecuencias bajas tomará la forma del filtro pasa altas

6.4.6 Filtro Biquad *(Digital Biquad Filter)*

Los filtros biquad o también llamados de *Tow-Thomas,* son filtros recursivos, es decir que usan una o varias salidas como entradas. Este tipo de filtros son bastantes estables cuando son de primer o segundo orden y menos estable conforme aumenta el orden del filtro. A diferencia de los filtros pasa banda de primer orden, los filtros de segundo orden se comportan de una forma más selectiva cerca de la frecuencia central.

La figura 6.4.6a representa el circuito característico de dicho filtro, el primer operacional actúa como filtro pasa banda de segundo orden, el segundo como pasa baja y el tercer y último operacional actúa como inversor *[8]*.

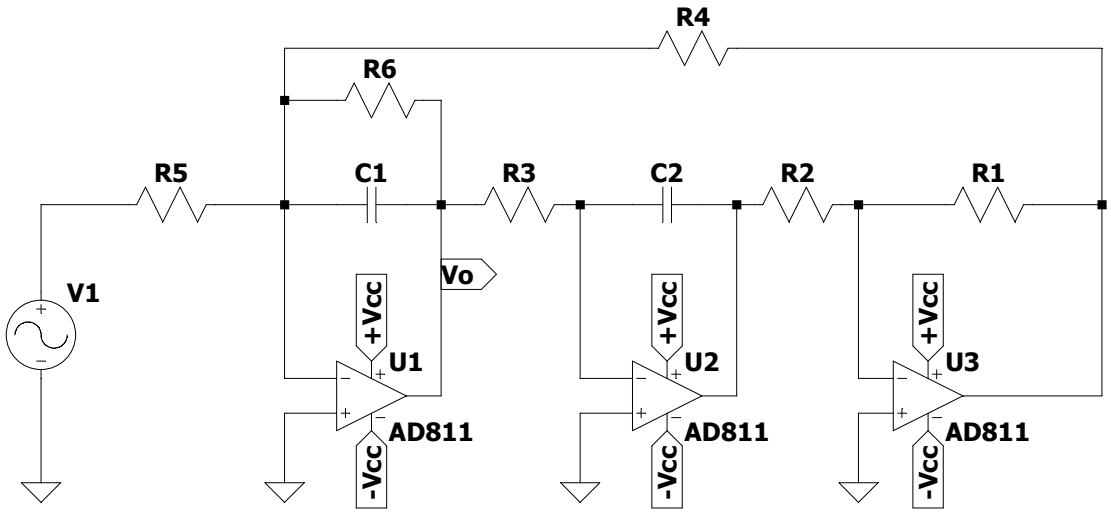


Figura 6.4.6a Circuito filtro Tow Thomas o filtro digital

Donde;

$$H_{0BP} = -\frac{R_2}{R_1} \qquad (6.4.6a)$$

$$H_{0LP} = \frac{R}{R_1} \qquad (6.4.6b)$$

$$\omega_0 = \frac{1}{RC} \qquad (6.4.6c)$$

$$Q = \frac{R_2}{R} \qquad (6.4.6d)$$

6.4.7 Ecualizadores gráficos *(Graphic Equalizer)*

Los ecualizadores gráficos son los sistemas de ecualización diseñados no solo para reducir o impulsar la ganancia de las frecuencias bajas o altas sino de cualquier banda de frecuencia. Una de las ventajas más importantes de este tipo de ecualizador es la posibilidad de trabajar con distintas bandas simultáneamente, teniendo el usuario la posibilidad al usuario de poder modificar la ganancia en cada banda.

Este tipo de ecualizador está compuesto de distintos filtros pasa banda (ver figura 6.4.6a.) en los cuales el valor que se modifica es el de la resistencia R_2 (Se trata de un potenciómetro) para variar la ganancia a la frecuencia central y se eligen el valor de los condensadores para que la frecuencia central tenga el valor que elige el usuario.

El circuito básico de un ecualizador gráfico emplea varias secciones, a continuación, se puede observar una sección del circuito del ecualizador gráfico véase la figura 6.4.7a.

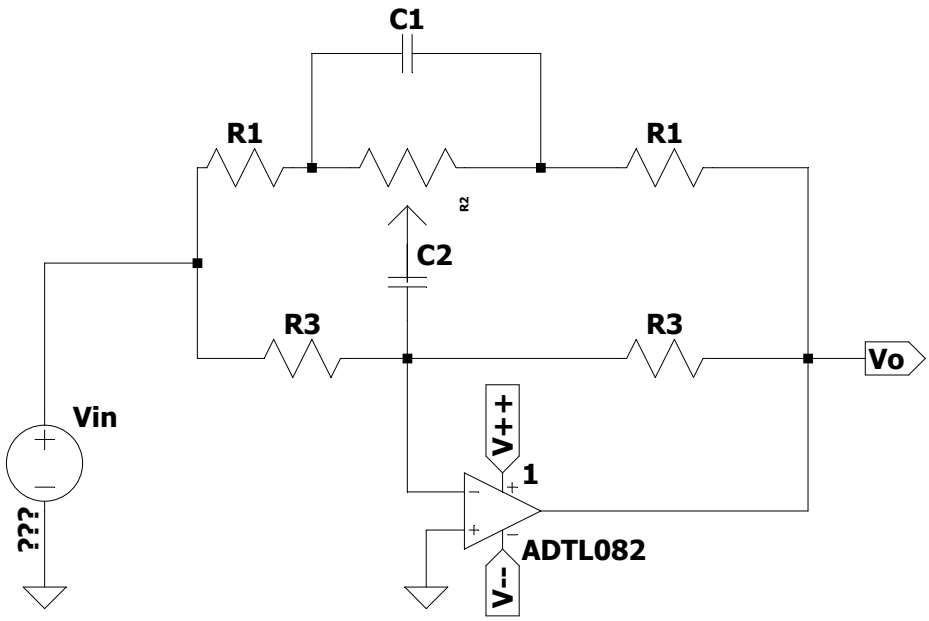


Figura 6.4.7a Sección de un circuito ecualizador gráfico

Según Sergio Franco, la frecuencia central viene dada por la siguiente expresión:

$$f_0 = \frac{\sqrt{2+R_2/R_1}}{20\pi R_2 C_2} \qquad (6.4.7\text{a})$$

A continuación, la ganancia de dicha frecuencia central viene dada por la siguiente expresión:

$$\frac{3R_1}{3R_1+R_2} \leq A_0 \leq \frac{3R_1+R_2}{3R_1} \qquad (6.4.7\text{b})$$

La respuesta en frecuencia de dicho tipo de ecualizador suele tener la siguiente forma:

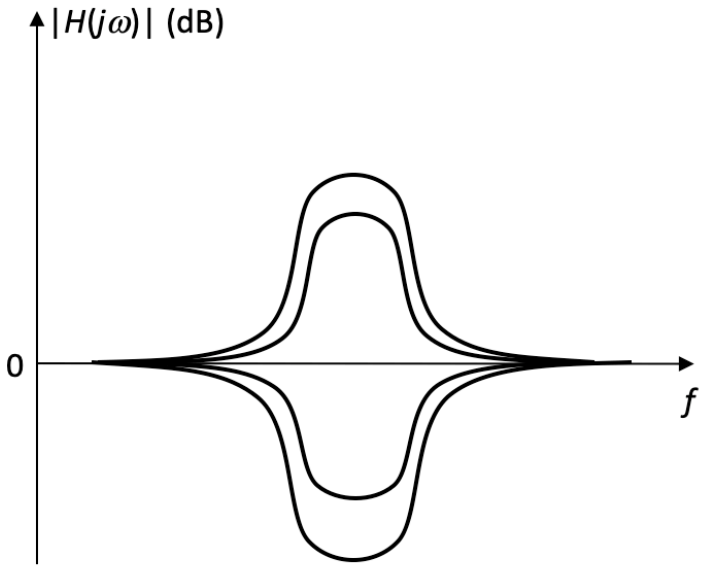


Figura 6.4.7b Respuesta en frecuencia Ecualizador Gráfico

Cuando la curva que se encuentra situada por encima del eje transversal (el eje de frecuencias) el circuito realza las frecuencias de la banda próximas a la frecuencia central; en caso contrario, las atenúa *[9]*.

7. DISEÑO E IMPLEMENTACIÓN DE UN ECUALIZADOR DE 4 BANDAS.

En este proyecto se va a llevar a cabo el diseño de un ecualizador de 4 bandas, por eso para este tipo de diseño se van a implementar ecualizadores gráficos y se va a llevar un estudio del prototipo tanto teórico como real, así como el resultado del mismo.

7.1 Diseño de las distintas secciones

El diseño que se va a llevar a diseñar un ecualizador de 4 bandas empleando ecualizadores gráficos en paralelo, cada uno con una frecuencia central cuyas salidas se

sumaran para obtener así un solo ecualizador el cual actúa sobre 4 frecuencias centrales independientes una de otras.

Para llevar a cabo el diseño de dicho ecualizador gráfico en primer lugar fijamos las ganancias máximas y mínimas que queremos obtener en cada una de las bandas, así como las distintas frecuencias centrales.

A la hora de elegir las frecuencias al tratarse del diseño de un ecualizador de audio de 4 bandas aplicado al campo sonoro, se ha seguido un criterio basado en el rango auditivo de los seres humanos, así como el rango sonoro que ocupan los distintos instrumentos musicales.

Por ejemplo, el bombo de una batería normal, sitúa su frecuencia central entre los 50 y 200 Hz. El bajo tiene sus frecuencias centrales comprendidas entre los 150 Hz y los 800 Hz, las guitarras comprenden las frecuencias comprendidas entre los 80 Hz y los 3.5 KHz.

Así, se puede afirmar que dependiendo del tono de sonido ocupará este un determinado rango de frecuencias dentro del el rango de frecuencias auditivas del ser humano.

Así, se ha procedido a elegir distintos rangos de frecuencias para que nuestro ecualizador actúe sobre distintas frecuencias.

Para la primera frecuencia central, hemos considerado que el ecualizador actué sobre la frecuencia fundamental de los bombos de las baterías.

$$f_{C1} = 115\ kHz$$

Para la segunda frecuencia central, se ha considerado la frecuencia fundamental de los bajos eléctricos.

$$f_{C2} = 330\ kHz$$

Para la tercera frecuencia central, se ha considerado la frecuencia fundamental de la voz humana, frecuencia a la que el oído humano es mucho más sensible a la hora de identificar variaciones en la ganancia.

$$f_{C3} = 3\ kHz$$

Para la cuarta y última frecuencia central, se ha considerado la frecuencia fundamental de los platillos u otros elementos los cuales tienen una tonalidad muy aguda.

$$f_{C4} = 9{,}9\ kHz$$

Para este diseño hemos elegido una ganancia máxima en la frecuencia central de +12dB y una ganancia mínima en la frecuencia central de -12dB.

7.1.1 Cálculo de componentes

A la hora de calcular los componentes reales necesarios para el diseño del ecualizador, se han tenido en cuenta ciertas expresiones citadas por Sergio Franco en su libro *Design with Operational Amplifiers and Analog Integrated Circuits*.

En primer lugar, la ganancias máximas y mínimas del ecualizador en la frecuencia central según Sergio Franco vienen dada por la expresión. *(6.4.7*b) del apartado 6 de este mismo proyecto.

En nuestro caso, la ganancia máxima será +12dB, luego teniendo en cuenta que:

$$\mathrm{dB} = 20 \log (\mathrm{A}_0)$$

La expresión *(6.4.7*b*)* quedara de la siguiente forma:

$$3{,}98 \leq \frac{3R_1 + R_2}{3R_1}$$

Por otro lado, la ganancia mínima será de -12dB luego:

$$0{,}25 \geq \frac{3R_1}{3R_1 + R_2}$$

Como queremos que nuestra ganancia este comprendida entre los siguientes valores;

$$-12\ dB < A_0 < +12\ dB$$

A continuación, implementamos lo anterior a la expresión. *(6.4.7*b) y despejando cualquiera de las ecuaciones anteriores obtenemos la siguiente relación:

$$R_2 > 9\ R_1 \qquad (6.4.7\text{b})$$

Luego para cada frecuencia central, debemos hallar el capacitor correspondiente. Para ello empleamos la expresión *(6.4.7*a).

FRECUENCIAS CENTRALES	
f_1 (Hz)	115
f_2 (Hz)	330
f_3 (Hz)	3000
f_4 (Hz)	9900

Tabla 7.1

Para hallar dichos capacitores debemos de fijar un valor para R1 provisional el cuál se verá modificado una vez hallados los capacitores. El hecho de que el cálculo se lleve a cabo de este modo es debido a que los valores nominales de los capacitores son más limitados que los valores nominales de los resistores, lo que nos conduce a hallar primero los valores nominales de los condensadores para posteriormente hallar los valores nominales de las resistencias.

Si suponemos que R_1= 10 kΩ y despejamos en cualquiera de las ecuaciones anteriores (Las de las inecuaciones) obtenemos que:

$$R_2 = 90\ k\Omega$$

En este caso usaremos una $R_2 = 100\ \text{k}\Omega$

Luego una vez tenemos los valores de R_1, R_2 y f_c, procedemos a hallar los valores de los capacitores despejando de la expresión *(6.4.7*a).

- Para $f_{C1} = 115\ \text{Hz}$

$$C_{12} = \frac{\sqrt{2 + R_2/R_1}}{20\pi R_2 f_{C1}}$$

Obtenemos que $C_{12} = 4{,}794\ \text{nF}$

- Para $f_{C2} = 330\ \text{Hz}$

$$C_{21} = \frac{\sqrt{2 + R_2/R_1}}{20\pi R_2 f_{C2}}$$

Obtenemos que $C_{22} = 1{,}671\ \text{nF}$

- Para $f_{C3} = 3\ \text{kHz}$

$$C_{31} = \frac{\sqrt{2 + R_2/R_1}}{20\pi R_2 f_{C3}}$$

Obtenemos que $C_{32} = 183{,}8\ \text{pF}$

- Para $f_{C4} = 9{,}9\ \text{kHz}$

$$C_{41} = \frac{\sqrt{2 + R_2/R_1}}{20\pi R_2 f_{C4}}$$

Obtenemos que C_{42} = 55,69 pF

Una vez obtenidos dichos valores, usamos los valores nominales superiores más próximos, luego:

- Para $f_{C1} = 115$ Hz usamos el valor nominal $C_{12} = 5{,}6$ nF
- Para $f_{C2} = 330$ Hz usamos el valor nominal $C_{22} = 1{,}8$ nF
- Para $f_{C3} = 3$ kHz usamos el valor nominal $C_{32} = 220$ pF
- Para $f_{C4} = 9{,}9$ kHz usamos el valor nominal $C_{42} = 56$ pF

Ahora que ya tenemos los valores nominales de los capacitores, podremos proceder a calcular los valores de los resistores usando la expresión de la frecuencia fundamental f_0 así como la que relaciona la Ganancia con R_1 y R_2 para ello emplearemos os valores nominales de los capacitores. Para cada una de las frecuencias centrales nos darán distintos valores de resistores.

Por otra parte, según Sergio Franco nos indica en su libro *Design with Operational Amplifiers and Integrated Circuits,* la relación que existe entre las resistencias R_2 y R_3, así como C_1 y C_2 es:

$$R_3 = 10\, R_2$$

$$C_1 = 10\, C_2$$

Luego ahora si podemos proceder a calcular los valores de todas las resistencias que componen el ecualizador gráfico:

- Para $f_{C1} = 115$ Hz

$$R_{11} = \frac{\sqrt{11}}{180\pi C_{12} F_{C1}}$$

Obtenemos que R_{11} = 9107,27 Ω

f_c = 115 Hz	
VALORES RESISITENCIAS (CALCULADAS)	
R_1 (Ω)	9107,27
R_2 (Ω)	81965,41
R_3 (Ω)	819654,08

Tabla 7.2

- Para $f_{C2} = 330$ Hz

$$R_{21} = \frac{\sqrt{11}}{180\pi C_{22} F_{C1}}$$

Obtenemos que $R_{21} = 9873{,}87\ \Omega$

f_c = 330 Hz	
VALORES RESISITENCIAS (CALCULADAS)	
$R_1\ (\Omega)$	9873,87
$R_2\ (\Omega)$	88864,85
$R_3\ (\Omega)$	888648,53

Tabla 7.3

- Para $f_{C3} = 3$ KHz

$$R_{31} = \frac{\sqrt{11}}{180\pi C_{32} F_{C3}}$$

Obtenemos que $R_{31} = 8886{,}49$

f_c = 3000 Hz	
VALORES RESISITENCIAS (CALCULADAS)	
$R_1\ (\Omega)$	8886,49
$R_2\ (\Omega)$	79978,37
$R_3\ (\Omega)$	799783,68

Tabla 7.4

- Para $f_{C4} = 9{,}9$ KHz Ω

$$R_{41} = \frac{\sqrt{11}}{180\pi C_{42} F_{C4}}$$

Obtenemos que $R_{41} = 10579{,}15\ \Omega$

f_c = 9900 Hz	
VALORES RESISITENCIAS (CALCULADAS)	
$R_1\ (\Omega)$	10579,15
$R_2\ (\Omega)$	95212,34
$R_3\ (\Omega)$	952123,43

Tabla 7.5

Una vez obtenidos los valores de R_1 disponemos de la expresión que relaciona R1 con R2 y R2 con R3, buscamos los valores nominales que más se aproximen. En este caso se ha empleado valores de resistores de la serie e 24, e48 y e96. Dichas tablas se encuentran el apartado de ANEXOS de este mismo documento.

A continuación, se muestran los valores de resistencias nominales elegidos, así como las frecuencias centrales calculadas nuevamente, en este caso con dichos componentes.

Para la primera $f_{C1} = 115$ Hz

f_c = 115 Hz	
VALORES (NOMINALES)	
R_1 (Ω)	9530
R_2 (Ω)	80600
R_3 (Ω)	806000
C_{11} (F)	5,60E-08
C_{12} (F)	5,60E-09

Tabla 7.6

Para la segunda $f_{C2} = 330$ Hz

f_c = 330 Hz	
VALORES (NOMINALES)	
R_1 (Ω)	11000
R_2 (Ω)	82500
R_3 (Ω)	825000
C_{21} (F)	1,80E-08
C_{22} (F)	1,80E-09

Tabla 7.7

Para la tercera $f_{C3} = 3000$ Hz

f_c = 3000 Hz	
VALORES (NOMINALES)	
R_1 (Ω)	9090
R_2 (Ω)	78700
R_3 (Ω)	787000
C_{31} (F)	2,20E-09
C_{32} (F)	2,20E-10

Tabla 7.8

Para la cuarta $f_{C4} = 9900$ Hz

f_c = 9900 Hz	
VALORES (NOMINALES)	
R_1 (Ω)	12100
R_2 (Ω)	86600
R_3 (Ω)	866000
C_{41} (F)	5,60E-10
C_{42} (F)	5,60E-11

Tabla 7.9

Así mismo se ha procedido a calcular cada una de las frecuencias centrales, en este caso con los compone reales cuyos valores nominales se recopilan en las tablas 5, 6, 7, 8 y 9

FRECUENCIAS CENTRALES (CALCULADAS)	
f_1 (Hz)	114,0
f_2 (Hz)	330,3
f_3 (Hz)	3000,9
f_4 (Hz)	9931,0

Tabla 7.10

Cabe destacar que al llevar a cabo diseños con componentes reales los cuales tienen valores limitados, las frecuencias centrales obtenidas respecto a las teóricas se ven desviadas. Por eso en este diseño se ha tenido en cuenta dicha desviación y se han elegido los componentes de forma que la frecuencia central obtenida con el diseño real sea lo más acotado al diseño teórico.

En la siguiente tabla se puede observar el porcentaje de desviación que presenta el resultado con los componentes reales frente al teórico.

% DE APROXIMACIÓN	
FRECUENCIAS CENTRALES (CALCULADAS)	
f_1 (Hz)	99,2
f_2 (Hz)	100,1
f_3 (Hz)	100,0
f_4 (Hz)	100,3

Tabla 7.11

Los valores de porcentaje de frecuencia menores a cien están muy próximos a la frecuencia central teórica y aquellos valores mayores de cien sobrepasan la frecuencia central teórica.

7.1.2 Implementación de las distintas secciones

Una vez obtenidos los componentes nominales del diseño, se implementan los diferentes circuitos al Ltspice.

En primer lugar, se implementan el circuito que comprende a cada banda para estudiar los resultados obtenidos de forma particular y a continuación se implementa el circuito completo del ecualizador de 4 bandas para estudiar su respuesta en frecuencia.

A la hora de llevar a cabo la implementación del circuito del ecualizador gráfico en LTSpice se ha observado que, independientemente de la versión del software, no existen potenciómetros. Por lo tanto, se ha llevado a cabo el diseño de dicho componente en primer lugar para posteriormente implementarlo en los distintos ecualizadores gráficos que se van a usar.

Para llevar a cabo el diseño de dicho componente se ha estudiado su funcionamiento y se ha llegado a la conclusión de que dicho componente resume su funcionamiento en la siguiente expresiones matemáticas estudiadas a continuación

Si separamos el valor de dicha resistencia en 2 resistencias, obtenemos que:

$$R_1 = R\frac{val}{100} \qquad (7.1.2a)$$

$$R_2 = R(1 - \frac{val}{100}) \qquad (7.1.2b)$$

A continuación, se ha implementado dicha ecuación en LTSpice, en primer lugar, creando el circuito representado en la figura 7.1.2a.

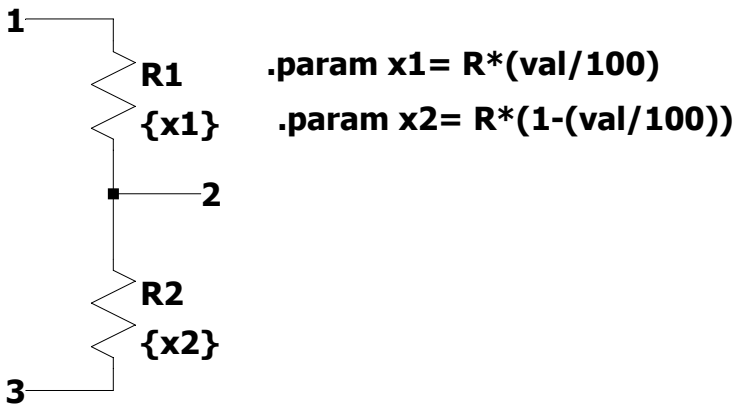


Figura 7.1.2a Circuito del potenciómetro implementado en LTSpice

Donde *Val* representa el porcentaje del potenciómetro (el valor de la resistencia variable en un punto determinado). A continuación, creando su correspondiente símbolo, véase la figura 7.1.2b.

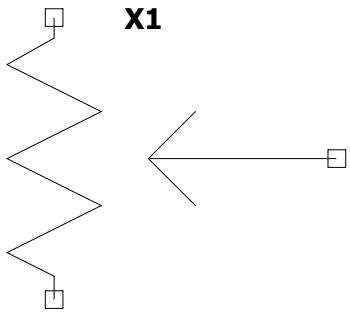


Figura 7.1.2b Símbolo del potenciómetro implementado en LTSpice

Una vez obtenido el potenciómetro se ha procedido a implementar el ecualizador gráfico correspondiente a cada una de las frecuencias centrales y se ha estudiado sus respuestas en frecuencia de forma individual, cabe destacar que el operacional empleado en este caso es el ADTL 082 cuya hoja de características se adjunta en el apartado de Anexos de este mismo proyecto.

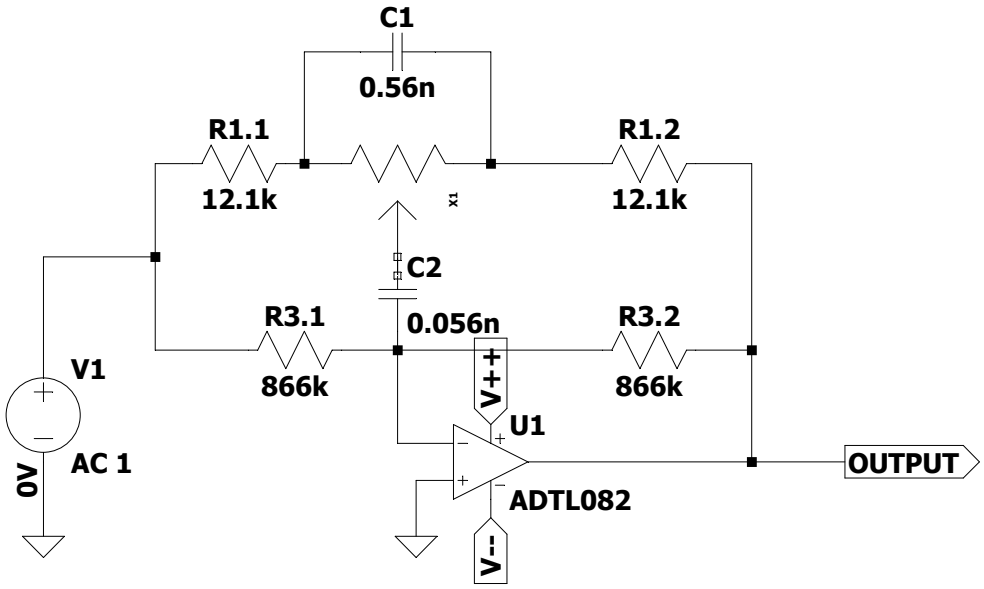


Figura 7.1.2c Sección de un circuito básico ecualizador gráfico

Una vez implementados los 4 ecualizadores gráficos, se han sumado las señales obtenidas a la salida de cada uno de los ecualizadores con un sumador Inversor.

El inconveniente principal de este tipo de solución es que la suma de las respuestas no da lugar a una respuesta plana, lo cual solo ocurriría si las respuestas pasa bajas y pasa altas no se vieran afectadas por las respuestas de las bandas adyacentes *[10]*.

En la figura 7.1.2d se puede apreciar cómo queda el circuito final.

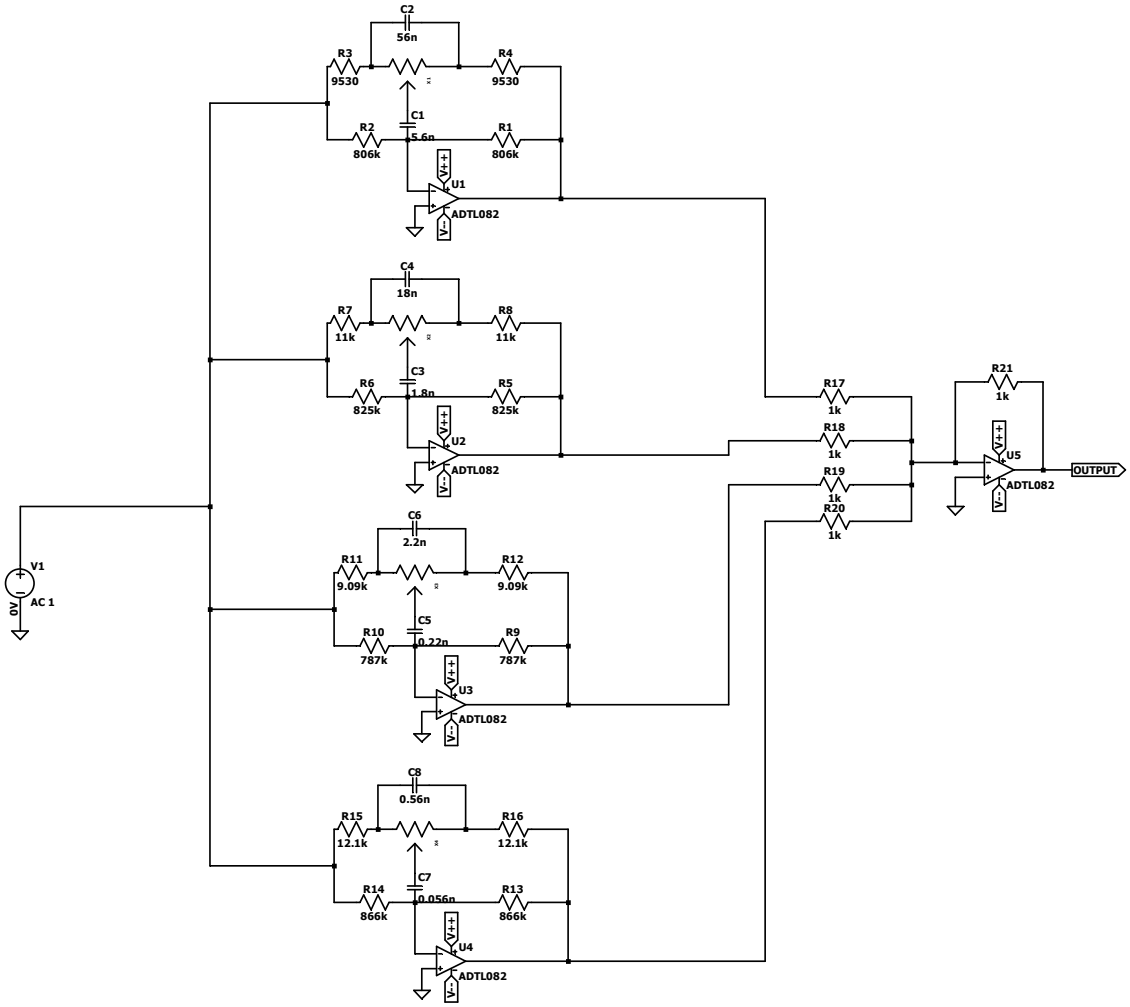


Figura 7.1.2d Circuito del ecualizador de 4 bandas

7.1.3 Simulaciones

En este apartado se estudiarán los resultados teóricos de la simulación de nuestro ecualizador, estudiando cada una de las bandas por separado para observar la respuesta en frecuencia obtenida. Se han llevado a cabo las simulaciones en cada una de las bandas con las ganancias en sus extremos, obteniendo, así como resultado la señal atenuada o amplificada.

A continuación, se representan algunos de los casos estudiados de las distintas bandas que componen nuestro ecualizador.

En primer lugar, para una frecuencia central f_c= 115 Hz, si tenemos nuestro potenciómetro (R_2) a 99% de la resistencia obtenemos la siguiente respuesta en

frecuencia representada en la figura 7.1.3a, donde se puede observar claramente una atenuación aproximadamente -12 dB.

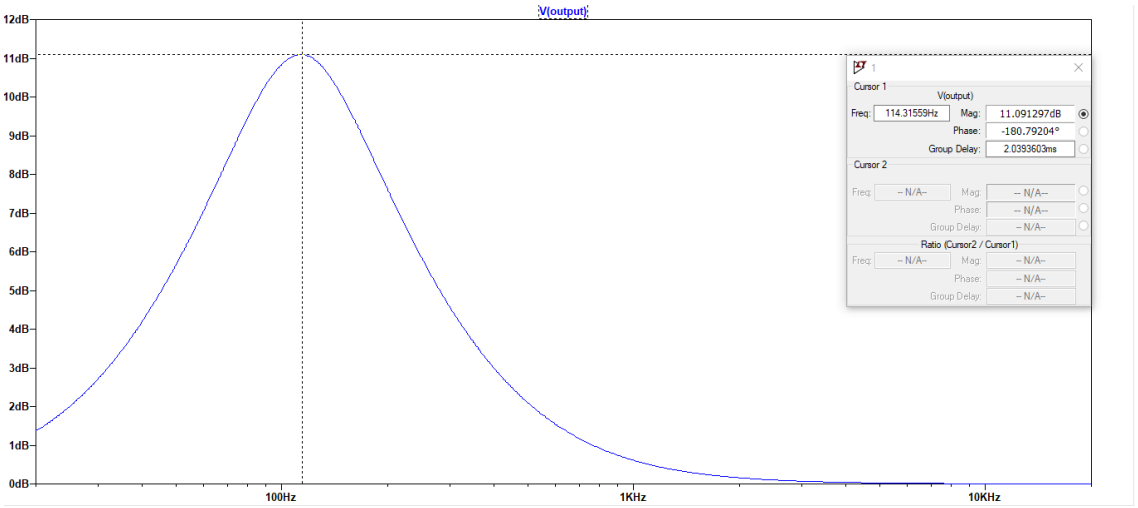


Figura 7.1.3a. Respuesta en frecuencia fc =115 Hz

A continuación, para la segunda frecuencia central, f_c = 330 Hz al poner el potenciómetro R_2 al 1% obtenemos la respuesta en frecuencia representada en la figura 7.1.3b, donde se puede apreciar que la ganancia toma el valor de -10,3 dB

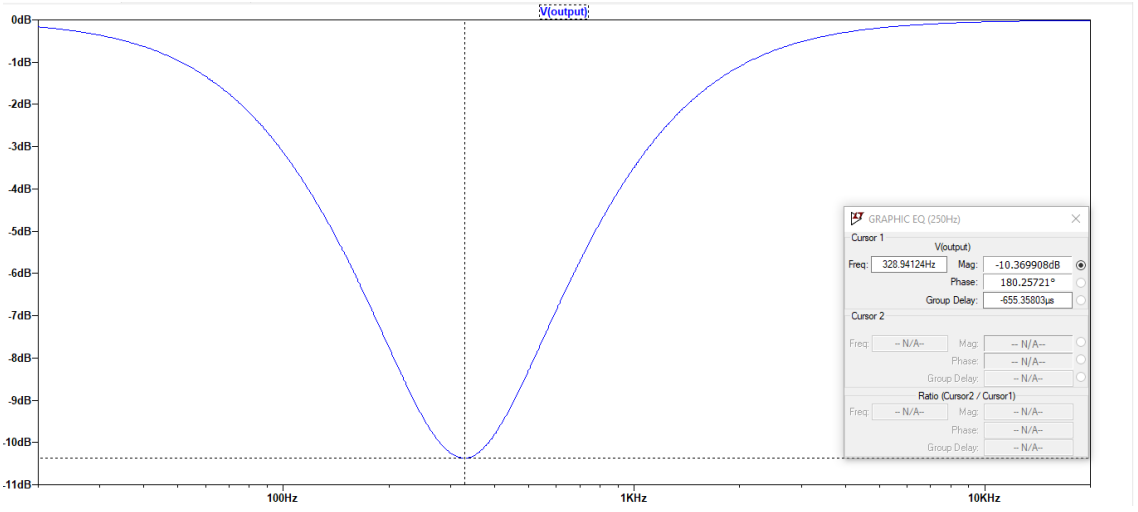


Figura 7.1.3b Respuesta en frecuencia con fc = 330 Hz

Para la tercera frecuencia central, f_c = 3 kHz, al tomar el potenciómetro R2 el valor de 20%, obtendremos a la salida que la ganancia toma el valor de -6,2 dB aproximadamente como bien se puede apreciar en la figura 7.1.3d.

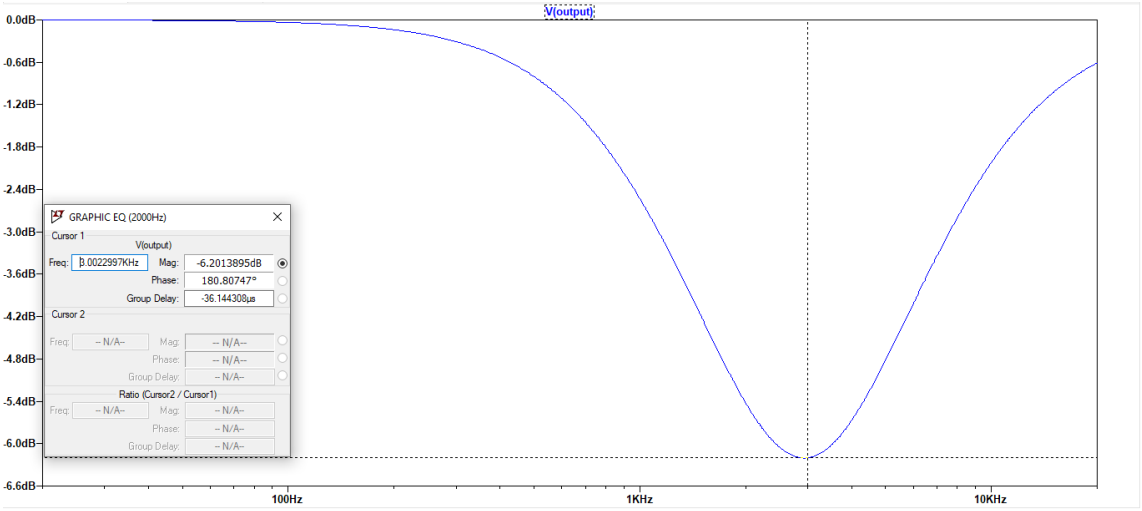


Figura 7.1.3d Respuesta en frecuencia con fc = 3000 Hz

Por último, para f_c= 9,9 kHz, al tomar el potenciómetro el 65% del valor de R_2, a la salida obtenemos una amplificación de la ganancia de aproximadamente 2,4 dB como se puede apreciar en la figura 7.1.3e.

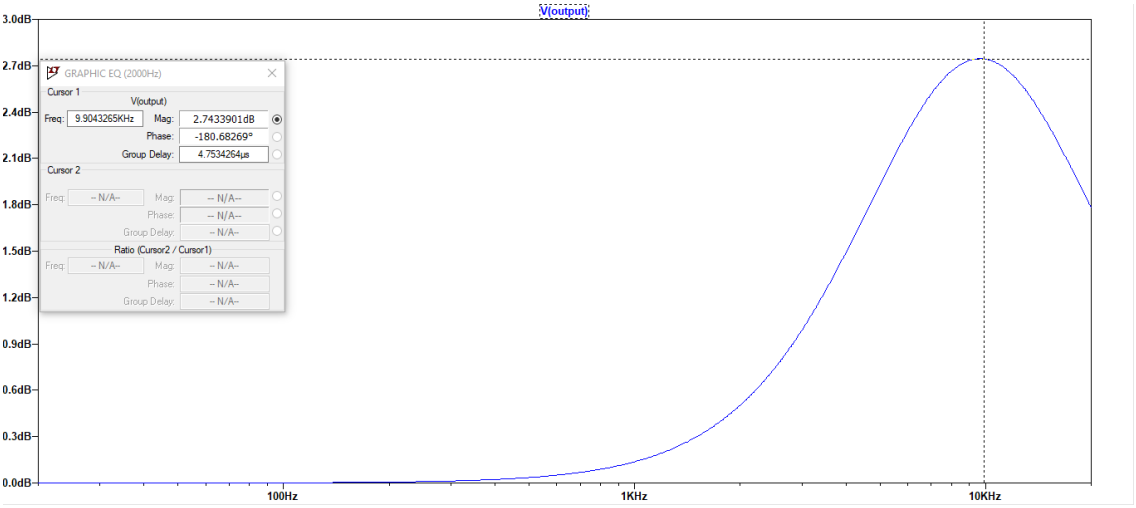


Figura 7.1.3e Respuesta en frecuencia con fc = 9090 Hz

Tras realizar las simulaciones cabe destacar que en aquellas simulaciones en las cuales se le han dado los valores de ganancia extremos se ha observado que estos no alcanzan los +12 dB -12 dB, sino que el valor de la ganancia es siempre ligeramente menor.

Para comprobar la efectividad del filtro biquad, se construyó en el laboratorio el circuito diseñado en el ejemplo 3.19 del libro de Sergio Franco, simulándose el circuito

y observándose una gran selectividad en torno a la frecuencia central. En la figura 7.1.3f se muestran los resultados del barrido en frecuencia con LTSpice.

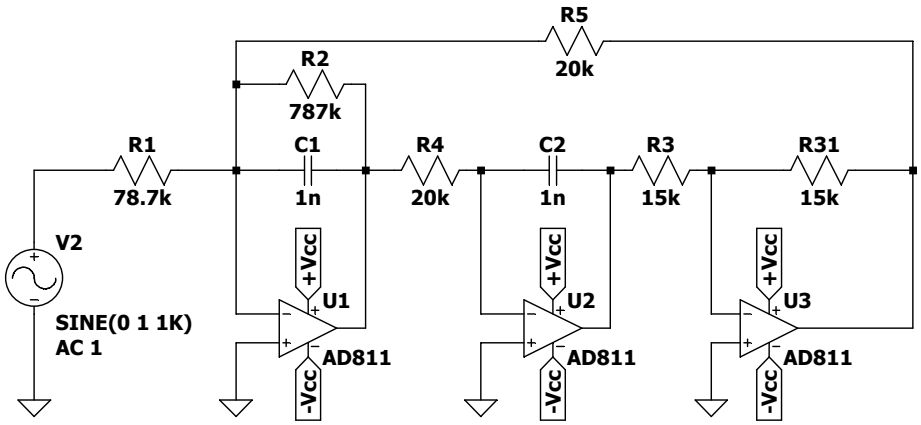


Figura 7.1.3f Circuito ejemplo 3.19

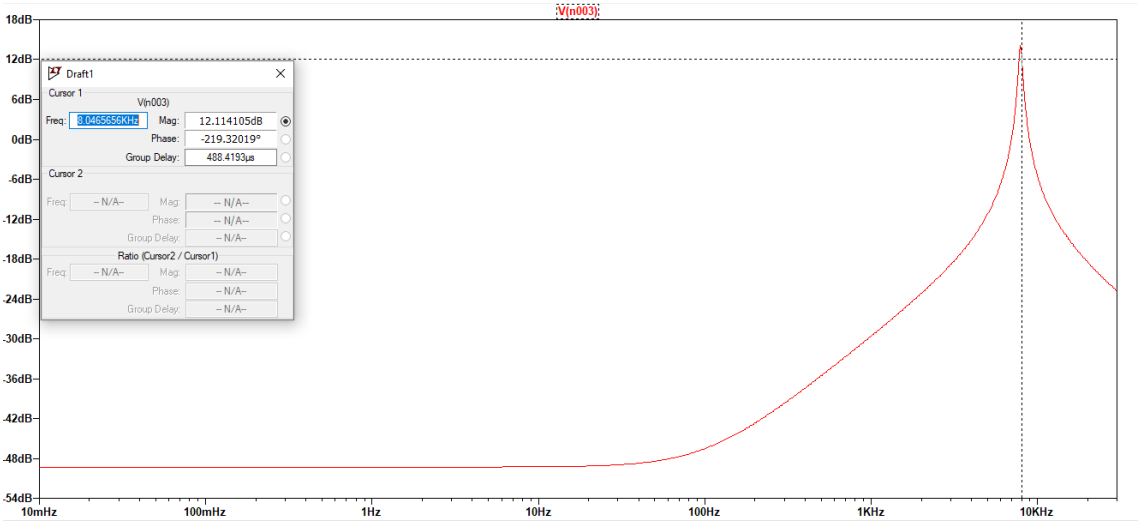


Figura 7.1.3g Respuesta circuito ejemplo 3.19

7.2 Diseño del prototipo.

Para llevar a cabo el diseño de forma analógica harían falta emplear 4 potenciómetros, 2 interruptores y 5 amplificadores operacionales. Cabe destacar el elevado precio de cada uno de los operacionales, el elevado espacio que requeriría llevar a cabo el diseño y el hecho de que uno de los principales problemas de los circuitos analógicos es el ruido el cual habrá que tratar una vez completado el diseño.

Por otro lado, otra alternativa que se puede contemplar es el desarrollo del proyecto de forma digital en una placa de desarrollo, la cual nos permitiría no solo cambiar las frecuencias centrales sin tener que modificar los circuitos sino también modificar el diseño sin tener que cambiar mucho el proyecto a nivel de circuitos.

Se ha llevado un pequeño estudio comparativo para observar las fortalezas de cada uno de los diseños, tanto el analógico como el digital y obtener así un diseño más optimizado.

Diseño	
Analógico	**Digital**
Problema con el ruido	No presenta excesivo ruido
Frecuencias ligadas a componentes	Frecuencias independientes
Elevado coste	Bajo coste
Optimización añadiendo componentes	Únicamente cambiando código
Diseño de etapa amplificadora	Añadir adaptador de audio
Muchos componentes	Pocos componentes

Tabla 7.2.

Debido al elevado coste que conllevaría llevar a cabo el diseño, la limitación de cara al futuro y el elevado ruido que podría presentar el diseño llevado a cabo de forma analógica se ha decidido en este punto llevar a cabo el diseño de forma digital usando 1 placa de desarrollo la cual incluso permite añadirle más prestaciones al diseño.

7.2.1 Componentes

Los componentes usados para llevar a cabo el diseño han sido los siguientes:

COMPONENTE	UDS
Placa Teensy 3.6	1
Encoders digitales	4
Pulsadores	2
Protoboard	1
Tarjeta micro SD	1
Teensy Audio Shield	1

Tabla 7.2.1

Placa Teensy 3.6

Se trata de una placa de desarrollo que cuenta con un procesador Córtex que puede ser usada para una gran variedad de proyectos. En este caso dicha placa se ha

usado como interfaz para conectar las entradas al adaptador de audio (Teensy Audio Shield o bien SGTL5000).

Este microcontrolador cuenta con 64 entradas y salidas digitales, 25 entradas analógicas, 2 salidas analógicas, así como 2 salida DAC.

Una de las grandes ventajas que presenta dicha placa de desarrollo es la capacidad de operar en modo stand-alone, es decir, una vez cargado el código podría trabajar simplemente con la fuente de alimentación y ejecutar el código de forma independiente sin la necesidad de tener un ordenador que este monitoreando continuamente.

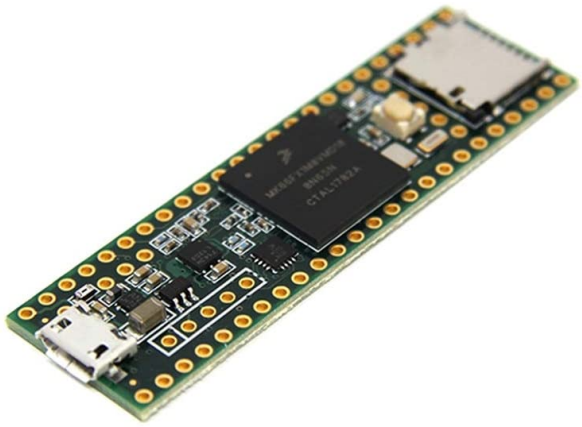


Figura 7.2.1a Placa de desarrollo Teensy 3.6

Encoders digitales

Los encoders empleados en este caso consisten en simples codificadores electromecánicos los cuales disponen de una conexión a tierra y 2 que van conectadas a las entradas digitales correspondientes en la placa Teensy 3.6.

Una característica destacable del encoder empleado es la posibilidad de usar el mismo encoder como pulsador ya que dispone del conexionado. Su funcionamiento básicamente se basa en sumar una unidad en caso de que la rotación sea en sentido horario y restar una unidad en caso de que la rotación sea en sentido anti horario.

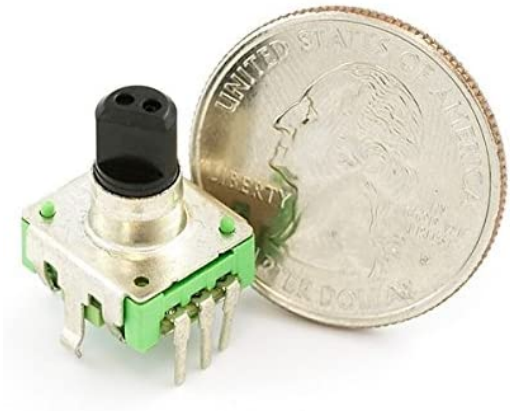


Figura 7.2.1b. Encoders

Protoboard

La protoboard es la placa de pruebas empleada en este caso para llevar a cabo el diseño del prototipo. Se caracteriza por tener la línea positiva conectada internamente al igual que la negativa, por otro lado, cada una de las filas tiene 5 pines conectados entre sí. Destaca por facilitar el montaje de prototipos experimentales facilitando las modificaciones necesarias que se llevan a cabo en esa fase del diseño.

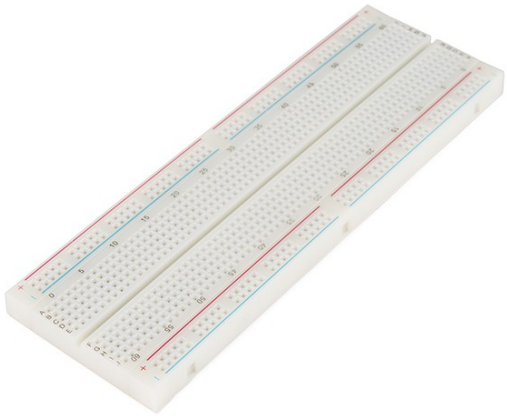


Figura 7.2.1c Protoboard

Tarjeta Micro SD

La tarjeta Micro-SD empleada consiste en una tarjeta de 8 GB, en la cual se guardarán los archivos de audio que se emplearán para llevar a cabo las pruebas del ecualizador. Los archivos de audio serán guardados en el formato *Waveform Audio Format* (WAV) para así conservar los archivos con la máxima calidad de audio.

Figura 7.2.1d Tarjeta de memoria Micro SD

Adaptador de Audio de Teensy

El adaptador de audio de Teensy, también llamado *Teensy Audio Shield* o *SGTL5000*, es un adaptador de audio para placas de desarrollo, principalmente creada para la gama Teensy 3.X. Dispone de 1 salida de auriculares estéreo, 2 salidas DAC, 2 entradas DAC, así como una entrada de micrófono. Así mismo dispone del procesador SGTL5000 el cual incluye prestaciones destacables que convierten en este adaptador ideal para emplearlo en proyectos relacionados con el sonido. Entre las prestaciones destacables se encuentra la amplia librería de filtros y ecualizadores que pone a disposición del usuario.

Figura 7.2.1e. Adaptador de Audio para Teensy 3.6

7.2.2 Implementación

Para llevar a cabo el ecualizador se ha decidido implementarlo en una placa de desarrollo, concretamente la Teensy 3.6 con un adaptador de audio, el Teensy Audio Shield cuyo procesador es el SGTL5000. Desde este se podrá reproducir una señal la cual tiene la posibilidad de ser ecualizada para obtener una señal personalizada a través del procesador de la placa y el procesador del correspondiente adaptador de audio que la placa. En la parte de ANEXOS de este mismo proyecto se adjunta la hoja de datos de la placa.

El diseño consiste básicamente en que la placa de desarrollo Teensy recibe la señal a ecualizar la cual puede ser desde o bien una tarjeta Micro Sd o bien desde la correspondiente entrada analógica de la placa Teensy 3.6.

A continuación, dicha señal es ruteada hacia el adaptador de audio donde éste se encargará de realizar la ecualización implementando uno de los ecualizadores que se encuentran disponibles en las librerías del sgtl500, en nuestro caso el ecualizador gráfico.

Una vez la señal llega a este punto, cabe aclarar que las ganancias del ecualizador gráfico están mapeadas a las correspondientes entradas digitales de la Placa Teensy 3.6 desde las cuales podremos modificar desde los Encoders digitales en tiempo real.
Por último, una vez ecualizada la señal esta es enviada a la salida la cual puede ser o bien la salida de auriculares (estéreo) que se encuentra en el adaptador de audio o bien la salida analógica de la misma.

Como bien se ha explicado en el apartado 6.4.6 de este mismo proyecto, la función de transferencia de cada una de las bandas del ecualizador gráfico del adaptador de audio dela placa Teensy se basa en la función de transferencia del filtro biquad, el cual es un filtro recursivo de segundo orden y contiene 2 polos y 2 ceros.

Así mismo para implementar el ecualizador en el Teensy 3.6 se ha usado e software Teensyduino proporcionado por el mismo fabricante. El lenguaje de programación usado en este caso para programar la placa de desarrollo es C. Se han empleado varias librerías que se encontraban disponibles para el Teensy 3.6 dentro de las cuales tenemos la siguiente:

```
#include <Audio.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <SerialFlash.h>
#include <Bounce2.h>
#include <Encoder.h>
```

Figura 7.2.2a. Librerías

A continuación, para implementarlo también se ha usado la herramienta virtual de Teensy la cual permite implementar los distintos bloques que componen el ecualizador y realizar el conexionado entre los mismos obteniendo como resultado la parte del código que corresponde a la declaración de los bloques y el conexionado entre los mismo.

Dicha herramienta ha sido creada por el fabricante única y exclusivamente para el desarrollo de proyectos relacionados con el sonido en los productos de PJRC. Al final de este proyecto se anexa el código completo que se ha generado para llevar a cabo el diseño del ecualizador de 4 bandas en Teensyduino.

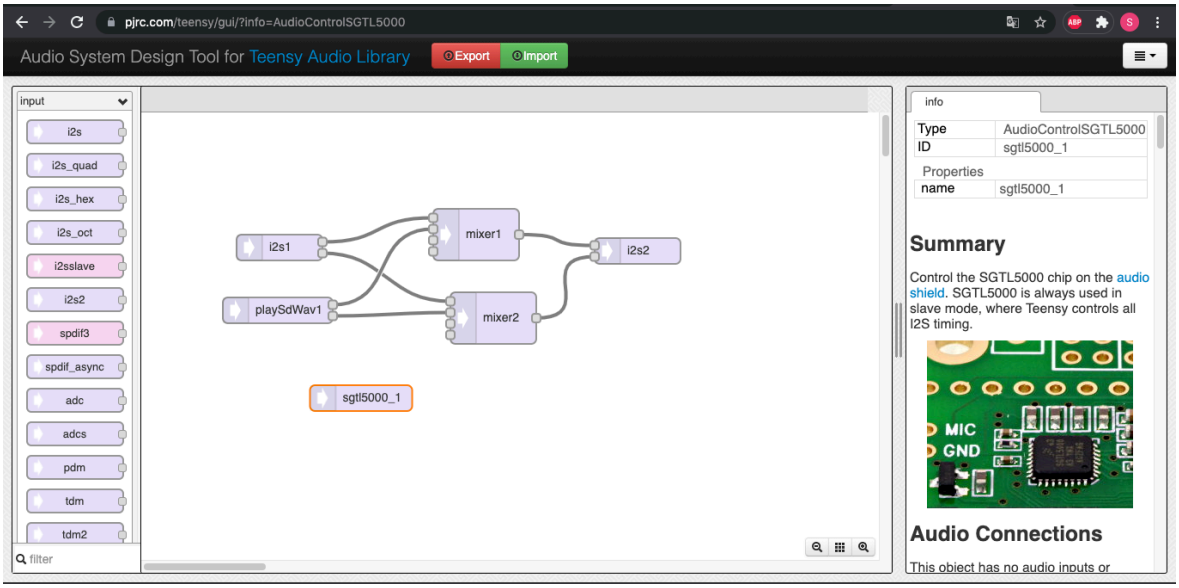


Figura 7.2.2b Herramienta de Diseño de PRJC

A continuación, se muestra la parte del código que corresponde a la recepción de los datos de uno de los Encoders y la ecualización de la banda correspondiente de la señal proporcionada. Para más detalles se anexa el código completo en el apartado de ANEXOS.

```
                                                    //ENCODER1
  current_value1 = enc1.read();
    if (current_value1 < -10) {
    current_value1 = -10;
     enc1.write(current_value1);
}
    else if (current_value1 > 10) {
    current_value1 = 10;
    enc1.write(current_value1);
}

  a1 = current_value1/10;
  sgtl5000_1.eqBand(4,a1);

  Serial.print("ENCODER 1 \n");
  Serial.println(a1);
```

Figura 7.2.2c Código del Encoder 1

Otro de los aspectos importantes a destacar del diseño es la función de transferencia empleada por el diseñador a la hora de implementar el ecualizador gráfico. Dicho diseñador emplea 5 filtros IIR de segundo orden en paralelo. A su vez dichos filtros han sido implementados usando filtros biquad cuyos coeficientes se calculan como se explican a continuación.

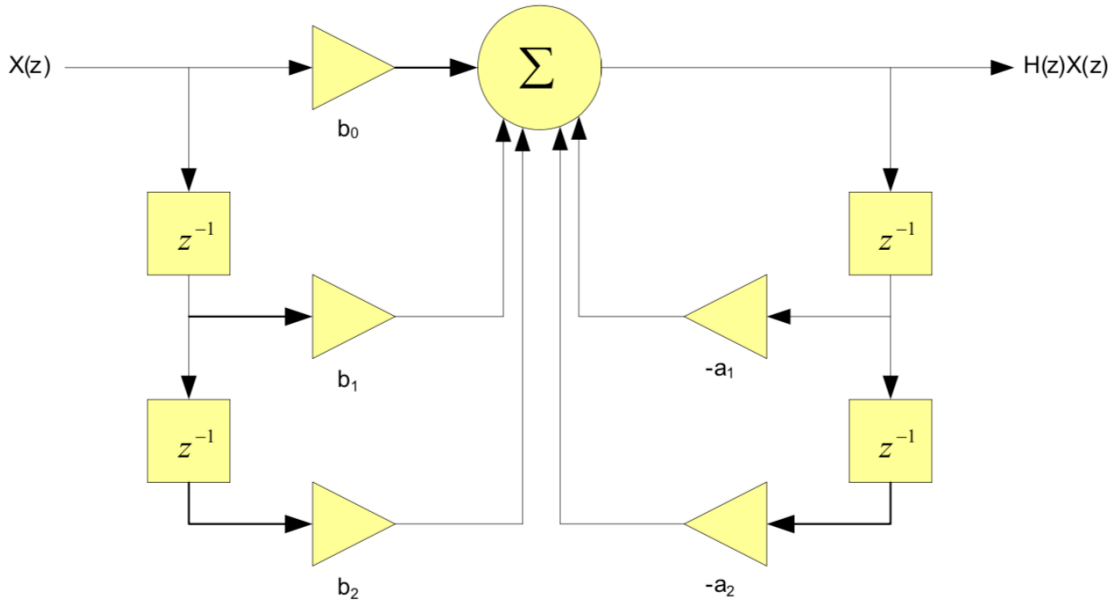


Figura 7.2.2d. Diagrama de bloques.

Cuya función de transferencia es la siguiente:

$$H(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \qquad (7.2.2a)$$

Así la función de transferencia de dicho ecualizador para cada banda en el Teensy tiene la siguiente forma:

$$H_{BP}(j\omega) = \frac{(j\omega/\omega_0)/Q}{1-(\omega/\omega_0)^2+(j\omega/\omega_0)/Q}$$

o bien:

$$H_{BP}(s) = \frac{(s/\omega_0)/Q}{1+(s/\omega_0)^2+(s/\omega_0)/Q}$$

como:

$$H(S) = H_0H_{BP}(s) \qquad (7.2.2b)$$

H(s) quedaría de la siguiente forma:

$$H(s) = H_0\frac{(s/\omega_0)/Q}{1+(s/\omega_0)^2+(s/\omega_0)/Q}$$

La función *H(s)* representa la relación entre las señales de salida y entrada en el dominio de la frecuencia compleja (el dominio de la transformada de Laplace). Las señales se obtienen en este caso aplicando la transformación a señales en el dominio del tiempo, que son las señales procesadas por los sistemas analógicos.

La situación cambia en los sistemas de procesado digital, ya que en este caso las señales se someten a un proceso de muestreo y cuantificación, obteniéndose señales en el dominio del tiempo discreto. La función de transferencia se determina en este caso aplicando la transformada Z a las señales de tiempo discreto, obteniéndose señales en el dominio de la frecuencia compleja. Dicha transformada se calcula aplicando la siguiente expresión:

$$X(z) = Z\{x[n]\} = \sum_{n=-\infty}^{\infty}x[n]z^{-n} \qquad (7.2.2c)$$

Como la multiplicación por la variable *s* en el dominio de Laplace equivale a una

diferenciación en el dominio del tiempo, una expresión en diferencias de *s* se puede obtener de forma aproximada empleando la *transformación bilineal [11]*:

$$s = \frac{1}{T}\frac{1-Z^{-1}}{\left(\frac{1+Z^{-1}}{2}\right)} = \frac{2}{T}\frac{1-Z^{-1}}{1+Z^{-1}},$$

donde $1 - Z^{-1}$ representa la diferencia entre la muestra de tensión presente y la muestra anterior, $(1 + Z^{-1})/2$ representa el promedio de las muestras presente y anterior, y el factor $1/T$ se introduce para normalizar respecto al periodo de muestreo, *T*. Esto nos permite, partiendo de las características del filtro analógico y aplicando la transformada bilineal, obtener la función de transferencia del filtro digital.

Aplicando la transformada bilineal, operando sobre nuestra función podremos obtener los coeficientes los cuales dependen de Q, w_0 y S de la función de transferencia implementada:

$$b_0 = 2T\omega_0$$

$$b_1 = 0$$

$$b_2 = -2T\omega_0$$

$$a_0 = T^2\omega_0 Q + 2T\omega_0 + 4Q$$

$$a_1 = 2T^2\omega_0 Q - 8Q$$

$$a_2 = T^2\omega_0 Q - 2T\omega_0 + 4Q$$

Por otro lado, cabe destacar la relación entre *Q* y ω_0;

$$Q = \frac{\omega_0}{BW} \qquad (7.2.2\text{d})$$

donde:

$$BW = \omega_H - \omega_L \qquad (7.2.2\text{e})$$

De este modo podríamos obtener los coeficientes de la función de transferencia de la banda deseada del ecualizador gráfico empleado cuya función de transferencia queda de la siguiente manera:

$$H(z) = \frac{(2T\omega_0) + 0 + (2T\omega_0)z^{-2}}{(T^2\omega_0 Q + 2T\omega_0 + 4Q) + (2T^2\omega_0 Q - 8Q)z^{-1} + (T^2\omega_0 Q - 2T\omega_0 + 4Q)z^{-2}}$$

En la actualidad existen bastantes programas que fijan los coeficientes de H(z) para así fijar la frecuencia la deseada. El diseñador del procesador SGTL5000 en este caso emplea un software para hallar dichos coeficientes y fijarlos, minimizando de este modo cualquier tipo de error debido a cálculos.

7.2.3 Resultados

En este apartado se van a representar la salida obtenidas para cuatro configuraciones distintas de nuestro ecualizador de 4 bandas. Para ello se conectará la salida del ecualizador a un analizador de espectro virtual a través del cual se podrán apreciar mejor las salidas. Dicha configuración viene detallada en el apartado 8.1 de este proyecto.

En primer lugar, se emplea se configura el ecualizador de la siguiente manera. A_1 = +12 dB, A_2 = 0 dB, A_3 = +12 dB y A_4 = +12 dB, obteniendo a la salida la siguiente respuesta.

Figura 7.2.3a. Respuesta en frecuencia configuración 1

A continuación, queremos realzar la presencia de los tonos medios, únicamente de los tonos que comprenden el ancho de banda que ocuparía un solo de un vocalista, para ello se procede a cambiar la configuración dándole los siguientes valores a las

ganancias de las frecuencias centrales, A_1 = -12 dB, A_2 = -12 dB, A_3 = +12 dB y A_4 = -12 dB.

Figura 7.2.3b. Respuesta en frecuencia con configuración 2

Por último, se va a realzar las frecuencias medias y altas mientras las bajas se mantienen sin modificar mediante los siguientes valores de ganancias; A_1 = 0 dB, A_2 = 0 dB, A_3 = +12 dB y A_4 = +12 dB

Figura 7.2.3c. Respuesta en frecuencia con configuración 3

Como bien se ha podido observar a través de las distintas configuraciones empleadas, el ecualizador opera de manera óptima, cumpliendo con las necesidades del usuario en lo que a respuesta en frecuencia comprende.

8. ANÁLISIS DE LA SEÑAL ECUALIZADA.

8.1 Configuración empleada

En primer lugar, Para llevar a cabo el análisis del comportamiento del ecualizador se ha optado por enviar la señal de audio a un analizador de espectro digital a través de un interfaz de audio y un DAW (Digital Audio Workstation). A continuación, se puede observar el diagrama de la configuración empleada para el análisis del comportamiento del ecualizador e 4 bandas diseñado.

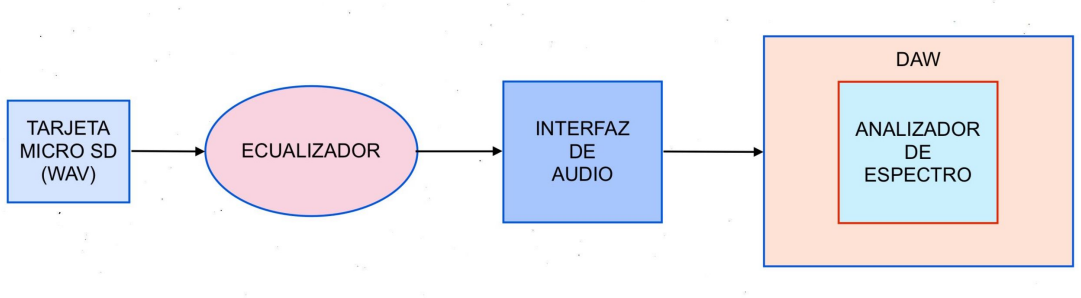


Figura 8.1.1 Diagrama de la configuración empleada.

En segundo lugar, se ha optado por usar el famoso ruido blanco, también llamado en ingles *White Noise*, el cuál se caracteriza por ocupar todo el espectro sonoro audible. Esta decisión se fundamenta en que al usar archivos de audio los cuales son grabaciones de instrumentos de distintos tipos, la señal de audio no ocupa completamente el espectro de audio.

Así empleando el archivo de ruido blanco para realizar pruebas podremos apreciar mejor cualquier variación a la salida, ya que el ruido blanco no presenta mucha dinámica en cuanto a volúmenes al contario que las canciones masterizadas y comercializadas que por norma general, si presentan bastante dinámica en lo que a volúmenes comprende, lo cual dificultaría el estudio de la ganancia de una banda determinada al presentar este continuos cambios.

Los archivos de audio que se cargan en la tarjeta Micro SD siempre tendrán el formato audio WAV. A continuación, se muestra una comparativa del espectro de audio

que ocupa el ruido blanco y el espectro de audio que ocupa una canción del género de música electrónica.

Figura 8.1.2 Ruido Blanco

Figura 8.1.3 Canción del genero de música electrónica

En este caso el interfaz de audio empleado para recopilarla a señal de audio analógica y convertirla en digital es el Focusrite 2i4. El analizador de espectro empleado

es el Voxengo Span de la marca Voxengo, el cual es usado como plugin externo del Ableton. Por otro lado, el DAW empleado en este caso es el Ableton Live 11 Suite. A continuación, se puede observar el conexionado llevado a cabo para realizar el análisis del comportamiento del ecualizador de 4 bandas diseñado.

Figura 8.1.4 Configuración empleada

8.2 Análisis de la señal ecualizada

En este apartado se va a realizar un análisis de los resultados obtenidos tras llevar a cabo la ecualización de una señal de audio en el apartado 7.5.

Como bien sabemos en el primero de los casos los valores de ganancias aplicados a través de los Encoders, se aplicaron con el fin de obtener a la salida una señal con los siguientes valores de ganancia para cada banda A_1 = +12 dB, A_2 = 0 dB, A_3 = +12 dB y A_4 = +12 dB. Una vez obtenida la salida se puede apreciar en primer lugar, que, para la f_{C1}, f_{C3} y f_{C4}, la ganancia no llega a los +12 dB teóricos, sino que se queda ligeramente por debajo de dicho valor, obteniendo aun así una notoria amplificación de dichas frecuencias, mientras que la f_{C2} se mantiene casi intacta.

Si comparamos la la f_{C1} a la salida del ecualizador gráfico mostrada en la figura 7.1.3a del apartado 7.1.3 de este mismo proyecto se puede apreciar que al igual que en el caso anterior la salida no alcanza los +12 dB, sino que se queda ligeramente por debajo de este valor, afirmando así el correcto funcionamiento del ecualizador diseñado.

Un aspecto a destacar de las salidas comparadas es la relación de fases, en el diseño analógico a la salida aparece siempre una inversión de fase de 180º mientras que a la salida del ecualizador digital dicha inversión de fase se corrige automáticamente.

9. PRESUPUESTO

Para llevar a cabo el diseño y ejecución del ecualizador de 4 bandas, han hecho falta una serie de recursos materiales y no materiales los cuales conllevan unos costes económicos. En este apartado se va a analizar el coste en primer lugar unitario para la fabricación de un único ecualizador de audio, seguido del estudio del coste para la fabricación de 100 unidades del mismo.

COMPONENTE	PRECIO € / Ud	Ud	SUBTOTAL
Teensy 3.6	36,27	1	26,37
Teensy Audioshield	13,85	1	12,84
Encoder	1,07	4	4,28
Protoboard	1,70	1	1,70
Cables	1,50	1	1,50
Ensamblaje*	13,87	8	110,96
Programación*	12,20	5	61,0
Documentación	13,87	30	416,1
TOTAL (CON IGIC)			**634,75 €**

Tabla 9.1

El coste para el diseño y fabricación de este prototipo ha sido de 634,75 euros, contando con la mano de obra de un Ingeniero Técnico y un programador cuyos honorarios han sido extraídos de las tablas salariales de los correspondientes convenios colectivos.

10. CONCLUSIONES

El presente proyecto nace con el fin de diseñar un sistema de ecualización de señales de audio de cuatro bandas con una ganancia variable entre menos doce y más doce decibelios, realizando el estudio desde un punto de vista teórico y llevando el diseño a la práctica a posteriori.

Dicho diseño se ha conseguido implementando en la parte teórica cuatro ecualizadores gráficos en paralelo cuyas respuestas son sumadas y obteniendo así una señal personalizada a la salida. Por otra parte, para el diseño del prototipo se ha optado por desarrollar el mismo en la placa de desarrollo Teensy 3.6 junto a su correspondiente adaptador de audio, el cual no solo presenta una respuesta óptima a las entradas proporcionadas por el usuario, sino que también presenta un amplio abanico de posibilidades, destacando entre otros la posibilidad de añadir distintos tipos de efectos de audio como pueden ser el *reverb* o el *delay* sin necesidad de añadir más componentes electrónicos al diseño.

Por otro lado, una de las grandes ventajas que presenta el diseño es la posibilidad de trabajar sin necesidad de estar conectado a un ordenador, sino siendo alimentado por sus entradas de 3.3 V. Así podría conectarse a las entradas analógicas de que dispone el adaptador de audio un instrumento como puede ser una guitarra eléctrica y podría ecualizarse la señal de salida de la guitarra, ajustando así la ecualización a las necesidades del usuario.

The aim of this project is to design a four-band audio signal equalization system with a variable gain between minus twelve and plus twelve decibels, carrying out the study from a theoretical point of view and putting the design into practice afterwards.

This design has been achieved by implementing in the theoretical part four graphic equalizers in parallel whose responses are summed and thus obtaining a customized signal at the output. On the other hand, for the design of the prototype we have chosen to develop it on the Teensy 3.6 development board together with its corresponding audio adapter, which not only presents an optimal response to the inputs provided by the user, but also presents a wide range of possibilities, highlighting among others the possibility of adding different types of audio effects such as *reverb* or *delay* without adding more electronic components to the design.

On the other hand, one of the great advantages of the design is the possibility of working without being connected to a computer, but being powered only by its 3.3 V inputs. Thus, an instrument such as an electric guitar could be connected to the analog inputs of the audio adapter and the output signal of the guitar could be equalized, thus adjusting the equalization to the user's needs.

11. REFERENCIAS BIBLIOGRÁFICAS

[1]. *https://en.wikipedia.org/wiki/Frequency*

[2]. *https://es.wikipedia.org/wiki/Frecuencia_de_corte*

[3]. *https://es.wikipedia.org/wiki/Respuesta_en_frecuencia*

[4]. *https://es.wikipedia.org/wiki/Funci%C3%B3n_de_transferencia*

[5]. Sergio Franco, *Design with Operational Amplifiers and Analog Integrated Circuit* 4º Edición McGraw Hill 2015 (Capitulo – 3)

[6]. https://es.wikipedia.org/wiki/Filtro_pasivo

[7]. Sergio Franco, *Design with Operational Amplifiers and Analog Integrated Circuit* 4º Edición McGraw Hill 2015 (Capitulo – 3.3)

[8]. Sergio Franco, *Design with Operational Amplifiers and Analog Integrated Circuit* 4º Edición McGraw Hill 2015 (Capitulo – 3.8)

[9]. Sergio Franco, *Design with Operational Amplifiers and Analog Integrated Circuit* 4º Edición McGraw Hill 2015 (Capitulo – 3.3)

[10]. S. Linkwitz e*n https://www.linkwitzlab.com/frontiers_5.htm#V* (Capitulo - Crossover topology issues)

[11]. Michael G.Ellis *Electronic filter analysis and synthesis* Artech House Publishers 1994

12. ANEXOS.

1. Hoja de datos del Amplificador Operacional TL082
2. Hoja de datos de TEENSY 3.6
3. Hoja de datos del adaptador de audio TEENSY 3.X AUDIO SHIELD
4. Hoja de datos del ENCODER
5. Tablas de valores de las resistencias SERIES E24, E48 Y E96
6. Código ecualizador de audio de 4 bandas

ANEXO 1

Hoja de datos del Amplificador Operacional TL082

- **Low Power Consumption**
- **Wide Common-Mode and Differential Voltage Ranges**
- **Low Input Bias and Offset Currents**
- **Output Short-Circuit Protection**
- **Low Total Harmonic Distortion . . . 0.003% Typ**
- **High Input Impedance . . . JFET-Input Stage**
- **Latch-Up-Free Operation**
- **High Slew Rate . . . 13 V/μs Typ**
- **Common-Mode Input Voltage Range Includes V_{CC+}**

description/ordering information

The TL08x JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient. Offset adjustment and external compensation options are available within the TL08x family.

The C-suffix devices are characterized for operation from 0°C to 70°C. The I-suffix devices are characterized for operation from –40°C to 85°C. The Q-suffix devices are characterized for operation from –40°C to 125°C. The M-suffix devices are characterized for operation over the full military temperature range of –55°C to 125°C.

ORDERING INFORMATION

T_J	V_{IO}max AT 25°C	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	15 mV	PDIP (P)	Tube of 50	TL081CP	TL081CP
			Tube of 50	TL082CP	TL082CP
		PDIP (N)	Tube of 25	TL084CN	TL084CN
		SOIC (D)	Tube of 75	TL081CD	TL081C
			Reel of 2500	TL081CDR	
			Tube of 75	TL082CD	TL082C
			Reel of 2500	TL082CDR	
			Tube of 50	TL084CD	TL084C
			Reel of 2500	TL084CDR	
		SOP (PS)	Reel of 2000	TL081CPSR	T081
			Reel of 2000	TL082CPSR	T082
		SOP (NS)	Reel of 2000	TL084CNSR	TL084
		TSSOP (PW)	Tube of 150	TL082CPW	T082
			Reel of 2000	TL082CPWR	
			Tube of 90	TL084CPW	T084
			Reel of 2000	TL084CPWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

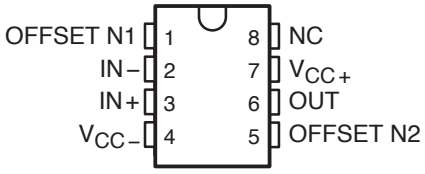
description/ordering information (continued)

ORDERING INFORMATION

T_J	V_{IO}max AT 25°C	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	6 mV	PDIP (P)	Tube of 50	TL081ACP	TL081ACP
			Tube of 50	TL082ACP	TL082ACP
		PDIP (N)	Tube of 25	TL084ACN	TL084ACN
		SOIC (D)	Tube of 75	TL081ACD	081AC
			Reel of 2500	TL081ACDR	
			Tube of 75	TL082ACD	082AC
			Reel of 2500	TL082ACDR	
			Tube of 50	TL084ACD	TL084AC
			Reel of 2500	TL084ACDR	
		SOP (PS)	Reel of 2000	TL082ACPSR	T082A
		SOP (NS)	Reel of 2000	TL084ACNSR	TL084A
	3 mV	PDIP (P)	Tube of 50	TL081BCP	TL081BCP
			Tube of 50	TL082BCP	TL082BCP
		PDIP (N)	Tube of 25	TL084BCN	TL084BCN
		SOIC (D)	Tube of 75	TL081BCD	081BC
			Reel of 2500	TL081BCDR	
			Tube of 75	TL082BCD	082BC
			Reel of 2500	TL082BCDR	
			Tube of 50	TL084BCD	TL084BC
			Reel of 2500	TL084BCDR	
–40°C to 85°C	6 mV	PDIP (P)	Tube of 50	TL081IP	TL081IP
			Tube of 50	TL082IP	TL082IP
		PDIP (N)	Tube of 25	TL084IN	TL081IN
		SOIC (D)	Tube of 75	TL081ID	TL081I
			Reel of 2500	TL081IDR	
			Tube of 75	TL082ID	TL082I
			Reel of 2500	TL082IDR	
			Tube of 50	TL084ID	TL084I
			Reel of 2500	TL084IDR	
		TSSOP (PW)	Reel of 2000	TL082IPWR	Z082
–40°C to 125°C	9 mV	SOIC (D)	Tube of 50	TL084QD	TL084QD
			Reel of 2500	TL084QDR	
–55°C to 125°C	9 mV	CDIP (J)	Tube of 25	TL084MJ	TL084MJ
		LCCC (FK)	Reel of 55	TL084FK	TL084FK
	6 mV	CDIP (JG)	Tube of 50	TL082MJG	TL082MJG
		LCCC (FK)	Tube of 55	TL082MFK	TL082MFK

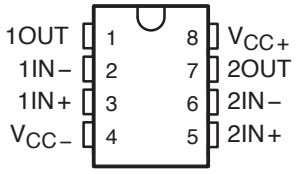
† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

TL081, TL081A, TL081B
D, P, OR PS PACKAGE
(TOP VIEW)

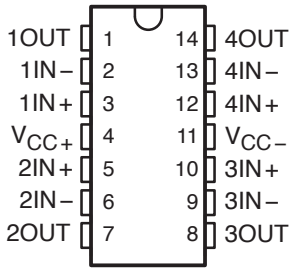


NC – No internal connection

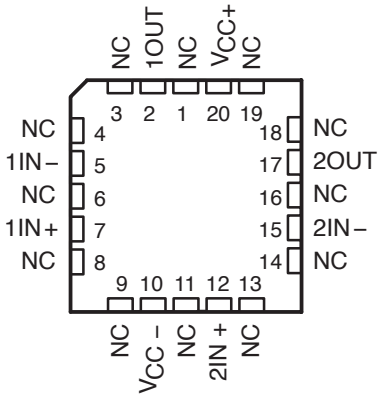
TL082, TL082A, TL082B
D, JG, P, PS, OR PW PACKAGE
(TOP VIEW)



TL084, TL084A, TL084B
D, J, N, NS, OR PW PACKAGE
(TOP VIEW)

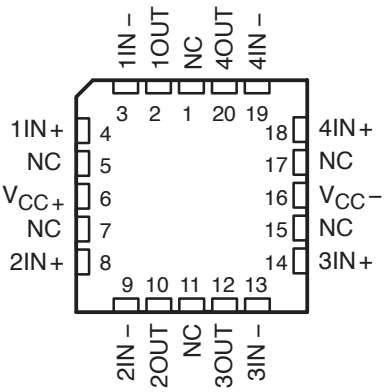


TL082M . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

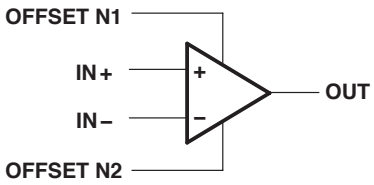
TL084M . . . FK PACKAGE
(TOP VIEW)



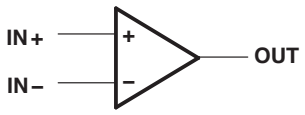
NC – No internal connection

symbols

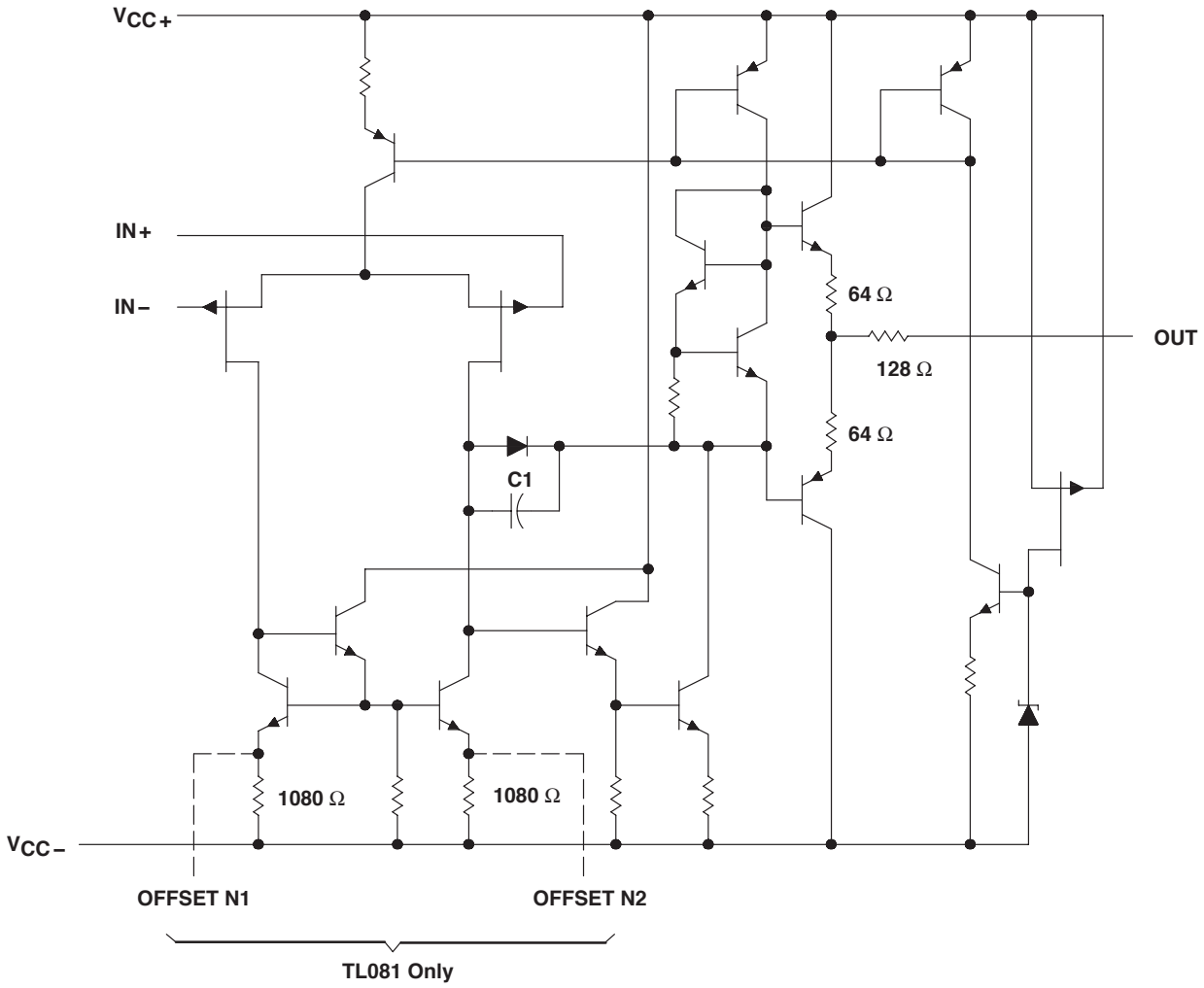
TL081



TL082 (EACH AMPLIFIER)
TL084 (EACH AMPLIFIER)



schematic (each amplifier)



Component values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

		TL08_C TL08_AC TL08_BC	TL08_I	TL084Q	TL08_M	UNIT
Supply voltage, V_{CC+} (see Note 1)		18	18	18	18	V
Supply voltage V_{CC-} (see Note 1)		–18	–18	–18	–18	V
Differential input voltage, V_{ID} (see Note 2)		±30	±30	±30	±30	V
Input voltage, V_I (see Notes 1 and 3)		±15	±15	±15	±15	V
Duration of output short circuit (see Note 4)		Unlimited	Unlimited	Unlimited	Unlimited	
Continuous total power dissipation		See Dissipation Rating Table				
Operating free-air temperature range, T_A		0 to 70	– 40 to 85	– 40 to 125	– 55 to 125	°C
Package thermal impedance, θ_{JA} (see Notes 5 and 6)	D package (8-pin)	97	97			°C/W
	D package (14-pin)	86	86			
	N package (14-pin)	76	76			
	NS package (14-pin)	80				
	P package (8-pin)	85	85			
	PS package (8-pin)	95	95			
	PW package (8-pin)	149				
	PW package (14-pin)	113	113			
Operating virtual junction temperature		150	150	150	150	°C
Case temperature for 60 seconds, T_C	FK package				260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	J or JG package				300	°C
Storage temperature range, T_{stg}		– 65 to 150	– 65 to 150	– 65 to 150	– 65 to 150	°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values, except differential voltages, are with respect to the midpoint between V_{CC+} and V_{CC-}.
2. Differential voltages are at IN+ with respect to IN–.
3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
4. The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.
5. Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
6. The package thermal impedance is calculated in accordance with JESD 51-7.

DISSIPATION RATING TABLE

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR	DERATE ABOVE T_A	$T_A = 70°C$ POWER RATING	$T_A = 85°C$ POWER RATING	$T_A = 125°C$ POWER RATING
D (14 pin)	680 mW	7.6 mW/°C	60°C	604 mW	490 mW	186 mW
FK	680 mW	11.0 mW/°C	88°C	680 mW	680 mW	273 mW
J	680 mW	11.0 mW/° C	88°C	680 mW	680 mW	273 mW
JG	680 mW	8.4 mW/°C	69°C	672 mW	546 mW	210 mW

electrical characteristics, $V_{CC\pm}$ = ±15 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS		T_A†	TL081C, TL082C, TL084C			TL081AC, TL082AC, TL084AC			TLO81BC, TL082BC, TL084BC			TL081I, TL082I, TL084I			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	V_O = 0	R_S = 50 Ω	25°C		3	15		3	6		2	3		3	6	mV
				Full range			20			7.5			5			9	
α_{VIO}	Temperature coefficient of input offset voltage	V_O = 0	R_S = 50 Ω	Full range		18			18			18			18		µV/°C
I_{IO}	Input offset current‡	V_O = 0		25°C		5	200		5	100		5	100		5	100	pA
				Full range			2			2			2			10	nA
I_{IB}	Input bias current‡	V_O = 0		25°C		30	400		30	200		30	200		30	200	pA
				Full range			10			7			7			20	nA
V_{ICR}	Common-mode input voltage range			25°C	±11	−12 to 15		±11	−12 to 15		±11	−12 to 15		±11	−12 to 15		V
V_{OM}	Maximum peak output voltage swing	R_L = 10 kΩ		25°C	±12	±13.5		±12	±13.5		±12	±13.5		±12	±13.5		V
		R_L ≥ 10 kΩ		Full range	±12			±12			±12			±12			
		R_L ≥ 2 kΩ			±10	±12		±10	±12		±10	±12		±10	±12		
A_{VD}	Large-signal differential voltage amplification	V_O = ±10 V,	R_L ≥ 2 kΩ	25°C	25	200		50	200		50	200		50	200		V/mV
		V_O = ±10 V,	R_L ≥ 2 kΩ	Full range	15			25			25			25			
B_1	Unity-gain bandwidth			25°C		3			3			3			3		MHz
r_i	Input resistance			25°C		10^{12}			10^{12}			10^{12}			10^{12}		Ω
CMRR	Common-mode rejection ratio	V_{IC} = V_{ICR}min, V_O = 0,	R_S = 50 Ω	25°C	70	86		75	86		75	86		75	86		dB
k_{SVR}	Supply-voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$)	V_{CC} = ±15 V to ± 9 V, V_O = 0,	R_S = 50 Ω	25°C	70	86		80	86		80	86		80	86		dB
I_{CC}	Supply current (per amplifier)	V_O = 0,	No load	25°C		1.4	2.8		1.4	2.8		1.4	2.8		1.4	2.8	mA
V_{O1}/V_{O2}	Crosstalk attenuation	A_{VD} = 100		25°C		120			120			120			120		dB

† All characteristics are measured under open-loop conditions with zero common-mode voltage, unless otherwise specified. Full range for T_A is 0°C to 70°C for TL08_C, TL08_AC, TL08_BC and –40°C to 85°C for TL08_I.

‡ Input bias currents of an FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive, as shown in Figure 17. Pulse techniques must be used that maintain the junction temperature as close to the ambient temperature as possible.

ANEXO 2

Hoja de datos de TEENSY 3.6

Kinetis K66 Sub-Family

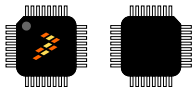
180 MHz ARM® Cortex®-M4F Microcontroller.

MK66FN2M0VMD18
MK66FX1M0VMD18
MK66FN2M0VLQ18
MK66FX1M0VLQ18

The K66 sub-family members provide greater performance, memory options up to 2 MB total flash and 256 KB of SRAM, as well as higher peripheral integration with features such as Dual USB and a 10/100 Mbit/s Ethernet MAC. These devices maintain hardware and software compatibility with the existing Kinetis family. This product also offers:

- Integration of a High Speed USB Physical Transceiver
- Greater performance flexibility with a High Speed Run mode
- Smarter peripherals with operation in Stop modes

144 MAPBGA (MD)
13 mm x 13 mm Pitch 1 mm



144 LQFP (LQ)
20 mm x 20 mm Pitch 0.5 mm

Performance

- Up to 180 MHz ARM Cortex-M4 based core with DSP instructions and Single Precision Floating Point unit

System and Clocks

- Multiple low-power modes to provide power optimization based on application requirements
- Memory protection unit with multi-master protection
- 3 to 32 MHz main crystal oscillator
- 32 kHz low power crystal oscillator
- 48 MHz internal reference

Security

- Hardware random-number generator
- Supports DES, AES, SHA accelerator (CAU)
- Multiple levels of embedded flash security

Timers

- Four Periodic interrupt timers
- 16-bit low-power timer
- Two 16-bit low-power timer PWM modules
- Two 8-channel motor control/general purpose/PWM timers
- Two 2-ch quad decoder/general purpose timers
- Real-time clock

Human-machine interface

- Low-power hardware touch sensor interface (TSI)
- General-purpose input/output

Memories and memory expansion

- Up to 2 MB program flash memory on non-FlexMemory devices with 256 KB RAM
- Up to 1 MB program flash memory and 256 KB of FlexNVM on FlexMemory devices
- 4 KB FlexRAM on FlexMemory devices
- FlexBus external bus interface and SDRAM controller

Analog modules

- Two 16-bit SAR ADCs and two 12-bit DAC
- Four analog comparators (CMP) containing a 6-bit DAC and programmable reference input
- Voltage reference 1.2V

Communication interfaces

- Ethernet controller with MII and RMII interface to external PHY and hardware IEEE 1588 capability
- USB high-/full-/low-speed On-the-Go with on-chip high speed transceiver
- USB full-/low-speed OTG with on-chip transceiver
- Two CAN, three SPI and four I2C modules
- Low Power Universal Asynchronous Receiver/ Transmitter 0 (LPUART0) and five standard UARTs
- Secure Digital Host Controller (SDHC)
- I2S module

Operating Characteristics

- Voltage/Flash write voltage range:1.71 to 3.6 V
- Temperature range (ambient): -40 to 105°C

Ordering Information [1]

Part Number	Memory		Maximum number of I\O's
	Flash	SRAM	
MK66FN2M0VMD18	2 MB	256 KB	100
MK66FX1M0VMD18	1.25 MB	256 KB	100
MK66FN2M0VLQ18	2 MB	256 KB	100
MK66FX1M0VLQ18	1.25 MB	256 KB	100

1. To confirm current availability of orderable part numbers, go to http://www.freescale.com and perform a part number search.

Related Resources

Type	Description	Resource
Selector Guide	The Freescale Solution Advisor is a web-based tool that features interactive application wizards and a dynamic product selector.	Solution Advisor
Reference Manual	The Reference Manual contains a comprehensive description of the structure and function (operation) of a device.	K66P144M180SF5RMV2[1]
Data Sheet	The Data Sheet includes electrical characteristics and signal connections.	This document.
Chip Errata	The chip mask set Errata provides additional or corrective information for a particular device mask set.	Kinetis_K_0N65N [1]
Package drawing	Package dimensions are provided in package drawings.	MAPBGA 144-pin : 98ASA00222D[1] LQFP 144-pin: 98ASS23177W[1]

1. To find the associated resource, go to http://www.freescale.com and perform a search using this term.

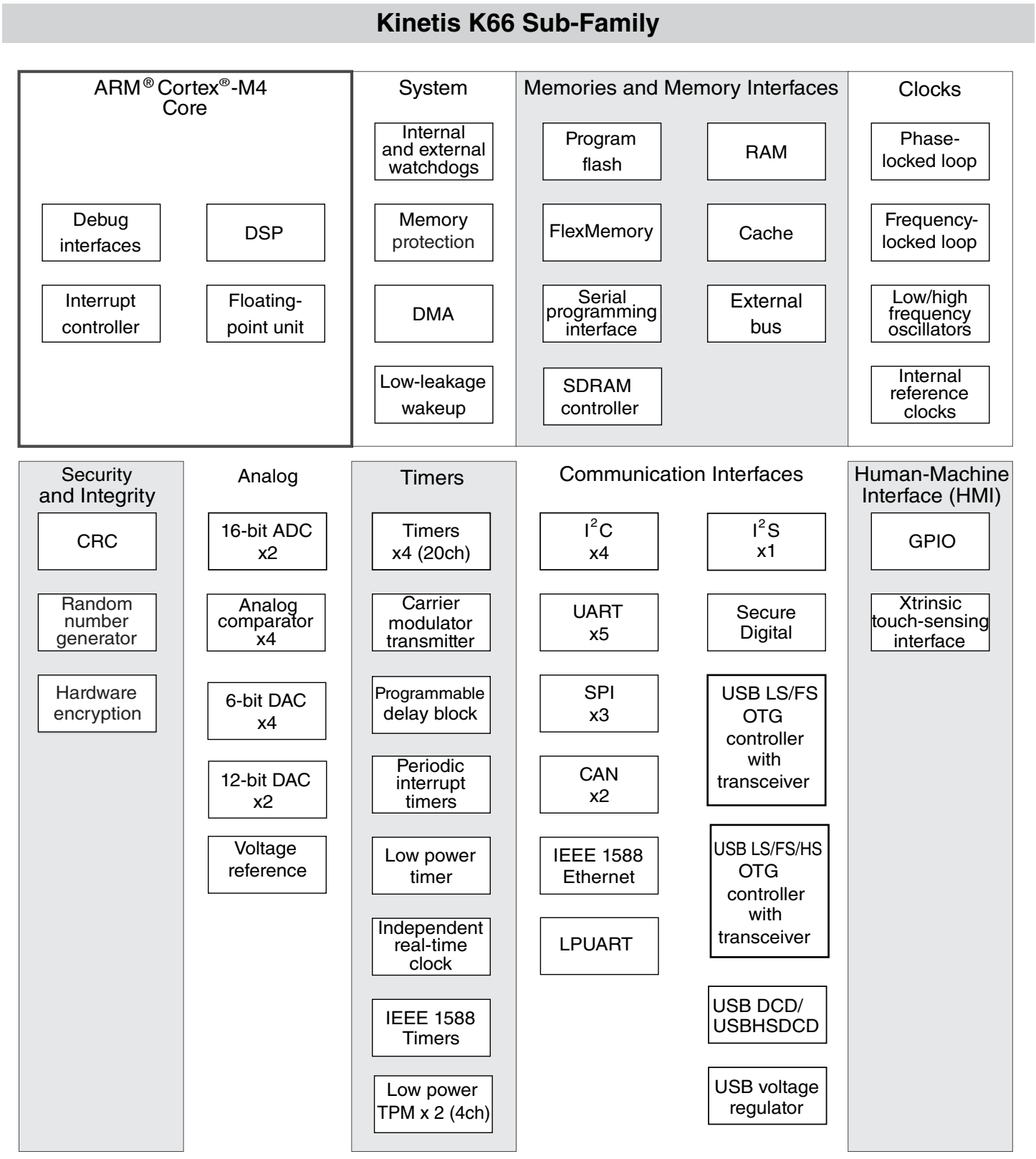


Figure 1. K66 Block Diagram

Table of Contents

ANEXO 3

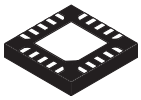
Hoja de datos del adaptador de audio TEENSY 3.X

AUDIOSHIELD

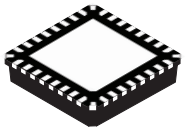
Low Power Stereo Codec with Headphone Amp

SGTL5000

AUDIO CODEC



PB-FREE
98ARE10742D
20-PIN QFN



PB-FREE
98ARE10739D
32-PIN QFN

The SGTL5000 is a Low Power Stereo Codec with Headphone Amp from Freescale, and is designed to provide a complete audio solution for products needing LINEIN, MIC_IN, LINEOUT, headphone-out, and digital I/O. Deriving it's architecture from best in class, Freescale integrated products that are currently on the market. The SGTL5000 is able to achieve ultra low power with very high performance and functionality, all in one of the smallest footprints available. Target markets include media players, navigation devices, smart phones, tablets, medical equipment, exercise equipment, consumer audio equipment, etc. Features such as capless headphone design and an internal PLL help lower overall system cost.

Features

Analog Inputs

- Stereo LINEIN - Support for external analog input
- Stereo LINEIN - Codec bypass for low power
- MIC bias provided
- Programmable MIC gain
- ADC - 85 dB SNR (-60 dB input) and -73 dB THD+N (VDDA = 1.8 V)

Analog Outputs

- HP Output - Capless design
- HP Output - 62.5 mW max, 1.02 kHz sine into 16 Ω load at 3.3 V
- HP Output - 100 dB SNR (-60 dB input) and -80 dB THD+N (V_{DDA} = 1.8 V, 16 Ω load, DAC to headphone)
- LINEOUT - 100 dB SNR (-60 dB input) and -85 dB THD+N (V_{DDIO} = 3.3 V)

Digital I/O

- I^2S port to allow routing to Application Processor

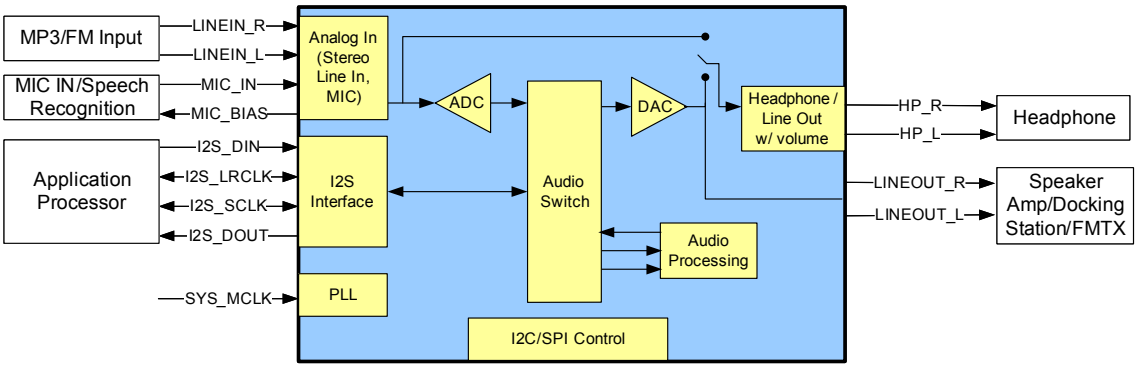
Integrated Digital Processing

- Freescale surround, Freescale bass, tone control/ parametric equalizer/graphic equalizer clocking/control
- PLL allows input of an 8.0 MHz to 27 MHz system clock - standard audio clocks are derived from PLL

Power Supplies

- Designed to operate from 1.62 to 3.6 volts

ORDERING INFORMATION		
Device	**Temperature Range (T_A)**	**Package**
SGTL5000XNLA3/R2	-40 to 85 °C	20 QFN
SGTL5000XNAA3/R2		32 QFN



Note: SPI is not supported in the 3.0 mm x 3.0 mm 20-pin QFN package

Figure 1. SGTL5000 Simplified Application Diagram

INTERNAL BLOCK DIAGRAM

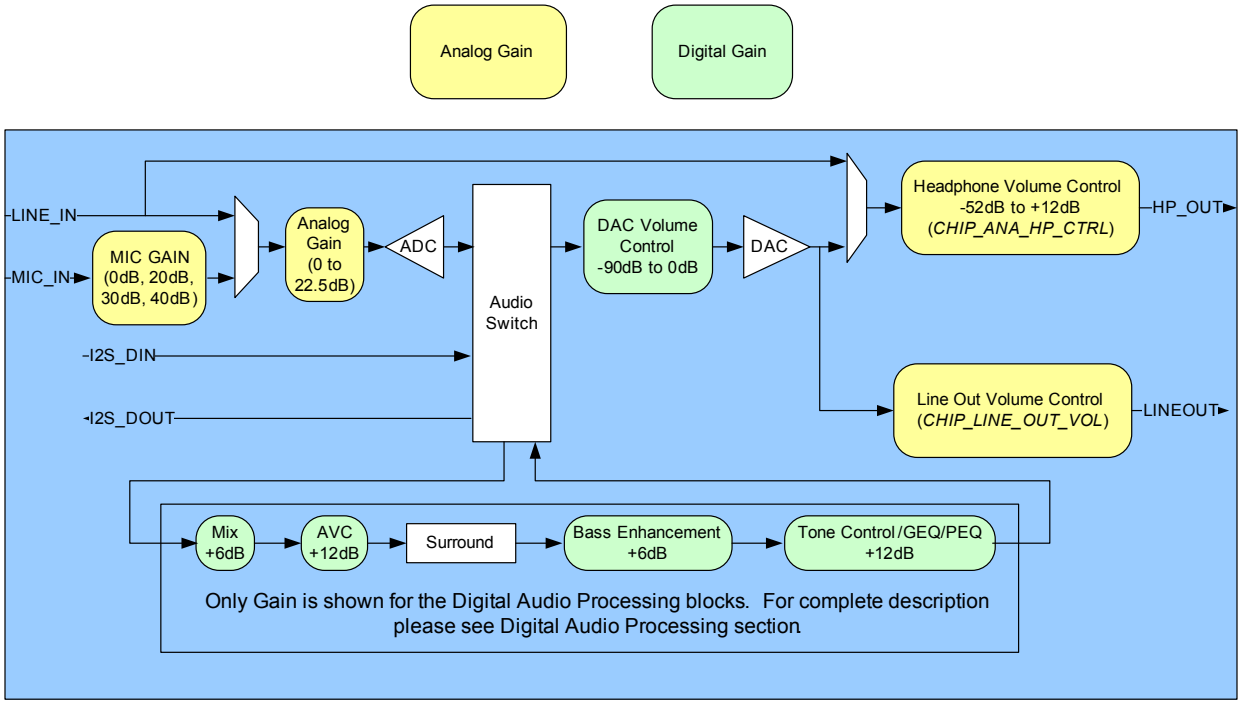


Figure 2. SGTL5000 Simplified Internal Block Diagram

PIN CONNECTIONS

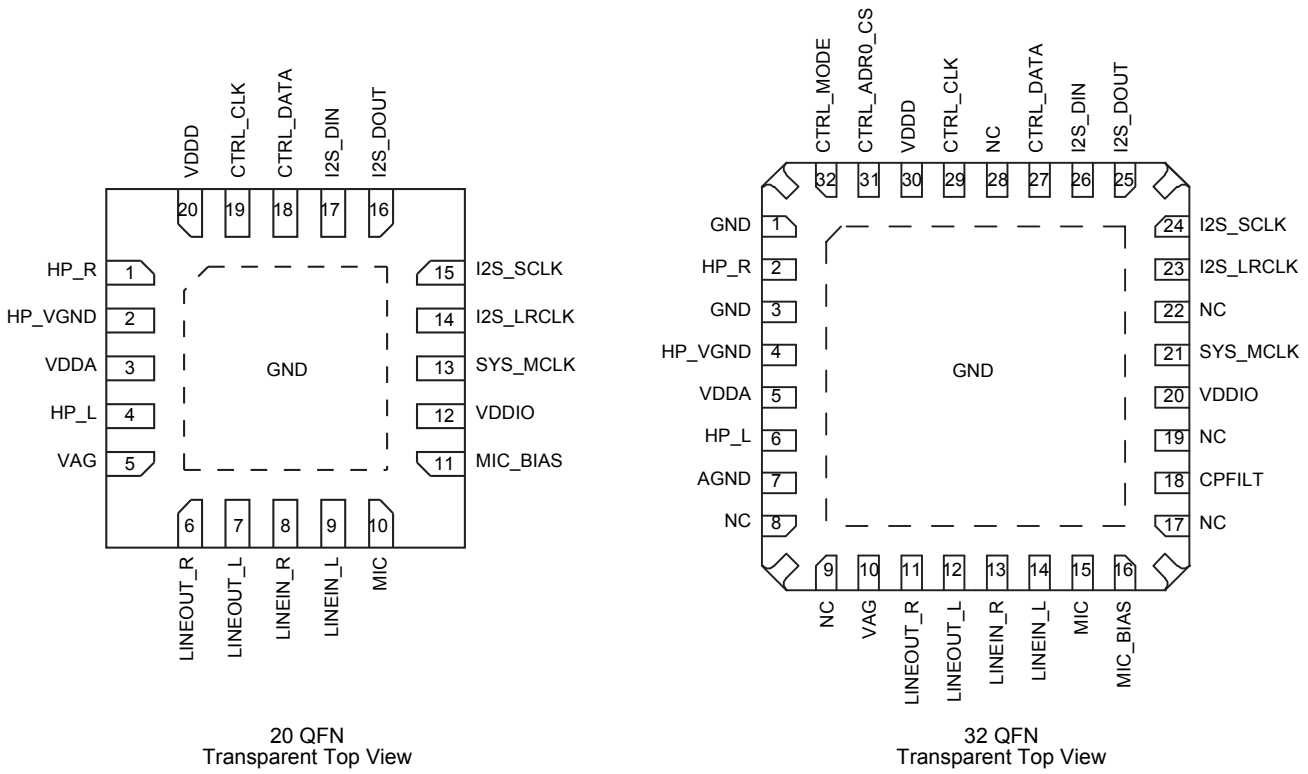


Figure 3. SGTL5000 Pin Connections

A functional description can be found in Functional Description, beginning on page 12.

Table 1. SGTL5000 Pin Definitions

20 Pin QFN	32 Pin QFN	Pin Name	Pin Function	Formal Name	Definition
1	2	HP_R	Analog	Right headphone output	
2	4	HP_VGND	Analog	Headphone virtual ground	Do not connect HP_VGND to system ground, even when unused. This is a virtual ground (DC voltage) that should never connect to an actual “0 Volt ground”. Use the widest, shortest trace possible for the HP_VGND.
3	5	VDDA	Power	Analog voltage	
4	6	HP_L	Analog	Left headphone output	
-	7	AGND	Analog Ground	Ground	
-	8, 9, 17, 19, 22, 28	NC	No Connect		
5	10	VAG	Analog	DAC VAG filter	
6	11	LINEOUT_R	Analog	Right LINEOUT	
7	12	LINEOUT_L	Analog	Left LINEOUT	
8	13	LINEIN_R	Analog	Right LINEIN	
9	14	LINEIN_L	Analog	Left LINEIN	

Table 1. SGTL5000 Pin Definitions (continued)

20 Pin QFN	32 Pin QFN	Pin Name	Pin Function	Formal Name	Definition
10	15	MIC	Analog	Microphone input	
11	16	MIC_BIAS	Analog	Mic bias	
—	18	CPFILT	Analog	Charge Pump Filter	The CPFILT cap value is 0.1 μF. If both VDDIO and VDDA are ≤ 3.0 V, the CPFILT pin must be connected to a 0.1 μF cap to GND. If either is > 3.0 V, the CPFILT cap MUST NOT be placed.
12	20	VDDIO	Power	Digital I/O voltage	
13	21	SYS_MCLK	Digital	System master clock	
14	23	I2S_LRCLK	Digital	I^2S frame clock	
15	24	I2S_SCLK	Digital	I^2S bit clock	
16	25	I2S_DOUT	Digital	I^2S data output	
17	26	I2S_DIN	Digital	I^2S data input	
18	27	CTRL_DATA	Digital	I^2C Mode: Serial Data (SDA); SPI Mode: Serial Data Input (MOSI)	
19	29	CTRL_CLK	Digital	I^2C Mode: Serial Clock (SCL); SPI Mode: Serial Clock (SCK)	
20	30	VDDD	Digital	Digital voltage	For new designs, connect VDDD to an external voltage source and to a 0.1 μF capacitor to GND.
-	31	CTRL_ADR0_CS	Digital	I^2C Mode: I^2C Address Select 0; SPI Mode: SPI Chip Select	
-	32	CTRL_MODE	Digital	Mode select for I^2C or SPI; When pulled low the control mode is I^2C, when pulled high the control mode is SPI	
PAD	1, 3, 4, PAD	GND	Ground	Ground	The PAD must be soldered to ground. Star the ground pins of the chip, VAG ground, and all analog inputs/outputs to a single point, then to the ground plane.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Table 2. Maximum Ratings

Exceeding the absolute maximum ratings shown in the following table could cause permanent damage to the part and is not recommended. Normal operation is not guaranteed at the absolute maximum ratings, and extended exposure could affect long term reliability.

Ratings	Symbol	Value	Unit
ELECTRICAL RATINGS			
Maximum Digital Voltage	V_{DDD}	1.98	V
Maximum Digital I/O Voltage	V_{DDIO}	3.6	V
Maximum Analog Supply Voltage	V_{DDA}	3.6	V
Maximum voltage on any digital input		GND-0.3 to V_{DDIO}+0.3	V
Maximum voltage on any analog input		GND-0.3 to V_{DDA}+0.3	V
THERMAL RATINGS			
Storage Temperature	T_{STG}	-55 to 125	°C
Operating Temperature Ambient	T_A	-40 to 85	°C

Table 3. Recommended Operating Conditions

Ratings	Symbol	Value	Unit
Digital Voltage (If supplied externally). External VDDD connection required for new designs.	V_{DDD}	1.1 to 2.0	V
Digital I/O Voltage	V_{DDIO}	1.62 to 3.6	V
Analog Supply Voltage	V_{DDA}	1.62 to 3.6	V

Table 4. Input/Output Electrical Characteristics

Test Conditions unless otherwise noted: V_{DDIO} = 3.3 V, V_{DDA} = 3.3 V, T_A = 25 °C, Slave mode, f_S = 48 kHz, MCLK = 256 f_S, 24 bit input, 1.02 kHz sine.

Characteristic	Symbol	Min	Typ	Max	Unit
LINEIN Input Level (3.3 V VDDA)		-	-	2.83	V_{PP}
LINEIN Input Level (1.8 V VDDA)		-	-	1.60	V_{PP}
MIC Input Level (3.3 V VDDA)		-	-	2.83	V_{PP}
MIC Input Level (1.8 V VDDA)		-	-	1.60	V_{PP}
LINEOUT Output level 0 dBFS at 1.031 kHz 12S input, 1.8 V LINEOUT supply (normally VDDIO), 10 kΩ load		1.46	1.52	1.68	V_{PP}
LINEOUT Output level 0 dBFS at 1.031 kHz 12S input, 3.3 V LINEOUT supply (normally VDDIO), 10 kΩ load		2.53	2.61	3.11	V_{PP}
LINEIN Input Impedance		-	29	-	kΩ
MIC Input Impedance		-	2.9	-	kΩ
LINEOUT Output Impedance		-	320	-	Ω
LINEOUT Load		10	-	-	kΩ
HP (headphone) Load		16	-	-	Ω
SYS_MCLK Input Voltage swing		-0.3	V_{DDIO}	V_{DDIO}+0.3	V
SYS_MCLK Rise/Fall Time		0.5	-	10	ns

STATIC ELECTRICAL CHARACTERISTICS

Table 5. Audio Performance 1

Test Conditions unless otherwise noted: V_{DDIO} = 1.8 V, V_{DDA} = 1.8 V, T_A = 25 °C, Slave mode, f_S = 48 kHz, MCLK = 256 f_S, 24 bit input

Characteristic	Symbol	Min	Typ	Max	Unit
AUDIO PERFORMANCE					
LINEIN Input Level		-	0.57	-	V_{RMS}
LINEIN Input Impedance (at 1.02 kHz)		-	29	-	kΩ
LINEIN -> ADC -> I^2S OUT					
SNR (-60 dB input)		-	85	-	dB
THD+N		-	-70	-	dB
Frequency Response		-	±0.11	-	dB
Channel Separation		-	79	-	dB
LINEIN -> HEADPHONE_LINEOUT (CODEC BYPASS MODE)					
SNR (-60 dB input)		-	98	-	dB
THD+N (10 kΩ load)		-	-87	-	dB
THD+N (16 Ω load)		-	-87	-	dB
Frequency Response		-	±0.05	-	dB
Channel Separation (at 1.0 kHz)			82		dB
I^2S IN -> DAC -> LINEOUT					
Output Level		-	0.6	-	V_{RMS}
SNR (-60 dB input)		-	95	-	dB
THD+N		-	-85	-	dB
Frequency Response		-	±0.12	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 16 Ω LOAD					
Output Power		-	17	-	mW
SNR (-60 dB input)		-	100	-	dB
THD+N		-	-80	-	dB
Frequency Response		-	±0.12	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 32 Ω LOAD					
Output Power		-	10	-	mW
SNR (-60 dB input)		-	95	-	dB
THD+N		-	-86	-	dB
Frequency Response		-	±0.11	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 10 KΩ LOAD					
SNR (-60 dB input)		-	96	-	dB
THD+N		-	-84	-	dB
Frequency Response		-	±0.11	-	dB
PSRR (200 mVp-p at 1.0 kHz on VDDA)		-	85	-	dB

Table 6. Audio Performance 2

Test Conditions unless otherwise noted: V_{DDIO} = 3.3 V, V_{DDA} = 3.3 V, T_A = 25°C, Slave mode, f_S = 48 kHz, MCLK = 256 f_S, 24 bit input. ADC tests were conducted with BIAS_CTRL = -37.5%, all other tests conducted with BIAS_CTRL = -50%.

Characteristic	Symbol	Min	Typ	Max	Unit
AUDIO PERFORMANCE					
LINEIN Input Level		-	1.0	-	V_{RMS}
LINEIN Input Impedance (at 1.02 kHz)		-	29	-	kΩ
LINEIN -> ADC -> I^2S OUT					
SNR (-60 dB input)		-	90	-	dB
THD+N		-	-72	-	dB
Frequency Response		-	±0.11	-	dB
Channel Separation		-	80	-	dB
LINEIN -> HEADPHONE_LINEOUT (CODEC BYPASS MODE)					
SNR (-60 dB input)		-	102	-	dB
THD+N (10 kΩ load)		-	-89	-	dB
THD+N (16 Ω load)		-	-87	-	dB
Frequency Response		-	±0.05	-	dB
Channel Separation (at 1.0 kHz)			81		dB
I^2S IN -> DAC -> LINEOUT					
Output Level		-	1.0	-	V_{RMS}
SNR (-60 dB input)		-	100	-	dB
THD+N		-	-85	-	dB
Frequency Response		-	±0.12	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 16 Ω LOAD					
Output Power		-	58	-	mW
SNR (-60 dB input)		-	98	-	dB
THD+N		-	-86	-	dB
Frequency Response		-	±0.12	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 32 Ω LOAD					
Output Power		-	30	-	mW
SNR (-60 dB input)		-	100	-	dB
THD+N		-	-88	-	dB
Frequency Response		-	±0.11	-	dB
I^2S IN -> DAC -> HEADPHONE OUT - 10 KΩ LOAD					
SNR (-60 dB input)		-	97	-	dB
THD+N		-	-85	-	dB
Frequency Response		-	±0.11	-	dB
PSRR (200 mVp-p at 1.0 kHz on VDDA)		-	89	-	dB

DYNAMIC ELECTRICAL CHARACTERISTICS

Table 7. Dynamic Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
POWER UP TIMING					
Time from all supplies powered up and SYS_MCLK present to initial communication. See Figure 4.	t_{PC}	1.0(2)	–	-	µs
I2C BUS TIMING(3) See Figure 5.					
I^2C Serial Clock Frequency	f_{I2C_CLK}	-	-	400	kHz
I^2C Start condition hold time	t_{I2CSH}	150	-	-	ns
I^2C Stop condition setup time	$t_{I2CSTSU}$	150	-	-	ns
I^2C Data input setup time to rising edge of CTRL_CLK	t_{I2CDSU}	125	-	-	ns
I^2C Data input hold time from falling edge of CTRL_CLK (receiving data)	t_{I2CDH}	5.0	-	-	ns
I^2C Data input hold time from falling edge of CTRL_CLK (driving data)	t_{I2CDH}	360	-	-	ns
I^2C CTRL_CLK low time	$t_{I2CCLKL}$	300	-	-	ns
I^2C CTRL_CLK high time	$t_{I2CCLKH}$	100	-	-	ns
SPI BUS TIMING(4) See Figure 6.					
SPI Serial Clock Frequency	f_{SPI_CLK}	-	-	TBD	MHz
SPI data input setup time	t_{SPIDSU}	10	-	-	ns
SPI data input hold time	t_{SPIDH}	10	-	-	ns
SPI CTRL_CLK low time	$t_{SPICLKL}$	TBD	-	-	ns
SPI CTRL_CLK high time	$t_{SPICLKH}$	TBD	-	-	ns
SPI clock to chip select	t_{CCS}	60	-	-	ns
SPI chip select to clock	t_{CSC}	20	-	-	ns
SPI chip select low	t_{CSL}	20	-	-	ns
SPI chip select high	t_{CSH}	20			ns
SPECIFICATIONS AND TIMING FOR THE I^2S PORT(5) See Figure 7.					
Frequency of I^2S_LRCLK	f_{LRCLK}	8.0	-	96	kHz
Frequency of I^2S_SCLK	f_{SCLK}	-	$32*f_{LRCLK}$ $64*f_{LRCLK}$	-	kHz
I^2S delay	t_{I2S_D}	-	-	10	ns
I^2S setup time	t_{I2S_S}	10	-	-	ns
I^2S hold time	t_{I2S_H}	10	-	-	ns

Notes

1. The SGTL5000 has an internal reset that is deasserted 8 SYS_MCLK cycles after all power rails have been brought up. After this time, communication can start.
2. 1.0µs represents 8 SYS_MCLK cycles at the minimum 8.0 MHz SYS_MCLK.
3. This section provides timing for the SGTL5000 while in I^2C mode (CTRL_MODE = 0).
4. This section provides timing for the SGTL5000 while in SPI mode (CTRL_MODE = 1)
5. The following are the specifications and timing for I^2S port. The timing applies to all formats.

TIMING DIAGRAMS

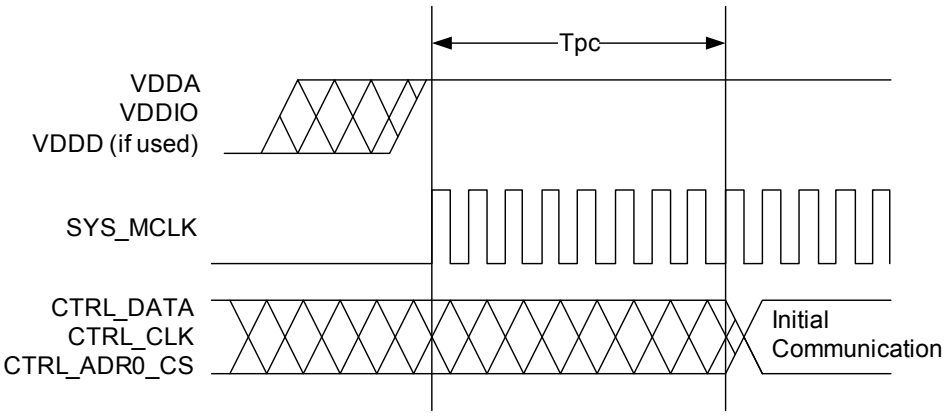


Figure 4. Power Up Timing

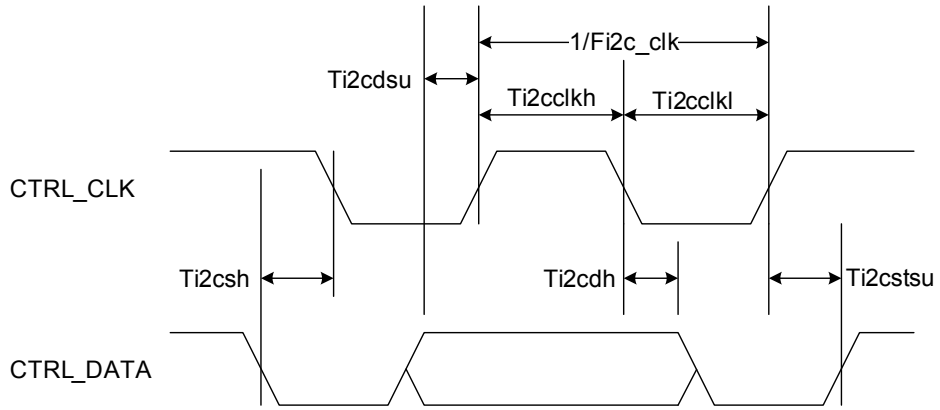


Figure 5. I²C Timing (CTRL_MODE == 0)

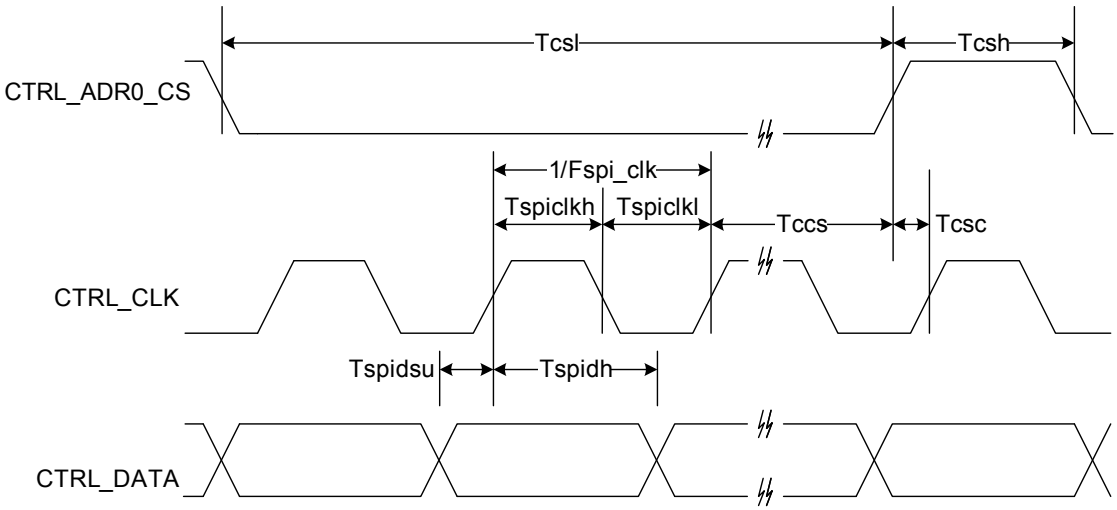


Figure 6. SPI Timing

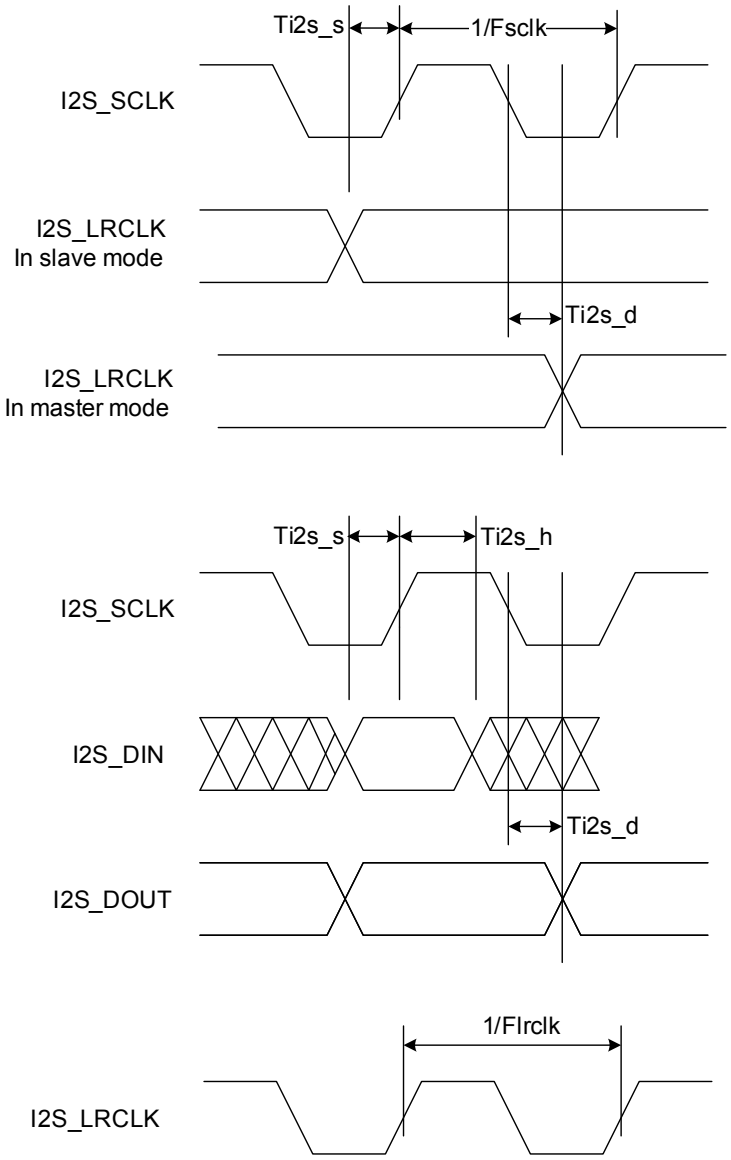


Figure 7. I²S Interface Timing

FUNCTIONAL DESCRIPTION

INTRODUCTION

The SGTL5000 is a low power stereo codec with integrated headphone amplifier. It is designed to provide a complete audio solution for portable products needing LINEIN, mic-in, LINEOUT, headphone-out, and digital I/O. Deriving it's architecture from best in class Freescale integrated products that are currently on the market, the SGTL5000 is able to achieve ultra low power with very high performance and functionality, all in one of the smallest footprints available. Target markets include portable media players, GPS units and smart phones. Features such as capless headphone design and USB clocking mode (12 MHz SYS_MCLK input) help lower overall system cost.

In summary, the SGTL5000 accepts the following inputs:

- Line input
- Microphone input, with mic bias
- Digital I^2S input

In addition, the SGTL5000 supports the following outputs:

- Line output
- Headphone output
- Digital I^2S output

The following digital audio processing is included to allow for product differentiation:

- Digital mixer
- Freescale Surround
- Freescale Bass Enhancement
- Tone Control, parametric equalizer, or graphic equalizer

The SGTL5000 can accept an external standard master clock at a multiple of the sampling frequency (i.e. 256*Fs, 385*Fs, 512*Fs). In addition it can take non-standard frequencies and use the internal PLL to derive the audio clocks. The device supports 8.0 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1kHz, 48 kHz, 96 kHz sampling frequencies.

FUNCTIONAL INTERNAL BLOCK DESCRIPTION

SYSTEM BLOCK DIAGRAM W/ SIGNAL FLOW AND GAIN MAP

Figure 8 shows a block diagram that highlights the signal flow and gain map for the SGTL5000.

To guarantee against clipping, it is important that the gain in a signal path in addition to the signal level does not exceed 0 dB at any point.

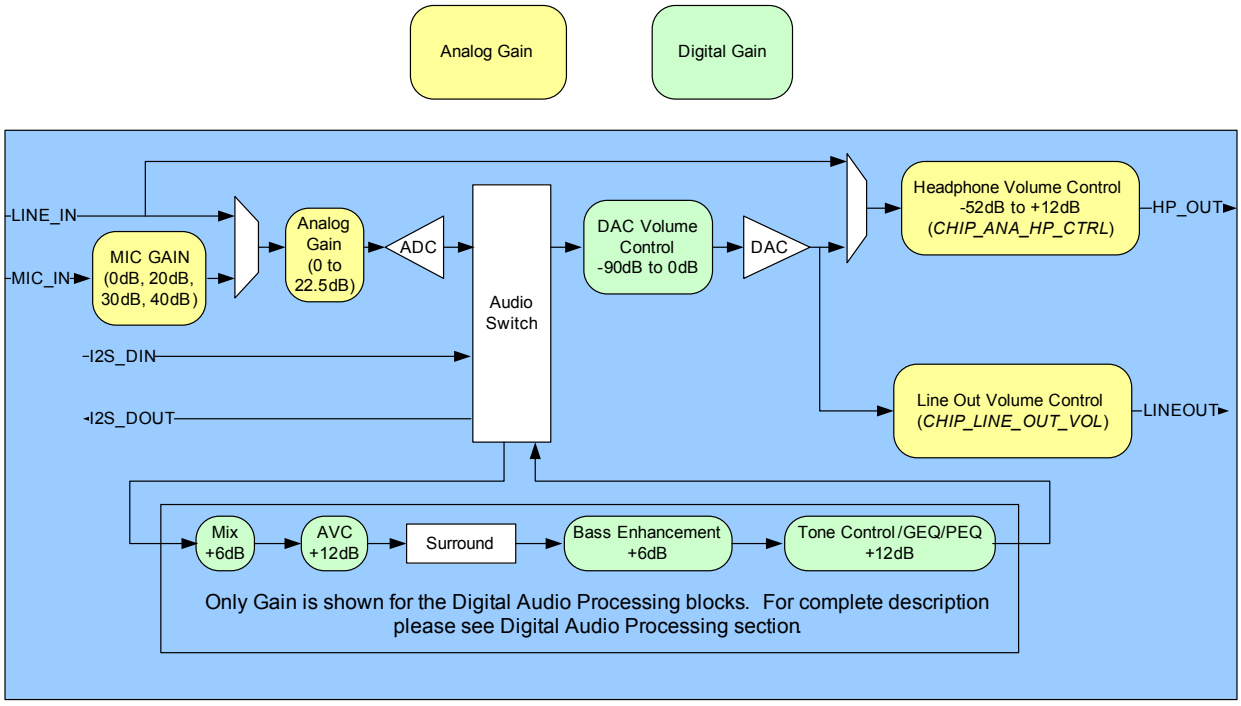


Figure 8. System Block Diagram, Signal Flow and Gain

POWER

The SGTL5000 has a flexible power architecture to allow the system designer to minimize power consumption and maximize performance at the lowest cost.

External Power Supplies

The SGTL5000 requires 2 external power supplies: VDDA and VDDIO. An optional third external power supply VDDD may be provided externally to achieve lower power. This external VDDD power supply is required for new designs. A description for the different power supplies is as follows:

- VDDA: This external power supply is used for the internal analog circuitry including ADC, DAC, LINE inputs, MIC inputs, headphone outputs and reference voltages. VDDA supply ranges are shown in Maximum Ratings. A decoupling cap should be used on VDDA, as shown in the typical application diagrams in Typical Applications.
- VDDIO: This external power supply controls the digital I/O levels as well as the output level of LINE outputs. VDDIO supply ranges are shown in Maximum Ratings. A decoupling cap should be used on VDDIO as shown in the typical application diagrams in Typical Applications.

Note that if VDDA and VDDIO are derived from the same voltage, a single decoupling capacitor can be used to minimize cost. This capacitor should be placed closest to VDDA.

- VDDD: This is a digital power supply that is used for internal digital circuitry. An external VDDD power supply is required for new designs. For lowest power, this supply can be driven at the lowest specified voltage given in Maximum Ratings. If an external supply is used for VDDD, a decoupling capacitor is recommended, as shown in the typical applications diagram. VDDD supply ranges are shown in Maximum Ratings for when externally driven. If the system drives VDDD externally, an efficient switching supply should be used or no system power savings is realized.

Internal Power Supplies

The SGTL5000 has two exposed internal power supplies, VAG and charge pump.

- VAG is the internal voltage reference for the ADC and DAC. After startup the voltage of VAG should be set to VDDA/2 by writing *CHIP_REF_CTRL->VAG_VAL*. Refer to programming Chip Powerup and Supply Configurations. The VAG pin should have an external filter capacitor as shown in the typical application diagram.
- Chargepump: This power supply is used for internal analog switches. If VDDA or VDDIO is greater than 2.7 V, this supply is automatically driven from the highest of VDDIO and VDDA. If both VDDIO and VDDA are less than 3.1 V, then the user should turn on the charge pump function to create the charge pump rail from VDDIO by writing *CHIP_ANA_POWER-> VDDC_CHRGPMP_POWERUP* register. Refer to programming Chip Powerup and Supply Configurations.
- LINE_OUT_VAG is the line output voltage reference. It should be set to VDDIO/2 by writing *CHIP_LINE_OUT_CTRL->LO_VAGCNTRL*.

Power Schemes

The SGTL5000 supports a flexible architecture and allows the system designer to minimize power or maximize BOM savings.

- For maximum cost savings, all supplies can be run at the same voltage.
- Alternatively for minimum power, the analog and digital supplies can be run at minimum voltage while driving the digital I/O voltage at the voltage needed by the system.
- To save power, independent supplies are provided for line outputs and headphone outputs. This allows for 1VRMS line outputs while using minimal headphone power.
- For best power, VDDA should be run at the lowest possible voltage required for the maximum headphone output level. For highest performance, VDDA should be run at 3.3 V. For most applications a lower voltage can be used for the best performance/power combination.

RESET

The SGTL5000 has an internal reset that is deasserted 8 SYS_MCLKs after all power rails have been brought up. After this time communication can start. See Dynamic Electrical Characteristics.

CLOCKING

Clocking for the SGTL5000 is provided by a system master clock input (SYS_MCLK). SYS_MCLK should be synchronous to the sampling rate (Fs) of the I^2S port. Alternatively any clock between 8.0 and 27 MHz can be provided on SYS_MCLK and the SGTL5000 can use an internal PLL to derive all internal and I^2S clocks. This allows the system to use an available clock such as 12 MHz (common USB clock) for SYS_MCLK to reduce overall system costs.

Synchronous SYS_MCLK input

The SGTL5000 supports various combinations of SYS_MCLK frequency and sampling frequency as shown in Table 8. Using a synchronous SYS_MCLK allows for lower power as the internal PLL is not used.

Table 8. Synchronous MCLK Rates and Sampling Frequencies

CLOCK	SUPPORTED RATES	UNITS
System Master Clock (SYS_MCLK)	256, 384, 512	Fs
Sampling Frequency (Fs)	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 96 (6)	kHz

Notes
6. For a sampling frequency of 96 kHz, only 256 Fs SYS_MCLK is supported

Using the PLL - Asynchronous SYS_MCLK input

An integrated PLL is provided in the SGTL5000 that allows any clock from 8.0 to 27 MHz to be connected to SYS_MCLK. This can help save system costs, as a clock available elsewhere in the system can be used to derive all audio clocks using the internal PLL. In this case, the clock input to SYS_MCLK can be asynchronous with the sampling frequency needed in the system. For example, a 12 MHz clock from the system processor could be used as the clock input to the SGTL5000.

Three register fields need to be configured to properly use the PLL. They are *CHIP_PLL_CTRL->INT_DIVISOR*, *CHIP_PLL_CTRL->FRAC_DIVISOR* and *CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2*. Figure 9 shows a flowchart that shows how to determine the values to program in the register fields.

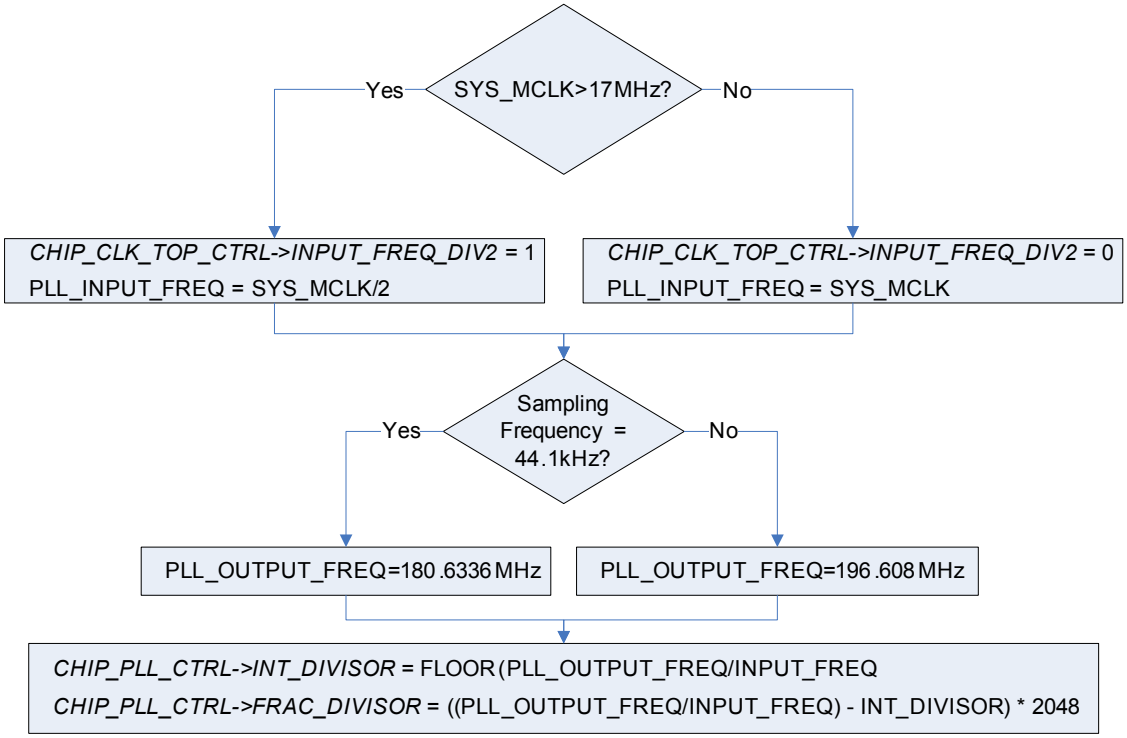


Figure 9. PLL Programming Flowchart

For example, when a 12 MHz digital signal is placed on MCLK, for a 48 kHz frame clock

CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0 // SYS_MCLK < 17 MHz

CHIP_PLL_CTRL->INT_DIVISOR = FLOOR (196.608 MHz/12 MHz) = 16 (decimal)

CHIP_PLL_CTRL->FRAC_DIVISOR = ((196.608 MHz/ 12 MHz) - 16) * 2048 = 786 (decimal)

Refer to PLL programming PLL Configuration.

AUDIO SWITCH (SOURCE SELECT SWITCH)

The audio switch is the central routing block that controls the signal flow from input to output. Any single input can be routed to any single or multiple outputs.

Any signal can be routed to the Digital Audio Processor (DAP). The output of the DAP (an input to the audio switch) can in turn be routed to any physical output. The output of the DAP can not be routed into itself. Refer to Digital Audio Processing, for DAP information and configuration.

It should be noted that the analog bypass from Line input to headphone output does not go through the audio switch.

To configure a route, the *CHIP_SSS_CTRL* register is used. Each output from the source select switch has its own register field that is used to select what input is routed to that output.

For example, to route the I^2S digital input through the DAP and then out to the DAC (headphone) outputs write *SSS_CTRL->DAP_SELECT* to 0x1 (selects I2S_IN) and *SSS_CTRL->DAC_SELECT* to 0x3 (selects DAP output).

ANALOG INPUT BLOCK

The analog input block contains a stereo line input and a microphone input with mic bias. Either input can be routed to the ADC. The line input can also be configured to bypass the CODEC and be routed directly to the headphone output.

Line Inputs

One stereo line input is provided for connection to line sources such as an FM radio or MP3 input.

The source should be connected to the left and right line inputs through series coupling capacitors. The suggested value is shown in the typical application diagram in Typical Applications.

As detailed in ADC, the line input can be routed to the ADC.

The line input can also be routed to the headphone output by writing *CHIP_ANA_CTRL->SELECT_HP*. This selection bypasses the ADC and audio switch and routes the line input directly to the headphone output to enable a very low power pass through.

Microphone Input

One mono microphone input is provided for uses such as voice recording.

Mic bias is provided. The mic bias is programmed with the *CHIP_MIC_CTRL->BIAS_VOLT* register field. Values from 1.25 V to 3.00 V are supported in 0.25 V steps. Mic bias should be set less than 200 mV from VDDA, e.g. with VDDA at 1.70 V, Mic bias should be set no greater than 1.50 V.

The microphone should be connected through a series coupling capacitor. The suggested value is shown in the typical connection diagram.

The microphone has programmable gain through the *CHIP_MIC_CTRL->GAIN* register field. Values of 0 dB, +20 dB, +30 dB and +40 dB are available.

ADC

The SGTL5000 contains an ADC, which takes its input from either the line input or a microphone. The register field *CHIP_ANA_CTRL->SELECT_ADC* controls this selection. The output of the ADC feeds the audio switch.

The ADC has its own analog gain stage that provides 0 to +22.5 dB of gain in 1.5 dB steps. A bit is available that shifts this range down by 6.0 dB to effectively provide -6.0 dB to +16.5 dB of gain. The ADC gain is controlled in the *CHIP_ANA_ADC_CTRL* register.

The ADC has an available zero cross detect (ZCD) that prevents any volume change until a zero-volt crossing of the audio signal is detected. This helps in eliminating pop or other audio anomalies. If the ADC is to be used, the chip reference bias current should not be set to -50% when in 3.0 V mode.

ANALOG OUTPUTS

The SGTL5000 contains a single stereo DAC that can be used to drive a headphone output and a line output. The DAC receives its input from the audio switch. The headphone output and the line output can be driven at the same time from the DAC.

The headphone output can also be driven directly by the line input bypassing the ADC and DAC for a very low power mode of operation.

The headphone output is powered by VDDA while the line output is powered by VDDIO. This allows the headphone output to be run at the lowest possible voltage while the line output can still meet line output level requirements.

DAC

The DAC output is routed to the headphone and the dedicated line output.

The DAC output has a digital volume control from -90 dB to 0 dB in ~0.5 dB step sizes. This volume is shared among headphone output and line output. The register *CHIP_DAC_VOL* controls the DAC volume.

Headphone

Stereo headphone outputs are provided which can be used to drive a headphone load or a line level output. The headphone output has its own independent analog volume control with a volume range of -52 dB to +12 dB in 0.5 dB step sizes. This volume control can be used in addition to the DAC volume control. For best performance the DAC volume control should be left at 0 dB until the headphone is brought to its lowest setting of -52 dB. The register *CHIP_ANA_HP_CTRL* is used to control the headphone volume.

The headphone output has an independent mute that is controlled by the register field *CHIP_ANA_CTRL->MUTE_HP*.

The line input is routed to the headphone output by writing *CHIP_ANA_CTRL->SELECT_HP*. This selection bypasses the ADC and audio switch and routes the line input directly to the headphone output to enable a very low power pass through. When the line input is routed to the headphone output, only the headphone analog volume and mute affects the headphone output.

The headphone has an available zero cross detect (ZCD) which, as previously described, prevents any volume change until a zero-volt crossing of the audio signal is detected. This helps in eliminating pop or other audio anomalies.

Line Outputs

The SGTL5000 contains a stereo line output. The line output has a dedicated gain stage that can be used to adjust the output level. The *CHIP_LINE_OUT_VOL* controls the line level output gain.

The line outputs also have a dedicated mute that is controlled by the register field *CHIP_ANA_CTRL->MUTE_LO.*

The line out volume is intended as maximum output level adjustment. It is intended to be used to set the maximum output swing. It does not have the range of a typical volume control and does not have a zero cross detect (ZCD). However the DAC digital volume could be used if volume control is desired.

FUNCTIONAL DEVICE OPERATION

POWER CONSUMPTION

Table 9. Power Consumption: V_{DDA}=1.8 V, V_{DDIO}=1.8 V

MODE	CURRENT CONSUMPTION (MA)			POWER (MW)
	V_{DDD}	V_{DDA}	V_{DDIO}	
Playback (I^2S->DAC->Headphone)	-	2.54	0.9	6.19
Playback with DAP ((I^2S->DAP->DAC->Headphone)	-	3.59	0.9	8.08
Playback/Record (I^2S->DAC->Headphone, ADC->I^2S)	-	3.71	1.10	8.67
Record (ADC->I^2S)	-	2.29	1.06	6.02
Analog playback, CODEC bypassed (LINEIN->HP)	-	1.48	0.89	4.27
Standby, all analog power off	-	0.019	0.002	0.038
Playback with PLL (I^2S->DAC->HP)	-	3.01	2.17	9.31

V_{DDD} derived internally at 1.2 V, slave mode except for PLL case, 32 Ω load on HP, Conditions: -100 dBFs signal input, slave mode unless otherwise noted, paths tested as indicated, unused paths turned off.

Table 10. Power Consumption: V_{DDA}=3.3 V, V_{DDIO}=3.3 V

MODE	CURRENT CONSUMPTION (MA)			POWER(MW)
	V_{DDD}	V_{DDA}	V_{DDIO}	
Playback (I^2S->DAC->Headphone)	-	3.45	0.067	11.60
Playback with DAP ((I^2S->DAP->DAC->Headphone)	-	4.49	0.067	15.03
Playback/Record (I^2S->DAC->Headphone, ADC->I^2S)	-	4.67	0.343	16.53
Record (ADC->I^2S)	-	2.90	0.296	10.56
Analog playback, CODEC bypassed (LINEIN->HP)	-	1.91	0.039	6.43
Standby, all analog power off	-	0.04	0.002	0.139
Playback with PLL (I^2S->DAC->HP)	-	3.92	2.76	22.05

DIGITAL INPUT & OUTPUT

One I^2S (Digital Audio) Port is provided which supports the following formats: I^2S, Left Justified, Right Justified, and PCM mode.

I^2S, Left Justified, and Right Justified Modes

I^2S, Left Justified and Right Justified modes are stereo interface formats. The I2S_SCLK frequency, I2S_SCLK polarity, I2S_DIN/DOUT data length, and I2S_LRCLK polarity can all be changed through the *CHIP_I2S_CTRL* register. For I2S, Left Justified and Right Justified formats,

the left subframe should always be presented first regardless of the *CHIP_I2S_CTRL->LRPOL* setting.

The I2S_LRCLK and I2S_SCLK can be programmed as master (driven to an external target) or slave (driven from an external source). When the clocks are in slave mode, they must be synchronous to SYS_MCLK. For this reason the SGTL5000 can only operate in synchronous mode (see Clocking) while in I^2S slave mode.

In master mode, the clocks are synchronous to SYS_MCLK or the output of the PLL when the part is running in asynchronous mode.

Figure 10 shows functional examples of different common digital interface formats and their associated register settings.

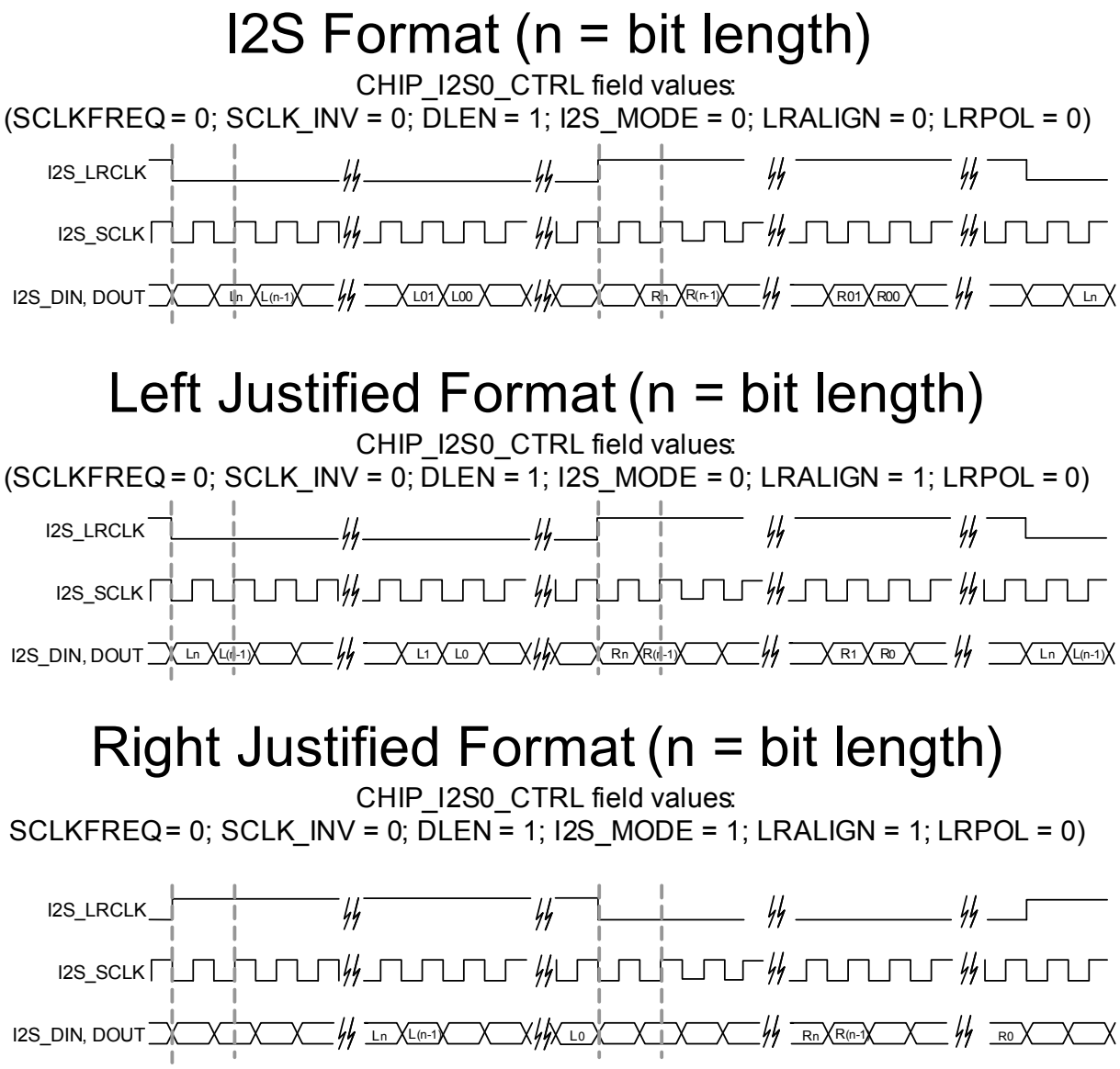


Figure 10. I^2S Port Supported Formats

PCM Mode

The I^2S port can also be configured in PCM mode (also known as DSP mode). This mode is provided to allow connectivity to external devices such as Bluetooth modules. PCM mode differs from other interface formats presented in I2S, Left Justified, and Right Justified Modes, in that the frame clock (I2S_LRCLK) does not represent a different channel when high or low. Instead, it is a bit-wide pulse that marks the start of a frame. Data is aligned such that the left channel data is immediately followed by right channel data. Zero padding is filled in for the remaining bits. The data and frame clock may be configured to clock in on the rising or falling edge of Bit Clock.

PCM Format A is a format in which the data word begins one SCLK bit following the I2S_LRCLK transition, as in I^2S Mode. PCM Format B is a format in which the data word begins after the I2S_LRCLK transition, as in Left Justified.

In slave mode, the pulse width of the I2S_LRCLK does not matter. The pulse can range from one cycle high to all but one cycle high. In master mode, it is driven one cycle high.

Figures 11 shows a functional drawing of the different formats in master mode.

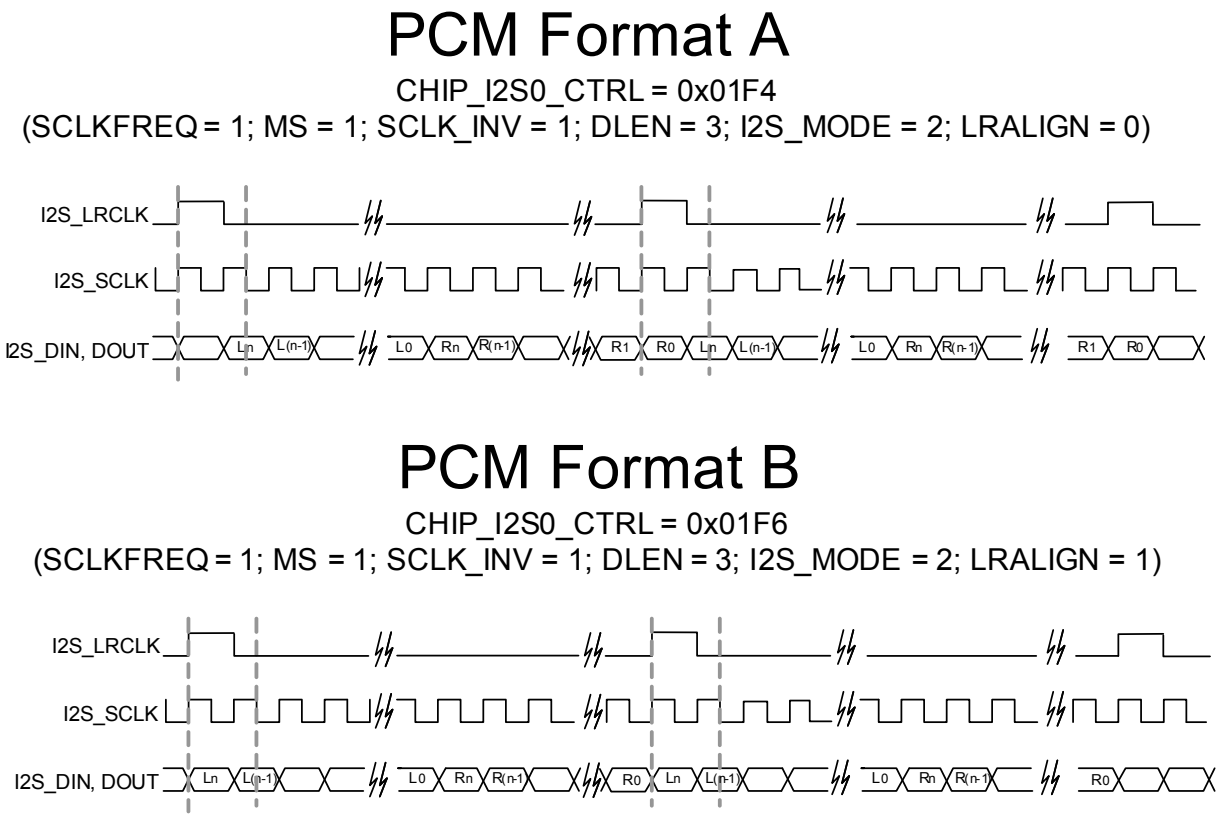


Figure 11. PCM Formats

DIGITAL AUDIO PROCESSING

The SGTL5000 contains a digital audio processing block (DAP) connected to the source select switch. The digitized signal from the source select switch can be routed into the DAP block for audio processing. The DAP has the following 5 sub blocks:

- Dual Input Mixer
- Freescale Surround
- Freescale Bass Enhancement
- 7-Band Parameter EQ / 5-Band Graphic EQ / Tone Control (only one can be used at a time)
- Automatic Volume Control (AVC)

The block diagram in Figure 12 shows the sequence in which the signal passes through these blocks.

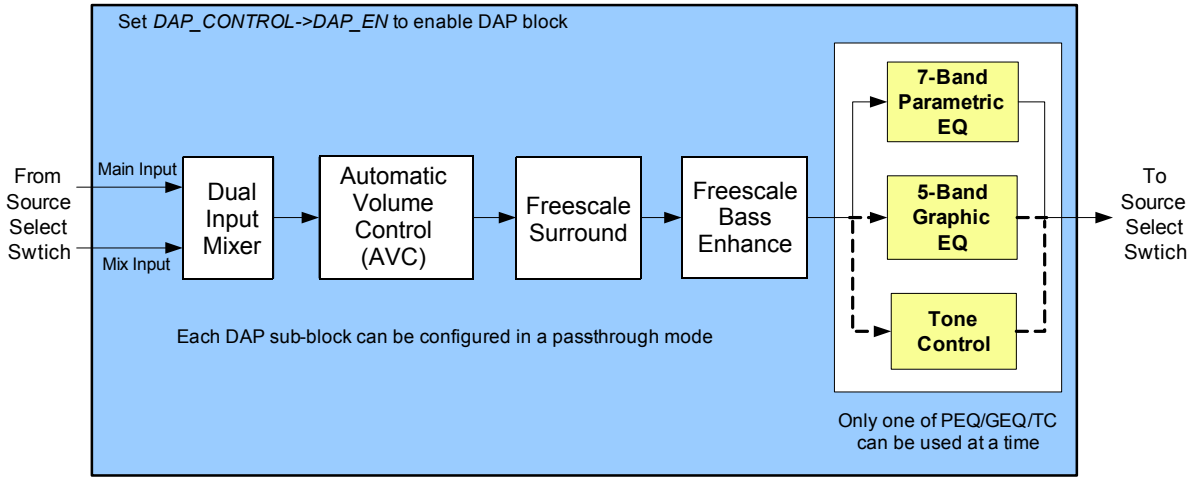


Figure 12. Digital Audio Processing Block Diagram

When the DAP block is added in the route, it must be enabled separately to get audio through. It is recommended to mute the outputs before enabling/disabling the DAP block to avoid any pops or clicks due to discontinuities in the output.

Refer to Digital Audio Processor Configuration for programming examples on how to enable/disable the DAP block.

Each sub-block of the DAP can be individually disabled if its processing is not required. The following sections describe the DAP sub-blocks and how to configure them.

Dual Input Mixer

The dual input digital mixer allows for two incoming streams from the source select switch as shown in DAP - Dual Input Mixer.

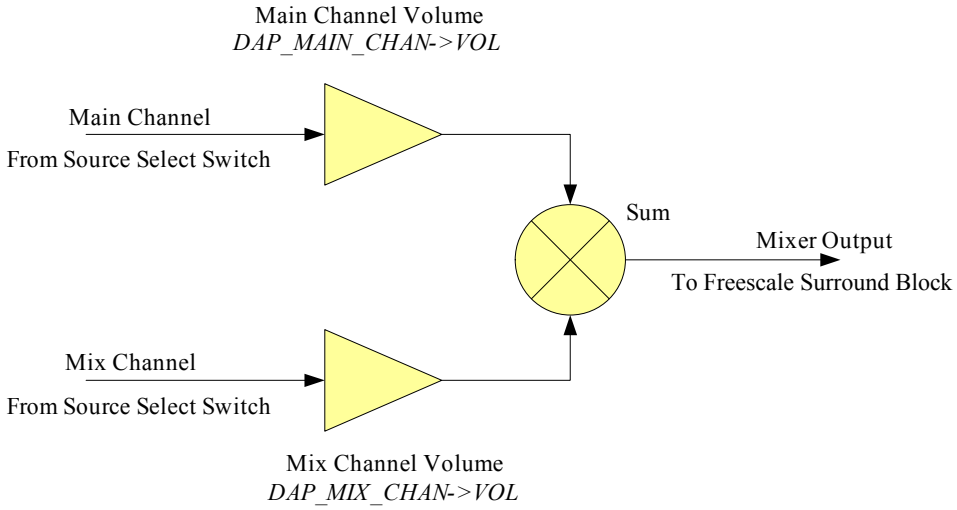


Figure 13. DAP - Dual Input Mixer

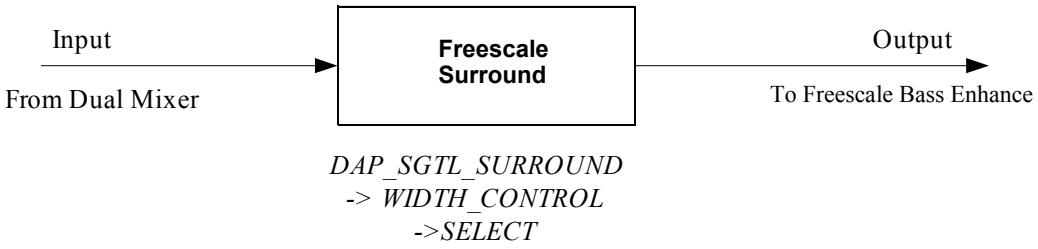
The Dual Input Mixer can be enabled or configured in a pass-through mode (Main channel is passed through without any mixing). When enabled, the volume of the main and mix channels can be independently controlled before they are mixed together.

The volume range allowed on each channel is 0% to 200% of the incoming signal level. The default is 100% (same as input signal level) volume on the main input and 0% (muted) on the mix input.

Refer to Dual Input Mixer for programming examples on how to enable/disable the mixer and also to set the main and mix channel volume.

Freescale Surround

Freescale Surround is a royalty free virtual surround algorithm for stereo or mono inputs. It widens and deepens the sound stage of the music input.



The Freescale Surround can be enabled or configured in pass-through mode (input is passed through without any processing). When enabling the Surround, mono or stereo input type must be selected based on the input signal. Surround width may be adjusted for the size of the sound stage.

Refer to Freescale Surround and Freescale Surround On/Off for a programming example on how to configure Surround width and how to enable/disable Surround.

Freescale Bass Enhance

Freescale Bass Enhance is a royalty-free algorithm that enhances natural bass response of the audio. Bass Enhance extracts bass content from right and left channels, adds bass and mixes this back up with the original signal. An optional complementary high pass filter is provided after the mixer.

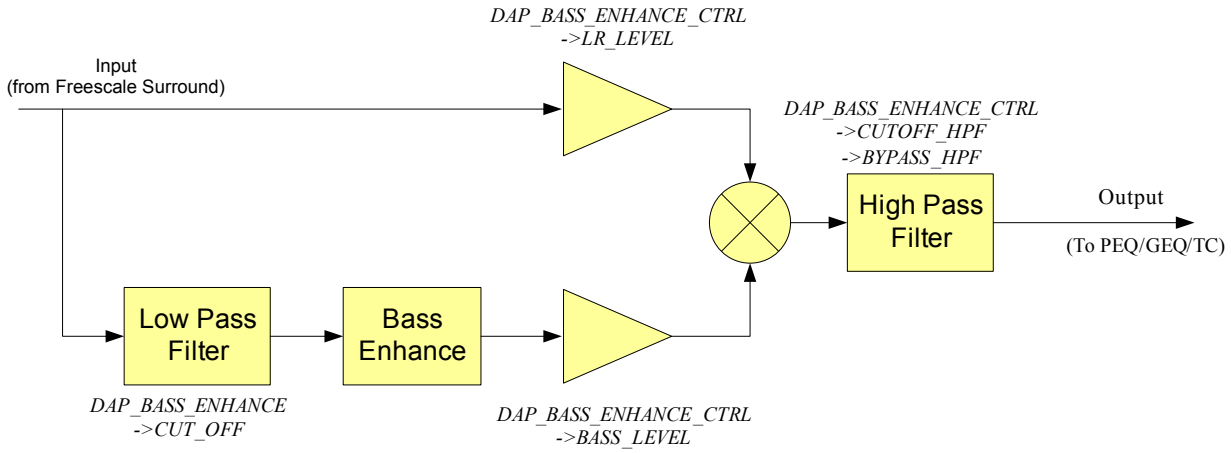


Figure 14. DAP- Freescale Bass Enhance

The Freescale Bass Enhance can be enabled or configured in pass-through mode (input is passed through without any processing).

The cutoff frequency of the low-pass filter (LPF) can be selected based on the speakers frequency response. The cutoff frequency of the low-pass and high-pass filters are selectable between 80 to 225 Hz. Also, the input signal and bass enhanced signal can be individually adjusted for level before the two signals are mixed.

Refer to Freescale Bass Enhance and Bass Enhance On/Off for a programming example on how to configure Bass Enhance and how to enable/disable this feature.

7-Band Parametric EQ / 5-Band Graphic EQ / Tone Control

One 7-band parametric equalizer (PEQ), one 5-band graphic equalizer (GEQ), and Tone Control (Bass and Treble control) blocks are implemented as mutually exclusive blocks. Only one block can be used at a given time.

Refer to 7-Band Parametric EQ / 5-Band Graphic EQ / Tone Control for a programming example that shows how to select the desired EQ mode.

7-Band Parametric EQ

The 7-band PEQ allows the designer to compensate for speaker response and to provide the ability to filter out resonant frequencies caused by the physical system design. The system designer can create custom EQ presets such as Rock, Speech, Classical, etc, which allows users the flexibility to customize their audio.

The 7-band PEQ is implemented using 7 cascaded second order IIR filters. All filters are implemented using programmable bi-quad filters. Figure 15 shows the transfer function and Direct Form 1 of the five coefficient biquadratic filter.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

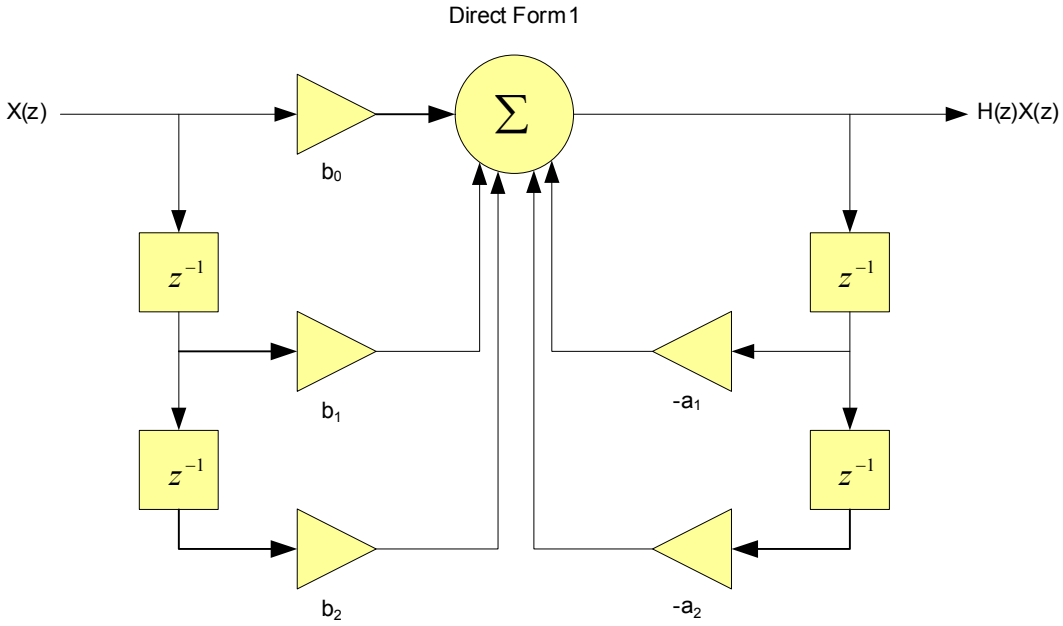


Figure 15. 5-Coefficient Biquad Filter and Transfer Function

If a band is enabled but is not being used (flat response), then a value of 0.5 should be put in b_0 and all other coefficients should be set to 0.0. Note that the coefficients must be converted to hex values before writing to the registers. By default, all the filters are loaded with coefficients to give a flat response.

In order to create EQ presets such as Rock, Speech, Classical, etc, the coefficients must be calculated, converted to 20-bit hex values and written to the registers. Note that coefficients are sample-rate dependent and separate coefficients must be generated for different sample rates. Please contact Freescale for assistance with generating the coefficients.

Refer to 7-Band PEQ Preset Selection for a programming example that shows how load the filter coefficients when the end-user changes the preset.

PEQ can be disabled (pass-through mode) by writing 0 to DAP_AUDIO_EQ->EN bits.

5-Band Graphic EQ

The 5-band graphic equalizer is implemented using 5 parallel second order IIR filters. All filters are implemented using biquad filters whose coefficients are programmed to set the bands at a specific frequency. The GEQ bands are fixed at 115 Hz, 330 Hz, 990 Hz, 3000 Hz, and 9900 Hz. The volume on each band is independently adjustable in the range of +12 dB to -11.75 dB in 0.25 dB steps.

Refer to 5-Band GEQ Volume Change for a programming example that shows how to change the GEQ volume.

Tone Control

Tone control comprises treble and bass controls. The tone control is implemented as one 2nd order low pass filter (bass) and one 2nd order high pass filter (treble).

Refer to Tone Control - Bass and Treble Change for a programming example that shows how to change Bass and Treble values.

Automatic Volume Control (AVC)

An Automatic Volume Control (AVC) block is provided to reduce loud signals and amplify low level signals for easier listening. The AVC is designed to compress audio when the measured level is above the programmed threshold or to expand the audio to the programmed threshold when the measured audio is below the threshold. The threshold level is programmable with an allowed range of 0 to -96 dB. Figure 16 shows the AVC block diagram and controls.

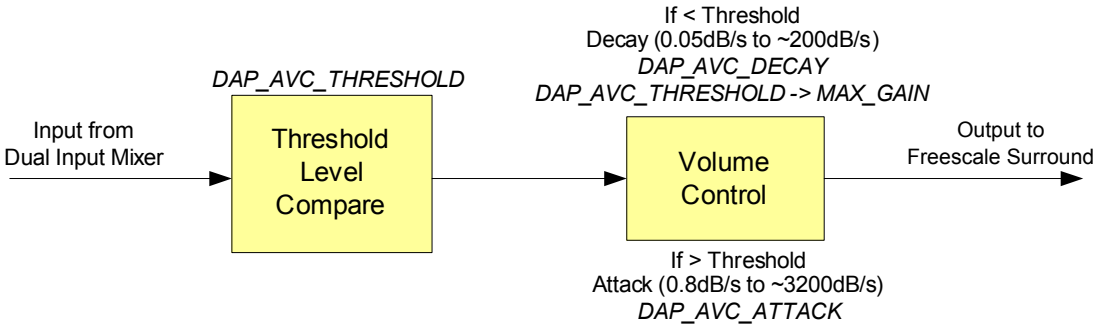


Figure 16. DAP AVC Block Diagram

When the measured audio level is below threshold, the AVC can apply a maximum gain of up to 12 dB. The maximum gain can be selected, either 0, 6, or 12 dB. When the maximum gain is set to 0 dB the AVC acts as a limiter. In this case the AVC only takes effect when the signal level is above the threshold.

The rate at which the incoming signal is attenuated down to the threshold is called the attack rate. Too high of an attack causes an unnatural sound as the input signal may be distorted. Too low of an attack may cause saturation of the output as the incoming signal is not compressed quickly enough. The attack rate is programmable with allowed range of 0.05 dB/s to 200 dB/s.

When the signal is below the threshold, AVC adjusts the volume up until either the threshold or the maximum gain is reached. The rate at which this volume is changed is called the decay rate. The decay rate is programmable with allowed range of 0.8 dB/s to 3200 dB/s. It is desirable to use very slow decay rate to avoid any distortion in the signal and prevent the AVC from entering a continuous attack-decay loop.

Refer to Automatic Volume Control (AVC) and Automatic Volume Control (AVC) On/Off for a programming example that shows how to configure AVC and how to enable/disable AVC respectively.

CONTROL

The SGTL5000 supports both I^{2}C and SPI control modes (note that SPI is not supported in the 20 QFN part). The CTRL_MODE pin chooses which mode is used. When CTRL_MODE is tied to ground, the control mode is I^{2}C. When CTRL_MODE is tied to VDDIO, the control mode is SPI.

Regardless of the mode, the control interface is used for all communication with the SGTL5000 including startup configuration, routing, volume, etc.

I^{2}C

The I^{2}C port is implemented according to the I^{2}C specification v2.0. The I^{2}C interface is used to read and write all registers.

For the 32 QFN version of the SGTL5000, the I^{2}C device address is 0n01010(R/W) where n is determined by CTRL_ADR0_CS and R/W is the read/write bit from the I^{2}C protocol.

For the 20 QFN version of the SGTL5000 the I^{2}C address is always 0001010(R/W).

The SGTL5000 is always the slave on all transactions, which means that an external master always drives CTRL_CLK.

In general, an I^{2}C transaction looks like the following.

All locations are accessed with a 16 bit address. Each location is 16 bits wide.

Example I^{2}C write

- Start condition
- Device address with the R/W bit cleared to indicate write
- Send two bytes for the 16 bit register address (most significant byte first)
- Send two bytes for the 16 bits of data to be written to the register (most significant byte first)
- Stop condition

Example I^{2}C read

- Start condition
- Device address with the R/W bit cleared to indicate write
- Send two bytes for the 16 bit register address (most significant byte first)
- Stop Condition followed by start condition (or a single restart condition)
- Device address with the R/W bit set to indicate read
- Read two bytes from the addressed register (most significant byte first)
- Stop condition

Figure 17 shows the functional I^{2}C timing diagram.

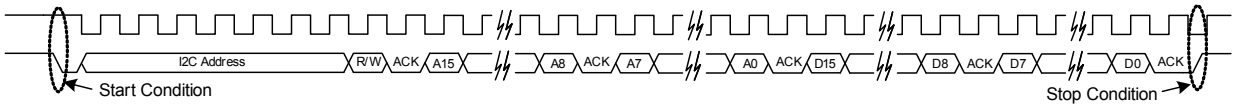


Figure 17. Functional I²C Diagram

The protocol has an auto increment feature. Instead of sending the stop condition after two bytes of data, the master may continue to send data byte pairs for writing, or it may send extra clocks for reading data byte pairs. In either case, the access address is incremented after every two bytes of data. A start or stop condition from the I²C master interrupts the current command. For reads, unless a new address is written, a new start condition with R/W=0 reads from the current address and continues to auto increment.

The following diagrams describe the different access formats. The gray fields are from the I²C master, and the white fields are the SGTL5000 responses. Data [n] corresponds to the data read from the address sent, data[n+1] is the data from the next register, and so on.

S = Start Condition

Sr = Restart Condition

A = Ack

N = Nack

P = Stop Condition

Table 11. Write Single Location

S	Device Address	W (0)	A	ADDR byte 1	A	ADDR byte 0	A	DATA byte 1	A	DATA byte 0	A	P

Table 12. Write Auto increment

S	Device Address	W (0)	A	start ADDR byte 1	A	start ADDR byte 0	A	DATA [n] byte 1	A	DATA [n] byte 0	A	DATA [n+1] byte 1	A	DATA [n+1] byte 0	A	P

Table 13. Read Single Location

S	Device Address	W (0)	A	ADDR byte 1	A	ADDR byte 0	A	Sr	Device Address	R (1)	A	DATA byte 1	A	DATA byte 0	N	P

Table 14. Read Auto increment

S	Device Address	W (0)	A	start ADDR byte 1	A	start ADDR byte 0	A	Sr	Device Address	R (1)	A	DATA [n] byte 1	A	DATA [n] byte 0	A	DATA [n+1] byte 1	A	DATA [n+1] byte 0	N	P

Table 15. Read Continuing Auto increment

S	Device Address	R	A	DATA [n+2] byte 1	A	DATA [n+2] byte 0	A	DATA [n+3] byte 1	A	DATA [n+3] byte 0	N	P

SPI

Serial Peripheral Interface (SPI) is a communications protocol supported by the SGTL5000 (not supported in the 20 QFN package). The SGTL5000 is always a slave. The CTRL_ADR0_CS is used as the slave select (SS) when the master wants to select the SGTL5000 for communication. CTRL_CLK is connected to master's SCLK and CTRL_DATA is connected to master's MOSI line. The part only supports SPI write operations and does not support read operations.

Figure 18 shows the functional timing diagram of the SPI communication protocol as supported by the SGTL5000 chip. Note that on the rising edge of the SS, the chip latches to the previous 32 bits of data. It interprets the latest 16-bits as register value and the 16-bits preceding it as register address.

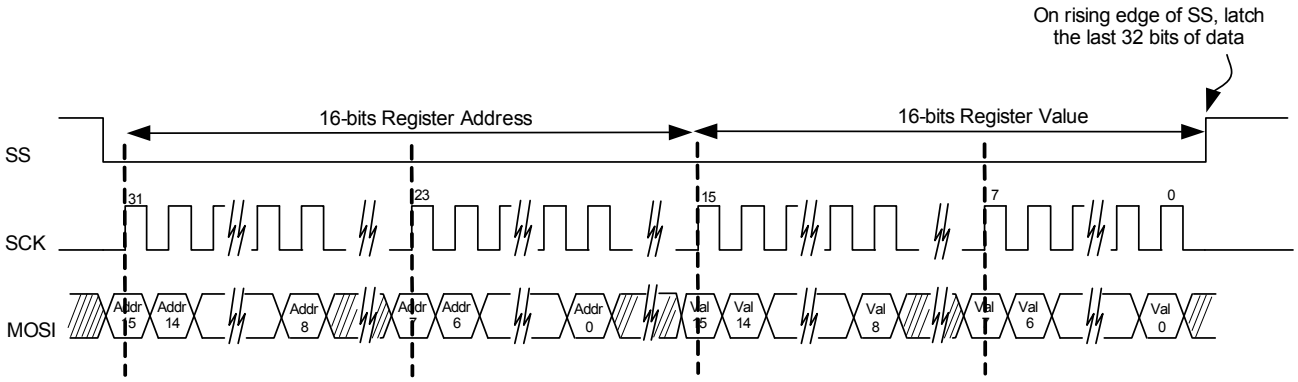


Figure 18. Functional Timing Diagram of SPI Protocol

PROGRAMMING EXAMPLES

This section provides programming examples showing how to configure the chip. The registers can be written/read by using I^2C communication protocol. The chip also supports SPI communication protocol (not supported in the 20 QFN package), but only register write operation is supported.

PROTOTYPE FOR READING AND WRITING A REGISTER

The generic register read write prototype is used throughout this section, as shown by the following. The I^2C or SPI implementation is specific to the I^2C/SPI hardware used in the system.

```
// This prototype writes a value to the entire register. All
// bit-fields of the register will be written.
Write REGISTER REGISTERVALUE
// This prototype writes a value only to the bit-field specified.
// In the actual implementation, the other bit-fields should be
// masked to prevent them from being written. Also, the
// actual implementation should left-shift the BITFIELDVALUE
// by appropriate number to match the starting bit location of
// the BITFIELD.
Modify REGISTER -> BITFIELD,   BITFIELDVALUE //Bitfield
Location
// Example implementation
// Modify DAP_EN (bit 0) bit to value 1 to enable DAP block
Modify(DAP_CONTROL_REG, 0xFFFE, 1 <<
DAP_EN_STARTBIT);
// Example Implementation of Modify
void Modify(unsigned short usRegister,
            unsigned short usClearMask,
            unsigned short usSetValue)
{
 unsigned short usData;
 // 1) Read current value
 ReadRegister(usRegister, &usData);
 // 2) Clear out old bits
 usData = usData & usClearMask;
 // 3) set new bit values
 usData = usData | usSetValue;
 // 4) Write out new value created
 WriteRegister(usRegister, usData);
}
```

CHIP CONFIGURATION

All outputs (LINEOUT, HP_OUT, I2S_OUT) are muted by default on power up. To avoid any pops/clicks, the outputs should remain muted during these chip configuration steps. Refer to Volume Control for volume and mute control.

Initialization

Chip Powerup and Supply Configurations

After the power supplies for the chip are turned on, the following initialization sequence should be followed. Please note that certain steps may be optional or different values may need to be written based on the power supply voltage used and desired configuration. The initialization sequence below assumes VDDIO = 3.3 V and VDDA = 1.8 V.

```
//--------------- Power Supply Configuration----------------
// NOTE: This next 2 Write calls is needed ONLY if VDDD is
// internally driven by the chip
// Configure VDDD level to 1.2V (bits 3:0)
Write CHIP_LINREG_CTRL    0x0008
// Power up internal linear regulator (Set bit 9)
Write CHIP_ANA_POWER      0x7260
// NOTE: This next Write call is needed ONLY if VDDD is
// externally driven
// Turn off startup power supplies to save power (Clear bit 12 and
13)
Write CHIP_ANA_POWER      0x4260
// NOTE: The next Write calls is needed only if both VDDA and
// VDDIO power supplies are less than 3.1V.
// Enable the internal oscillator for the charge pump (Set bit 11)
Write CHIP_CLK_TOP_CTRL   0x0800
// Enable charge pump (Set bit 11)
Write CHIP_ANA_POWER      0x4A60
// NOTE: The next modify call is only needed if both VDDA and
// VDDIO are greater than 3.1 V
// Configure the charge pump to use the VDDIO rail (set bit 5 and
bit 6)
Write CHIP_LINREG_CTRL    0x006C
//---- Reference Voltage and Bias Current Configuration----
// NOTE: The value written in the next 2 Write calls is dependent
// on the VDDA voltage value.
// Set ground, ADC, DAC reference voltage (bits 8:4). The value
should
// be set to VDDA/2.   This example assumes VDDA = 1.8 V.
VDDA/2 = 0.9 V.
// The bias current should be set to 50% of the nominal value (bits
3:1)
Write CHIP_REF_CTRL 0x004E
// Set LINEOUT reference voltage to VDDIO/2 (1.65 V) (bits 5:0)
and bias current (bits 11:8) to the recommended value of 0.36 mA
for 10 kOhm load with 1.0 nF capacitance
Write CHIP_LINE_OUT_CTRL 0x0322
//------------Other Analog Block Configurations-------------
// Configure slow ramp up rate to minimize pop (bit 0)
Write CHIP_REF_CTRL       0x004F
// Enable short detect mode for headphone left/right
// and center channel and set short detect current trip level
// to 75 mA
Write CHIP_SHORT_CTRL     0x1106
// Enable Zero-cross detect if needed for HP_OUT (bit 5) and ADC
(bit 1)
```

```
Write CHIP_ANA_CTRL        0x0133
//------------Power up Inputs/Outputs/Digital Blocks---------
// Power up LINEOUT, HP, ADC, DAC
Write CHIP_ANA_POWER 0x6AFF
// Power up desired digital blocks
// I2S_IN (bit 0), I2S_OUT (bit 1), DAP (bit 4), DAC (bit 5),
// ADC (bit 6) are powered on
Write CHIP_DIG_POWER       0x0073
//----------------Set LINEOUT Volume Level------------------
// Set the LINEOUT volume level based on voltage reference
(VAG)
// values using this formula
// Value = (int)(40*log(VAG_VAL/LO_VAGCNTRL) + 15)
// Assuming VAG_VAL and LO_VAGCNTRL is set to 0.9 V and
1.65 V respectively, the // left LO vol (bits 12:8) and right LO
volume (bits 4:0) value should be set // to 5
Write CHIP_LINE_OUT_VOL 0x0505
```

System MCLK and Sample Clock

```
// Configure SYS_FS clock to 48 kHz
// Configure MCLK_FREQ to 256*Fs
Modify CHIP_CLK_CTRL->SYS_FS 0x0002 // bits 3:2
Modify CHIP_CLK_CTRL->MCLK_FREQ 0x0000 // bits 1:0
// Configure the I2S clocks in master mode
// NOTE: I2S LRCLK is same as the system sample clock
Modify CHIP_I2S_CTRL->MS 0x0001 // bit 7
```

PLL Configuration

These programming steps are needed only when the PLL is used. Refer to Using the PLL - Asynchronous SYS_MCLK input for details on when to use the PLL.

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to Volume Control for volume and mute control.

```
// Power up the PLL
Modify CHIP_ANA_POWER->PLL_POWERUP 0x0001 // bit 10
Modify CHIP_ANA_POWER->VCOAMP_POWERUP 0x0001 //
bit 8
// NOTE: This step is required only when the external SYS_MCLK
// is above 17 MHz. In this case the external SYS_MCLK clock
// must be divided by 2
Modify CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 0x0001 //
bit 3
Sys_MCLK_Input_Freq = Sys_MCLK_Input_Freq/2;
// PLL output frequency is different based on the sample clock
// rate used.
if (Sys_Fs_Rate == 44.1 kHz)
  PLL_Output_Freq = 180.6336 MHz
else
  PLL_Output_Freq = 196.608 MHz
// Set the PLL dividers
Int_Divisor = floor(PLL_Output_Freq/Sys_MCLK_Input_Freq)
Frac_Divisor = ((PLL_Output_Freq/Sys_MCLK_Input_Freq) -
Int_Divisor)*2048
Modify CHIP_PLL_CTRL->INT_DIVISOR Int_Divisor // bits 15:11
Modify CHIP_PLL_CTRL->FRAC_DIVISOR Frac_Divisor // bits
10:0
```

Input/Output Routing

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to Volume Control for volume and mute control.

A few example routes are shown below:

```
// Example 1: I2S_IN -> DAP -> DAC -> LINEOUT, HP_OUT
// Route I2S_IN to DAP
Modify CHIP_SSS_CTRL->DAP_SELECT 0x0001 // bits 7:6
// Route DAP to DAC
Modify CHIP_SSS_CTRL->DAC_SELECT 0x0003 // bits 5:4
// Select DAC as the input to HP_OUT
Modify CHIP_ANA_CTRL->SELECT_HP 0x0000 // bit 6
// Example 2: MIC_IN -> ADC -> I2S_OUT
// Set ADC input to MIC_IN
Modify CHIP_ANA_CTRL->SELECT_ADC 0x0000 // bit 2
// Route ADC to I2S_OUT
Modify CHIP_SSS_CTRL->I2S_SELECT 0x0000 // bits 1:0
// Example 3: LINEIN -> HP_OUT
// Select LINEIN as the input to HP_OUT
Modify CHIP_ANA_CTRL->SELECT_HP 0x0001 // bit 6
```

DIGITAL AUDIO PROCESSOR CONFIGURATION

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to Volume Control for volume and mute control.

```
// Enable DAP block
// NOTE: DAP will be in a pass-through mode if none of DAP
// sub-blocks are enabled.
Modify DAP_CONTROL->DAP_EN 0x0001 // bit 0
```

Dual Input Mixer

These programming steps are needed only if dual input mixer feature is used.

```
// Enable Dual Input Mixer
Modify DAP_CONTROL->MIX_EN 0x0001 // bit 4
// NOTE: This example assumes mix level of main and mix
// channels as 100% and 50% respectively
// Configure main channel volume to 100% (No change from input
// level)
Write DAP_MAIN_CHAN 0x4000
// Configure mix channel volume to 50% (attenuate the mix
// input level by half)
Write DAP_MIX_CHAN 0x4000
```

Freescale Surround

The Freescale Surround on/off function is typically controlled by the end-user. End-user driven programming steps are shown in End-user Driven Chip Configuration.

The default WIDTH_CONTROL of 4 should be appropriate for most applications. This optional programming step shows how to configure a different width value.

```
// Configure the surround width
// (0x0 = Least width, 0x7 = Most width). This example shows
// a width setting of 5
Modify DAP_SGTL_SURROUND->WIDTH_CONTROL 0x0005
// bits 6:4
```

Freescale Bass Enhance

The Freescale Bass Enhance on/off function is typically controlled by the end-user. End-user driven programming steps are shown in **End-user Driven Chip Configuration**.

The default LR_LEVEL value of 0x0005 results in no change in the input signal level and BASS_LEVEL value of 0x001F adds some harmonic boost to the main signal. The default settings should work for most applications. This optional programming step shows how to configure a different value.

```
// Gain up the input signal level
Modify DAP_BASS_ENHANCE_CTRL->LR_LEVEL 0x0002
// bits 7:4
// Add harmonic boost
Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL 0x003F);
// bits 6:0
```

7-Band Parametric EQ / 5-Band Graphic EQ / Tone Control

Only one audio EQ block can be used at a given time. The pseudocode in this section shows how to select each block.

Some parameters of the audio EQ are typically controlled by the end-user. End-user driven programming steps are shown in **End-user Driven Chip Configuration**.

```
// 7-Band PEQ Mode
// Select 7-Band PEQ mode and enable 7 PEQ filters
Write DAP_AUDIO_EQ 0x0001
Write DAP_PEQ 0x0007
// Tone Control mode
Write DAP_AUDIO_EQ 0x0002
// 5-Band GEQ Mode
Write DAP_AUDIO_EQ 0x0003
```

Automatic Volume Control (AVC)

The AVC on/off function is typically controlled by the end-user. End-user driven programming steps are shown in **End-user Driven Chip Configuration**.

The default configuration of the AVC should work for most applications. However, the following example shows how to change the configuration if needed.

```
// Configure threshold to -18dB
Write DAP_AVC_THRESHOLD 0x0A40
// Configure attack rate to 16dB/s
Write DAP_AVC_ATTACK 0x0014
// Configure decay rate to 2dB/s
Write DAP_AVC_DECAY 0x0028
```

I²S CONFIGURATION

By default the I²S port on the chip is configured for 24-bits of data in I²S format with SCLK set for 64*Fs. This can be modified by setting various bit-fields in the CHIP_I2S_CTRL register.

VOLUME CONTROL

The outputs should be unmuted after all the configuration is complete.

```
//---------- Input Volume Control----------
// Configure ADC left and right analog volume to desired default.
// Example shows volume of 0dB
Write CHIP_ANA_ADC_CTRL 0x0000
// Configure MIC gain if needed. Example shows gain of 20dB
Modify CHIP_MIC_CTRL->GAIN 0x0001
// bits 1:0
//---------- Volume and Mute Control----------
// Configure HP_OUT left and right volume to minimum, unmute
// HP_OUT and ramp the volume up to desired volume.
Write CHIP_ANA_HP_CTRL 0x7F7F
Modify CHIP_ANA_CTRL->MUTE_HP 0x0000
// bit 4
// Code assumes that left and right volumes are set to same value
// So it only uses the left volume for the calculations
usCurrentVolLeft = 0x7F;
usNewVolLeft = usNewVol & 0xFF;
usNumSteps = usNewVolLeft - usCurrentVolLeft;
if (usNumSteps == 0) return;
// Ramp up
for (int i = 0; i < usNumSteps; i++)
{
  ++usCurrentVolLeft;
  usCurrentVol = (usCurrentVolLeft << 8) | (usCurrentVolLeft);
  Write CHIP_ANA_HP_CTRL usCurrentVol;
}
// LINEOUT and DAC volume control
Modify CHIP_ANA_CTRL->MUTE_LO 0x0000
// bit 8
// Configure DAC left and right digital volume. Example shows
// volume of 0dB
Write CHIP_DAC_VOL 0x3C3C
Modify CHIP_ADCDAC_CTRL->DAC_MUTE_LEFT 0x0000
// bit 2
Modify CHIP_ADCDAC_CTRL->DAC_MUTE_RIGHT 0x0000
// bit 3
// Unmute ADC
Modify CHIP_ANA_CTRL->MUTE_ADC 0x0000
// bit 0
```

END-USER DRIVEN CHIP CONFIGURATION

End-users control features like volume up/down, and audio EQ parameters such as Bass and Treble. This requires programming the chip without introducing any pops/clicks or any other disturbance to the output. This section shows examples on how to program these features.

VOLUME AND MUTE CONTROL

Refer to **Volume Control** for examples on how to program volume when end-user changes the volume or mutes/ unmutes the output. Note that the DAC volume ramp is automatically handled by the chip.

7-BAND PEQ PRESET SELECTION

This programming example shows how to load the filter coefficients when the end-user changes PEQ presets such as Rock, Speech, Classical etc.

```
// Load the 5 coefficients for each band and write them to
// appropriate filter address. Repeat this for all enabled
// filters (this example shows 7 filters)
for (i = 0; i < 7; i++)
{
// Note that each 20-bit coefficient is broken into 16-bit MSB
// (unsigned short usXXMSB) and 4-bit LSB (unsigned short
// usXXLSB)
Write DAP_COEF_WR_B0_LSB usB0MSB[i]
Write DAP_COEF_WR_B0_MSB usB0LSB[i]
Write DAP_COEF_WR_B1_LSB usB1MSB[i]
Write DAP_COEF_WR_B1_MSB usB1LSB[i]
Write DAP_COEF_WR_B2_LSB usB2MSB[i]
Write DAP_COEF_WR_B2_MSB usB2LSB[i]
Write DAP_COEF_WR_A1_LSB usA1MSB[i]
Write DAP_COEF_WR_A1_MSB usA1LSB[i]
Write DAP_COEF_WR_A2_LSB usA2MSB[i]
Write DAP_COEF_WR_A2_MSB usA2LSB[i]
// Set the index of the filter (bits 7:0) and load the
// coefficients
Modify DAP_FILTER_COEF_ACCESS->INDEX (0x0101 + i)
// bit 8
}
```

5-BAND GEQ VOLUME CHANGE

This programming example shows how to program the GEQ volume when end-user changes the volume on any of the 5 bands.

GEQ volume should be ramped in 0.5 dB steps in order to avoid any pops. The example assumes that volume is ramped on Band 0. Other bands can be programmed similarly.

```
// Read current volume set on Band 0
usCurrentVol = Read DAP_AUDIO_EQ_BASS_BAND0
// Convert the new volume to hex value
usNewVol = 4*dNewVolDb + 47;
// Calculate the number of steps
usNumSteps = abs(usNewVol - usCurrentVol);
if (usNumSteps == 0) return;
for (int i = 0; i++; usNumSteps)
{
 if (usNewVol > usCurrentVol)
++usCurrentVol;
 else
--usCurrentVol;
Write DAP_AUDIO_EQ_BASS_BAND0 usCurrentVol;
}
```

TONE CONTROL - BASS AND TREBLE CHANGE

This programming example shows how to program the Tone Control Bass and Treble when end-user changes it on the fly.

Tone Control Bass and Treble volume should be ramped in 0.5 dB steps in order to avoid any pops. The example assumes that Treble is changed to a new value. Bass can be programmed similarly.

```
// Read current Treble value
usCurrentVal = Read DAP_AUDIO_EQ_TREBLE_BAND4
// Convert the new Treble value to hex value
usNewVol = 4*dNewValDb + 47;
// Calculate the number of steps
usNumSteps = abs(usNewVal - usCurrentVal);
if (usNumSteps == 0) return;
for (int i = 0; i++; usNumSteps)
{
if (usNewVal > usCurrentVal)
++usCurrentVal;
 else
--usCurrentVal;
 Write DAP_AUDIO_EQ_TREBLE_BAND4 usCurrentVal;
}
```

FREESCALE SURROUND ON/OFF

This programming example shows how to program the Surround when end-user turns it on/off on their device.

The Surround width should be ramped up to highest value before enabling/disabling the Surround to avoid any pops.

```
// Read current Surround width value
// WIDTH_CONTROL bits 6:4
usOriginalVal = (Read DAP_SGTL_SURROUND >> 4) &&
0x0003;
usNextVal = usOriginalVal;
// Ramp up the width to maximum value of 7
for (int i = 0; i++; (7 - usOriginalVal)
{
 ++usNextVal;
 Modify DAP_SGTL_SURROUND->WIDTH_CONTROL
usNextVal;
}
// Enable (To disable, write 0x0000) Surround
```

```
// SELECT bits 1:0
Modify DAP_SGTL_SURROUND->SELECT 0x0003;
// Ramp down the width to original value
for (int i = 0; i++; (7 - usOriginalVal)
{
 --usNextVal;
 Modify DAP_SGTL_SURROUND->WIDTH_CONTROL
usNextVal;
}
```

BASS ENHANCE ON/OFF

This programming example shows how to program the Bass Enhance on/off when end-user turns it on/off on their device.

The Bass level should be ramped down to the lowest Bass before Bass Enhance feature is turned on/off.

```
// Read current Bass level value
// BASS_LEVEL bits 6:0
usOriginalVal = Read DAP_BASS_ENHANCE_CTRL &&
0x007F;
usNextVal = usOriginalVal;
// Ramp Bass level to lowest bass (lowest bass = 0x007F)
usNumSteps = abs(0x007F - usOriginalVal);
for (int i = 0; i++; usNumSteps)
{
 ++usNextVal;
 Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL
usNextVal;
}
// Enable (To disable, write 0x0000) Bass Enhance
// EN bit 0
Modify DAP_BASS_ENHANCE->EN 0x0001;
// Ramp Bass level back to original value
for (int i = 0; i++; usNumSteps)
{
 --usNextVal;
 Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL
usNextVal;
}
```

AUTOMATIC VOLUME CONTROL (AVC) ON/OFF

This programming example shows how to program the AVC on/off when end-user turns it on/off on their device.

```
// Enable AVC (To disable, write 0x0000)
Modify DAP_AVC_CTRL->EN 0x0001
// bit 0
```

Register description

CHIP_ID 0x0000

Table 16. CHIP_ID 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTID								REVID							

BITS	FIELD	RW	RESET	DEFINITION
15:8	PARTID	RO	0xA0	SGTL5000 Part ID 0xA0 - 8 bit identifier for SGTL5000
7:0	REVID	RO	0x00	SGTL5000 Revision ID 0xHH - revision number for SGTL5000.

Table 17. CHIP_DIG_POWER 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									ADC_POWERUP	DAC_POWERUP	DAP_POWERUP	RSVD		I2S_OUT_POWERUP	I2S_IN_POWERUP

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6	ADC_POWERUP	RW	0x0	Enable/disable the ADC block, both digital and analog 0x0 = Disable 0x1 = Enable

BITS	FIELD	RW	RESET	DEFINITION
5	DAC_POWERUP	RW	0x0	Enable/disable the DAC block, both analog and digital 0x0 = Disable 0x1 = Enable
4	DAP_POWERUP	RW	0x0	Enable/disable the DAP block 0x0 = Disable 0x1 = Enable
3:2	RSVD	RW	0x0	Reserved
1	I2S_OUT_POWERUP	RW	0x0	Enable/disable the I2S data output 0x0 = Disable 0x1 = Enable
0	I2S_IN_POWERUP	RW	0x0	Enable/disable the I2S data input 0x0 = Disable 0x1 = Enable

Table 18. CHIP_CLK_CTRL 0x0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										RATE_MODE		SYS_FS		MCLK_FREQ	

BITS	FIELD	RW	RESET	DEFINITION
15:6	RSVD	RO	0x0	Reserved
5:4	RATE_MODE	RW	0x0	Sets the sample rate mode. MCLK_FREQ is still specified relative to the rate in SYS_FS 0x0 = SYS_FS specifies the rate 0x1 = Rate is 1/2 of the SYS_FS rate 0x2 = Rate is 1/4 of the SYS_FS rate 0x3 = Rate is 1/6 of the SYS_FS rate
3:2	SYS_FS	RW	0x2	Sets the internal system sample rate 0x0 = 32 kHz 0x1 = 44.1 kHz 0x2 = 48 kHz 0x3 = 96 kHz
1:0	MCLK_FREQ	RW	0x0	Identifies incoming SYS_MCLK frequency and if the PLL should be used 0x0 = 256*Fs 0x1 = 384*Fs 0x2 = 512*Fs 0x3 = Use PLL The 0x3 (Use PLL) setting must be used if the SYS_MCLK is not a standard multiple of Fs (256, 384, or 512). This setting can also be used if SYS_MCLK is a standard multiple of Fs. Before this field is set to 0x3 (Use PLL), the PLL must be powered up by setting CHIP_ANA_POWER->PLL_POWERUP and CHIP_ANA_POWER->VCOAMP_POWERUP. Also, the PLL dividers must be calculated based on the external MCLK rate and CHIP_PLL_CTRL register must be set (see CHIP_PLL_CTRL register description details on how to calculate the divisors).

Table 19. CHIP_I2S_CTRL 0x0006

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SCLKFREQ	MS	SCLK_INV	DLEN		I2S_MODE		LRALIGN	LRPOL

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	SCLKFREQ	RW	0x0	Sets frequency of I2S_SCLK when in master mode (MS=1). When in slave mode (MS=0), this field must be set appropriately to match SCLK input rate. 0x0 = 64Fs 0x1 = 32Fs - Not supported for RJ mode (I2S_MODE = 1)
7	MS	RW	0x0	Configures master or slave of I2S_LRCLK and I2S_SCLK. 0x0 = Slave: I2S_LRCLK and I2S_SCLK are inputs 0x1 = Master: I2S_LRCLK and I2S_SCLK are outputs NOTE: If the PLL is used (CHIP_CLK_CTRL->MCLK_FREQ==0x3), the SGTL5000 must be a master of the I^2S port (MS==1)
6	SCLK_INV	RW	0x0	Sets the edge that data (input and output) is clocked in on for I2S_SCLK 0x0 = data is valid on rising edge of I2S_SCLK 0x1 = data is valid on falling edge of I2S_SCLK
5:4	DLEN	RW	0x1	I^2S data length 0x0 = 32 bits (only valid when SCLKFREQ=0), not valid for Right Justified Mode 0x1 = 24 bits (only valid when SCLKFREQ=0) 0x2 = 20 bits 0x3 = 16 bits
3:2	I2S_MODE	RW	0x0	Sets the mode for the I^2S port 0x0 = I^2S mode or Left Justified (Use LRALIGN to select) 0x1 = Right Justified Mode 0x2 = PCM Format A/B 0x3 = RESERVED
1	LRALIGN	RW	0x0	I2S_LRCLK Alignment to data word. Not used for Right Justified mode 0x0 = Data word starts 1 I2S_SCLK delay after I2S_LRCLK transition (I^2S format, PCM format A) 0x1 = Data word starts after I2S_LRCLK transition (left justified format, PCM format B)
0	LRPOL	RW	0x0	I2S_LRCLK Polarity when data is presented. 0x0 = I2S_LRCLK = 0 - Left, 1 - Right 1x0 = I2S_LRCLK = 0 - Right, 1 - Left The left subframe should be presented first regardless of the setting of LRPOL.

Table 20. CHIP_SSS_CTRL 0x000A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DAP_MIX_LRSWAP	DAP_LRSWAP	DAC_LRSWAP	RSVD	I2S_LRSWAP	DAP_MIX_SELECT		DAP_SELECT		DAC_SELECT		RSVD		I2S_SELECT	

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RW	0x0	Reserved
14	DAP_MIX_LRSWAP	RW	0x0	DAP Mixer Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAP MIXER Input are swapped.
13	DAP_LRSWAP	RW	0x0	DAP Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAP Input are swapped
12	DAC_LRSWAP	RW	0x0	DAC Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAC are swapped
11	RSVD	RW	0x0	Reserved
10	I2S_LRSWAP	RW	0x0	I2S_DOUT Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the I2S_DOUT are swapped
9:8	DAP_MIX_SELECT	RW	0x0	Select data source for DAP mixer 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = Reserved
7:6	DAP_SELECT	RW	0x0	Select data source for DAP 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = Reserved
5:4	DAC_SELECT	RW	0x1	Select data source for DAC 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = DAP
3:2	RSVD	RW	0x0	Reserved
1:0	I2S_SELECT	WO	0x0	Select data source for I2S_DOUT 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = DAP

Table 21. CHIP_ADCDAC_CTRL 0x000E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		VOL_BUSY_DAC_RIGHT	VOL_BUSY_DAC_LEFT	RSVD		VOL_RAMP_EN	VOL_EXPO_RAMP	RSVD				DAC_MUTE_RIGHT	DAC_MUTE_LEFT	ADC_HPF_FREEZE	ADC_HPF_BYPASS

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RO	0x0	Reserved
13	VOL_BUSY_DAC_RIGHT	RO	0x0	Volume Busy DAC Right 0x0 = Ready 0x1 = Busy - This indicates the channel has not reached its programmed volume/mute level
12	VOL_BUSY_DAC_LEFT	RO	0x0	Volume Busy DAC Left 0x0 = Ready 0x1 = Busy - This indicates the channel has not reached its programmed volume/mute level
11:10	RSVD	RO	0x0	Reserved
9	VOL_RAMP_EN	RW	0x1	Volume Ramp Enable 0x0 = Disables volume ramp. New volume settings take immediate effect without a ramp 0x1 = Enables volume ramp This field affects DAC_VOL. The volume ramp effects both volume settings and mute. When set to 1 a soft mute is enabled.
8	VOL_EXPO_RAMP	RW	0x0	Exponential Volume Ramp Enable 0x0 = Linear ramp over top 4 volume octaves 0x1 = Exponential ramp over full volume range This bit only takes effect if *VOL_RAMP_EN* is 1.
7:4	RSVD	RW	0x0	Reserved
3	DAC_MUTE_RIGHT	RW	0x1	DAC Right Mute 0x0 = Unmute 0x1 = Muted If *VOL_RAMP_EN* = 1, this is a soft mute.
2	DAC_MUTE_LEFT	RW	0x1	DAC Left Mute 0x0 = Unmute 0x1 = Muted If *VOL_RAMP_EN* = 1, this is a soft mute.
1	ADC_HPF_FREEZE	RW	0x0	ADC High Pass Filter Freeze 0x0 = Normal operation 0x1 = Freeze the ADC high-pass filter offset register. The offset continues to be subtracted from the ADC data stream.
0	ADC_HPF_BYPASS	RW	0x0	ADC High Pass Filter Bypass 0x0 = Normal operation 0x1 = Bypassed and offset not updated

Table 22. CHIP_DAC_VOL 0x0010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC_VOL_RIGHT								DAC_VOL_LEFT							

BITS	FIELD	RW	RESET	DEFINITION
15:8	DAC_VOL_RIGHT	RW	0x3C	DAC Right Channel Volume Set the Right channel DAC volume with 0.5017 dB steps from 0 to -90 dB 0x3B and less = Reserved 0x3C = 0 dB 0x3D = -0.5 dB 0xF0 = -90 dB 0xFC and greater = Muted If *VOL_RAMP_EN* = 1, there is an automatic ramp to the new volume setting.
7:0	DAC_VOL_LEFT	RW	0x3C	DAC Left Channel Volume Set the Left channel DAC volume with 0.5017 dB steps from 0 to -90 dB 0x3B and less = Reserved 0x3C = 0 dB 0x3D = -0.5 dB 0xF0 = -90 dB 0xFC and greater = Muted If *VOL_RAMP_EN* = 1, there is an automatic ramp to the new volume setting.

Table 23. CHIP_PAD_STRENGTH 0x0014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						I2S_LRCLK		I2S_SCLK		I2S_DOUT		CTRL_DATA		CTRL_CLK	

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RW	0x0	Reserved
9:8	I2S_LRCLK	RW	0x1	I^2S LRCLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8 V 2.5 V 3.3 V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
7:6	I2S_SCLK	RW	0x1	I^2S SCLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8 V 2.5 V 3.3 V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA

BITS	FIELD	RW	RESET	DEFINITION
5:4	I2S_DOUT	RW	0x1	I^2S DOUT Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8 V 2.5 V 3.3 V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
3:2	CTRL_DATA	RW	0x3	I^2C DATA Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8 V 2.5 V 3.3 V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
1:0	CTRL_CLK	RW	0x3	I^2C CLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8 V 2.5 V 3.3 V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA

Table 24. CHIP_ANA_ADC_CTRL 0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							ADC_VOL_M6DB	ADC_VOL_RIGHT				ADC_VOL_LEFT			

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	ADC_VOL_M6DB	RW	0x0	ADC Volume Range Reduction This bit shifts both right and left analog ADC volume range down by 6.0 dB. 0x0 = No change in ADC range 0x1 = ADC range reduced by 6.0 dB

BITS	FIELD	RW	RESET	DEFINITION
7:4	ADC_VOL_RIGHT	RW	0x0	ADC Right Channel Volume Right channel analog ADC volume control in 1.5.0 dB steps. 0x0 = 0 dB 0x1 = +1.5 dB ... 0xF = +22.5 dB This range is -6.0 dB to +16.5 dB if *ADC_VOL_M6DB* is set to 1.
3:0	ADC_VOL_LEFT	RW	0x0	ADC Left Channel Volume Left channel analog ADC volume control in 1.5 dB steps. 0x0 = 0 dB 0x1 = +1.5 dB ... 0xF = +22.5 dB This range is -6.0 dB to +16.5 dB if *ADC_VOL_M6DB* is set to 1.

Table 25. CHIP_ANA_HP_CTRL 0x0022

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	HP_VOL_RIGHT							RSVD	HP_VOL_LEFT						

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14:8	HP_VOL_RIGHT	RW	0x18	Headphone Right Channel Volume Right channel headphone volume control with 0.5 dB steps. 0x00 = +12 dB 0x01 = +11.5 dB 0x18 = 0 dB ... 0x7F = -51.5 dB
7	RSVD	RO	0x0	Reserved
6:0	HP_VOL_LEFT	RW	0x18	Headphone Left Channel Volume Left channel headphone volume control with 0.5 dB steps. 0x00 = +12 dB 0x01 = +11.5 dB 0x18 = 0 dB ... 0x7F = -51.5 dB

Table 26 is an analog control register that includes mutes, input selects, and zero-cross-detectors for the ADC, headphone, and LINEOUT.

Table 26. 7.0.0.11. CHIP_ANA_CTRL 0x0024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							MUTE_LO	RSVD	SELECT_HP	EN_ZCD_HP	MUTE_HP	RSVD	SELECT_ADC	EN_ZCD_ADC	MUTE_ADC

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	MUTE_LO	RW	0x1	LINEOUT Mute 0x0 = Unmute 0x1 = Mute
7	RSVD	RO	0x0	Reserved
6	SELECT_HP	RW	0x0	Select the headphone input. 0x0 = DAC 0x1 = LINEIN
5	EN_ZCD_HP	RW	0x0	Enable the headphone zero cross detector (ZCD) 0x0 = HP ZCD disabled 0x1 = HP ZCD enabled
4	MUTE_HP	RW	0x1	Mute the headphone outputs 0x0 = Unmute 0x1 = Mute
3	RSVD	RO	0x0	Reserved
2	SELECT_ADC	RW	0x0	Select the ADC input. 0x0 = Microphone 0x1 = LINEIN
1	EN_ZCD_ADC	RW	0x0	Enable the ADC analog zero cross detector (ZCD) 0x0 = ADC ZCD disabled 0x1 = ADC ZCD enabled
0	MUTE_ADC	RW	0x1	Mute the ADC analog volume 0x0 = Unmute 0x1 = Mute

The Table 27, CHIP_LINREG_CTRL 0x0026 register controls the VDDD linear regulator and the charge pump.

Table 27. CHIP_LINREG_CTRL 0x0026

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VDDC_MAN_ASSN	VDDC_ASSN_OVRD	RSVD	D_PROGRAMMING			

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6	VDDC_MAN_ASSN	RW	0x0	Determines chargepump source when VDDC_ASSN_OVRD is set. 0x0 = VDDA 0x1 = VDDIO
5	VDDC_ASSN_OVRD	RW	0x0	Charge pump Source Assignment Override 0x0 = Charge pump source is automatically assigned based on higher of VDDA and VDDIO 0x1 = the source of charge pump is manually assigned by VDDC_MAN_ASSN If VDDIO and VDDA are both the same and greater than 3.1 V, VDDC_ASSN_OVRD and VDDC_MAN_ASSN should be used to manually assign VDDIO as the source for charge pump.
4	RSVD	RW	0x0	Reserved
3:0	D_PROGRAMMING	RW	0x0	Sets the VDDD linear regulator output voltage in 50 mV steps. Must clear the LINREG_SIMPLE_POWERUP and STARTUP_POWERUP bits in the 0x0030 register after power-up, for this setting to produce the proper VDDD voltage. 0x0 = 1.60 0xF = 0.85

The Table 28. CHIP_REF_CTRL 0x0028 register controls the bandgap reference bias voltage and currents.

Table 28. CHIP_REF_CTRL 0x0028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							VAG_VAL					BIAS_CTRL			SMALL_POP

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8:4	VAG_VAL	RW	0x0	Analog Ground Voltage Control These bits control the analog ground voltage in 25 mV steps. This should usually be set to VDDA/2 or lower for best performance (maximum output swing at minimum THD). This VAG reference is also used for the DAC and ADC voltage reference. So changing this voltage scales the output swing of the DAC and the output signal of the ADC. 0x00 = 0.800 V 0x1F = 1.575 V

BITS	FIELD	RW	RESET	DEFINITION
3:1	BIAS_CTRL	RW	0x0	Bias control These bits adjust the bias currents for all of the analog blocks. By lowering the bias current a lower quiescent power is achieved. It should be noted that this mode can affect performance by 3-4 dB. 0x0 = Nominal 0x1-0x3=+12.5% 0x4=-12.5% 0x5=-25% 0x6=-37.5% 0x7=-50%
0	SMALL_POP	RW	0x0	VAG Ramp Control Setting this bit slows down the VAG ramp from ~200 to ~400 ms to reduce the startup pop, but increases the turn on/off time. 0x0 = Normal VAG ramp 0x1 = Slow down VAG ramp

The Table 29, CHIP_MIC_CTRL 0x002A register controls the microphone gain and the internal microphone biasing circuitry.

Table 29. CHIP_MIC_CTRL 0x002A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BIAS_RESISTOR		RSVD	BIAS_VOLT			RSVD		GAIN	

BITS	FIELD	RW	RESET	DEFINITION
15:10	RSVD	RO	0x0	Reserved
9:8	BIAS_RESISTOR	RW	0x0	MIC Bias Output Impedance Adjustment Controls an adjustable output impedance for the microphone bias. If this is set to zero the micbias block is powered off and the output is highZ. 0x0 = Powered off 0x1 = 2.0 kΩ 0x2 = 4.0 kΩ 0x3 = 8.0 kΩ
7	RSVD	RO	0x0	Reserved
6:4	BIAS_VOLT	RW	0x0	MIC Bias Voltage Adjustment Controls an adjustable bias voltage for the microphone bias amp in 250 mV steps. This bias voltage setting should be no more than VDDA-200 mV for adequate power supply rejection. 0x0 = 1.25 V ... 0x7 = 3.00 V
3:2	RSVD	RO	0x0	Reserved
1:0	GAIN	RW	0x0	MIC Amplifier Gain Sets the microphone amplifier gain. At 0 dB setting the THD can be slightly higher than other paths- typically around ~65 dB. At other gain settings the THD are better. 0x0 = 0 dB 0x1 = +20 dB 0x2 = +30 dB 0x3 = +40 dB

Table 30. CHIP_LINE_OUT_CTRL 0x002C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				OUT_CURRENT				RSVD		LO_VAGCNTRL					

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved
11:8	OUT_CURRENT	RW	0x0	Controls the output bias current for the LINEOUT amplifiers. The nominal recommended setting for a 10 kΩ load with 1.0 nF load cap is 0x3. There are only 5 valid settings. 0x0=0.18 mA, 0x1=0.27 mA, 0x3=0.36 mA, 0x7=0.45 mA, 0xF=0.54 mA
7:6	RSVD	RO	0x0	Reserved
5:0	LO_VAGCNTRL	RW	0x0	LINEOUT Amplifier Analog Ground Voltage Controls the analog ground voltage for the LINEOUT amplifiers in 25 mV steps. This should usually be set to VDDIO/2. 0x00 = 0.800 V ... 0x1F = 1.575 V ... 0x23 = 1.675 V 0x24-0x3F are invalid

Table 31. CHIP_LINE_OUT_VOL 0x002E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			LO_VOL_RIGHT					RSVD			LO_VOL_LEFT				

BITS	FIELD	RW	RESET	DEFINITION
15:13	RSVD	RO	0x0	Reserved
12:8	LO_VOL_RIGHT	RW	0x4	LINEOUT Right Channel Volume Controls the right channel LINEOUT volume in 0.5 dB steps. Higher codes have more attenuation. See programming information for Left channel.
7:5	RSVD	RO	0x0	Reserved
4:0	LO_VOL_LEFT	RW	0x4	LINEOUT Left Channel Output Level The LO_VOL_LEFT is used to normalize the output level of the left line output to full scale based on the values used to set LINE_OUT_CTRL -> LO_VAGCNTRL and CHIP_REF_CTRL -> VAG_VAL. In general this field should be set to: $40*\log((VAG\_VAL)/(LO\_VAGCNTRL)) + 15$ Table 32 shows suggested values based on typical VDDIO and VDDA voltages. After setting to the nominal voltage, this field can be used to adjust the output level in +/-0.5 dB increments by using values higher or lower than the nominal setting.

Table 32. LINEOUT Output Level Values

VDDA	VAG_VAL	VDDIO	LO_VAGCNTRL	LO_VOL_*
1.8 V	0.9	3.3 V	1.55	0x06
1.8 V	0.9	1.8 V	0.9	0x0F
3.3 V	1.55	1.8 V	0.9	0x19
3.3 V	1.55	3.3 V	1.55	0x0F

The Table 33, CHIP_ANA_POWER 0x0030 register contains all of the power down controls for the analog blocks. The only other power-down controls are BIAS_RESISTOR in the MIC_CTRL register and the EN_ZCD control bits in ANA_CTRL.

Table 33. CHIP_ANA_POWER 0x0030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DAC_MONO	LINREG_SIMPLE_POWERUP	STARTUP_POWERUP	VDDC_CHRGPMP_POWERUP	PLL_POWERUP	LINREG_D_POWERUP	VCOAMP_POWERUP	VAG_POWERUP	ADC_MONO	REFTOP_POWERUP	HEADPHONE_POWERUP	DAC_POWERUP	CAPLESS_HEADPHONE_POWERUP	ADC_POWERUP	LINEOUT_POWERUP

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RW	0x0	Reserved
14	DAC_MONO	RW	0x1	While DAC_POWERUP is set, this allows the DAC to be put into left only mono operation for power savings. 0x0 = Mono (left only) 0x1 = Stereo
13	LINREG_SIMPLE_POWERUP	RW	0x1	Power up the simple (low power) digital supply regulator. After reset, this bit can be cleared IF VDDD is driven externally OR the primary digital linreg is enabled with LINREG_D_POWERUP 0x0 = Power down 0x1 = Power up
12	STARTUP_POWERUP	RW	0x1	Power up the circuitry needed during the power up ramp and reset. After reset this bit can be cleared if VDDD is coming from an external source. 0x0 = Power down 0x1 = Power up
11	VDDC_CHRGPMP_POWERUP	RW	0x0	Power up the VDDC charge pump block. If neither VDDA or VDDIO is 3.0 V or larger this bit should be cleared before analog blocks are powered up. 0x0 = Power down 0x1 = Power up Note that for charge pump to function, either the PLL must be powered on and programmed correctly (refer to CHIP_CLK_CTRL->MCLK_FREQ description) or the internal oscillator (set CLK_TOP_CTRL->ENABLE_INT_OSC) must be enabled
10	PLL_POWERUP	RW	0x0	PLL Power Up 0x0 = Power down 0x1 = Power up When cleared, the PLL is turned off. This must be set before CHIP_CLK_CTRL -> MCLK_FREQ is programmed to 0x3. The CHIP_PLL_CTRL register must be configured correctly before setting this bit.
9	LINREG_D_POWERUP	RW	0x0	Power up the primary VDDD linear regulator. 0x0 = Power down 0x1 = Power up

BITS	FIELD	RW	RESET	DEFINITION
8	VCOAMP_POWERUP	RW	0x0	Power up the PLL VCO amplifier. 0x0 = Power down 0x1 = Power up
7	VAG_POWERUP	RW	0x0	Power up the VAG reference buffer. Setting this bit starts the power up ramp for the headphone and LINEOUT. The headphone (and/or LINEOUT) powerup should be set BEFORE clearing this bit. When this bit is cleared the power-down ramp is started. The headphone (and/or LINEOUT) powerup should stay set until the VAG is fully ramped down (200 to 400 ms after clearing this bit). 0x0 = Power down 0x1 = Power up
6	ADC_MONO	RW	0x1	While ADC_POWERUP is set, this allows the ADC to be put into left only mono operation for power savings. This mode is useful when only using the microphone input. 0x0 = Mono (left only) 0x1 = Stereo
5	REFTOP_POWERUP	RW	0x1	Power up the reference bias currents 0x0 = Power down 0x1 = Power up This bit can be cleared when the part is a sleep state to minimize analog power.
4	HEADPHONE_POWERUP	RW	0x0	Power up the headphone amplifiers 0x0 = Power down 0x1 = Power up
3	DAC_POWERUP	RW	0x0	Power up the DACs 0x0 = Power down 0x1 = Power up
2	CAPLESS_HEADPHONE_POWERUP	RW	0x0	Power up the capless headphone mode 0x0 = Power down 0x1 = Power up
1	ADC_POWERUP	RW	0x0	Power up the ADCs 0x0 = Power down 0x1 = Power up
0	LINEOUT_POWERUP	RW	0x0	Power up the LINEOUT amplifiers 0x0 = Power down 0x1 = Power up

The Table 34, CHIP_PLL_CTRL 0x0032 register may only be changed after reset, and before PLL_POWERUP is set.

Table 34. CHIP_PLL_CTRL 0x0032

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_DIVISOR					FRAC_DIVISOR										

BITS	FIELD	RW	RESET	DEFINITION
15:11	INT_DIVISOR	RW	0xA	This is the integer portion of the PLL divisor. To determine the value of this field, use the following calculation: INT_DIVISOR = FLOOR(PLL_OUTPUT_FREQ/INPUT_FREQ) PLL_OUTPUT_FREQ = 180.6336 MHz if System sample rate = 44.1 kHz else PLL_OUTPUT_FREQ = 196.608 MHz if System sample rate!= 44.1 kHz INPUT_FREQ = Frequency of the external MCLK provided if CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x0 else INPUT_FREQ = (Frequency of the external MCLK provided/2) If CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x1
10:0	FRAC_DIVISOR	RW	0x0	This is the fractional portion of the PLL divisor. To determine the value of this field, use the following calculation: FRAC_DIVISOR = ((PLL_OUTPUT_FREQ/INPUT_FREQ) - INT_DIVISOR)*2048 PLL_OUTPUT_FREQ = 180.6336 MHz if System sample rate = 44.1 kHz else PLL_OUTPUT_FREQ = 196.608 MHz if System sample rate!= 44.1 kHz INPUT_FREQ = Frequency of the external MCLK provided if CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x0 else INPUT_FREQ = (Frequency of the external MCLK provided/2) If CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x1

Table 35, CHIP_CLK_TOP_CTRL 0x0034 has the miscellaneous controls for the clock block.

Table 35. CHIP_CLK_TOP_CTRL 0x0034

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				ENABLE_INT_OSC	RSVD							INPUT_FREQ_DIV2	RSVD		

BITS	FIELD	RW	RESET	DEFINITION
15:12	RESERVED	RO	0x0	Reserved
11	ENABLE_INT_OSC	RW	0x0	Setting this bit enables an internal oscillator to be used for the zero cross detectors, the short detect recovery, and the charge pump. This allows the I^2S clock to be shut off while still operating an analog signal path. This bit can be kept on when the I^2S clock is enabled, but the I^2S clock is more accurate so it is preferred to clear this bit when I^2S is present.
10:4	RSVD	RW	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
3	INPUT_FREQ_DIV2	RW	0x0	SYS_MCLK divider before PLL input 0x0 = pass through 0x1 = SYS_MCLK is divided by 2 before entering PLL This must be set when the input clock is above 17 MHz. This has no effect when the PLL is powered down.
2:0	RSVD	RW	0x0	Reserved

Status bits for analog blocks are found in Table 36, CHIP_ANA_STATUS 0x0036

Table 36. CHIP_ANA_STATUS 0x0036

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						LRSHORT_STS	CSHORT_STS	RSVD			PLL_IS_LOCKED	RSVD			

BITS	FIELD	RW	RESET	DEFINITION
15:10	RSVD	RO	0x0	Reserved
9	LRSHORT_STS	RO	0x0	This bit is high whenever a short is detected on the left or right channel headphone drivers. 0x0 = Normal 0x1 = Short detected
8	CSHORT_STS	RO	0x0	This bit is high whenever a short is detected on the capless headphone common/ center channel driver. 0x0 = Normal 0x1 = Short detected
7:5	RSVD	RO	0x0	Reserved
4	PLL_IS_LOCKED	RO	0x0	This bit goes high after the PLL is locked. 0x0 = PLL is not locked 0x1 = PLL is locked
3:0	RSVD	RO	0x0	Reserved

Table 37, CHIP_ANA_TEST1 0x0038 and Table 38, CHIP_ANA_TEST2 0x003A register controls are intended only for debug.

Table 37. CHIP_ANA_TEST1 0x0038

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HP_IALL_ADJ		HP_I1_ADJ		HP_ANTIPOP			HP_CLASSAB	HP_HOLD_GND_CENTER	HP_HOLD_GND	VAG_DOUB_CURRENT	VAG_CLASSA	TM_ADCIN_TOHP	TM_HPCOMMON	TM_SELECT_MIC	TESTMODE

BITS	FIELD	RW	RESET	DEFINITION
15:14	HP_IALL_ADJ	RW	0x0	These bits control the overall bias current of the headphone amplifier (all stages including first and output stage). 0x0=nominal, 0x1=-50%, 0x2=+50%, 0x3=-40%
13:12	HP_I1_ADJ	RW	0x0	These bits control the bias current for the first stage of the headphone amplifier. 0x0=nominal, 0x1=-50%, 0x2=+100%, 0x3=+50%
11:9	HP_ANTIPOP	RW	0x0	These bits control the headphone output current in classA mode and also the pull-down strength while powering off. These bits normally are not needed.
8	HP_CLASSAB	RW	0x1	This defaults high. When this bit is high the headphone is in classAB mode. ClassA mode would normally not be used.
7	HP_HOLD_GND_CENTER	RW	0x1	This defaults high. When this bit is high and the capless headphone center channel is powered off, the output is tied to ground. This is the preferred mode of operation for best antipop performance.
6	HP_HOLD_GND	RW	0x1	This defaults high. When this bit is high and the headphone is powered off, the output is tied to ground. This is the preferred mode of operation for best antipop performance.
5	VAG_DOUB_CURRENT	RW	0x0	Double the VAG output current when in classA mode.
4	VAG_CLASSA	RW	0x0	Turn off the classAB output current for the VAG buffer. The classA current is limited so this may cause clipping in some modes.
3	TM_ADCIN_TOHP	RW	0x0	Put ADCmux output onto the headphone output pin. Must remove headphone load and any external headphone compensation for this mode.
2	TM_HPCOMMON	RW	0x0	Enable headphone common to be used in ADCmux for testing
1	TM_SELECT_MIC	RW	0x0	Enable the mic-adc-dac-HP path
0	TESTMODE	RW	0x0	Enable the analog test mode paths

Table 38. CHIP_ANA_TEST2 0x003A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	LINEOUT_TO_VDDA	SPARE	MONOMODE_DAC	VCO_TUNE_AGAIN	LO_PASS_MASTERVAG	INVERT_DAC_SAMPLE_CLOCK	INVERT_DAC_DATA_TIMING	DAC_EXTEND_RTZ	DAC_DOUBLE_I	DAC_DIS_RTZ	DAC_CLASSA	INVERT_ADC_SAMPLE_CLOCK	INVERT_ADC_DATA_TIMING	ADC_LESSI	ADC_DITHEROFF

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14	LINEOUT_TO_VDDA	RW	0x0	Changes the LINEOUT amplifier power supply from VDDIO to VDDA. Typically LINEOUT should be on the higher power supply. This bit is useful when VDDA is ~3.3 V and VDDIO is ~1.8 V.
13	SPARE	RW	0x0	Spare registers to analog.
12	MONOMODE_DAC	RW	0x0	Copy the left channel DAC data to the right channel. This allows both left and right to play from MONO dac data.
11	VCO_TUNE_AGAIN	RW	0x0	When toggled high then low forces the PLL VCO to retune the number of inverters in the ring oscillator loop.

BITS	FIELD	RW	RESET	DEFINITION
10	LO_PASS_MASTERVAG	RW	0x0	Tie the main analog VAG to the LINEOUT VAG. This can improve SNR for the LINEOUT when both are the same voltage.
9	INVERT_DAC_SAMPLE_CLOCK	RW	0x0	Change the clock edge used for the DAC output sampling.
8	INVERT_DAC_DATA_TIMING	RW	0x0	Change the clock edge used for the digital to analog DAC data crossing.
7	DAC_EXTEND_RTZ	RW	0x0	Extend the return-to-zero time for the DAC.
6	DAC_DOUBLE_I	RW	0x0	Double the output current of the DAC amplifier when it is in classA mode.
5	DAC_DIS_RTZ	RW	0x0	Turn off the return-to-zero in the DAC. In mode cases, this hurts the SNDR of the DAC.
4	DAC_CLASSA	RW	0x0	Turn off the classAB mode in the DAC amplifier. This mode should normally not be used. The output current is not high enough to support a full scale signal in this mode.
3	INVERT_ADC_SAMPLE_CLOCK	RW	0x0	Change the clock edge used for the ADC sampling.
2	INVERT_ADC_DATA_TIMING	RW	0x0	Change the clock edge used for the analog to digital ADC data crossing
1	ADC_LESSI	RW	0x0	Drops ADC bias currents by 20%
0	ADC_DITHEROFF	RW	0x0	Turns off the ADC dithering.

The Table 39, CHIP_SHORT_CTRL 0x003C register contains controls for the headphone short detectors.

Table 39. CHIP_SHORT_CTRL 0x003C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	LVLADJR			RSVD	LVLADJL			RSVD	LVLADJC			MODE_LR		MODE_CM	

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14:12	LVLADJR	RW	0x0	These bits adjust the sensitivity of the right channel headphone short detector in 25 mA steps.This trip point can vary by ~30% over process so leave plenty of guard band to avoid false trips. This short detect trip point is also effected by the bias current adjustments made by CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=25 mA 0x2=50 mA 0x1=75 mA 0x0=100 mA 0x4=125 mA 0x5=150 mA 0x6=175 mA 0x7=200 mA
11	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
10:8	LVLADJL	RW	0x0	These bits adjust the sensitivity of the left channel headphone short detector in 25 mA steps.This trip point can vary by ~30% over process so leave plenty of guard band to avoid false trips. This short detect trip point is also effected by the bias current adjustments made by CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=25 mA 0x2=50 mA 0x1=75 mA 0x0=100 mA 0x4=125 mA 0x5=150 mA 0x6=175 mA 0x7=200 mA
7	RSVD	RO	0x0	Reserved
6:4	LVLADJC	RW	0x0	These bits adjust the sensitivity of the capless headphone center channel short detector in 50 mA steps. This trip point can vary by ~30% over process so leave plenty of guard band to avoid false trips. This short detect trip point is also effected by the bias current adjustments CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=50 mA 0x2=100 mA 0x1=150 mA 0x0=200 mA 0x4=250 mA 0x5=300 mA 0x6=350 mA 0x7=400 mA
3:2	MODE_LR	RW	0x0	These bits control the behavior of the short detector for the capless headphone central channel driver. This mode should be set prior to powering up the headphone amplifier. When a short is detected the amplifier output switches to classA mode internally to avoid excessive currents. 0x0 = Disable short detector, reset short detect latch, software view non-latched short signal 0x1 = Enable short detector and reset the latch at timeout (every ~50 ms) 0x2 = This mode is not used/invalid 0x3 = Enable short detector with only manual reset (have to return to 0x0 to reset the latch)
1:0	MODE_CM	RW	0x0	These bits control the behavior of the short detector for the capless headphone central channel driver. This mode should be set prior to powering up the headphone amplifier. When a short is detected the amplifier output switches to classA mode interally to avoid excessive currents. 0x0 = Disable short detector, reset short detect latch, software view non-latched short signal 0x1 = Enable short detector and reset the latch at timeout (every ~50 ms) 0x2 = Enable short detector and auto reset when output voltage rises (preferred mode) 0x3 = Enable short detector with only manual reset (have to return to 0x0 to reset the latch)

Table 40. DAP_CONTROL 0x0100

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											MIX_EN	RSVD			DAP_EN

BITS	FIELD	RW	RESET	DEFINITION
15:5	RSVD	RO	0x0	Reserved
4	MIX_EN	RW	0x0	Enable/Disable the DAP mixer path 0x0 = Disable 0x1 = Enable When enabled, DAP_EN must also be enabled to use the mixer.
3:1	RSVD	RO	0x0	Reserved
0	DAP_EN	RW	0x0	Enable/Disable digital audio processing (DAP) 0x0 = Disable. When disabled, no audio passes through. 0x1 = Enable. When enabled, audio can pass through DAP even if none of the DAP functions are enabled.

Table 41. DAP_PEQ 0x0102

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													EN		

BITS	FIELD	RW	RESET	DEFINITION
15:3	RSVD	RO	0x0	Reserved
2:0	EN	RW	0x0	Set to Enable the PEQ filters 0x0 = Disabled 0x1 = 1 Filter Enabled 0x2 = 2 Filters Enabled ..... 0x7 = Cascaded 7 Filters DAP_AUDIO_EQ->EN bit must be set to 1 in order to enable the PEQ

Table 42. DAP_BASS_ENHANCE 0x0104

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							BYPASS_HPF	RSVD	CUTOFF			RSVD			EN

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	BYPASS_HPF	RW	0x0	Bypass high pass filter 0x0 = Enable high pass filter 0x1 = Bypass high pass filter
7	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
6:4	CUTOFF	RW	0x4	Set cut-off frequency 0x0 = 80 Hz 0x1 = 100 Hz 0x2 = 125 Hz 0x3 = 150 Hz 0x4 = 175 Hz 0x5 = 200 Hz 0x6 = 225 Hz
3:1	RSVD	RO	0x0	Reserved
0	EN	RW	0x0	Enable/Disable Bass Enhance 0x0 = Disable 0x1 = Enable

Table 43. DAP_BASS_ENHANCE_CTRL 0x0106

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		LR_LEVEL						RSVD	BASS_LEVEL						

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RO	0x0	Reserved
13:8	LR_LEVEL	RW	0x5	Left/Right Mix Level Control 0x00= +6.0 dB for Main Channel ..... 0x3F= Least L/R Channel Level
7	RSVD	RO	0x0	
6:0	BASS_LEVEL	RW	0x1f	Bass Harmonic Level Control 0x00= Most Harmonic Boost ..... 0x7F=Least Harmonic Boost

Table 44. DAP_AUDIO_EQ 0x0108

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														EN	

BITS	FIELD	RW	RESET	DEFINITION
15:2	RSVD	RO	0x0	Reserved
1:0	EN	RW	0x0	Selects between PEQ/GEQ/Tone Control and Enables it. 0x0 = Disabled. 0x1 = Enable PEQ. NOTE: DAP_PEQ->EN bit must also be set to the desired number of filters (bands) in order for the PEQ to be enabled. 0x2 = Enable Tone Control 0x3 = Enable 5 Band GEQ

Table 45. DAP_SGTL_SURROUND 0x010A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									WIDTH_CONTROL			RSVD		SELECT	

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:4	WIDTH_CONTROL	RW	0x4	Freescale Surround Width Control - The width control changes the perceived width of the sound field. 0x0 = Least Width ..... 0x7 = Most Width
3:2	RSVD	RO	0x0	Reserved
1:0	SELECT	RW	0x0	Freescale Surround Selection 0x0 = Disabled 0x1 = Disabled 0x2 = Mono input Enable 0x3 = Stereo input Enable

Table 46. DAP_FILTER_COEF_ACCESS 0x010C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							WR	INDEX							

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	WR	WO	0x0	When set, the coefficients written in the ten coefficient data registers are loaded into the filter specified by INDEX
7:0	INDEX	RW	0x0	Specifies the index for each of the seven bands of the filter coefficient that needs to be written to. Each filter has 5 coefficients that need to be loaded into the 10 coefficient registers (MSB, LSB) before setting the index and WR bit. Steps to write coefficients: 1. Write the five 20-bit coefficient values to DAP_COEF_WR_XX_MSB and DAP_COEF_WR_XX_LSB registers (XX= B0,B1,B2,A1,A2) 2. Set INDEX of the coefficient from the table below. 3. Set the WR bit to load the coefficient. NOTE: Steps 2 and 3 can be performed with a single write to DAP_FILTER_COEF_ACCESS register. Coefficient address: Band 0 = 0x00 Band 1 = 0x01 Band 2 = 0x02 Band 3 = 0x03 Band 4 = 0x04 ... Band 7 = 0x06

Table 47. DAP_COEF_WR_B0_MSB 0x010E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT_19	BIT_18	BIT_17	BIT_16	BIT_15	BIT_14	BIT_13	BIT_12	BIT_11	BIT_10	BIT_9	BIT_8	BIT_7	BIT_6	BIT_5	BIT_4

BITS	FIELD	RW	RESET	DEFINITION
15	BIT_19	WO	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written
14	BIT_18	WO	0x0	
13	BIT_17	WO	0x0	
12	BIT_16	WO	0x0	
11	BIT_15	WO	0x0	
10	BIT_14	WO	0x0	
9	BIT_13	WO	0x0	
8	BIT_12	WO	0x0	
7	BIT_11	WO	0x0	
6	BIT_10	WO	0x0	
5	BIT_9	WO	0x0	
4	BIT_8	WO	0x0	
3	BIT_7	WO	0x0	
2	BIT_6	WO	0x0	
1	BIT_5	WO	0x0	
0	BIT_4	WO	0x0	

Table 48. DAP_COEF_WR_B0_LSB 0x0110

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												BIT_3	BIT_2	BIT_1	BIT_0

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	
3	BIT_3	WO	0x0	
2	BIT_2	WO	0x0	
1	BIT_1	WO	0x0	
0	BIT_0	WO	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

Table 49. DAP_AUDIO_EQ_BASS_BAND0 0x0116 115 Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets Tone Control Bass/GEQ Band0 0x5F = sets to 12 dB 0x2F = sets to 0 dB 0x00 = sets to -11.75 dB Each LSB is 0.25 dB

Table 50. DAP_AUDIO_EQ_BAND1 0x0118 330 Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets GEQ Band1 0x5F = sets to 12 dB 0x2F = sets to 0 dB 0x00 = sets to -11.75 dB Each LSB is 0.25 dB

Table 51. DAP_AUDIO_EQ_BAND2 0x011A 990 Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets GEQ Band2 0x5F = sets to 12 dB 0x2F = sets to 0 dB 0x00 = sets to -11.75 dB Each LSB is 0.25 dB

Table 52. DAP_AUDIO_EQ_BAND3 0x011C 3000 Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets GEQ Band3 0x5F = sets to 12 dB 0x2F = sets to 0 dB 0x00 = sets to -11.75 dB Each LSB is 0.25 dB

Table 53. DAP_AUDIO_EQ_TREBLE_BAND4 0x011E 9900 Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets Tone Control Treble/GEQ Band4 0x5F = sets to 12 dB 0x2F = sets to 0 dB 0x00 = sets to -11.75 dB Each LSB is 0.25 dB

Table 54, DAP_MAIN_CHAN 0x0120 sets the main channel volume level

Table 54. DAP_MAIN_CHAN 0x0120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOL															

BITS	FIELD	RW	RESET	DEFINITION
15:0	VOL	RW	0x8000	DAP Main Channel Volume 0xFFFF = 200% 0x8000 (default) = 100% 0x0000 = 0%

Table 55, DAP_MIX_CHAN 0x0122 sets the mix channel volume level

Table 55. DAP_MIX_CHAN 0x0122

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOL															

BITS	FIELD	RW	RESET	DEFINITION
15:0	VOL	RW	0x0000	DAP Mix Channel Volume 0xFFFF = 200% 0x8000 = 100% 0x0000 (default) = 0%

Table 56. DAP_AVC_CTRL 0x0124

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	MAX_GAIN		RSVD		LBI_RESPONSE		RSVD		HARD_LIMIT_EN	RSVD				EN

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14	RSVD	RW	0x1	Reserved.
13:12	MAX_GAIN	RW	0x1	Maximum gain that can be applied by the AVC in expander mode. 0x0 = 0 dB gain 0x1 = 6.0 dB of gain 0x2 = 12 dB of gain
11:10	RSVD	RO	0x0	Reserved
9:8	LBI_RESPONSE	RW	0x1	Integrator Response 0x0 = 0 mS LBI 0x1 = 25 mS LBI 0x2 = 50 mS LBI 0x3 = 100 mS LBI
7:6	RSVD	RO	0x0	Reserved
5	HARD_LIMIT_EN	RW	0x0	Enable Hard Limiter Mode 0x0 = Hard limit disabled. AVC Compressor/Expander is enabled. 0x1 = Hard limit enabled. The signal is limited to the programmed threshold. (Signal saturates at the threshold)
4:1	RSVD	RO	0x0	Reserved
0	EN	RW	0x0	Enable/disable AVC 0x0 = Disable 0x1 = Enable

Table 57. DAP_AVC_THRESHOLD 0x0126

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRESH															

BITS	FIELD	RW	RESET	DEFINITION
15:0	THRESH	RW	0x1473	AVC Threshold Value Threshold is programmable. Use the following formula to calculate hex value: Hex Value = ((10^(THRESHOLD_dB/20))*0.636)*2^15 Threshold can be set in the range of 0 dB to -96 dB Example Values: 0x1473 = Set Threshold to -12 dB 0x0A40 = Set Threshold to -18 dB

Table 58. DAP_AVC_ATTACK 0x0128

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				RATE											

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved
11:0	RATE	RW	0x28	AVC Attack Rate This is the rate at which the AVC applies attenuation to the signal to bring it to the threshold level. AVC Attack Rate is programmable. To use a custom rate, use the formula below to convert from dB/S to hex value: Hex Value = (1 - (10^(-(Rate_dBs/(20*SYS_FS)))) * 2^19 where, SYS_FS is the system sample rate configured in CHIP_CLK_CTRL register. Example values: 0x28 = 32 dB/s 0x10 = 8.0 dB/s 0x05 = 4.0 dB/s 0x03 = 2.0 dB/s

Table 59. DAP_AVC_DECAY 0x012A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				RATE											

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved
11:0	RATE	RW	0x50	AVC Decay Rate This is the rate at which the AVC releases the attenuation previously applied to the signal during attack. AVC Decay Rate is programmable. To use a custom rate, use the formula below to convert from dB/S to hex value: Hex Value = (1 - (10^(-(Rate_dBs/(20*SYS_FS)))) * 2^23 where, SYS_FS is the system sample rate configured in CHIP_CLK_CTRL register. Example values: 0x284 = 32 dB/s 0xA0 = 8.0 dB/s 0x50 = 4.0 dB/s 0x28 = 2.0 dB/s

Table 60. DAP_COEF_WR_B1_MSB 0x012C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

Table 61. DAP_COEF_WR_B1_LSB 0x012E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

Table 62. DAP_COEF_WR_B2_MSB 0x0130

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

Table 63. DAP_COEF_WR_B2_LSB 0x0132

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

Table 64. DAP_COEF_WR_A1_MSB 0x0134

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

Table 65. DAP_COEF_WR_A1_LSB 0x0136

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

Table 66. DAP_COEF_WR_A2_MSB 0x0138

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

Table 67. DAP_COEF_WR_A2_LSB 0x013A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

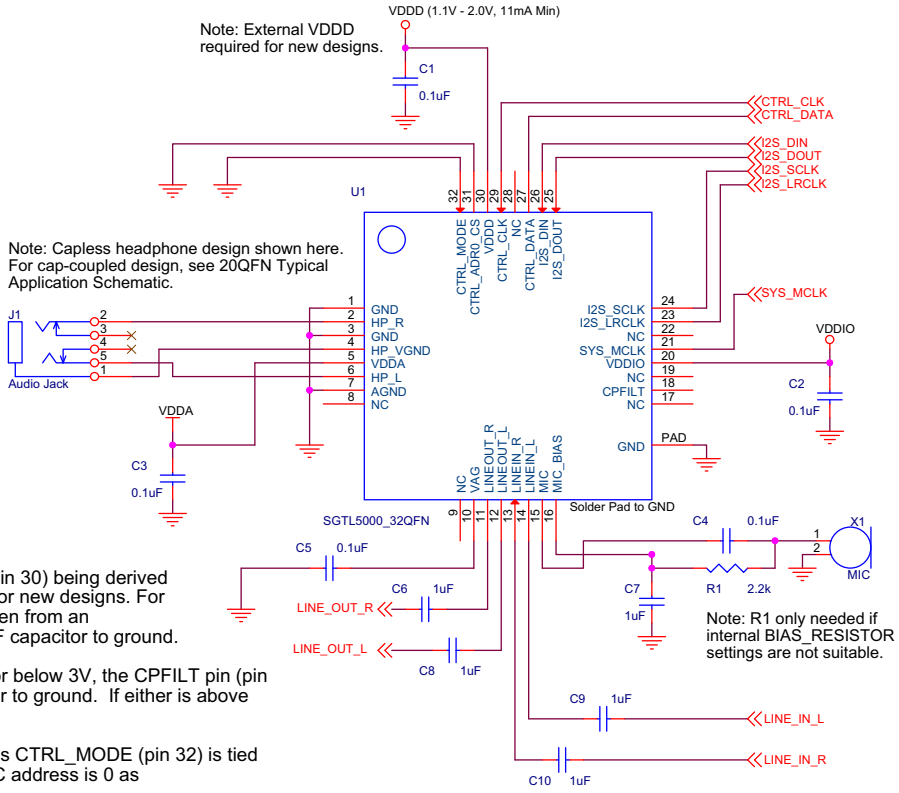
TYPICAL APPLICATIONS

INTRODUCTION

Typical connections are shown in the following application diagrams. For new designs, and for either the 20 QFN or 32 QFN part, an external VDDD power supply connection is required along with a 0.1 μF cap connection from VDDD to ground.

CPFILT Note: The CPFILT cap value is 0.1 μF. If both VDDIO and VDDA are ≤ 3.0 V, the CPFILT pin must be connected to a 0.1 μF cap to GND. If either is > 3.0 V, the CPFILT cap MUST NOT be placed.

HP_VGND Note: Do not connect HP_VGND to system ground, even when unused. This is a virtual ground (DC voltage) that should never connect to an actual “0 Volt ground”. Use the widest, shortest trace possible for the HP_VGND.



Notes:

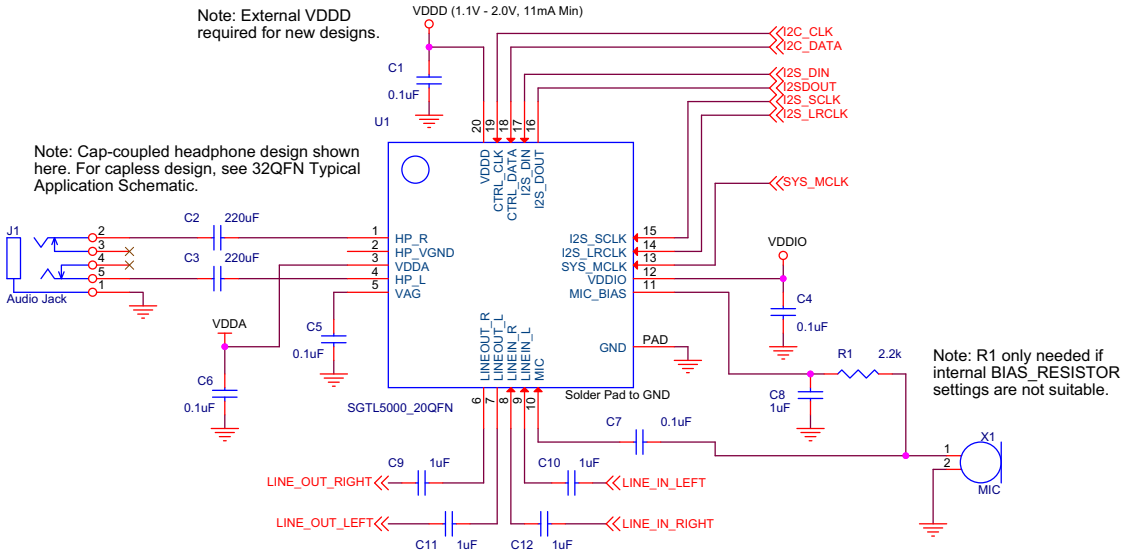
1. This 32QFN schematic shows VDDD (pin 30) being derived externally. An external VDDD is required for new designs. For lowest power operation, VDDD can be driven from an external 1.2V switching supply with a 0.1uF capacitor to ground.

2. If both VDDIO and VDDA are equal to or below 3V, the CPFILT pin (pin 17) must be connected to a 0.1uF capacitor to ground. If either is above 3V, this capacitor must not be placed.

3. The above shows I2C implementation as CTRL_MODE (pin 32) is tied to ground. In addition, address 0 of the I2C address is 0 as CTRL_ADR0_CS (pin 31) is tied to ground.

4. AGND (pin 7) should be "star" connected to the jack grounds for LINEIN and LINEOUT, and to the VAG capacitor ground. This node should via to the ground plane (or connected to ground) at a single point.

Figure 19. 32 QFN Typical Application Schematic



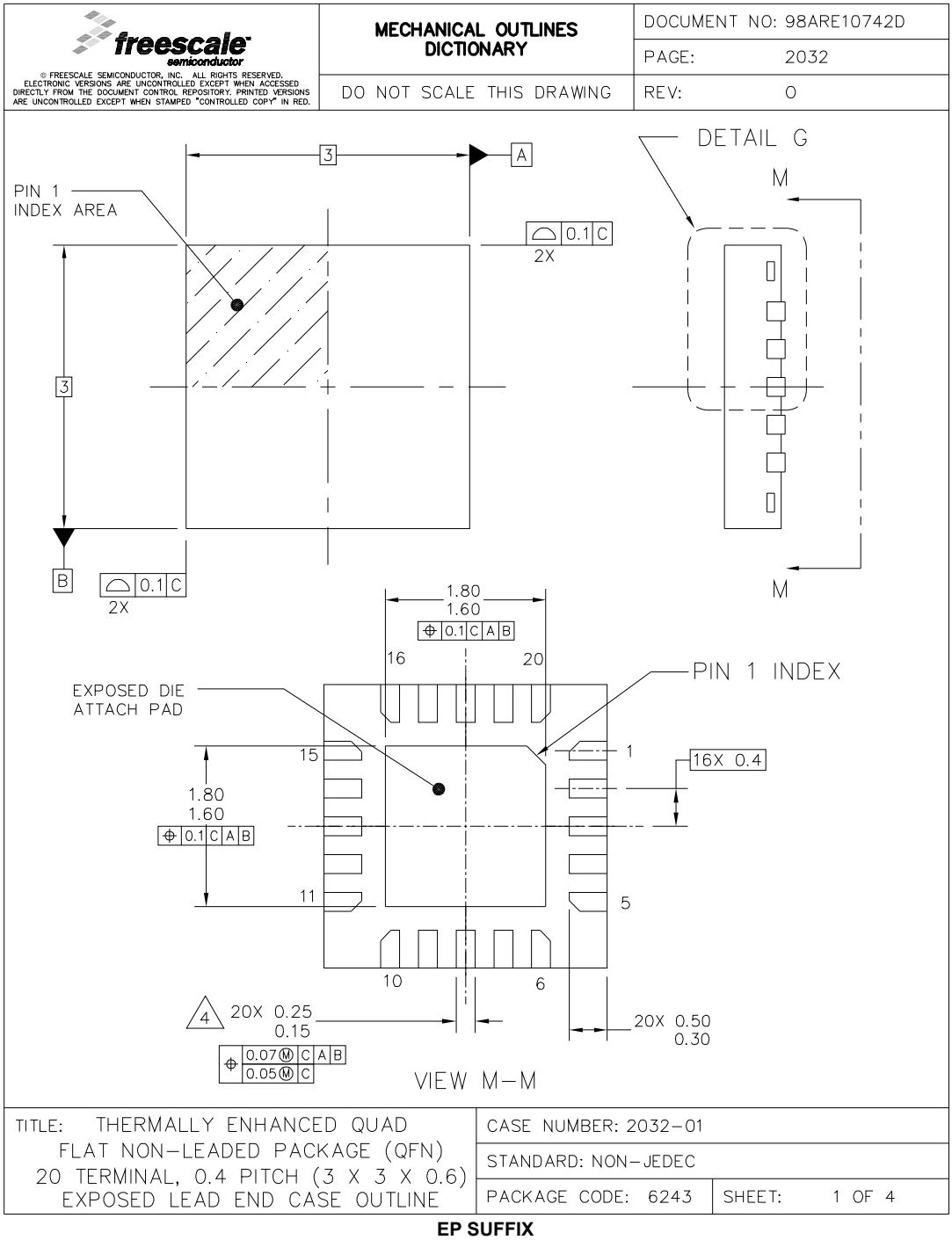
Note: Bottom PAD/FLAG/Paddle MUST be connected to ground.

Figure 20. 20 QFN Typical Application Schematic

PACKAGING

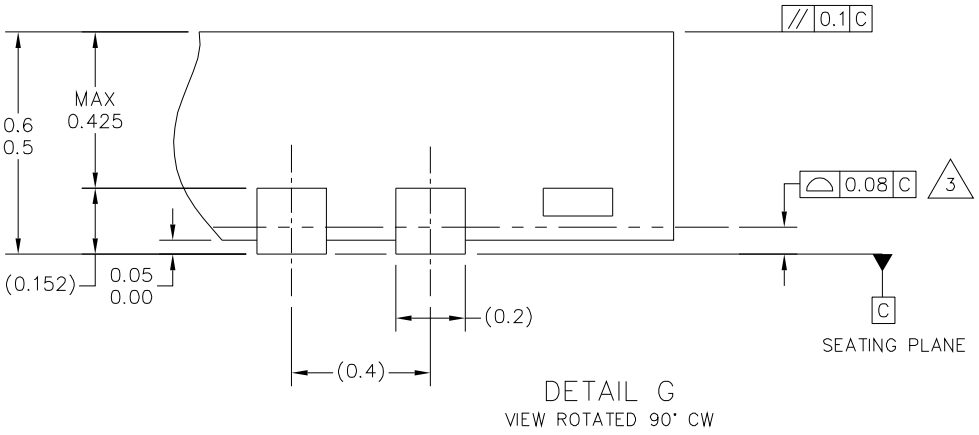
PACKAGE DIMENSIONS

For the most current package revision, visit **www.freescale.com** and perform a keyword search using the 98Axxxxxxxxx listed on the following pages.



EP SUFFIX
20-PIN
98ARE10742D
REVISION 0

MECHANICAL OUTLINES DICTIONARY	DOCUMENT NO: 98ARE10742D
	PAGE: 2032
DO NOT SCALE THIS DRAWING	REV: O



DETAIL G
VIEW ROTATED 90° CW

TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 20 TERMINAL, 0.4 PITCH (3 X 3 X 0.6) EXPOSED LEAD END CASE OUTLINE	CASE NUMBER: 2032–01	
	STANDARD: NON–JEDEC	
	PACKAGE CODE: 6243	SHEET: 2 OF 4

EP SUFFIX
20-PIN
98ARE10742D
REVISION 0

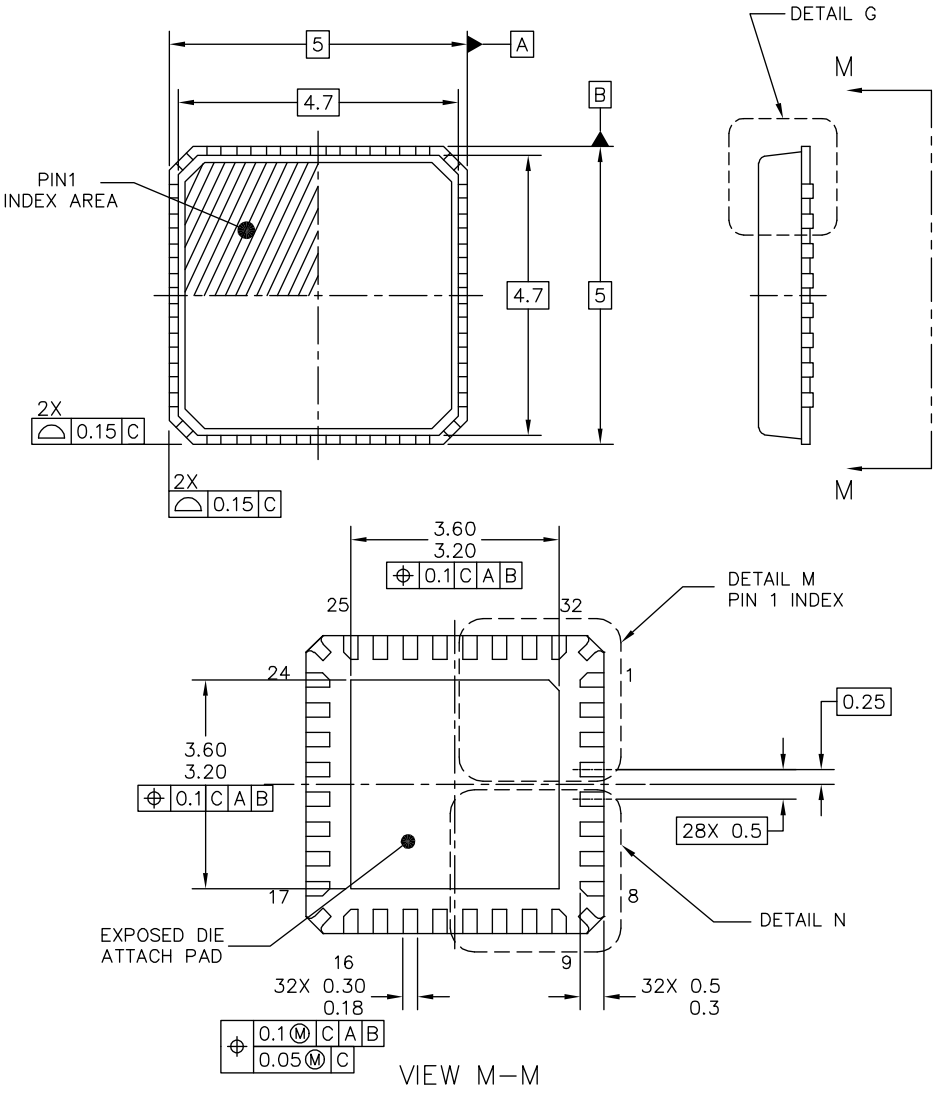
freescale semiconductor	MECHANICAL OUTLINES DICTIONARY	DOCUMENT NO: 98ARE10742D
		PAGE: 2032
	DO NOT SCALE THIS DRAWING	REV: O

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M–1994.
3. COPLANARITY APPLIES TO LEADS, CORNER LEADS, AND DIE ATTACH PAD.
4. DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.20 MM AND 0.25 MM FROM TERMINAL TIP.
5. MIN. METAL GAP SHOULD BE 0.2MM.

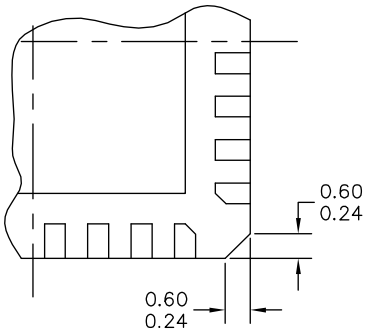
TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 20 TERMINAL, 0.4 PITCH (3 X 3 X 0.6) EXPOSED LEAD END CASE OUTLINE	CASE NUMBER: 2032–01	
	STANDARD: NON–JEDEC	
	PACKAGE CODE: 6243	SHEET: 3 OF 4

EP SUFFIX
20-PIN
98ARE10742D
REVISION 0



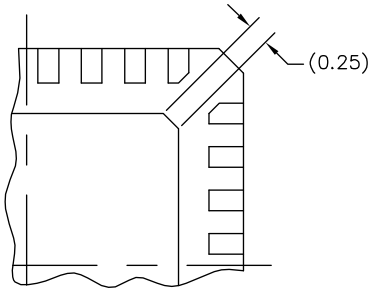
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	**MECHANICAL OUTLINE**	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 32 TERMINAL, 0.5 PITCH (5 X 5 X 1)	DOCUMENT NO: 98ARE10739D	REV: O
	CASE NUMBER: 2029–01	15 MAY 2008
	STANDARD: JEDEC MO–220 VHHD–5	

FC SUFFIX
32-PIN
98ARE10739D
REVISION 0

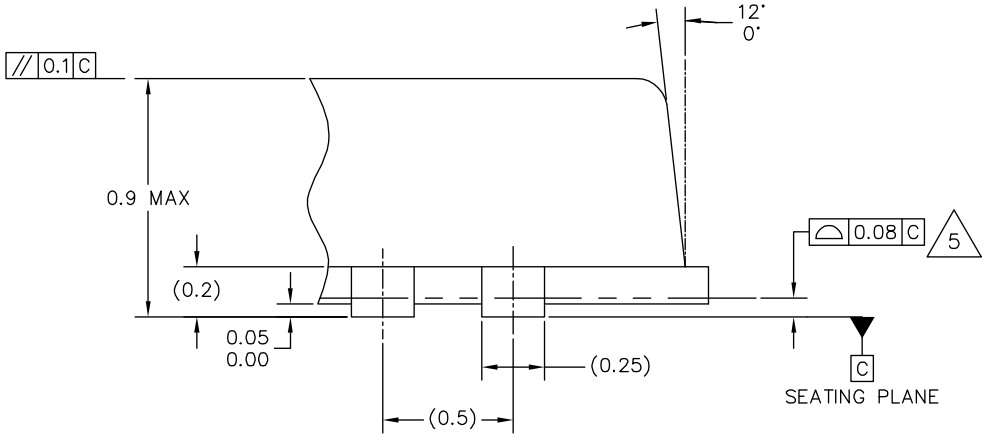


DETAIL N
CORNER CONFIGURATION OPTION

4



DETAIL M
PREFERRED BACKSIDE PIN 1 INDEX



DETAIL G
VIEW ROTATED 90° CW

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	**MECHANICAL OUTLINE**	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 32 TERMINAL, 0.5 PITCH (5 X 5 X 1)	DOCUMENT NO: 98ARE10739D	REV: O	
	CASE NUMBER: 2029–01	15 MAY 2008	
	STANDARD: JEDEC MO–220 VHHD–5		

FC SUFFIX
32-PIN
98ARE10739D
REVISION 0

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. THE COMPLETE JEDEC DESIGNATOR FOR THIS PACKAGE IS: HF–PQFN.
4. DIMENSIONS OF OPTIONAL FEATURES ARE FOR REFERENCE ONLY.
5. COPLANARITY APPLIES TO LEADS, AND DIE ATTACH PAD.
6. MIN METAL GAP SHOULD BE 0.2MM.

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 32 TERMINAL, 0.5 PITCH (5 X 5 X 1)	DOCUMENT NO: 98ARE10739D	REV: O
	CASE NUMBER: 2029–01	15 MAY 2008
	STANDARD: JEDEC MO–220 VHHD–5	

FC SUFFIX
32-PIN
98ARE10739D
REVISION 0

REVISION HISTORY

REVISION	DATE	DESCRIPTION
3.0	6/2010	• Conversion from the old Freescale form and style to the current version. No existing content has been added, altered, or removed.
4.0	9/2010	• Corrected Pin 4 explanation (32-pin package) and added Pin 3 (32-Pin package) to Table 1.
5.0	5/2013	• Corrected LINEOUT - 100 dB SNR (-60 dB input) and -85 dB THD+N (VDDIO = 3.3 V) in features • Added note for HP_VGND and CPFILT in pin definition table • Moved Recommended Operating Conditions to separate table • Added Input/Output Electrical Characteristics • Corrected LINEIN Input Level from 0.75 to 0.57 • Corrected Table 7 Test Conditions unless otherwise noted: VDDIO = 1.8 V, VDDA = 1.8 V, TA = 25 °C, Slave mode, fS = 48 kHz, MCLK = 256 fS, 24 bit input • Added note for HP_VGND and CPFILT to Typical Applications introduction • Corrected pin nomenclature as required for consistency • Clarified Bits 3:0 in Figure 27 • Corrected pin name in Figure 3 and Table 1 • Corrected address name in Figure 6, I2C, SPI • Changed limits on LINEOUT Output level • Changed 0x00 = sets to 12 dB to 11.75 dB, and deleted “To convert dB to hex value, use Hex Value = 4* dB value + 47” on tables 49, 50, 51, 52 and 53. • Revised back page. Updated document properties. Added SMARTMOS sentence to first paragraph. • Added comment for “new designs” where applicable • Corrected pin designations in the Pin Connections section • Changed limits and conditions for LINEOUT Output level and LINEOUT Output level • Added two new application diagrams in Typical Applications section
6.0	11/2013	• Modified front page intro text to include more target markets and to remove type of IC technology • Increased HP max output power from 45 mW to 62.5 mW at 1.02 kHz based on bench measurements • Changed TYP LINEIN input impedance from 100kohm to 29 kohm at 1.02 kHz based on bench measurements • Added MIC input impedance based on bench measurements • Removed 10 kohm MIN LINEIN input impedance, and added 29 kohm as TYP in Table 5 and Table 6 • Added 12 kHz sample rate to Functional Description Introduction, and added 12 kHz and 24 kHz sample rates to Table 8

RoHS

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: SGTL5000
Rev. 6.0
11/2013

freescale™

ANEXO 4

Hoja de datos del ENCODER

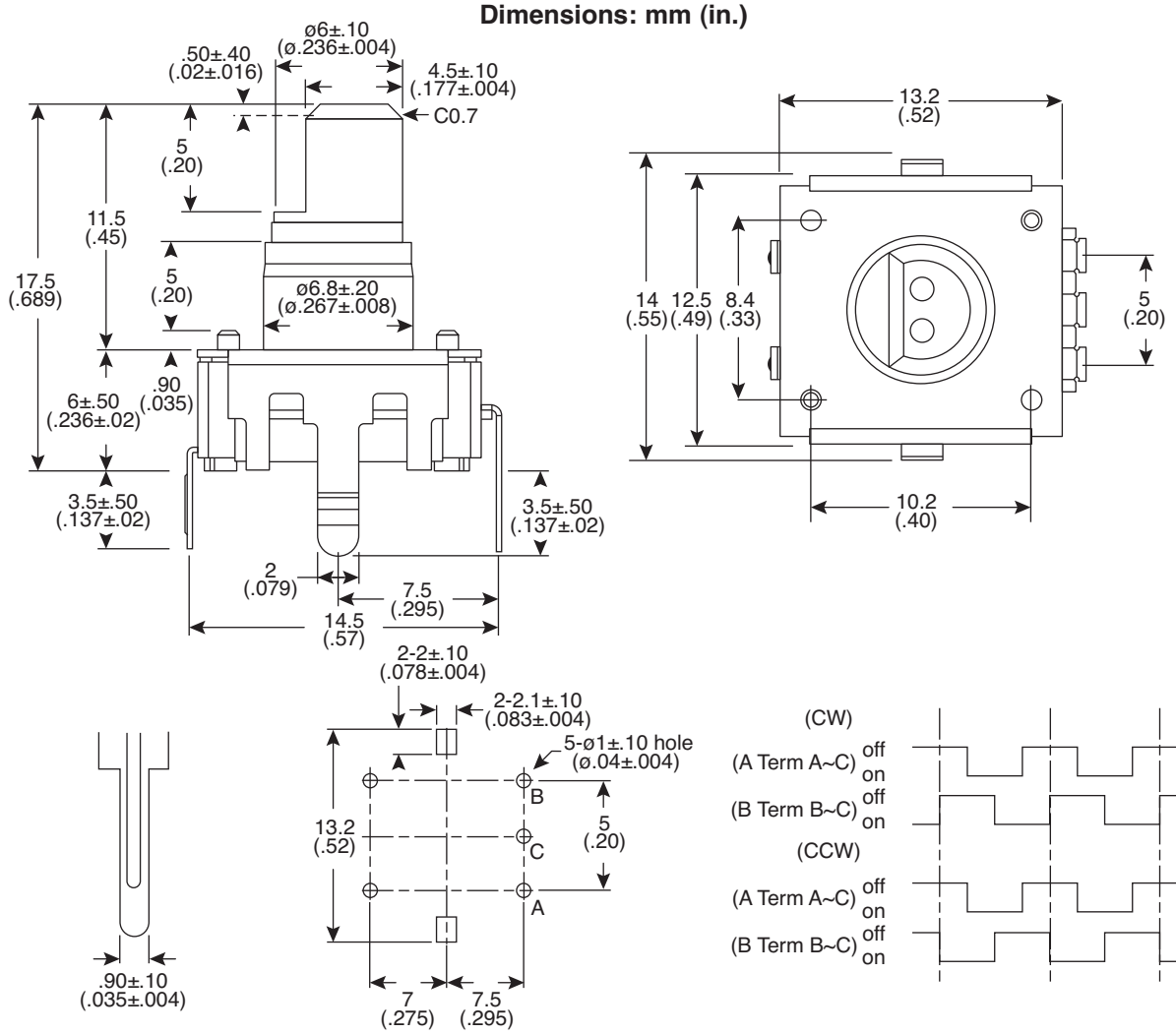
Date: 07-16-08 **Ref.#:** TW-700198

ALPHA

Rotary Encoder

318-ENC130175F-12PS

Dimensions: mm (in.)



Mechanical Specifications:

- Operating Temp: -10°C to 70°C
- Storage Temp: -40°C to 85°C
- Rotational Torque: 50gf.cm max.
- No. and Pos. of detents: 12 detents (Step angle 30°±3°)
- Terminal Strength: A static load for 300gf.cm shall be applied to the tip of the terminals for 10 sec. in any direction
- Shaft push-pull strength: 5.1kgf
- Rotational life: 30,000 cycles

Note:

- RoHS Compliant

Electrical Specifications:

- Rating: 1mA/10VDC
- Insulation Resistance: 50VDC 10MΩ Min.
- Dielectric Strength: 50VAC for 1 min.
- Resolution: 12 pulses/360° for each phase

Soldering Specifications:

- Soldering: To be performed in 5 seconds within 260±5°C
- Manual Soldering: To be performed in 3 seconds within 350±5°C
- Preheating: The entire flow duration should not exceed 2 min., and soldering surface temperature (undersurface of PCB) shall be settled within 100°C

Push-on Switch Specifications:

- Type: Single Pole Single Throw (Push on)
- Rating: 10mA/5VDC
- Switch Travel (mm): 0.5±0.4
- Operating Force: 200~460gf
- Operating Life: 20,000 times

ANEXO 5

Tablas de valores de las Resistencias SERIE E24, E48, E96

Valores normalizados de resistencia y capacitores

SERIE	E192	E96	E48	E24	E12	E6
TOLERANCIA	+/- 0,5%	+/- 1%	+/- 2%	+/- 5%	+/- 10%	+/- 20%

TABLA 1

E192	E96	E48	E24	E12	E6	E3	E192	E96	E48	E24	E12	E6	E3
1,00	1,00	1,00	1,0	1,0	1,0	1,0	1,01	-	-	-	-	-	-
1,02	1,02	-	-	-	-	-	1,04	-	-	-	-	-	-
1,05	1,05	1,05	-	-	-	-	1,06	-	-	-	-	-	-
1,07	1,07	-	-	-	-	-	1,09	-	-	-	-	-	-
1,10	1,10	1,10	1,1	-	-	-	1,11	-	-	-	-	-	-
1,13	1,13	-	-	-	-	-	1,14	-	-	-	-	-	-
1,15	1,15	1,15	-	-	-	-	1,17	-	-	-	-	-	-
1,18	1,18	-	-	-	-	-	1,20	-	-	1,2	1,2	1,2	-
1,21	1,21	1,21	-	-	-	-	1,23	-	-	-	-	-	-
1,24	-	-	-	-	-	-	1,26	-	-	-	-	-	-
1,27	1,27	1,27	-	-	-	-	1,29	-	-	-	-	-	-
1,30	1,30	-	1,3	-	-	-	1,32	-	-	-	-	-	-
1,33	1,33	1,33	-	-	-	-	1,35	-	-	-	-	-	-
1,37	1,37	-	-	-	-	-	1,38	-	-	-	-	-	-
1,40	1,40	1,40	-	-	-	-	1,42	-	-	-	-	-	-
1,43	1,43	-	-	-	-	-	1,45	-	-	-	-	-	-
E192	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**	**E192**	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**
1,47	1,47	1,47	-	-	-	-	1,49	-	-	-	-	-	-
1,50	1,50	-	1,5	1,5	1,5	-	1,52	-	-	-	-	-	-
1,54	1,54	1,54	-	-	-	-	1,56	-	-	-	-	-	-
1,58	1,58	-	-	-	-	-	1,60	-	-	1,6	-	-	-
1,62	1,62	1,62	-	-	-	-	1,64	-	-	-	-	-	-
1,65	1,65	-	-	-	-	-	1,67	-	-	-	-	-	-
1,69	1,69	1,69	-	-	-	-	1,72	-	-	-	-	-	-
1,74	1,74	-	-	-	-	-	1,76	-	-	-	-	-	-
1,78	1,78	1,78	-	-	-	-	1,80	-	-	1,8	1,8	1,8	-
1,82	1,82	-	-	-	-	-	1,84	-	-	-	-	-	-
1,87	1,87	1,87	-	-	-	-	1,89	-	-	-	-	-	-
1,91	1,91	-	-	-	-	-	1,93	-	-	-	-	-	-
1,96	1,96	1,96	-	-	-	-	1,98	-	-	-	-	-	-
2,00	2,00	-	2,0	-	-	-	2,03	-	-	-	-	-	-
2,05	2,05	2,05	-	-	-	-	2,08	-	-	-	-	-	-
E192	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**	**E192**	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**
2,10	2,10	-	-	-	-	-	2,13	-	-	-	-	-	-

2,15	2,15	2,15	-	-	-	-	2,18	-	-	-	-	-	-
2,21	2,21	-	-	-	-	-	2,23	-	-	-	-	-	-
2,26	2,26	2,26	-	-	-	-	2,29	-	-	-	-	-	-
2,32	2,32	-	-	-	-	-	2,34	-	-	-	-	-	-
2,37	2,37	2,37	-	-	-	-	2,40	-	-	2,4	-	-	-
-	-	-	-	-	-	-	2,43	2,43	-	-	-	-	-
2,46	-	-	-	-	-	-	2,49	2,49	2,49	-	-	-	-
2,52	-	-	-	-	-	-	2,55	2,55	-	-	-	-	-
2,58	-	-	-	-	-	-	2,61	2,61	2,61	-	-	-	-
2,64	-	-	-	-	-	-	2,67	2,67	-	-	-	-	-
-	-	-	2,7	2,7	-	-	2,71	-	-	-	-	-	-
2,74	2,74	2,74	-	-	-	-	2,77	-	-	-	-	-	-
2,80	-	-	-	-	-	-	2,84	-	-	-	-	-	-
2,87	2,87	2,87	-	-	-	-	2,91	-	-	-	-	-	-
2,94	2,94	-	-	-	-	-	2,98	-	-	-	-	-	-
-	-	-	3,0	-	-	-	3,01	3,01	3,01	-	-	-	-
3,05	-	-	-	-	-	-	3,09	3,09	-	-	-	-	-
3,12	-	-	-	-	-	-	3,16	3,16	3,16	-	-	-	-
3,20	-	-	-	-	-	-	3,24	3,24	-	-	-	-	-

TABLA 2

E192	E96	E48	E24	E12	E6	E3	E192	E96	E48	E24	E12	E6	E3
3,28	-	-	-	-	-	-	-	-	-	3,3	3,3	3,3	-
3,32	3,32	3,32	-	-	-	-	3,36	-	-	-	-	-	-
3,40	3,40	-	-	-	-	-	3,44	-	-	-	-	-	-
3,48	3,48	3,48	-	-	-	-	3,52	-	-	-	-	-	-
3,57	3,57	-	-	-	-	-	-	-	-	3,6	-	-	-
3,61	-	-	-	-	-	-	3,65	3,65	3,65	-	-	-	-
3,70	-	-	-	-	-	-	3,74	3,74	-	-	-	-	-
3,79	-	-	-	-	-	-	3,83	3,83	3,83	-	-	-	-
3,88	-	-	-	-	-	-	-	-	-	3,9	3,9	-	-
E192	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**	**E192**	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**
3,92	3,92	-	-	-	-	-	3,97	-	-	-	-	-	-
4,02	4,02	4,02	-	-	-	-	4,07	-	-	-	-	-	-
4,12	4,12	-	-	-	-	-	4,17	-	-	-	-	-	-
4,22	4,22	4,22	-	-	-	-	4,27	-	-	-	-	-	-
-	-	-	4,3	-	-	-	4,32	4,32	-	-	-	-	-
4,37	-	-	-	-	-	-	4,42	4,42	4,42	-	-	-	-

4,48	-	-	-	-	-	-	4,53	4,53	-	-	-	-	-
4,59	-	-	-	-	-	-	4,64	4,64	4,64	-	-	-	-
4,70	-	-	4,7	4,7	4,7	4,7-	4,75	4,75	-	-	-	-	-
4,81	-	-	-	-	-	-	4,87	4,87	4,87	-	-	-	-
4,93	-	-	-	-	-	-	4,99	4,99	-	-	-	-	-
5,05	-	-	-	-	-	-	-	-	-	5,1	-	-	-
5,11	5,11	5,11	-	-	-	-	5,17	-	-	-	-	-	-
5,23	5,23	-	-	-	-	-	5,30	-	-	-	-	-	-
5,36	5,36	5,36	-	-	-	-	5,42	-	-	-	-	-	-
E192	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**	**E192**	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**
5,49	5,49	-	-	-	-	-	5,56	-	-	-	-	-	-
-	-	-	5,6	5,6	-	-	5,62	5,62	5,62	-	-	-	-
5,69	-	-	-	-	-	-	5,76	5,76	-	-	-	-	-
5,83	-	-	-	-	-	-	5,90	5,90	5,90	-	-	-	-
5,97	-	-	-	-	-	-	6,04	6,04	-	-	-	-	-
6,12	-	-	-	-	-	-	6,19	6,19	6,19	-	-	-	-
-	-	-	6,2	-	-	-	6,26	-	-	-	-	-	-
6,34	6,34	-	-	-	-	-	6,42	-	-	-	-	-	-
6,49	6,49	6,49	-	-	-	-	6,57	-	-	-	-	-	-
6,65	6,65	-	-	-	-	-	6,73	-	-	-	-	-	-
-	-	-	6,8	6,8	6,8	-	6,81	6,81	6,81	-	-	-	-
6,90	-	-	-	-	-	-	6,98	6,98	-	-	-	-	-
7,06	-	-	-	-	-	-	7,15	7,15	7,15	-	-	-	-
7,23	-	-	-	-	-	-	7,32	7,32	-	-	-	-	-
7,41	-	-	-	-	-	-	7,50	7,50	7,50	7,5	-	-	-
7,59	-	-	-	-	-	-	7,68	7,68	-	-	-	-	-
E192	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**	**E192**	**E96**	**E48**	**E24**	**E12**	**E6**	**E3**
7,77	-	-	-	-	-	-	7,87	7,87	7,87	-	-	-	-
7,96	-	-	-	-	-	-	8,06	8,06	-	-	-	-	-
8,16	-	-	-	-	-	-	-	-	-	8,2	8,2	-	-
8,25	8,25	8,25	-	-	-	-	8,35	-	-	-	-	-	-
8,45	8,45	-	-	-	-	-	8,56	-	-	-	-	-	-
8,66	8,66	8,66	-	-	-	-	8,76	-	-	-	-	-	-
8,87	8,87	-	-	-	-	-	8,98	-	-	-	-	-	-
9,09	9,09	9,09	-	-	-	-	-	-	-	9,1	-	-	-
9,20	-	-	-	-	-	-	9,31	9,31	-	-	-	-	-
9,42	-	-	-	-	-	-	9,53	9,53	9,53	-	-	-	-
9,65	-	-	-	-	-	-	9,76	9,76	-	-	-	-	-
9,88	-	-	-	-	-	-	-	-	-	-	-	-	-

ANEXO 6

Código ecualizador de audio de 4 bandas

CODIGO ECUALIZADOR DE AUDIO DE 4 BANDAS

```
#include <Audio.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <SerialFlash.h>
#include <Bounce2.h>
#include <Encoder.h>


AudioPlaySdWav          playWav1;
AudioOutputI2S          audioOutput;
AudioConnection         patchCord1(playWav1, 0, audioOutput, 0);
AudioConnection         patchCord2(playWav1, 1, audioOutput, 1);
AudioControlSGTL5000    sgtl5000_1;

float a1;
float a2;
float a3;
float a4;
int current_value1;
int current_value2;
int current_value3;
int current_value4;


Encoder enc1(26, 25);
Encoder enc2(28, 27);
Encoder enc3(30, 29);
Encoder enc4(32, 31);

#define SDCARD_CS_PIN    BUILTIN_SDCARD
#define SDCARD_MOSI_PIN  11
#define SDCARD_SCK_PIN   13



void setup() {
  Serial.begin(9600);
  AudioMemory(8);

  sgtl5000_1.enable();
  sgtl5000_1.volume(0.5);
  sgtl5000_1.audioPostProcessorEnable();
  sgtl5000_1.eqSelect(3);
```

```
  SPI.setMOSI(SDCARD_MOSI_PIN);
  SPI.setSCK(SDCARD_SCK_PIN);
  if (!(SD.begin(SDCARD_CS_PIN))) {
    while (1) {
      Serial.println("Unable to access the SD card");
      delay(500);
    }
  }

}

void playFile(const char *filename)
{
  Serial.print("Playing file: ");
  Serial.println(filename);
  playWav1.play(filename);

  delay(5);

  while (playWav1.isPlaying()) {

                                //ENCODER1
    current_value1 = enc1.read();
      if (current_value1 < -10) {
      current_value1 = -10;
       enc1.write(current_value1);
  }
      else if (current_value1 > 10) {
      current_value1 = 10;
      enc1.write(current_value1);
  }

    a1 = current_value1/10;
    sgtl5000_1.eqBand(4,a1);

    Serial.print("ENCODER 1 \n");
    Serial.println(a1);


                                //ENCODER2

   current_value2 = enc2.read();
      if (current_value2 < -10) {
      current_value2 = -10;
       enc2.write(current_value2);
  }
```

```
   else if (current_value2 > 10) {
   current_value2 = 10;
   enc2.write(current_value2);
}

 a2 = current_value2/10;
 sgtl5000_1.eqBand(3,a2);

 Serial.print("ENCODER 2 \n");
 Serial.println(a2);


                          //ENCODER3

 current_value3 = enc3.read();
   if (current_value3 < -10) {
   current_value3 = -10;
    enc3.write(current_value3);
}
   else if (current_value3 > 10) {
   current_value3 = 10;
   enc3.write(current_value3);
}

 a3 = current_value3/10;
 sgtl5000_1.eqBand(1,a3);

 Serial.print("ENCODER 3 \n");
 Serial.println(a3);

                          //ENCODER4

 current_value4 = enc4.read();
   if (current_value4 < -10) {
   current_value4 = -10;
    enc4.write(current_value4);
}
   else if (current_value4 > 10) {
   current_value4 = 10;
   enc4.write(current_value4);
}

 a4 = current_value4/10;
 sgtl5000_1.eqBand(0,a4);

 Serial.print("ENCODER 4 \n");
 Serial.println(a4);
```

```
  }


  delay(100);


}


void loop() {
  playFile("SDTEST1.WAV");
  delay(10);
  playFile("SDTEST2.WAV");
  delay(1);
  playFile("SDTEST3.WAV");
  delay(1);
  playFile("SDTEST4.WAV");
  delay(1);
}
```