

Implementación de un Vúmetro de 8 Bandas con la EDU-CIAA

1.° Maximiliano Alexandroff
Universidad Nacional de La Plata
Facultad de Ingeniería
La Plata, Buenos Aires
malexandroff@linti.unlp.edu.ar

2.° Tomás Rosales
Universidad Nacional de La Plata
Facultad de Ingeniería
La Plata, Buenos Aires
tomasjoaquinrosales@gmail.com

3.° José Juárez
Universidad Nacional de La Plata
Facultad de Ingeniería
La Plata, Buenos Aires
UNQ Dpto. CyT
Bernal, Buenos Aires
jjuares@ing.unlp.edu.ar

Resumen—En el presente trabajo se describe la implementación de un vúmetro de 8 bandas utilizando la plataforma EDU-CIAA. El objetivo principal es el procesamiento digital de una señal de audio en tiempo real, su análisis espectral y la visualización del resultado en una matriz de LEDs de 8x8. Se presenta el diseño de hardware para acondicionar la señal analógica y para controlar la matriz, se describe el desarrollo de software y presentan los resultados obtenidos.

Palabras claves—vúmetro, audio, EDU-CIAA, DSP, diseño de PCB, FFT.

I. INTRODUCCIÓN

En los últimos años con el avance de la tecnología se ha producido una migración cada vez mayor desde el procesamiento analógico hacia el procesamiento digital. En consecuencia, para realizar el procesamiento digital es necesario realizar varias tareas, desde el acondicionamiento de señales hacia el microcontrolador, la conversión analógico-digital, el procesamiento de la misma y su posterior visualización. El presente artículo muestra la implementación de un vúmetro de 8 bandas sobre la placa de desarrollo EDU-CIAA [1] en el cual se realiza el procesamiento digital de audio en tiempo real, proyecto propuesto por la materia *Taller de Proyecto I* de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la UNLP.

II. ETAPA DE ANÁLISIS

El proceso de desarrollo del sistema actual, comenzó con una exploración de los componentes necesarios. Luego, el diseño y fabricación de la placa de expansión o “Poncho” que cumple las funcionalidades principales del proyecto. Las cuales son: recibir y acondicionar la señal de audio para enviarla a la placa EDU-CIAA, brindar una interfaz al usuario mediante el uso de teclado y posteriormente al análisis de la señal de audio, presentar por medio de la matriz de leds 8x8 la potencia promedio de dicha señal en cada banda. Por último, se realizó el desarrollo de software el cual incluye: muestreo de la señal entrante, análisis espectral, comunicación SPI, ingreso de teclado y planificación de tareas compartidas.

Las 8 bandas que se eligieron para realizar el presente proyecto se muestran en el *Cuadro I*.

Cuadro I
SELECCIÓN DE BANDAS

Banda	Rango [Hz]	Banda	Rango [Hz]
1	0 a 160	2	160 a 300
3	300 a 600	4	600 a 1200
5	1200 a 2400	6	2400 a 5000
7	5000 a 10000	8	10000 a 16000

Para realizar el análisis en el dominio de frecuencia se optó por utilizar la transformada rápida de Fourier (*FFT, Fast Fourier Transform*) [2][3] que es un algoritmo optimizado para facilitar el cálculo computacional.

En la Figura 1, se ilustra un esquema sintetizado del proceso que se utilizó para llevar a cabo la implementación del sistema de procesamiento de audio en tiempo real.

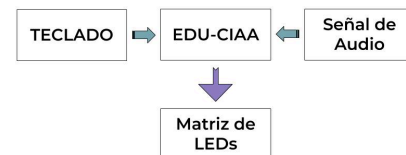


Figura 1. Diagrama de bloques del sistema.

III. ACONDICIONAMIENTO DE LA SEÑAL

Antes de muestrear y procesar, fue necesario conocer las características de la señal de entrada al sistema. Como entrada, se dispuso de una señal audio provista por un celular, tablet o PC que está estandarizada para trabajar con tensiones entre $\pm 1V$, lo cual fue confirmado midiendo con un osciloscopio, su señal a máximo volumen. Por otro lado, para el muestreo de la señal se utilizó el conversor analógico-digital (ADC) de 10 bits de la EDU-CIAA, el cual posee un rango de tensiones entre 0V y +3,3V. Como primer paso, se aplicó un offset a la señal entrante, para que la misma trabaje en tensiones positivas. Esto se logró mediante la utilización de un divisor de tensión desacoplado en continua, seguido por un filtro anti-aliasing cuya frecuencia de corte es de 16kHz.

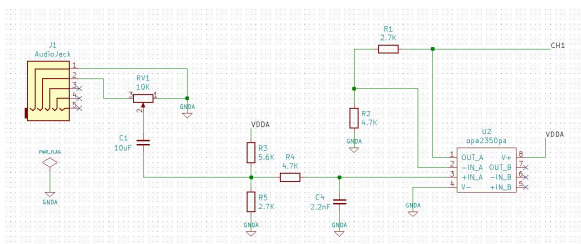


Figura 2. Esquemático - Entrada de Audio.

Luego, para utilizar el rango completo de entrada del ADC de la EDU-CIAA, se utilizó un amplificador operacional rail to rail (OPA2350) utilizando un circuito no-inversor para obtener una ganancia de $G = 1,5745$. Con lo cual, como resultado se obtuvo una señal que variaba aproximadamente entre un valor mínimo de $+0,12V$ y un valor máximo de $+3,27V$, centrada en $+1,695V$. El circuito de acondicionamiento de la señal se muestra en la Figura 2.

IV. DESARROLLO DE PONCHO

Para este sistema, se diseñó un circuito impreso con los siguientes componentes: audio jack de entrada, resistencias, capacitores, amplificador operacional OPA2350, controlador de matriz de led MAX7219, matriz de leds de 8×8 y, por último, los terminales adecuados para los pines de la placa EDU-CIAA. Luego de la etapa de diseño, se pasó a la etapa de fabricación del PCB, en la cual, se realizó la impresión del diseño sobre una placa de cobre para luego realizar la soldadura de los componentes sobre la placa. Para esta etapa se utilizó el paquete de software libre de KiCad [4] y el modelo de poncho [7] mediano (planilla PCB con las interfaces de conexión de la EDU-CIAA) como base para el diseño del PCB.

V. DESARROLLO DE FIRMWARE

Previamente a comenzar a escribir el código, se realizó un estudio de la arquitectura de la placa de desarrollo a utilizar y de los módulos que se comunican con la misma.

V-A. Filtrado Digital

Para realizar el procesamiento digital de audio en tiempo real en la EDU-CIAA se optó, como se mencionó anteriormente, por utilizar la FFT en vez de utilizar 8 filtros digitales mediante la convolución. El motivo de esta decisión, fue la optimización de tiempo de ejecución del firmware y poder realizar el procesamiento en tiempo real, ya que, la FFT es más eficiente que la utilización de filtros digitales. Para realizar el muestreo, se escogió una frecuencia de muestreo de 32.768 Hz para cumplir con el teorema de Nyquist-Shannon y un tamaño de muestra de 2048 elementos debido al espacio reducido de memoria. Con estos parámetros, se calculó la resolución de la FFT mediante la siguiente ecuación:

$$Resolucion = \frac{Fs}{N} \quad (1)$$

Esto significa que, se cuenta con una resolución de 16 Hz . De esta manera, el filtrado digital se realizó tomando porciones del buffer de 2048 elementos. A modo de ejemplo, si se toman los primeros 10 elementos del mismo, se obtiene el rango de frecuencias de 0 Hz a 160 Hz . Mediante este análisis, se pudo realizar la división del espectro de frecuencia en 8 bandas y realizar el cálculo de potencia para cada una en particular y expresarlo mediante una escala de leds.

V-B. Cálculo de Potencia

Una vez realizado el filtrado de la señal, se calculó la potencia de cada banda mediante el uso de la librería **CMSIS-DSP** [5]. Para esto, se procesó la FFT calculada por la función: `void arm_cmplx_rfft_fast_f32()`. Ya que, retornaba un arreglo de números complejos. Por lo tanto, se calculó la magnitud de ese arreglo mediante la función: `void arm_cmplx_mag_f32()`. Este arreglo de magnitudes solo presentaba la mitad de la FFT, pero, como se estaba procesando una señal real, la magnitud de la transformada era simétrica. De esta manera, se utilizó la función: `void arm_power_f32()`. La cual, realizaba el cálculo de la energía de la señal. Finalmente, aplicando la relación de Parseval, y multiplicando por dos, debido a la simetría, se obtuvo la potencia.

VI. RESULTADO

A lo largo del proyecto, se desarrollaron diferentes ensayos llevados a cabo en el laboratorio ATEI de la Facultad de Ingeniería. Se comenzó con un ensayo de prueba sobre el Firmware v2 [6] que provee el proyecto CIAA, el cual, hace utilización del conversor analógico digital y se comprobó la correcta configuración del ADC y periféricos. Luego, se le realizaron pruebas al poncho para verificar que la señal estaba bien acondicionada antes de conectarse al ADC de la placa. Esto se realizó utilizando el osciloscopio y observando en diferentes niveles de la señal de entrada, esto es: señal de entrada, señal luego de pasar por el divisor de tensión, señal después de ser amplificada y por último la señal resultante al pasar por el circuito antialiasing. Posteriormente, se realizaron ensayos de volumen y determinación de la escala de visualización junto con ensayos de tonos senoidales puros barriendo diferentes frecuencias. Por último, se comprobó el funcionamiento utilizando la señal proveniente del reproductor de música de un smartphone.

En la Figura 3, se puede observar el sistema en funcionamiento.

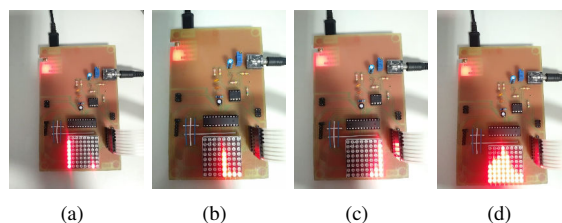


Figura 3. Tono de (a) 80Hz (b) 1800 Hz (c) 15000Hz (d) Música

REFERENCIAS

- [1] (2018, Nov.) CIAA website. [Online]. Disponible en internet: <http://www.proyecto-ciaa.com.ar/>
- [2] A.V. Oppenheim, A. S. Willsky, S.H. Nawab, Señales y Sistemas, 2ª edición, Pearson Educación, 1998.
- [3] M.J. Roberts, Señales y Sistemas, McGraw - Hill Interamericana, 2005.
- [4] (2018, Nov.) KiCad website. [Online]. Disponible en internet: <http://kicad-pcb.org/>
- [5] (2018, Nov.) Keil website. [Online]. Disponible en internet: <https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>
- [6] (2018, Nov.) Firmware v2 website. [Online]. Disponible en internet: https://github.com/ciaa/firmware_v2
- [7] (2018, Nov.) CIAA GitHub. [Online]. Disponible en internet: <https://github.com/ciaa/Ponchos/tree/master/modelos>