

DECIBELÍMETRO DIGITAL

con SD/MMC y vúmetro.

Agresta, Bruno
brunoagresta@gmail.com

Duje, Nicolás
nicoduje@hotmail.com

Moyano, Gabriel
vgmoyano@gmail.com

Mugnaini, Guillermo
mugnainiguillermo@gmail.com

28 de Noviembre de 2010

Resumen

En el siguiente informe el lector encontrará la metodología utilizada para la realización del decibelímetro. Al hablar de metodología nos referimos a la búsqueda de información, caminos elegidos, y el porque de todas las decisiones tomadas.

El proyecto es un decibelímetro con posibilidad de guardar los valores obtenidos en una memoria SD/MMC y mostrar los mismos por medio de un vúmetro de leds. El trabajo puede dividirse en 2 bloques, software y hardware. El microcontrolador utilizado es el LPC2124 de núcleo ARM7 fabricado por Philips Semiconductors. El hardware está compuesto por una serie de filtros analógicos conectados en cascada y un micrófono genérico de tipo electret, los cuales son los encargados de leer cuatro canales de diferente rango de frecuencias.

Por último, este decibelímetro surgió para ser presentado como proyecto final de 4^{to} año en la carrera de Ingeniería Electrónica de la Universidad Tecnológica Nacional - Facultad Regional Córdoba. En el mismo, aplicamos los conocimientos adquiridos en las materias de Electrónica Aplicada II, Técnicas Digitales II y Medidas Electrónicas I.

Capítulo 1

Introducción

1.1. Marco Teórico

1.1.1. Sonido

El sonido, en física, es cualquier fenómeno que involucre la propagación en forma de ondas elásticas (sean audibles o no), generalmente a través de un fluido (u otro medio elástico) que esté generando el movimiento vibratorio de un cuerpo.

El sonido humanamente audible consiste en ondas sonoras consistentes en oscilaciones de la presión del aire, que son convertidas en ondas mecánicas en el oído humano y percibidas por el cerebro. La propagación del sonido es similar en los fluidos, donde el sonido toma la forma de fluctuaciones de presión. En los cuerpos sólidos la propagación del sonido involucra variaciones del estado tensional del medio.

1.1.2. Presión Sonora

La presión sonora o acústica es producto de la propia propagación del sonido. La energía provocada por las ondas sonoras generan un movimiento ondulatorio de las partículas del aire, provocando la variación alterna en la presión estática del aire (pequeñas variaciones en la presión atmosférica. La presión atmosférica es la presión del aire sobre la superficie terrestre). La razón de estas variaciones de presión atmosférica es que se producen áreas donde se concentran estas partículas (zonas de concentración) y otras áreas quedan menos saturadas (zonas de rarefacción). Las zonas con mayor concentración de moléculas tienen mayor densidad y las zonas de menor concentración tienen menor densidad. Cuando estas ondas se encuentran en su camino con el oído la presión que ejercen sobre el mismo no es igual para toda la longitud de onda.

Así pues, la presión acústica queda definida como la diferencia de presión instantánea (cuando la onda sonora alcanza al oído) y la presión atmosférica estática.

La presión atmosférica se mide en pascales (Pa). En el SI (Sistema Internacional) 1 pascal es igual a una fuerza de 1 newton actuando sobre una superficie de 1 metro cuadrado. La presión atmosférica se sitúa en torno a los 100.000 Pa (estableciéndose como valor normalizado los 101.325 Pa).

Como en Pa las cifras obtenidas son muy grandes, normalmente, la presión atmosférica, se expresa en hectopascales hPa (igual dimensión que los milibares, que quedan en desuso) y se establecen 1.013 hPa como presión atmosférica normalizada a nivel del mar.

La presión sonora también se puede medir en pascales, no obstante, su valor es muy inferior al de la atmosférica. El umbral de dolor se sitúa en los 200 Pa,[1] mientras que el umbral de audición se sitúa en los 20 micropascales (correspondiente a 0 dB).

Además del pascal, para medir la presión sonora se utiliza el microbar (μbar), que es la millonésima parte del bar ($1 \text{ Pa} = 1 \text{ N/m}^2 = 10 \mu\text{bar}$ y $1 \mu\text{bar} = 10^{-6} \text{ bar}$).

La principal diferencia entre presión atmosférica y presión sonora es que, mientras que la presión atmosférica cambia muy lentamente, la presión sonora, alterna muy rápidamente entre valores negativos (menores que la presión atmosférica) y positivos (mayores). El número de veces que se repite un fenómeno por unidad de tiempo es lo que en física se denomina frecuencia. El hombre no tiene sensibilidad ante todas las frecuencias. El margen de frecuencias que pueden producir la sensación de sonido cuando impresiona el oído humano es lo que se conoce como audiofrecuencias y va de los 20 a los 20.000 Hz. No hay que confundir presión acústica

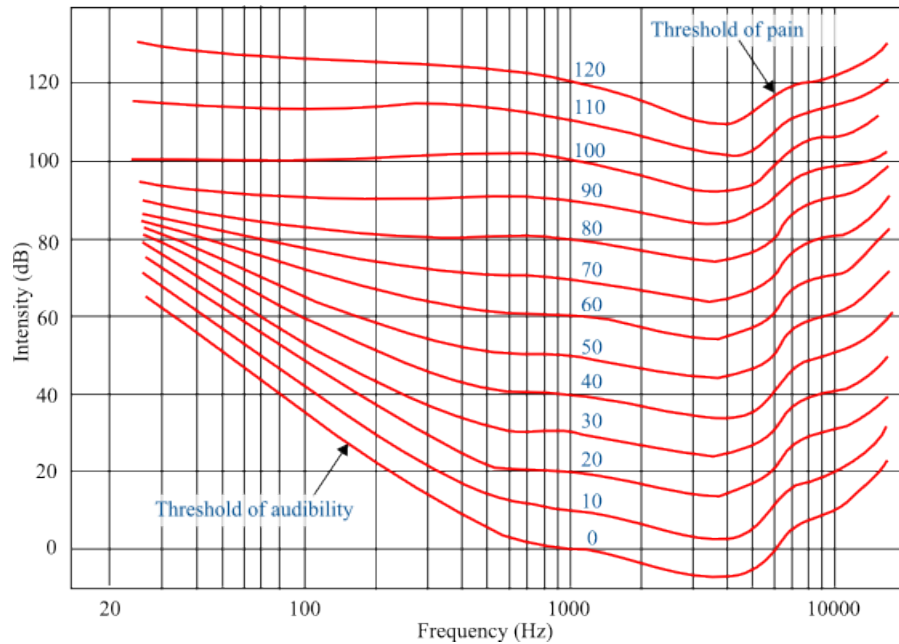


Figura 1.1: Curvas Isofónicas

con potencia acústica. La confusión viene por el hecho de que la presión sonora es la responsable directa de la amplitud de la onda y la amplitud determinara la cantidad de energía (potencia acústica) que contiene una señal sonora.

Para diferenciar entre sonidos más intensos (el oído soporta mayor cantidad de presión sonora), de sonidos débiles, se utiliza el llamado nivel de presión sonora.

1.1.3. Curvas Isofónicas

Las curvas isofónicas son curvas de igual sonoridad.

Estas curvas calculan la relación existente entre la frecuencia y la intensidad (en decibelios) de dos sonidos para que éstos sean percibidos como igual de fuertes, con lo que todos los puntos sobre una misma curva isofónica tienen la misma sonoridad.

Así, si 0 fon corresponden a una sonoridad con una intensidad de 0 dB con una frecuencia de 1 kHz, también una sonoridad de 0 fon podría corresponder a una sonoridad con una intensidad de 60 dB con una frecuencia de 70 Hz.

Las primeras curvas de igual sonoridad fueron establecidas por Fletcher y Munson en 1930.

En estas curvas isofónicas se observa como, a medida que aumenta la intensidad sonora, las curvas se hacen, cada vez, más planas. Esto se traduce en que la dependencia de la frecuencia es menor a medida que aumenta el nivel de presión sonora, lo que significa que si disminuye la intensidad sonora los últimos sonidos perceptibles en desaparecer serían los agudos (altas frecuencias).

Entre estos valores, se definen ciertas curvas:

- Curva A (dBA). Mide la respuesta del oído, ante un sonido de intensidad baja. Es la más semejante a la percepción logarítmica del oído humano, aunque los estudios de psicoacústica modernos cuestionan esta afirmación. Se utiliza para establecer el nivel de contaminación acústica y el riesgo que sufre el hombre al ser expuesto a la misma. Por ello, es la curva que se utiliza a la hora de legislar
- Curva B (dBB). Su función era medir la respuesta del oído ante intensidades para intensidades medias. Como no tiene demasiadas aplicaciones prácticas es una de las menos utilizadas. Muchos sonómetros no la contemplan
- Curva C (dBC). Mide la respuesta del oído ante sonidos de gran intensidad. Es tanto, o más empleada que la curva A a la hora de medir los niveles de contaminación acústica. También se utiliza para medir los sonidos más graves (entre 80dB y 110dB)

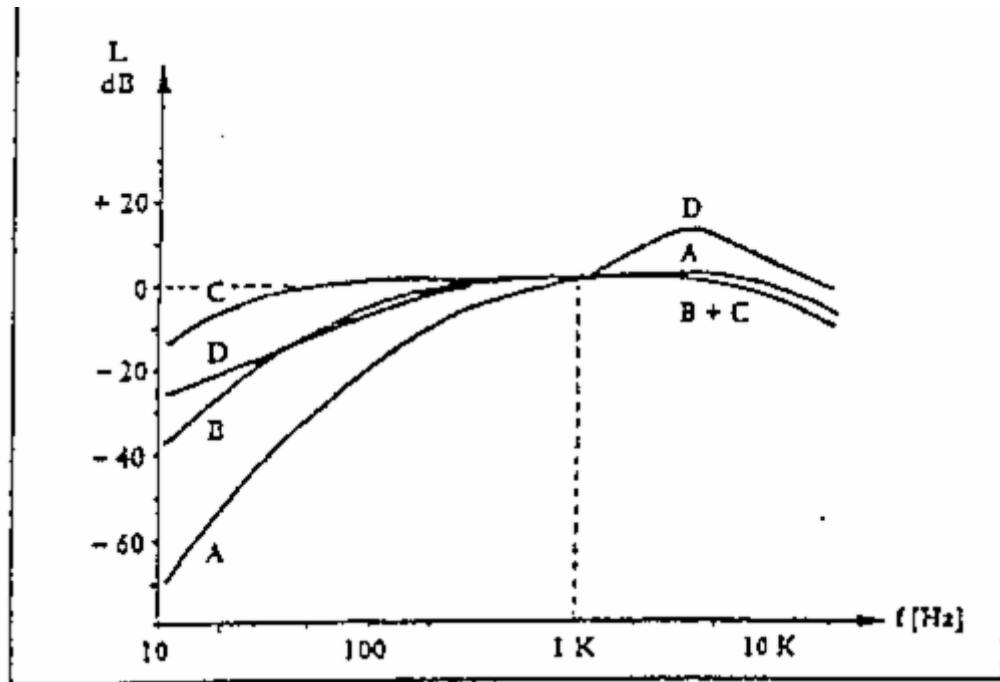


Figure 1.2: Curvas Isofonicas A, B, C, D

- Curva D (dBD). Se utiliza, casi exclusivamente, para estudiar el nivel de ruido generado por los aviones
- Curva U(dBU). Es la curva de más reciente creación y se utiliza para medir ultrasonidos, no audibles por los seres humanos.

Por ultimo, la curva de sonoridad del oido Nivel de presion sonora P esta dada en dB y se relaciona con la siguiente ecuacion:

$$dB = 20 \log \left(\frac{P_s [Pa]}{20 [\mu Pa]} \right)$$

Realizando un escalamiento con los niveles de tensión se pueden relacionar las tensiones obtenidas con las variaciones de presion sonora en dB

1.2. ¿Que es un Decibelímetro?

Existe una gran variedad de instrumentos para medir el ruido llamados decibelímetros o sonómetros. Cuando es necesario medir el promedio de nivel sonoro al que ha estado expuesto un individuo durante un tiempo determinado o para evaluar la exposición al ruido utilizamos los decibelímetros integradores. En el caso que se necesite conocer el espectro del ruido, es decir, la distribución del nivel de presión sonora en las diferentes frecuencias usamos el decibelímetro con banda de octavas.

El decibelímetro es un instrumento que permite medir el nivel de presión acústica, expresado en dB. Está diseñado para responder al sonido casi de la misma forma que le oído humano y proporcionar mediciones objetivas y reproducibles del nivel de presión acústica. En esencia, consta de un micrófono, una sección de procesamiento de señal y una unidad de lectura.

El funcionamiento de un decibelímetro consiste en un micrófono de medición que convierte la señal acústica en una señal eléctrica equivalente, la cual se procesa a través de amplificadores que adecuan la sensibilidad de la señal dentro del sistema de medición. Posteriormente la señal entra en un detector, cuya función es obtener los valores representativos de la señal. Seguidamente la señal es enviada a un convertidor lineal- logarítmico que permite la conversión de una escala lineal (presión en Pa) a una escala logarítmica (nivel de presión acústica, en dB), de modo que la tensión eléctrica de esta etapa es proporcional al nivel de presión considerado.

Cuando se requiere información más detallada de una señal compleja, puede dividirse el intervalo de frecuencia audible en bandas de frecuencia. Esto se realiza con filtros electrónicos para banda, los cuales rechazan señales que contengan frecuencias fuera de la banda seleccionada. Al proceso de dividir de esta manera la señal compleja se llama análisis de frecuencia. A los decibelímetros convencionales se les pueden adosar analizadores de bandas de octavas para obtener dicho propósito.

Capítulo 2

Modo de uso del Dispositivo

El decibelímetro tiene un modo de uso muy sencillo. Está pensado para funcionar como datalogger de forma autónoma, con la condición de que no se corte el suministro de energía eléctrica.

El usuario en un primer momento tendrá que configurar el instrumento. Las configuraciones a realizar son las siguientes:

1. Configurar fecha y hora por medio de una computadora conectada al instrumento. Para la conexión del instrumento con la PC es necesario el cable mostrado en la fig 2.1. Se puede acceder a la configuración con el programa Cutecom para linux. Configurando la velocidad en 9600 baudios, 8 bits de datos, sin bit de paridad y un bit de stop. Presionar la tecla "c", para entrar al menú de Configuración.



Figura 2.1: Cable para conexión con PC

2. Elección del modo de trabajo del mismo, ya sea como funcionamiento normal, funcionamiento sin vúmetro y utilización del vúmetro para mostrar el porcentaje ocupado en en la memoria. Esta configuración se realiza por medio del Dip-Switch.

Switch 1	Switch 2	Vúmetro	Porcentaje de Memoria
0	0	ON	OFF
0	1	OFF	ON
1	0	OFF	OFF
1	1	OFF	ON

Cuadro 2.1: Tabla de verdad del Dip-Switch

3. Colocar el decibelímetro en el lugar adecuado y solo acudir al mismo cuando lo crea conveniente.

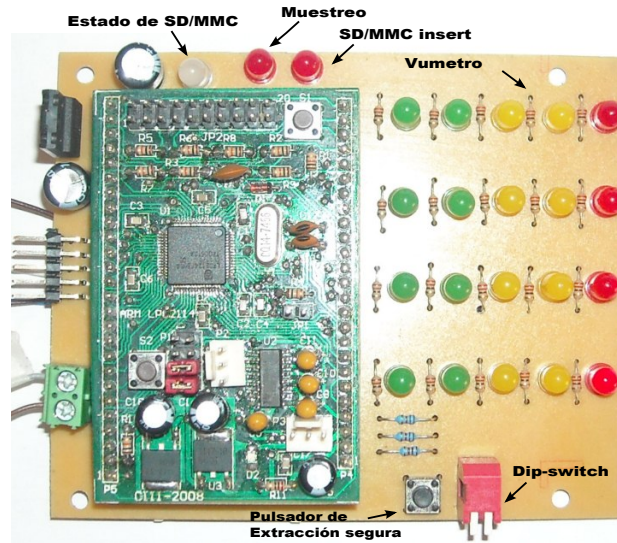


Figura 2.2: Fotografía de Placa de Adquisición

4. La extracción de la SD/MMC tiene que ser extraída de forma segura, para eso hay que presionar el pulsador de extracción segura y cuando el led bicolor pase a color rojo se puede extraer la SD. Para volver a insertarla, primero se coloca la SD en el conector adecuado y luego se presiona el mismo pulsador.
5. Para ver los valores medidos y grabados en la SD por el instrumento hay que recurrir a una PC. El decibelímetro creará un sistema de carpetas para organizar la información como muestra la siguiente imagen.

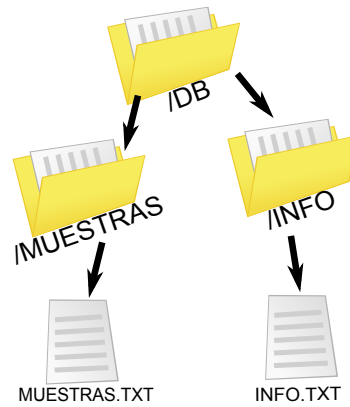


Figura 2.3: Árbol de carpetas.

En el archivo Muestras.txt se encuentran seis columnas, donde la primera es el ID de cada medición, la segunda es la fecha y hora de la misma y las cuatro restantes son las medidas de cada canal respectivamente. Si el usuario lo desea, puede exportar este archivo a un archivo .xls (Microsoft Excel) y generar su propio gráfico.

Capítulo 3

Software

3.1. Microcontrolador y Herramienta de Programación

El microcontrolador utilizado es un LPC2124 basado en un núcleo ARM7TDMI-STM fabricado por Philips Semiconductors. Algunas características del mismo son:

- 16/32-bits
- 64 pines.
- 16 kB de memoria RAM estática.
- 256 kB de memoria flash de programa.
- 4 canales de A/D de 10 bits cada uno.

En el año 2006 Philips Semiconductors cambia de nombre por NXP Semiconductors, esta nueva empresa continúa fabricando microcontroladores LPC2124, pero les agrega una sufijo (01 / 02) para diferenciarlos de los anteriores. Estos nuevos LPC2124/01/02 tienen algunas diferencias con los anteriores, por lo que hay que estar atento.

Por otro lado, para la programación del microcontrolador se utilizó el proyecto GNU-ARM bajo Linux que es una herramientas de programación libre. Para su uso se dispone de un instalador que se podrá descargar de la página:

“<http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/HerramientasProgramacion>”, proveyendo así todas las herramientas necesarias para construir un proyecto.

3.2. Firmware del microcontrolador.

Al final del trabajo, se encuentra la sección Bibliografía y en ella el lector encontrará las direcciones de páginas de Internet utilizadas para el software. No se explicará en detalle todo el programa utilizado en microcontrolador, debido a que escapa el objetivo de este informe, si se desea ampliar la información consultar con las referencias de la Bibliografía.

3.2.1. Main

El main.c es el programa principal en el cual incluimos una serie de librerías, las cuales nombramos y explicamos más abajo. Este archivo es creación nuestra y contiene todas las declaraciones de variables y funciones que permiten la interconexión de todas las librerías utilizadas para el proyecto.

Para simplificar la interpretación del programa se creó un archivo llamado integer.h en el cual están todas las definiciones .

El programa fue comentado donde era necesario para que el lector pueda entenderlo rápidamente y así como nosotros utilizamos librerías de otras personas, poder compartir la nuestro proyecto de forma clara y ordenada.

Las funciones dentro del main son las siguientes:

- void spi_init(void): Inicialización de módulo SPI (Serial Peripheral Interface).
- void LogFres(FRESULT): Muestra en el log el resultado de una operación con archivos.
- void LogDres(DRESULT): Muestra en log el resultado de una operación de conexión.
- void LogClusterFree(FATFS *): Función para mostrar el porcentaje ocupado de la memoria, como así también los Kb libres y totales.
- void Interrupción_10ms(): Interrupción que se realiza con el timer cada 10ms., para el uso de la memoria.
- void Interrupción_50ms(): Interrupción que se realiza con el timer cada 50ms. Barre lso canales del ADC, promedia, actualiza leds y lee los switches.
- void imprimir_hora () __attribute__ ((interrupt,noinline)): Interrupción que se realiza con el RTC cada un segundo. Guarda la muestra en la SD y imprime en el log la muestra tomada.
- void ExtraccionSegura(void): Cierra el entorno de trabajo y el archivo en donde se están almacenando las muestras para extraer la memoria de forma segura.
- void perifericos_init(void): Configura los pines como GPIO para poder utilizar los switches y leds.
- void sd_init(void): Inicializa la memoria, crea carpetas y el archivo info.txt. Abre la corriente del archivo muestras.txt para poder guardar la información obtenida.
- while(true): Bucle infinito, se encarga del manejo del menú de configuración.

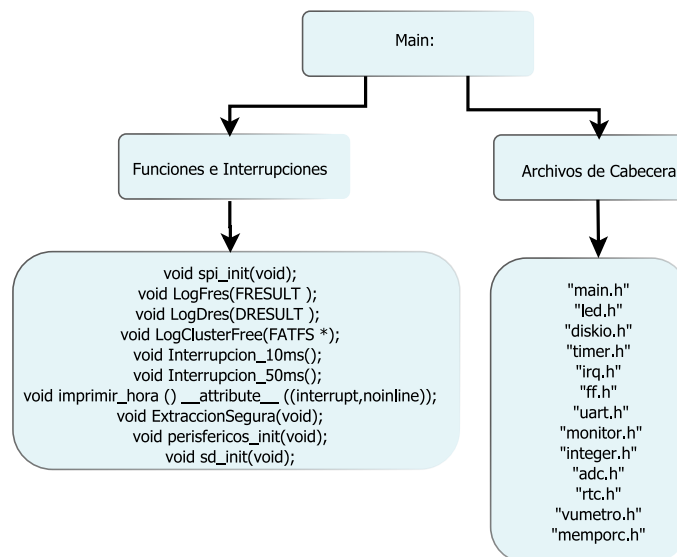


Figura 3.1: Diagrama general del archivo Main.

3.2.2. UART

En primer lugar se creó una función para la UART con el fin de poder depurar todo lo realizado con el microcontrolador. Luego, esta función fue de mucha utilidad para poder configurar la fecha y hora del instrumento. Este archivo llamado uart.c fue provisto por la Wiki de Técnicas Digitales 2 y modificado por nosotros. Fue enlazado con otro archivo llamado monitor.c obtenido de una página web, su creador es el Sr.ChaN, este código permite utilizar funciones de “printf(“.”)”, de manera que se hace más fácil la depuración del programa, teniendo en cuenta que no tuvimos la posibilidad de obtener un depurador Jtag.

La librería uart.c es la encargada de inicializar y configurar la UART, de enviar y recibir un Byte y por medio de la librería monitor.c transformamos los datos obtenidos a variables, las cuales podemos utilizar dentro de nuestro programa principal. La librería otorgada por ChaN también nos permite enviar y recibir tanto un solo caracter como una cadena. Utilizamos la UART, en el funcionamiento normal del decibelímetro para realizar un log del avance del programa y configurar la fecha y hora del reloj de tiempo Real, con un sencillo menú.

3.2.3. A/D

El software para el conversor analógico - digital utiliza los cuatro canales de 10 bits provistos por el microcontrolador. Para la lectura de los mismo se utiliza un archivo llamado `adc.c` que se basa en el funcionamiento descrito en la Wiki de Técnicas Digitales 2, y este tiene las funciones para inicializar y leer el canal del ADC deseado.

3.2.4. RTC

El microcontrolador posee un Reloj de Tiempo Real internamente, este nos sirvió para saber en que momento (fecha y hora) se estaba tomando la muestra. El mismo tiene la posibilidad de crear alarmas e interrupciones, las cuales no son utilizadas por en este programa.

El archivo `rtc.c` es el encargado de inicializar el RTC y en el mismo se encuentran 2 funciones, una para configurar y otra para obtener la fecha y hora.

Esta librería fue provista por Sr. ChaN la cual modificamos a nuestra conveniencia.

3.2.5. Interrupciones

La librería encargada de las interrupciones fue obtenida de la página del CIII y se llama `irq.c`. Este archivo inicializa, habilita y deshabilita las interrupciones vectorizadas deseadas, las cuales tienen que ser pasadas como parámetro de la función.

En el programa hay 2 interrupciones, por desborde del timer, una cada 10ms y otro cada 50ms.

3.2.6. SD/MMC

La conexión de la SD puede ser de forma nativa o por medio del protocolo SPI, nosotros hemos elegido la segunda opción. La inicialización de este modo se realiza en el archivo `main.c` dentro de una función.

Las librerías utilizadas para la SD/MMC es la `sdmmc.c` y la `ff.c`, las cuales fueron obtenidas de la página del Sr.ChaN.

Para comprender un poco el sistema utilizado para guardar archivos en la memoria, podemos dividir el problema en tres partes o niveles.

1. El primer nivel es el SPI, maneja el módulo interno del microcontrolador para realizar una comunicación serie con la memoria. Este nivel es dependiente del microcontrolador utilizado y si se desea migrar este proyecto a uno con diferente microcontrolador se deberá cambiar este nivel.
2. El segundo nivel y colocado por encima del anterior es el manejado por la librería `sdmmc.c`, el cual es el encargado de inicializar la SD/MMC, la memoria utiliza ciertos comandos para ser inicializada en modo spi. Cada versión de SD posee diferentes comandos y lo que realizamos en este nivel es manejar todos estos comandos para poder acceder a la memoria y poder guardar datos.
3. El tercer nivel es el sistema de archivos FAT, que utiliza una tabla de asignación de archivos. Este nivel fue necesario para poder leer la SD fuera del instrumento y que el tipo de archivo sea de un tipo estándar. Esta librería es de alto nivel, pudiendo ser aplicada a un PenDrive, a una memoria SD/MMC o a un disco duro. El archivo encargado de manejar este nivel es el `ff.c` y por ser de alto nivel es independiente de la plataforma utilizada. Esta librería tiene la posibilidad de trabajar con FAT16 o FAT32.

Capítulo 4

Hardware

4.1. Placa ARM 2009

La Cooperativa de Estudiantes de Electrónica (CEE) diseñó, fabricó y vendió a los alumnos la Placa ARM 2009, la cual adquirimos.

Ésta, contiene el microcontrolador LPC2124 y varios componentes para programarlo. Tiene conexión con la uart y varios reguladores para no dañar el mismo.

Por otro lado, tiene todos los pines del dispositivo conectados a 2 conectores tipo tira de pines macho 1X25 y otro 1X26, los cuales utilizamos para adjuntar esta placa a la de nuestro decibelímetro.

4.2. Placa Decibelímetro

La placa del decibelímetro en sí, es la placa ARM 2009 más un conjunto de leds y otros componentes. El vúmetro contiene cinco leds por cada uno de los cuatro canales, lo que suma un total de veinte. Por otro lado, en esta placa hay conectados un regulador de tensión, un pulsador para extracción segura de la SD y un Dip-Switch para configuración del decibelímetro. Por último, en un costado de la misma se encuentran tres leds utilizados para indicar al usuario, si el decibelímetro está midiendo, está grabando en la SD y si el ultimo indica la inserción de la memoria.

4.2.1. Esquemático Placa Decibelímetro

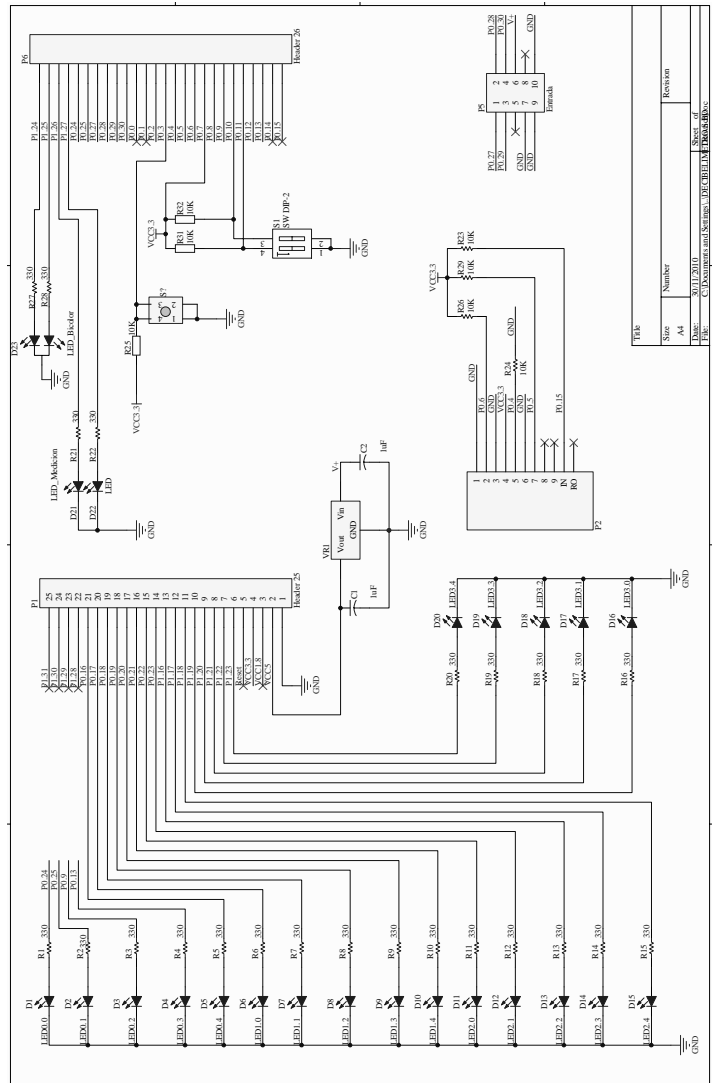


Figura 4.1: Esquemático Placa Decibelímetro

4.2.2. Top Layer

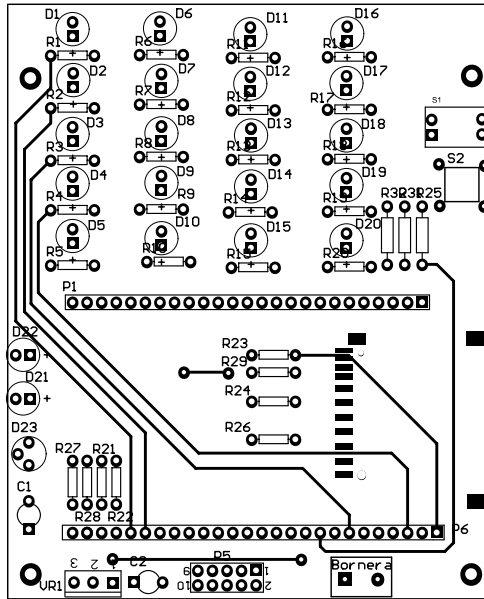


Figura 4.2: Top Placa Decibelímetro

4.2.3. Bottom Layer

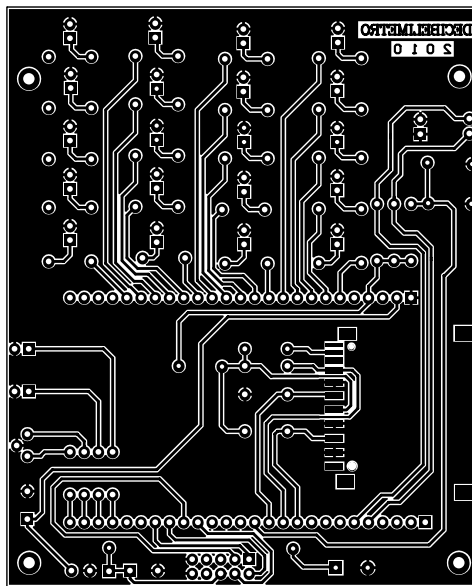


Figura 4.3: Bottom Placa Decibelímetro

4.3. Placa Filtros

En el uso del decibelímetro es importante poder discernir a que frecuencias o que rango de las mismas corresponde la medición realizada. Para llevar a cabo lo dicho anteriormente se pueden elegir dos opciones, la primera es utilizar el algoritmo de FFT (Transformada Rápida de Fourier) en el microcontrolador y la segunda es realizar varios filtros pasa banda. En nuestro decibelímetro se opto por realizar la segunda opción.

Los anchos de banda elegidos son de 64Hz-250Hz, 250Hz-2kHz, 2kHz-4kHz y 4kHz-8kHz. Esta elección tiene una justificación. El primer ancho de banda (64Hz-250Hz) corresponde a la tercera y cuarta octava donde se perciben los tonos graves y medios. En el segundo (250Hz-2kHz) correspondiente a la quinta, sexta y séptima octava, se encuentran los tonos fundamentales y primeros armónicos de la mayoría de las fuentes sonoras. El tercero (2kHz-4kHz), siendo la octava número ocho, comprende el margen en que el oído humano tiene mayor sensibilidad. Finalmente el último ancho de banda elegido (4kHz-8kHz), correspondiente a la novena octava, contiene una importante parte de los tonos más agudos del espectro audible.

4.3.1. Elección de parámetros y topología a realizar

4.3.1.1. Tipo y orden de los filtros

Para la aplicación se ha decidido llevar a cabo los filtros mediante la aproximación de Butterworth. Esta elección se ve justificada con el hecho de que esta aproximación posee una banda de paso plana y una buena pendiente en la banda de transición. Otras aproximaciones tienen algunas características mejores que la utilizada pero en detrimento de otras, por ejemplo la aproximación de Bessel tiene una respuesta de fase mejor que la de Butterworth pero tiene una pendiente más chica en la banda de transición, y la aproximación de Chebyshev tiene una mejor pendiente en la banda de transición pero aparece un rizo en la banda de paso.

Debido a que la aproximación de Butterworth tiene una pendiente menor que la de Chebyshev con el mismo orden, se decidió realizar cada filtro pasa banda de octavo orden. Además de esta forma, como con un amplificador operacional se puede lograr un filtro de segundo orden como máximo, se utiliza un circuito integrado con cuatro amplificadores por cada filtro.

4.3.1.2. Topología

La restricción de orden del filtro por amplificador operacional nos lleva a realizar cuatro filtros y conectarlos en cascada para lograr un filtro de octavo orden.

De las topologías existentes las que mejores se adaptan a nuestros requerimientos son las llamadas Sallen-Key (o filtros KRC) y Realimentación Múltiples. Esta última a diferencia de la primera aprovecha toda la ganancia de lazo abierto. Este motivo más el hecho de que la topología de Realimentación Múltiples es menos sensible a la variación de los componentes utilizados que la Sallen-Key, nos llevó a elegir como topología a utilizar la de Realimentación Múltiples. Aunque si bien lo dicho hasta ahora es todo en favor de la topología de Realimentación Múltiples, esta posee la desventaja de utilizar más componentes.

4.3.2. Cálculos de los filtros

Debido a que varios de los cálculos a realizar eran similares se empleó la herramienta de cálculo MatLab[®] mediante la utilización de scripts para la optimización de los mismos. Estos scripts fueron realizados por nosotros, por lo que se amoldan perfectamente a nuestras necesidades. Para el caso del cálculo de las resistencias necesarias antes de mostrar los scripts utilizados se desarrollará el cálculo necesario a realizar, porque consideramos que es más fácil para la interpretación del mismo, además que de esta forma no se pierde la teoría que conlleva el mismo. Luego de expuesto el script se mostrará un ejemplo de la devolución del script, seguido de una tabla con los valores obtenidos para cada filtro.

4.3.2.1. Funciones de Transferencia

Antes de mostrar el script utilizado en este caso, se describirá lo que realiza el mismo.

La herramienta de cálculo MatLab[®] nos provee de una función que nos devuelve la función de transferencia de un filtro, según la aproximación de Butterworth, indicándole el orden y las pulsaciones de corte del mismo. En base a esto, se factoriza la función de transferencia obtenida en lo que serían las cuatro diferentes etapas del filtro para poder llevarla a cabo con amplificadores operacionales. Luego de esto se calcula la pulsación angular y la selectividad de cada etapa. Finalmente se obtiene la ganancia a frecuencias medias y se muestra un gráfico de bode de la función de transferencia total.

Script FiltroButtPB8 - %FiltroButtPB8 realiza un filtro pasa banda de orden 8 según la aproximación
- %Butterworth. Además divide el filtro en cuatro etapas pasa banda para
- %poder ser sintetizado con amplificadores operacionales.

```

fprintf('\n—Cálculo de un filtro Butterworth Pasa Banda de 8to orden—\n\n')
fci=input('Ingrese el valor de la frecuencia de corte inferior(fci)\n');
fcs=input('Ingrese el valor de la frecuencia de corte superior(fcs)\n');
n=4; %orden del polo
wn=[2*pi*fci 2*pi*fcs]; %cálculo de las pulsaciones de corte
[z,p,k]=butter(n,wn,'s'); %cálculo de los ceros(z), polos(p) y ganancia(k)
[b,a]=butter(n,wn,'s'); %cálculo del numerador(b) y el denominador(a) de la función de transferencia del
filtro
num=[b(5)^(1/4) 0]; %cálculo del nuevo numerador para c/etapa del filtro
p1=[p(1) p(2)]; %División de los polos para c/etapa del filtro
p2=[p(3) p(4)];
p3=[p(5) p(6)];
p4=[p(7) p(8)];
dem1=poly(p1); %cálculo de los denominadores para c/etapa del filtro
dem2=poly(p2);
dem3=poly(p3);
dem4=poly(p4);
HoBP1=num(1)/dem1(2); %cálculo de la ganancia a frecuencias medias de c/etapa
HoBP2=num(1)/dem2(2);
HoBP3=num(1)/dem3(2);
HoBP4=num(1)/dem4(2);
fprintf('\n—————Etapa_1—————')
filtro1=tf(num,dem1); %se obtiene la función de transferencia de la etapa
filtro1=filtro1/HoBP1 %y se le extrae la ganancia a frecuencias medias
fprintf('-Pulsación angular central de la etapa-')
wo1=dem1(3)^(1/2)
fprintf('-Selectividad de la etapa-')
Q1=wo1/dem1(2)
fprintf('\n—————Etapa_2—————')
filtro2=tf(num,dem2); %se obtiene la función de transferencia de la etapa
filtro2=filtro2/HoBP2 %y se le extrae la ganancia a frecuencias medias
fprintf('-Pulsación angular central de la etapa-')
wo2=dem2(3)^(1/2)
fprintf('-Selectividad de la etapa-')
Q2=wo2/dem2(2)
fprintf('\n—————Etapa_3—————')
filtro3=tf(num,dem3); %se obtiene la función de transferencia de la etapa
filtro3=filtro3/HoBP3 %y se le extrae la ganancia a frecuencias medias
fprintf('-Pulsación angular central de la etapa-')
wo3=dem3(3)^(1/2)
fprintf('-Selectividad de la etapa-')
Q3=wo3/dem3(2)
fprintf('\n—————Etapa_4—————')
filtro4=tf(num,dem4); %se obtiene la función de transferencia de la etapa
filtro4=filtro4/HoBP4 %y se le extrae la ganancia a frecuencias medias
fprintf('-Pulsación angular central de la etapa-')
wo4=dem4(3)^(1/2)
fprintf('-Selectividad de la etapa-')
Q4=wo4/dem4(2)
fprintf('\n————Ganancia a frecuencias medias del Filtro para obtener en la banda de paso 0dB————')
HoBP=HoBP1*HoBP2*HoBP3*HoBP4 %cálculo de la ganancia a frecuencias medias del filtro
bode(HoBP*filtro1*filtro2*filtro3*filtro4) %bode de la función de transferencia del filtro
clear a b dem1 dem2 dem3 dem4 fci fcs k n num p p1 p2 p3 p4 wn z HoBP1 HoBP2 HoBP3 HoBP4

```

Ejemplo de devolución del script FiltroButtPB8

Filtro 64Hz-250Hz >> FiltroButtPB8

—Cálculo de un filtro Butterworth PasaBanda de 8to orden—

Ingrese el valor de la frecuencia de corte inferior(fci)

64

Ingrese el valor de la frecuencia de corte superior(fcs)

250

-----Etapa_1-----

Transfer function:

703.2 s

 $s^2 + 703.2 s + 2.323e006$

-Pulsación angular central de la etapa-

wo1 =

1.5241e+003

-Selectividad de la etapa-

Q1 =

2.1673

-----Etapa_2-----

Transfer function:

1450 s

 $s^2 + 1450 s + 1.292e006$

-Pulsación angular central de la etapa-

wo2 =

1.1366e+003

-Selectividad de la etapa-

Q2 =

0.7837

-----Etapa_3-----

Transfer function:

709.2 s

 $s^2 + 709.2 s + 3.089e005$

-Pulsación angular central de la etapa-

wo3 =

555.7609

-Selectividad de la etapa-

Q3 =

0.7837

-----Etapa_4-----

Transfer function:

191.2 s

 $s^2 + 191.2 s + 1.718e005$

-Pulsación angular central de la etapa-

wo4 =

414.4463

-Selectividad de la etapa-

Q4 =

2.1673

-----Ganancia a frecuencias medias del Filtro para obtener en la banda de paso 0dB-----

HoBP =

13.4873

Filtro	Etapa 1		Etapa 2		Etapa 3		Etapa 4		Ho
	Q	ω [rad/s]	Q	ω [rad/s]	Q	ω [rad/s]	Q	ω [rad/s]	
64Hz-250Hz	2.1673	1.5241e+003	0.7837	1.1366e+003	0.7837	555.7609	2.1673	414.4463	13.4873
250Hz-2kHz	1.6491	1.2269e+004	0.5882	9.9700e+003	0.5882	1.9799e+003	1.6491	1.6088e+003	35.2937
2kHz-4kHz	3.8908	2.4564e+004	1.5463	2.0493e+004	1.5463	1.5411e+004	3.8908	1.2857e+004	9.0492
4kHz-8kHz	3.8908	4.9128e+004	1.5463	4.0986e+004	1.5463	3.0823e+004	3.8908	2.5715e+004	9.0492

Cuadro 4.1: Valores Obtenidos para cada filtro

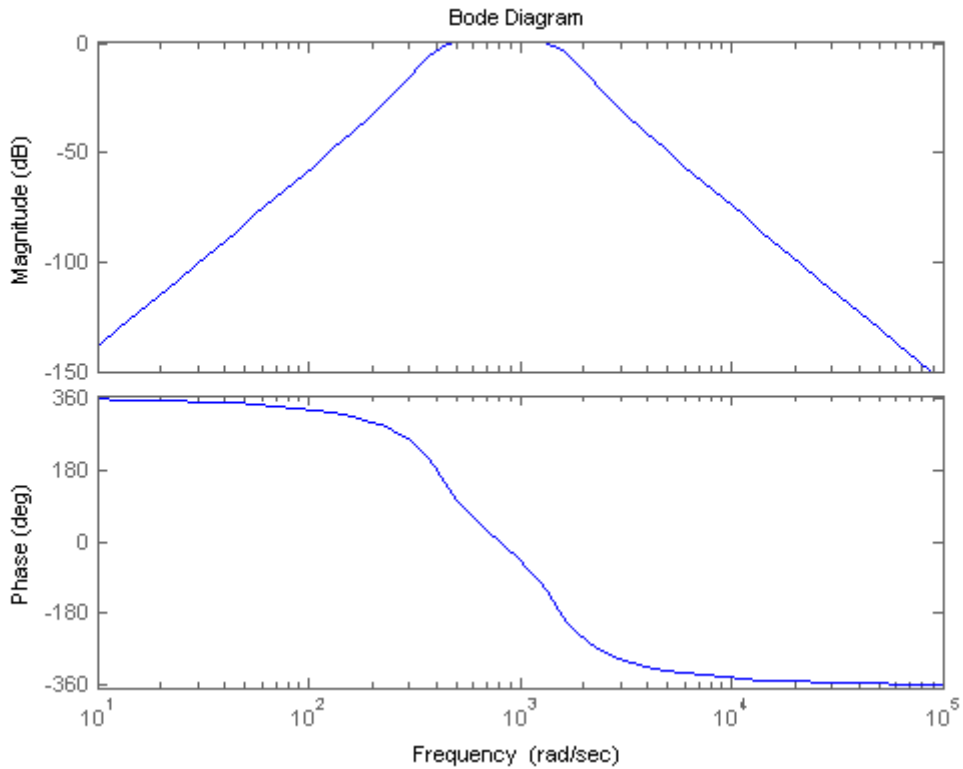


Figura 4.4: Bode filtro 64Hz-250Hz

Tabla de Valores Obtenidos A continuación se muestran en una tabla los valores obtenidos anteriormente.

4.3.2.2. Cálculo de Resistencias

Como se menciono anteriormente, antes de mostrar el correspondiente script para el cálculo de las resistencias, se expondrán los cálculos necesarios a realizar para la topología de Múltiple Realimentación.

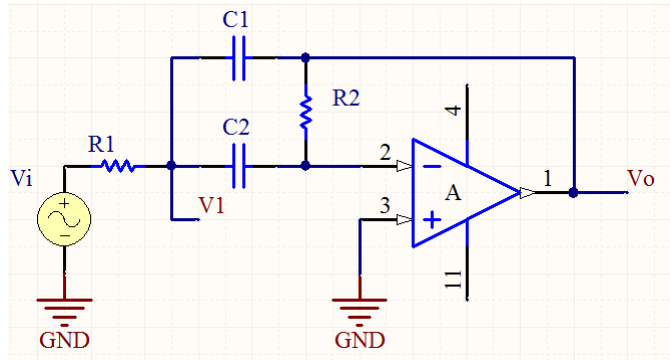


Figura 4.5: Circuito de Múltiple Realimentación

Topología de Múltiple Realimentación Del circuito de la figura se puede interpretar que el amplificador operacional actúa como una diferenciador con respecto a $V1$:

$$V_o = -\frac{R2}{\frac{1}{s \cdot C2}} \cdot V1 = -s \cdot R2 \cdot C2 \cdot V1$$

Al sumar las corrientes en el nodo $V1$:

$$\frac{V_i - V1}{R1} + \frac{V_o - V1}{1/sC1} + \frac{0 - V1}{1/sC2} = 0$$

Despejando $V1$ de la primera ecuación, reemplazando en la segunda y acomodando:

$$H(j\omega) = \frac{V_o}{V_i} = \frac{-j\omega \cdot R2 \cdot C2}{1 - \omega^2 \cdot R1 \cdot R2 \cdot C1 \cdot C2 + j\omega \cdot R1 \cdot (C1 + C2)}$$

Para pasar esta función a la forma estándar de $H(j\omega) = H_{0BP} \cdot H_{BP}(j\omega)$:

$$H(j\omega) = H_{0BP} \cdot \frac{(j\omega/\omega_0)/Q}{1 - (\omega/\omega_0)^2 + (j\omega/\omega_0)/Q}$$

Se debe igualar:

$$\omega^2 \cdot R1 \cdot R2 \cdot C1 \cdot C2 = (\omega/\omega_0)^2$$

Despejando de esta última ecuación ω_0 :

$$\omega_0 = \frac{1}{\sqrt{R1 \cdot R2 \cdot C1 \cdot C2}}$$

Además también se iguala:

$$j\omega \cdot R1 \cdot (C1 + C2) = (j\omega/\omega_0)/Q$$

Luego se despeja Q :

$$Q = \frac{\sqrt{R2/R1}}{\sqrt{C1/C2} + \sqrt{C2/C1}}$$

Por último se iguala:

$$-j\omega \cdot R2 \cdot C2 = H_{0BP} \cdot (j\omega/\omega_0)/Q$$

Despejando H_{0BP} y reemplazando los valores obtenidos de Q y ω_0 :

$$H_{0BP} = \frac{-R2/R1}{1 + C1/C2}$$

Para el cálculo se hace $C1 = C2 = C$, con lo cual las expresiones anteriores quedan como:

$$\omega_0 = \frac{1}{\sqrt{R1 \cdot R2 \cdot C}}$$

$$Q = 0,5 \cdot \sqrt{R2/R1}$$

$$H_{0BP} = -2 \cdot Q^2$$

De estas ecuaciones se pueden despejar R1 y R2:

$$R1 = \frac{1}{2 \cdot \omega_0 \cdot Q \cdot C}$$

$$R2 = \frac{2 \cdot Q}{\omega_0 \cdot C}$$

Si se desea que $H_0 < 2 \cdot Q^2$ se debe reemplazar la resistencia R1 por un divisor resistivo como se indica a continuación (utilizando el teorema de Thevenin):

$$A_{nueva} = \frac{R1B}{R1A + R1B} \cdot A_{antigua}$$

$$R1 = \frac{R1A \cdot R1B}{R1A + R1B}$$

Despejando R1A y R1B se obtiene:

$$R1A = R1 \cdot \frac{A_{antigua}}{A_{nueva}}$$

$$R1B = \frac{1}{1 - A_{nueva}/A_{antigua}}$$

Los valores de ganancias son:

$$A_{antigua} = 2 \cdot Q^2$$

$$A_{nueva} = H_0$$

De donde H_0 es el valor entregado por el script FiltroButtPB8. En base a lo anteriormente expuesto se puede finalizar mostrando las ecuaciones para el cálculo de R1A y R1B:

$$R1A = \frac{Q}{H_0 \cdot \omega_0 \cdot C}$$

$$R1B = \frac{R1A}{\frac{2 \cdot Q^2}{H_0} - 1}$$

Finalmente, y antes de mostrar el script, se hace notar que el mismo realiza el cálculo de R1A y R1B solo en el caso de que sea necesario.

Script CalcR_MFeedback - %CalcR_MFeedback calcula las resistencias necesarias para los valores
- %entregados por FiltroButtPB8 y los capacitores indicados por el usuario,
- %para de esta forma realizar el circuito(filtro pasa banda) con amplificadores
- %operacionales según la configuración de Múltiple realimentación.

```
fprintf('\n\n\n—Cálculo de las R para Filtros con AO según Realimentación Multiple—\n\n')
fprintf('\n—————Etapa_1—————')
C_1=input('\ningrese el valor del capacitor a utilizar\n');
R1_1=1/(2*Q1*C_1*wo1)
R2_1=(2*Q1)/(C_1*wo1)
fprintf('\n—————Etapa_2—————')
C_2=input('\ningrese el valor del capacitor a utilizar\n');
R1_2=1/(2*Q2*C_2*wo2)
R2_2=(2*Q2)/(C_2*wo2)
fprintf('\n—————Etapa_3—————')
C_3=input('\ningrese el valor del capacitor a utilizar\n');
R1_3=1/(2*Q3*C_3*wo3)
R2_3=(2*Q3)/(C_3*wo3)
fprintf('\n—————Etapa_4—————')
C_4=input('\ningrese el valor del capacitor a utilizar\n');
Hactual=2^3*Q1^2*Q2^2*Q3^2;
if((Hactual*2*Q4^2)>HoBP) %si la ganancia total del filtro es más grande de la que necesito, la última
etapa debera atenuar
HoBP4=HoBP/Hactual; %cálculo de la atenuación necesaria
R1A_4=Q4/(HoBP4*wo4*C_4)
R1B_4=R1A_4/(2*Q4^2/HoBP4-1)
else
fprintf('\nEl circuito necesitará una amplificador de ganancia:')
Hnesaria=HoBP/(Hactual*2*Q4^2)
R1_4=1/(2*Q4*C_4*wo4)
end
R2_4=(2*Q4)/(C_4*wo4)
```

Ejemplo de devolución del script CalcR_MFeedback

```
Filtro 64Hz-250Hz >> CalcR_MFeedback
—Cálculo de las R para Filtros con AO según Realimentación Multiple—
—————Etapa_1—————
Ingrese el valor del capacitor a utilizar
10e-9
R1_1 =
1.5137e+004
R2_1 =
2.8440e+005
—————Etapa_2—————
Ingrese el valor del capacitor a utilizar
10e-9
R1_2 =
5.6135e+004
R2_2 =
1.3791e+005
—————Etapa_3—————
Ingrese el valor del capacitor a utilizar
10e-9
R1_3 =
1.1480e+005
```

Filtro	Etapa 1			Etapa 2			Etapa 3			Etapa 4			
	C[nF]	R1[kΩ]	R2[kΩ]	C[nF]	R1[kΩ]	R2[kΩ]	C[nF]	R1[kΩ]	R2[kΩ]	C[nF]	R1_A[kΩ]	R1_B[kΩ]	R2[kΩ]
64 Hz-250Hz	10	15,137	284,40	10	56,135	137,91	10	114,80	282,02	100	54,955	6,1940	104,59
250Hz-2k Hz	10	2,4712	26,882	10	8,5265	11,799	10	42,937	59,415	10	18,846		205
2k Hz-4k Hz	10	0,5231523	31,679	10	1,5778	15,091	10	2,0982	20,067	100	231,54	0,0999928	6,0523
4k Hz-8k Hz	10	0,2615761	15,839	10	0,7889241	7,5454	10	1,0491	10,034	100	115,77	0,0499964	3,0262

Cuadro 4.2: Valores de resistencias obtenidos

R2_3 =
2.8202e+005

-----Etapa_4-----

Ingrese el valor del capacitor a utilizar

100e-9

R1A_4 =
5.4955e+004

R1B_4 =
6.1940e+003

R2_4 =
1.0459e+005

Tabla de Valores Obtenidos Antes de mostrar la tabla de los valores de resistencias, se hace notar que el filtro cuyo ancho de banda es de 250Hz-2kHz, necesita un amplificador adicional con ganancia de 2,4923 veces para lograr 0dB en la banda de paso de dicho filtro.

4.3.3. Simulación en Pspice

4.3.3.1. Circuito de simulación

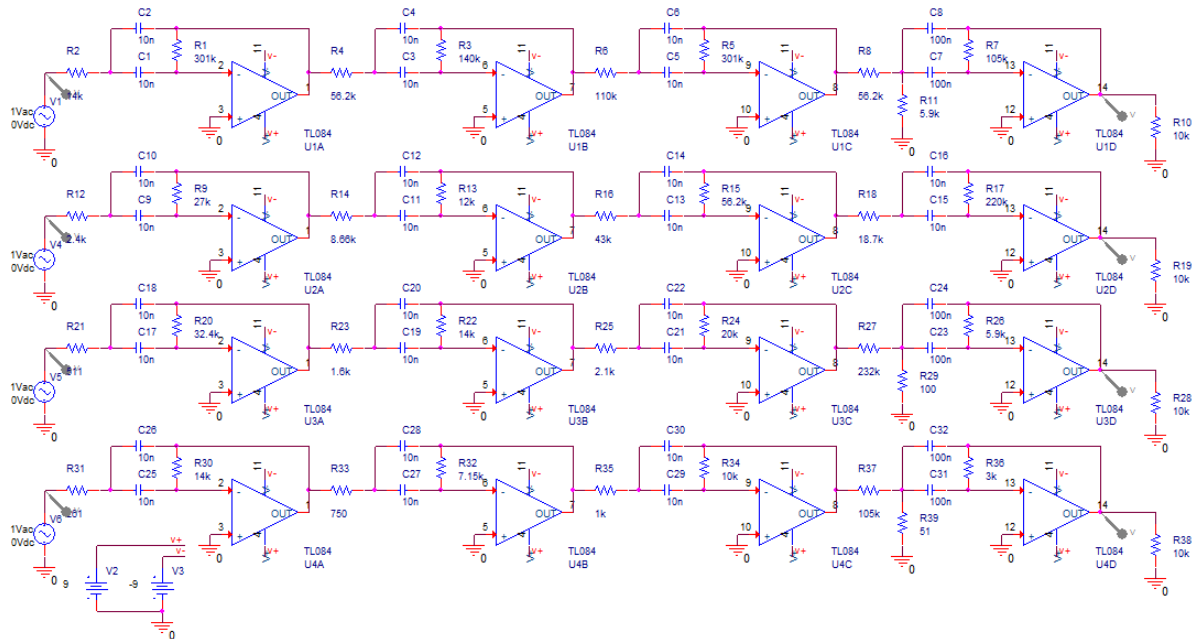


Figura 4.6: Circuito de simulación

4.3.3.2. Bodes de la simulación

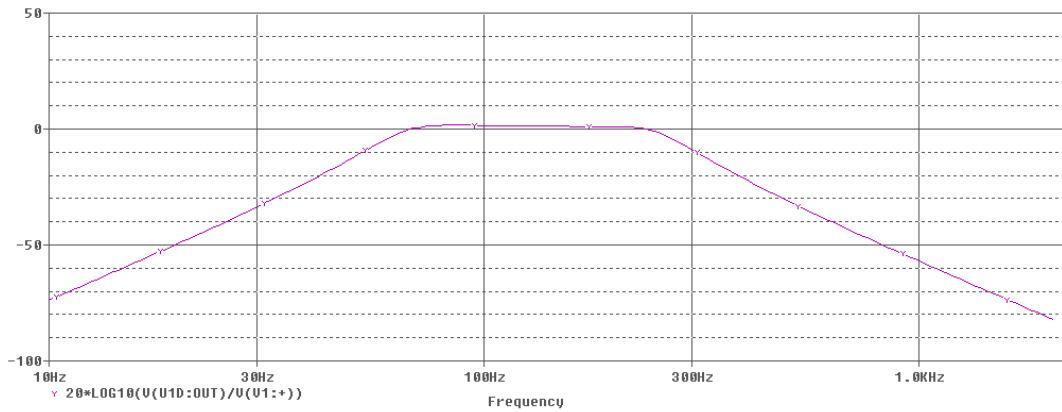


Figura 4.7: Bode del filtro 64Hz-250Hz

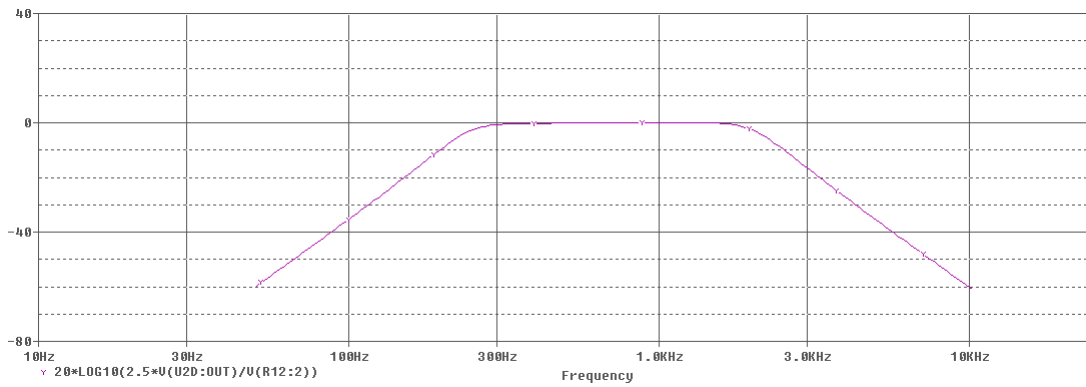


Figura 4.8: Bode del filtro 250Hz-2kHz

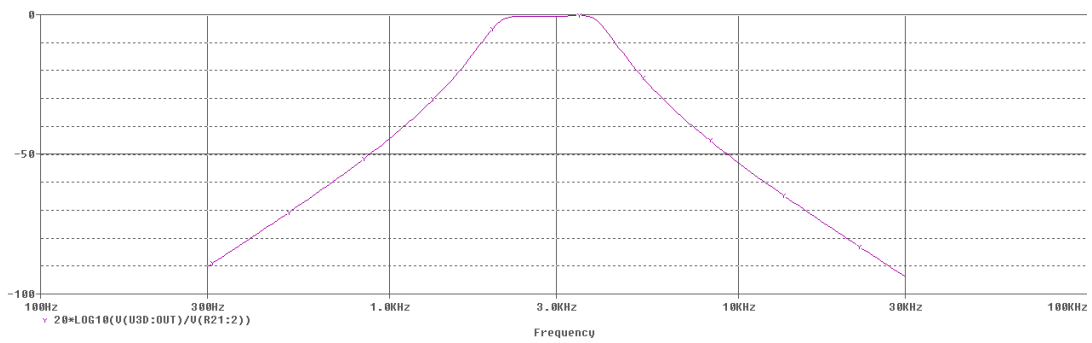


Figura 4.9: Bode del filtro 2kHz-4kHz

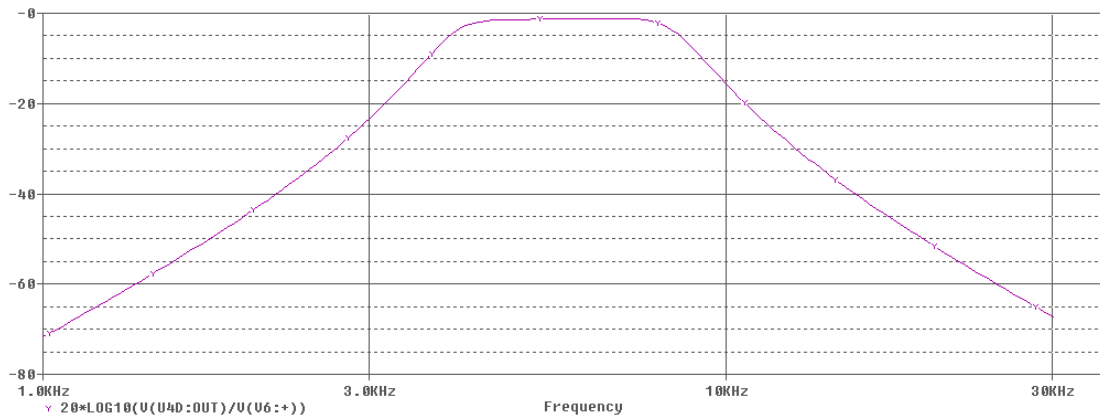


Figura 4.10: Bode del filtro 4kHz-8kHz

4.3.4. Realización de la placa

4.3.4.1. Circuito implementado

Antes de mostrar el circuito implementado se mencionaran ciertos aspectos que se tuvieron en cuenta en la practica. Lo primero que se añadió al circuito práctico fue una etapa de buffer en la entrada de cada filtro, excepto en el filtro de 250Hz-2kHz en la que el amplificador tiene una ganancia de 2,5 veces aproximadamente, esto se realizo para lograr una alta impedancia de entrada.

Desde un punto de vista matemático, el orden en que se conecten los filtros es irrelevante. En la práctica, para evitar la pérdida de rango dinámico y la exactitud del filtro debida a un posible recortamiento de la señal en las secciones con Q grande, se acomodan en orden creciente de Q. Sin embargo, esta forma no toma en cuenta el ruido interno, lo cual resulta preocupante en las etapas de Q grande, en las que cualquier componente de ruido que cae en el pico de resonancia puede amplificarse de manera significativo. Por esto, las etapas de Q grande deben ir primero, con el fin de minimizar el ruido.

Por ultimo se hace notar que los valores de resistencias utilizados son los más próximos encontrados en el mercado, correspondientes a las tolerancias del 1 y 2 %.

4.3.4.2. Vista del TopLayer

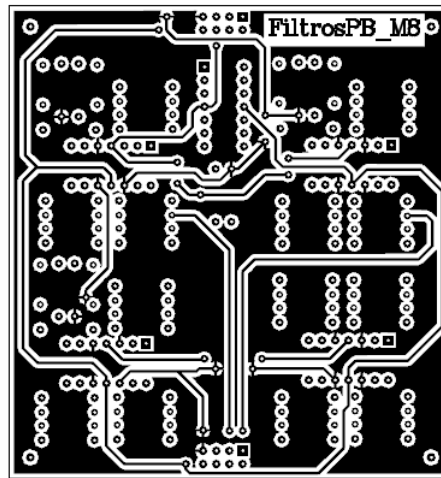


Figura 4.12: TopLayer

4.3.4.3. Vista del BottomLayer

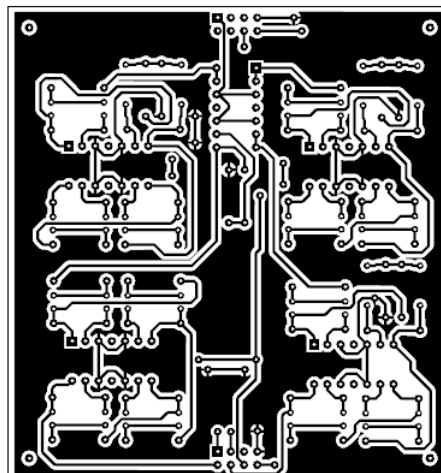


Figura 4.13: BottomLayer

4.3.4.4. Vista del TopOverlay

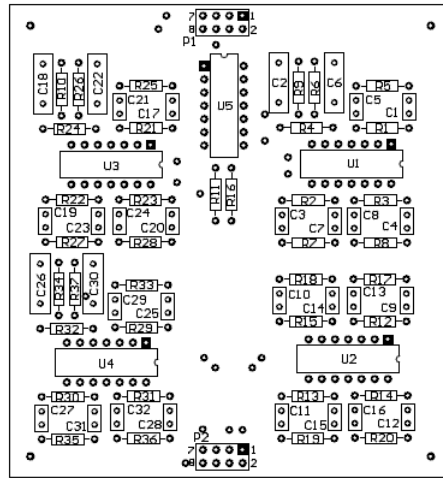


Figura 4.14: TopOverlay

4.4. Placa Micrófono

La etapa de adquisición consiste principalmente de 3 etapas: La etapa de polarización de micrófono, amplificación de la señal fuente y la etapa de aproximación de la respuesta en frecuencia a las Curva Isofonicas A y C.

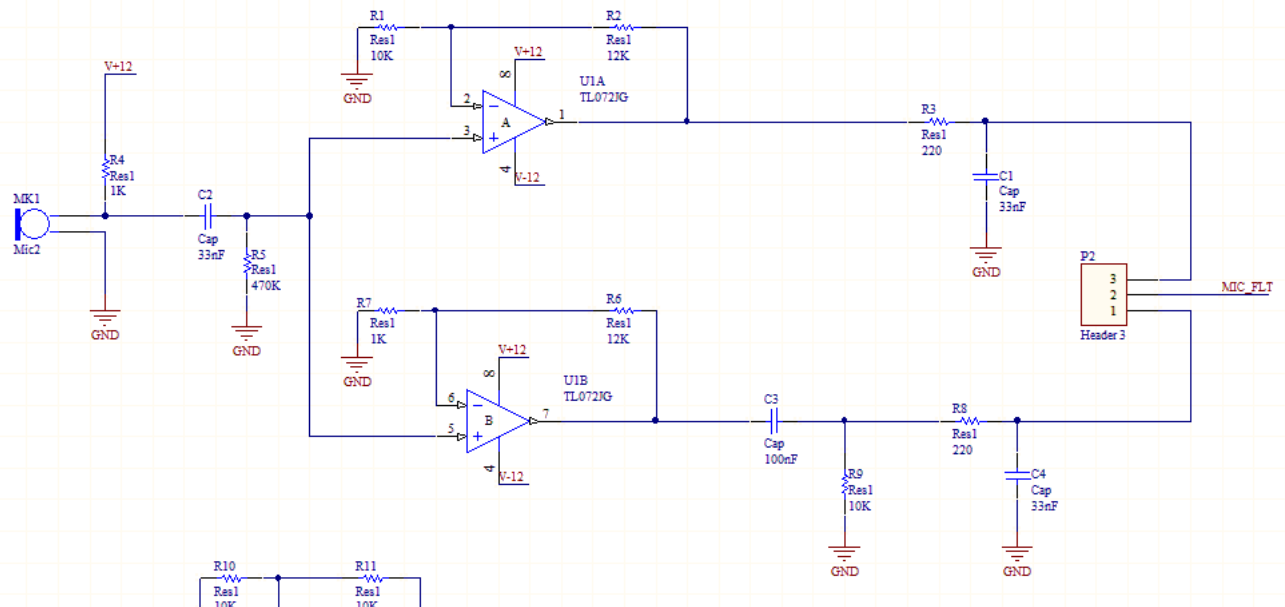


Figure 4.15: Esquema circuital de etapa de micrófono

4.4.1. Polarización de micrófono, amplificación y adaptación a curvas A y C

En primer lugar, el primer eslabón de la cadena es el micrófono a utilizar, ya que este limita desde un primer momento el ancho de banda y el rango dinámico de las variaciones de presión sonora que se desean registrar. El micrófono utilizado ha sido uno de tipo Electret. El único inconveniente es que no existe documentación

o referencia sobre el mismo, siendo imposible conocer su respuesta en frecuencia o su rango dinámico, por lo tanto se construyo un circuito preliminar para luego medir con un instrumento patrón.

El circuito preliminar consiste de la etapa de polarización y de amplificación, el micrófono Electret es un dispositivo que necesita ser excitado con una corriente fija para funcionar. Simplemente con conectarlo en serie con cualquier valor de resistencia, el micrófono esta listo para entregar una señal acorde. Se eligió el valor de 1KOhm ya que hace que este maneje un rango dinámico dentro de los valores necesarios.

Al no tener valores estándar de salida del micrófono, se utilizo un valor arbitrario de ganancia en la etapa del amplificador, y empíricamente se adaptaron los valores necesarios.

Para realizar las mediciones, se utilizo un instrumento patrón el cual dispone de un receptáculo(para ser utilizado con un micrófono Electret) con un pequeño parlante realimentado con un micrófono, en donde el mismo hace variar la presión sonora a un valor determinado de dB y frecuencia, siendo estos valores lo suficientemente fieles para calibrar otros instrumentos. La gráfica de las mediciones es la siguiente

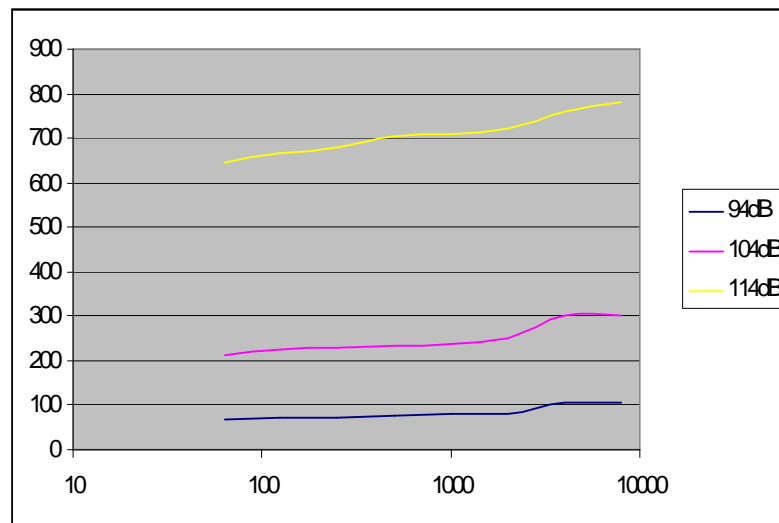


Figura 4.16: Respuesta medida de micrófono Electret

Este rango se adapta a la curva de ponderación C, que es para ambientes ruidosos, por lo tanto se deberá aproximar nuestra respuesta del micrófono a la curva de ponderación C. La respuesta intrínseca del micrófono no es del todo aproximada en las frecuencias altas, por lo tanto se decidió implementar un filtro pasabajos de orden 1 alrededor de los 4000Hz.

En el caso de la aproximación a la curva A, aparte del filtro PB de 4000Hz, se le debe adicionar un filtro pasa altos en 1000Hz, ya que en la curva A, las frecuencias bajas se atenúan considerablemente.

Por ultimo se dispuso de un jumper para seleccionar un rango o otro, siendo este cambio, también configurable en el software del microcontrolador.

Luego de comprobada esta etapa se procedió a la realización del circuito completo.

Finalmente se definieron dos amplificadores: Uno correspondiente al rango de 50dB-80dB, con ganancia de 13; y otro correspondiente al rango de 80dB-130dB con ganancia 2.2 ; ya que con solo una conflagración no se puede abarcar un rango muy extenso.

Adicionalmente se aplico en la entrada de ambos un filtro pasa alto para desacoplar el nivel de tensión continua de polarización.

Luego de esta etapa, la señal es entregada a los filtros diferenciadores de octavadas, tratada en el capítulo anterior

4.4.2. Adaptación a niveles de tensión ADC

La señal de entrada del ADC debe estar entre 0V y 3V, por lo tanto, se implementa al final un sumador no inversor con ganancia 1, en el que se suma una señal de continua de 1.5V a la señal del micrófono amplificada, para que la Vpp de la señal varíe entre 0V y 3V, con centro en 1.5V.

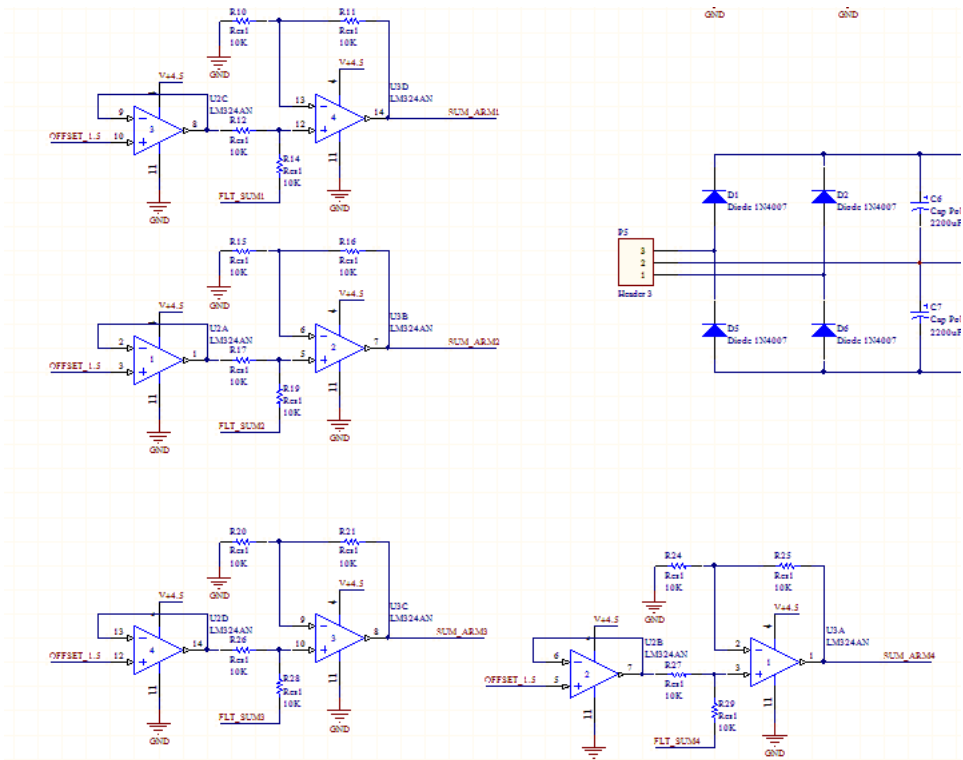


Figura 4.17: Esquema circuitual de etapa de micrófono

Al existir cuatro señales diferentes, se deben implementar 4 sumadores no inversores distintos. Para evitar problemas con la señal, la señal de offset ingresa al sumador mediante un buffer no inversor.

Como medida de seguridad se utilizaron LM324, ya que estos pueden ser alimentados con 4.5V y 0V evitando así que la señal de salida excursione fuera de 0V o 3V ($V_{cc} - 1.5V$) pudiendo así evitar un posible daño al ADC del microcontrolador.

4.4.3. Fuente de alimentación

Por ultimo, la fuente de alimentación de todo el dispositivo se implemento en la placa junto al micrófono y a los sumadores, para aprovechar el tamaño de la placa.

Es de notar que se utilizar 2 potenciómetros para regular la tensiones de alimentación de los LM324, y de la señal de offset para los sumadores.

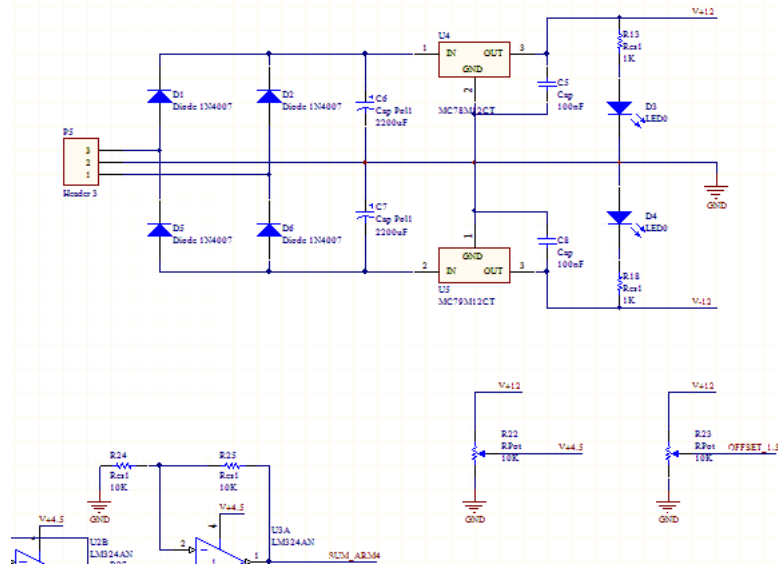


Figura 4.18: Esquema circuitual Fuente de alimentación

4.4.4. Top Layer

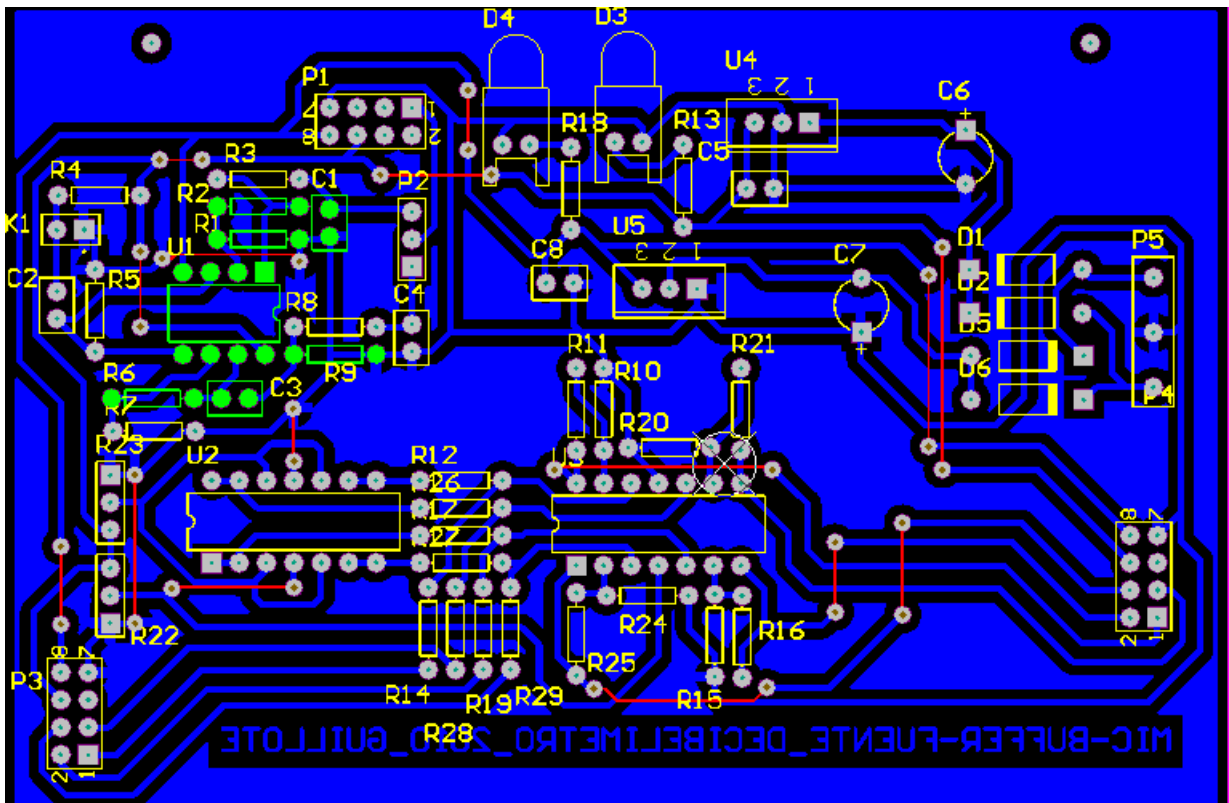


Figura 4.19: Top Placa Micrófono

4.4.5. Bottom Layer

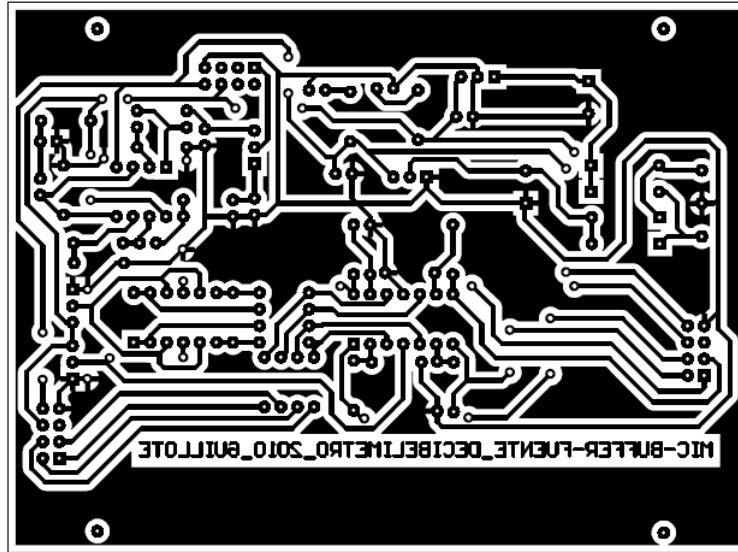


Figura 4.20: Esquemático Placa Micrófono

Capítulo 5

Método Experimental

5.1. Mediciones del Filtro

A continuación se mostraran en forma de tablas las mediciones realizadas para obtener los bodes de los filtros reales, para luego mostrar los mismos.

5.1.1. Tablas de mediciones

Filtro 64Hz-250Hz			
Vin[Vpp]	Vout[Vpp]	Frecuencia[Hz]	dB
3,84	0,12	20,2	-30,1
4,04	0,48	40,5	-18,5
4,04	1,24	50,2	-10,25
4,04	3,06	60,6	-2,41
4,04	3,68	64	-0,81
4,06	4,72	101,2	1,3
4,08	4,5	150,2	0,85
4,04	4,46	180,4	0,85
4,08	4,4	200,3	0,65
4,08	2,88	252	-3,02
4,08	1,28	300,5	-10,06
4,08	0,66	350	-15,82
4,08	0,36	401,9	-21,08
4,08	0,14	651	-29,29

Cuadro 5.1: Mediciones del filtro 64Hz-250Hz

Filtro 250Hz-2kHz			
Vin[Vpp]	Vout[Vpp]	Frecuencia[Hz]	dB
4	0,16	100	-27,95
4,08	1,72	200	-7,5
4,04	2,2	215	-5,27
4,04	2,68	230	-3,56
4,04	3,2	250	-2,02
4,04	4	500	-0,08
4,04	4,2	1000	0,33
4,04	3,12	1500	-2,24
4,04	1,9	1900	-6,55
4,04	1,72	2000	-7,41
4,04	1,56	2100	-8,26
4,04	1,32	2250	-9,71
4,04	1	2500	-12,12
4,04	0,6	3000	-16,56
4,04	0,24	4000	-24,52

Cuadro 5.2: Mediciones del filtro 250Hz-2kHz

Filtro 2kHz-4kHz			
Vin[Vpp]	Vout[Vpp]	Frecuencia[Hz]	dB
4,16	0,3	1000	-22,83
4,12	0,44	1500	-19,42
4,16	1,36	1800	-9,71
4,12	2,96	2000	-2,87
4,08	4	2520	-0,17
4,08	3,92	3000	-0,34
4,12	3,96	3500	-0,34
4,12	2,16	4000	-5,6
4,08	0,88	4500	-13,32
4,12	0,48	5000	-18,67

Cuadro 5.3: Mediciones del filtro 2kHz-4kHz

Filtro 4kHz-8kHz			
Vin[Vpp]	Vout[Vpp]	Frecuencia[Hz]	dB
4,08	0,48	3000	-18,58
4,12	1,28	3500	-10,15
4,12	2,96	4000	-2,87
4,12	3,28	4500	-1,98
4,08	3,36	5000	-1,68
4,12	3,6	6000	-1,17
4,12	3,32	7000	-1,87
4,12	3,12	7500	-2,41
4,08	2,6	8000	-3,91
4,12	1,96	8500	-6,45
4,12	1,24	9000	-10,42
4,12	0,6	10000	-16,73
4,08	0,4	15000	-20,17

Cuadro 5.4: Mediciones del filtro 4kHz-8kHz

5.1.2. Bodes de los filtros reales

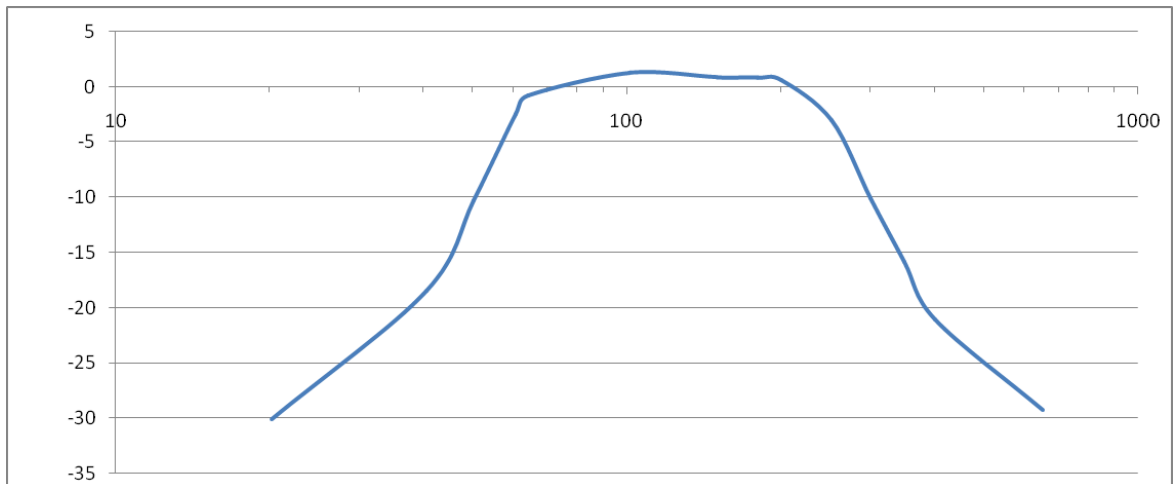


Figura 5.1: Bode del filtro 64Hz-250Hz

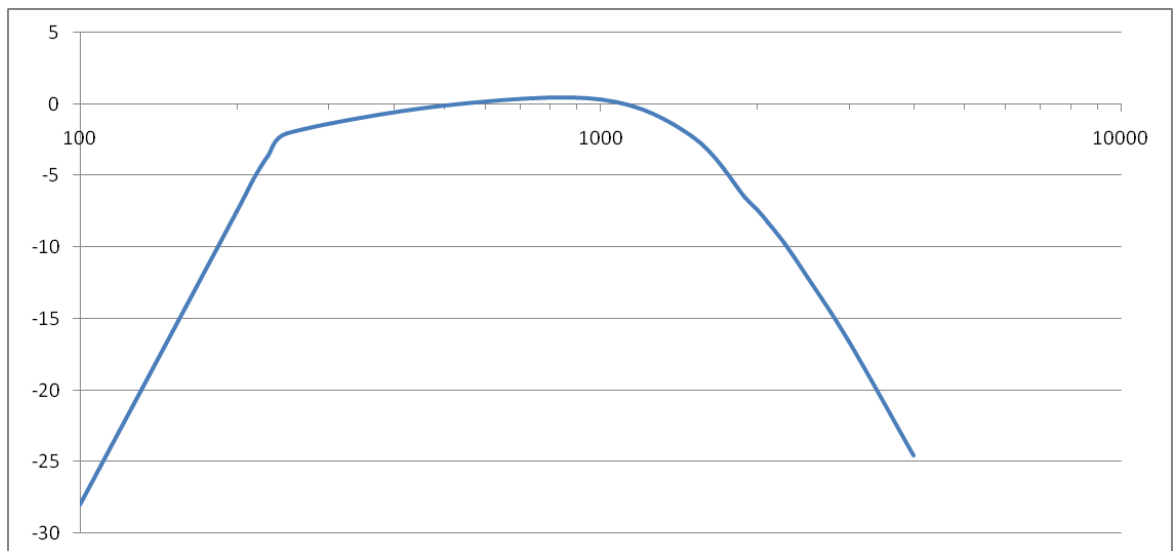


Figura 5.2: Bode del filtro 250Hz-2kHz

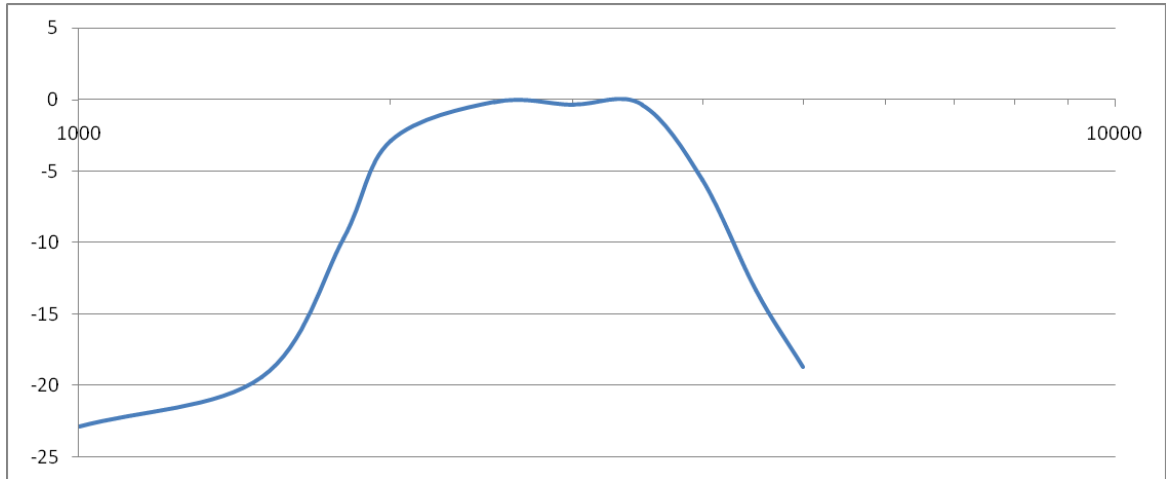


Figura 5.3: Bode del filtro 2kHz-4kHz

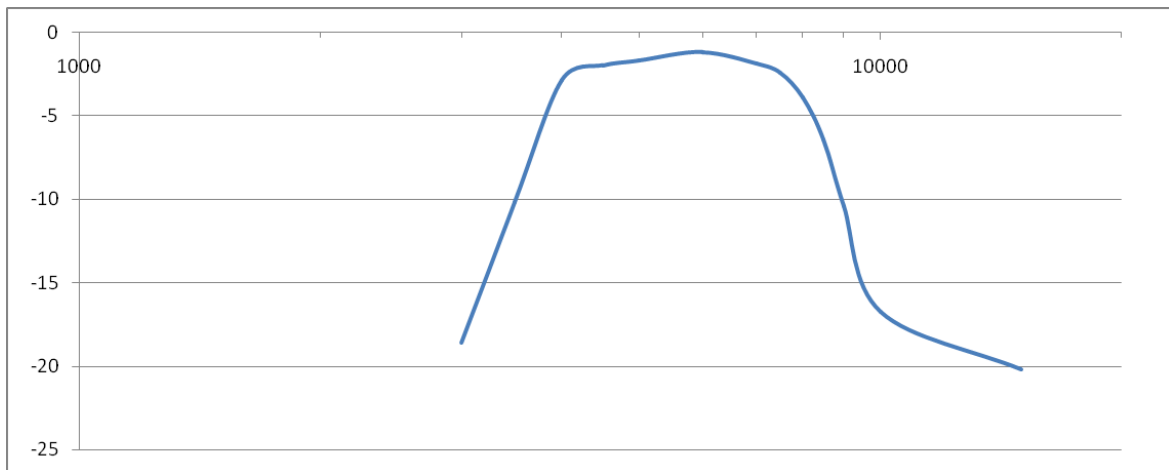


Figura 5.4: Bode del filtro 4kHz-8kHz

5.2. Mediciones del Micrófono

Principalmente se realizaron mediciones en el Centro de Investigación y Transferencia en Acústica (CIN-TRA) dentro de una cámara acustizada. Consistió en la medición de la señal de salida de la etapa completa de micrófono cuando se excitaba un parlante en varias frecuencia e intensidades diferentes (conocidas), en la cual se obtuvieron los niveles de tensión referentes a las diferentes excitaciones.

Con estas tablas, se puede deducir que el error del decibelímetro es de por lo menos $\pm 2dB$.

5.2.1. Tablas de mediciones

X	dB Medidos	dB Calculados	Tensión Vpp
63Hz	104dB	105,7dB	276mV
125Hz	115dB	116,1dB	699mV
250Hz	112dB	112,7dB	417mV
500Hz	119dB	120dB	994mV
1KHz	114dB	113,1dB	424mV
2KHz	121dB	121,6dB	1,13V
4KHz	119dB	117,8dB	736mV
8KHz	94dB	96dB	120mv

Cuadro 5.5: Tabla de Contrastación

Capítulo 6

Conclusiones

En primer lugar, se pudo aprender a programar una nueva familia de microcontroladores basados en el núcleo ARM7. Gracias al proyecto elegido, tuvimos que darle un uso de más de un 80 % al microcontrolador lo que nos llevó a aprender varias características del mismo y llegar a la conclusión de que este es muy poderoso, se encuentra mucha información al respecto y es relativamente económico para cualquier estudiante.

Por otra parte, hemos notado que cada vez es mayor la gente que comparte sus conocimientos, permitiendo que nuevas personas (como nosotros) puedan utilizarlos para ganar tiempo, para generar un mejor proyecto y para muchas cosas más.

La posibilidad de respaldar los datos en la SD/MMC fue el tema más importante y más costoso a nivel software, pero como antípoda de lo anterior fue el ítem que más contentos nos dejó al verlo funcionando. El entender los 3 niveles que necesitaba la SD/MMC era lo principal para poder empezar a jugar con ella y hablar de niveles siempre fue un punto teórico e intangible, pero el tener que aplicar esos conocimientos a la memoria SD era bastante nuevo.

Por otro lado, la configuración del SPI del micro también la realizamos totalmente nosotros. No conseguimos librerías para esto y las notas de aplicación que ofrece el fabricante son para los microcontroladores más nuevos (los que tienen el sufijo /01/02). Por lo que tuvimos que leer a fondo la hoja de datos, comenzar a probar y testear con un osciloscopio los pines configurados, ya sea el clock del SPI, envío y recepción de datos.

Para el envío de comandos a la SD conseguimos una librería pero era necesario estudiarla al 100 % ya que necesitábamos entender que comando enviábamos y que recibíamos de la SD para saber si esta estaba lista, era de solo lectura, y varias cosas más.

Para finalizar, la alegría que nos dio al ver un dato grabado en la SD fue saber que habíamos realizado un proyecto que no está en la WEB, que lo hicimos completamente solos y ahora lo podemos compartir para que algún estudiante lo pueda adjuntar a su próximo proyecto.

Respecto a los filtros, podemos decir que a la hora de elegir los anchos de banda, se encontraron rangos de frecuencias ya determinados (64Hz-250Hz, 250Hz-2kHz, 2kHz-4kHz y 4kHz-8kHz), los cuales tienen importantes características y están definidos por octavas, por lo que se eligió los mismos para desarrollar los filtros pasa banda para nuestro decibelímetro.

Al momento de elegir la aproximación del filtro a realizar, se estudiaron cuales eran, las ventajas y desventajas de cada una, y se llegó a la elección de la que nos pareció más correcta (Butterworth) para nuestro decibelímetro.

Entre las dos topologías de filtros que se adaptaban mejor a nuestros requerimientos (Sallen-Key y Múltiples Realimentación) se encontró que si bien la Sallen-Key usa menos componentes, la de Múltiples Realimentación es más estable a las variaciones de los componentes por lo que se eligió esta última.

Se utilizó el programa de cálculo MatLab[®] debido a que nos entrega la función de transferencia de un filtro, según la aproximación deseada, y además porque el uso de scripts nos facilitó mucho los cálculos.

Cuando se estudió el circuito de Múltiples Realimentación, se observó que podía ser necesario una atenuación en cada filtro y se realizó el cálculo correspondiente para llevarla a cabo con un divisor resistivo.

A la hora de realizar el circuito que se iba a llevar a la práctica, se tubo en cuenta el orden de las etapas (Q decreciente) para minimizar el ruido. Además también se añadió una etapa de buffer al principio de cada filtro para obtener una alta impedancia de entrada.

Por último, en la contrastación de las mediciones con los valores encontrados teóricamente, se obtuvieron pequeñas discrepancias, las cuales son debidas a que los valores encontrados de las resistencias difieren de los

calculados, y en algunos casos esta diferencia es bastante grande.

Capítulo 7

Bibliografía.

- <http://ciiii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/HerramientasProgramacion>
- <http://elm-chan.org/>
- <http://ciiii.frc.utn.edu.ar/>
- <http://www.organizaciones.frc.utn.edu.ar/cee/>
- Diseño con amplificadores operacionales y circuitos integrados analógicos. Autor: Sergio Franco.
- Datasheet Lpc2124.

Anexo

Lista de Componentes

Componente	Cantidad	Observaciones
Cap 1uF	2	
Cap 10nF	26	
Cap 100nF	6	
Res 330	24	
Res 10K	7	
Resistencias	varias	1.6K, 1K, 2.1K, 2.4K, 3K, 5.9K, 7.15K, 8.66K, 10K, 12K, 14K, 18.7K, 20K, 27K, 32.4K, 43K, 51, 56.2K, 100, 105K, 110K, 140K, 220K, 232K, 261, 301K, 511, 750
TL074N	5	
SW DIP-2	1	
Pulsador	1	
LM7805	1	
Tira Pines 2X5	1	
Tira Pines 2X4	2	
Tira Pines 26	1	
Tira Pines 25	1	
Con SD/MMC	1	
LED	24	
LED_Bicolor	1	
MC78M12CT	1	
MC79M12CT	1	
TL072JG	1	
LM324AN	2	
Diode 1N4007	4	
Micrófono Electret	1	

Cuadro 7.1: Lista de Componentes

Índice general

1. Introducción	1
1.1. Marco Teórico	1
1.1.1. Sonido	1
1.1.2. Presión Sonora	1
1.1.3. Curvas Isofónicas	2
1.2. ¿Que es un Decibelímetro?	3
2. Modo de uso del Dispositivo	5
3. Software	7
3.1. Microcontrolador y Herramienta de Programación	7
3.2. Firmware del microcontrolador.	7
3.2.1. Main	7
3.2.2. UART	8
3.2.3. A/D	9
3.2.4. RTC	9
3.2.5. Interrupciones	9
3.2.6. SD/MMC	9
4. Hardware	10
4.1. Placa ARM 2009	10
4.2. Placa Decibelímetro	10
4.2.1. Esquemático Placa Decibelímetro	11
4.2.2. Top Layer	12
4.2.3. Bottom Layer	12
4.3. Placa Filtros	12
4.3.1. Elección de parámetros y topología a realizar	13
4.3.1.1. Tipo y orden de los filtros	13
4.3.1.2. Topología	13
4.3.2. Cálculos de los filtros	13
4.3.2.1. Funciones de Transferencia	13
4.3.2.2. Cálculo de Resistencias	16
4.3.3. Simulación en Pspice	20
4.3.3.1. Circuito de simulación	20
4.3.3.2. Bodes de la simulación	21
4.3.4. Realización de la placa	22
4.3.4.1. Circuito implementado	22
4.3.4.2. Vista del TopLayer	24
4.3.4.3. Vista del BottomLayer	24
4.3.4.4. Vista del TopOverlay	25
4.4. Placa Micrófono	25
4.4.1. Polarización de micrófono, amplificación y adaptación a curvas A y C	25
4.4.2. Adaptación a niveles de tensión ADC	26
4.4.3. Fuente de alimentación	27
4.4.4. Top Layer	28

4.4.5. Bottom Layer	29
5. Método Experimental	30
5.1. Mediciones del Filtro	30
5.1.1. Tablas de mediciones	30
5.1.2. Bodes de los filtros reales	32
5.2. Mediciones del Micrófono	33
5.2.1. Tablas de mediciones	34
6. Conclusiones	35
7. Bibliografía.	37