

DISEÑO Y CONSTRUCCIÓN DE UN ANALIZADOR DE ESPECTROS PORTÁTIL UTILIZANDO UN LCD GRÁFICO Y UN DSP56F807 DE FREESCALE

OMAR JAVIER TÍJARO ROJAS

*Ingeniero Electrónico
Magister en Ingenierías Área Electrónica
Universidad Industrial de Santander
maepe22@uis.edu.co*

YAIR DE JESÚS RUIDÍAZ PALOMINO

*Ingeniero Electrónico
yair0414@hotmail.com*

JORGE EDUARDO HIGUERA PORTILLA

*Ingeniero Electrónico
Magister en Ingenierías Área Electrónica
Universidad Industrial de Santander
maepe3@uis.edu.co*

JAIME GUILLERMO BARRERO PÉREZ

*Magíster en Potencia Eléctrica
Profesor Asociado Escuela de Ingenierías Eléctrica, Electrónica
y de Telecomunicaciones
Universidad Industrial de Santander
jbarrero@uis.edu.co*

RESUMEN

Fecha Recepción: 19 de septiembre de 2005

Fecha Aceptación: 23 de mayo de 2006

En el presente artículo se hallan contenido los resultados de la investigación realizada para el diseño y construcción de un equipo analizador de espectros portátil utilizando el procesador de señales digitales DSP56F807 miembro de la familia 56F800 de Freescale Semiconductor y una pantalla LCD gráfica. Se resalta este diseño como un ejemplo de aplicación directa de la teoría de tratamiento digital de señales y, se observa finalmente que los errores obtenidos son menores al 1% al implementar el algoritmo de radix-2 para calcular la Transformada Rápida de Fourier con diversas formas de onda.

PALABRAS CLAVE: *DSP, FFT, Aliasing, FIR, Enventanado, Throughput, MIPS, PLL, CMOS, GPIO, ADC, JTAG, Flash, LCD, Touch screen, ASCII.*

ABSTRACT

This article presents the results of the investigation carried out for the design and construction of a portable spectrum analyzer using the digital signal processor DSP56F807, a member of the family 56F800 of Freescale Semiconductor, and a screen LCD graph. This design is an example of a direct application of the theory of digital signal processing. It is observed finally that the obtained errors are smaller than 1% when implementing the radix-2 algorithm to calculate the fast fourier transform with diferents waveforms.

KEYWORDS: *DSP, FFT, Aliasing, FIR, Windowing, Throughput, MIPS, PLL, CMOS, GPIO, ADC, JTAG, Flash, LCD, Touch screen, ASCII.*

INTRODUCCIÓN

El desarrollo vertiginoso de la Electrónica en el mundo actual, ejerce una amplia influencia en todas las esferas de la vida moderna. Los avances recientes de la Microelectrónica en el desarrollo de nuevos procesadores digitales de señales (DSP) más versátiles, con la capacidad de procesar simultáneamente millones de operaciones por segundo y con capacidades matemáticas acordes para implementar algoritmos matemáticos complejos, proporcionan la plataforma para efectuar aplicaciones de la teoría del tratamiento de señales tanto en el dominio del tiempo como la frecuencia. Estos nuevos avances están siendo aprovechados ampliamente en aplicaciones de telecomunicaciones, la robótica y el control de procesos en tiempo real.

El objetivo del presente trabajo es el de llevar a cabo un equipo analizador de espectros portátil aprovechando las ventajas de los procesadores digitales de señales (DSP) de la familia 56F80X de FREESCALE SEMICONDUCTOR para el uso en la industria eléctrica y la automatización en la industria automotriz.

DESCRIPCIÓN GENERAL DEL EQUIPO

Las características técnicas del equipo observado en la **Figura 1** son:

- Alimentación con una fuente de 9 V DC y 500mA.
- Ancho de banda de 20 kHz.
- Span de 2343,75 Hz.
- Resolución en frecuencia de 585,9375 Hz.
- Baterías recargables de 5 V.
- Tamaño 17 x 14 x 6 cms.
- Corriente de consumo del sistema 238,5 mA.
- Impedancia de entrada de la señal 56k Ω .
- Expansión de memoria y conexiones al PC por puerto USB 2.0.
- Dos canales de ADC con resolución de 12 "bits" de adquisición de la señal.
- Procesado de la señal con 16 "bits".
- Despliegue de la FFT en una pantalla LCD de 320 x 240 píxeles.
- Interfaz gráfica de fácil entendimiento para el usuario.
- Puerto JTAG para actualización del firmware.

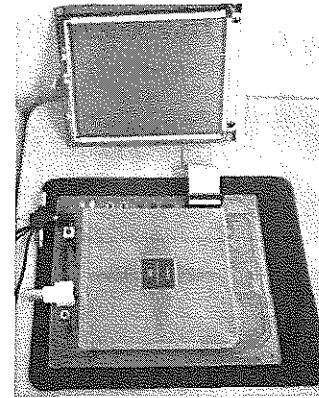


Figura 1. Equipo analizador de espectros.

HARDWARE IMPLEMENTADO

Para el diseño del equipo electrónico se planteó una etapa de filtrado antisolapamiento por hardware y una por software que se encarga de evitar errores en la adquisición de datos debido al ruido presentado en el ambiente y a la alimentación de los diferentes circuitos integrados; además se construyó una etapa de acondicionamiento de señales para luego desarrollar el algoritmo de la transformada de Fourier. En la **Figura 2** se observa el diagrama de bloques referente a la implementación física del analizador de espectros.

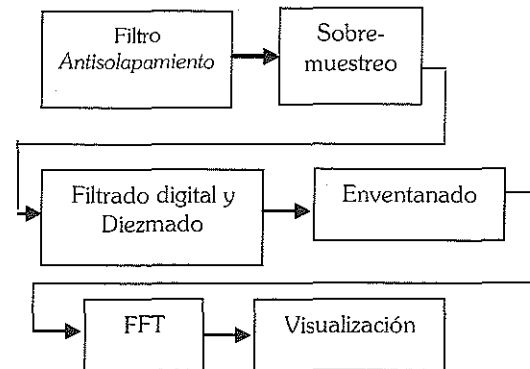


Figura 2. Diagrama de bloques general de un sistema de tratamiento digital de señales.

Filtro antisolapamiento. Cuando se muestrea una señal continua en el dominio del tiempo, existe la posibilidad de encontrar solapamiento (*aliasing*¹) de frecuencias. Para evitar este problema, se debe implementar un filtro analógico pasabajas que evite el muestreo de señales que no correspondan a las componentes de la frecuencia de la señal de entrada. [1].

¹ Aliasing: Fenómeno que se presenta cuando se muestrea una señal a una frecuencia menor que la mitad de ella. En éste, las frecuencias altas se pueden observar como bajas.

Se diseñó para esta etapa en “hardware” un filtro Butterworth pasabajas, ya que la respuesta es máximamente plana en amplitud en el ancho de banda deseado (23 kHz). El cálculo de los parámetros de éste filtro se planteó utilizando el “software” FilterPro de Texas Instruments. Esta herramienta permite diseñar el circuito esquemático del filtro y la respuesta en frecuencia (magnitud y fase).

El orden del filtro para lograr las atenuaciones adecuadas fue de orden cuarto, cuya frecuencia de paso (-3dB) fue de 20kHz, y la frecuencia de corte (-80dB) de 280kHz. En un diseño de detección de armónicos de la señal eléctrica existe el criterio de -98dB de atenuación, pero la razón por la que se escogió la atenuación de -80dB y no la de -98dB (que corresponde a una ganancia de 12,5 uV/V), se debe a que el nivel de cuantificación del conversor A/D del DSP es de 806 uV aproximadamente, es decir, cualquier señal de menor valor a este valor se pierde y por lo tanto el filtro no tiene que ser tan estricto. Este calculo se puede observar claramente en la ecuación (1). [3]

$$\Delta = \frac{V_{ref}}{4096} = \frac{3.3}{4096} = 805,6640 \text{ uV}$$

Por lo tanto:

$$\Delta dB = -61.8769 \text{ dB (1)}$$

Sobremuestreo. En esta etapa se adquiere la señal por medio del conversor A/D del DSP a una frecuencia de muestreo de 600kHz ± 12kHz, tomando aproximadamente 29 muestras en un periodo de la señal de más alta frecuencia(20kHz).

Además, se toman 1442 muestras para tener una mejor resolución en frecuencia.

Filtrado digital. Teniendo las muestras almacenadas en la memoria RAM del DSP, se procede a realizar un filtrado. El filtrado tiene como objetivo atenuar de forma más selectiva las frecuencias que deja pasar el filtro analógico. Este filtro se escogió FIR ya que estos presentan fase lineal, y se obtuvo utilizando las funciones de Matlab 6.5 **fir2** y **buttord**.

La función **buttord** se utilizó para encontrar el orden del filtro a partir de los argumentos: frecuencia de paso, frecuencia de corte, atenuación frecuencia de paso y de frecuencia de corte. El resultado se utilizó como argumento de la función **fir2** para hallar los coeficientes del filtro digital.

El orden del filtro obtenido fue de N=417, dándole al comando **buttord** los siguientes argumentos: frecuencia de paso 20 kHz y frecuencia de corte de 20,5kHz, atenuación de -3db y -90 dB respectivamente, estos últimos parámetros se utilizaron para darle más selectividad al filtro. [1]

Diezmado. En este sistema, el diezmado tiene como fin eliminar muestras de la señal para que la resolución en frecuencia se visualice mejor. El resultado obtenido de esta operación es un vector más pequeño, en donde se encuentra distribuido de una mejor manera el ancho especificado.

El diezmado realizado es por 8, teniéndose como vector final un arreglo 128 elementos para calcular la FFT.

Enventanado. Esta técnica se utiliza para obtener una mejor respuesta en la amplitud de los espectros de las muestras. Se implementó la ventana de Hanning, ya que permitió la mejor adaptación al sistema para la visualización de los resultados en la pantalla LCD gráfica en comparación con la ventana rectangular, es decir, permitía diferenciar dos componentes espectrales en regiones cercanas, y además su ecuación fue de fácil ejecución ya que realiza menos operaciones que ventanas como la Blackman que tienen una respuesta similar con mayor carga computacional.

La respuesta al impulso de la ventana de hanning se observa en (2) [6].

N: número de muestras.

N: variable independiente.

$$h[n] = 0.5 - 0.5 \cos\left(\frac{2\pi \cdot n}{N+1}\right) \quad (2)$$

Transformada de Fourier. Para realizar un análisis en frecuencia se necesita conocer la teoría básica de señales y por ende la Transformada Discreta de Fourier (DFT del inglés Discrete Fourier Transform), que es el equivalente discreto de las Series de Fourier. La Transformada de Fourier X(w) de una señal analógica x(t) es la vista en (3).

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j\omega t} dt \quad (3)$$

La Transformada Discreta de Fourier (4) es un método muy eficiente para determinar el espectro en frecuencia de una señal. Esta, permite convertir una secuencia de valores en el dominio del tiempo a una secuencia de valores equivalente en el dominio de la frecuencia. La Inversa de la Transformada Discreta de Fourier (IDFT) realiza el proceso contrario (5).

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W^{nk} \quad (4) \quad k=0,1,\dots,N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot W^{-nk} \quad (5) \quad n=0,1,\dots,N-1$$

La constante “W” es conocida como **factor de Fourier** y definida en (6)

$$W = e^{-j2\pi / N} \quad (6)$$

Se observa que 'W' es una función que depende de N, por ello, también suele expresarse como W_N .

Un algoritmo que realiza eficientemente la DFT es la Transformada Rápida de Fourier (FFT del inglés *Fast Fourier Transform*). Con la FFT se obtienen los mismos resultados que la DFT pero más rápidamente debido a que reduce el número de cálculos requeridos.

En la ecuación de la Transformada Discreta de Fourier, obtener X(k) para un 'k' determinado requiere aproximadamente N sumas complejas y N productos complejos (7).

$$X(k) = x(0) + x(1) \cdot W^k + \dots + x(N-1) \cdot W^{(N-1)k} \quad (7)$$

Para $k = 0, 1, \dots, N-1$.

Si lo que se desea es obtener X(0), X(1), ..., X(N-1) entonces se necesitarían un total de aproximadamente N^2 sumas complejas y N^2 productos complejos. Esto quiere decir que los requerimientos computacionales de la DFT pueden ser excesivos especialmente si el tamaño de N es grande.

La FFT aprovecha la periodicidad y simetría del factor de Fourier 'W' para el cálculo del Transformada Discreta de Fourier. La periodicidad de 'W' implica (8) y su simetría implica (9).

$$W^k = W^{k+N} \quad (8)$$

$$W^k = -W^{k+N/2} \quad (9)$$

La FFT descompone la DFT de N puntos en transformadas más pequeñas. Una DFT de N puntos es descompuesta en dos DFT's de (N/2) puntos. Cada DFT de (N/2) puntos se descompone a su vez en dos DFT's de (N/4) puntos y así sucesivamente. Al final de la descomposición se obtienen (N/2) DFT's de 2 puntos cada una. La transformada más pequeña viene determinada por la base de la FFT. Para una FFT de base 2, N debe ser una potencia de 2 y la transformada más pequeña es la DFT de 2 puntos. Para implementar la FFT existen dos procedimientos: diezmado en frecuencia (DIF del inglés *Decimation In Frequency*) y diezmado en el tiempo (DIT del inglés *Decimation In Time*). [5].

Visualización. Para la visualización de los resultados se utilizó una pantalla gráfica de cristal líquido fabricada por *Powertip Technology Corporation* [9], cuya resolución fue de 320x240 píxeles. La interfaz (gráfica y carácter) de la pantalla fue controlada con el DSP, en donde se diseñaron funciones para facilitar la programación y optimizar el algoritmo. En esta pantalla se mostró la FFT con una resolución en frecuencia de 585,9375 Hz (600k/1024) y un SPAM de 2343,75 Hz (585,9375 x 4). Para garantizar que se pueda observar la componente espectral de mayor potencia,

el vector de la FFT fue normalizado con el elemento de mayor valor absoluto, siendo el valor de la unidad 160 píxeles.

Éste se utilizó para poder cumplir con el despliegue vertical de la LCD (240 píxeles), de los cuales se utilizan 72 píxeles para mostrar texto.

Procesador digital de señales (DSP). Los procesadores digitales de señales (DSP-Sigla en Inglés) son dispositivos que se caracterizan por su alto throughput², bajo consumo de potencia y bajo costo, en comparación con los microcontroladores; por estas razones estos procesadores se convierten en una de las mejores herramientas para el procesamiento digital de señales, los cuales son empleados muy comúnmente en sistemas telefónicos, aplicaciones de reducción de ruidos, filtros digitales, etc. [3]

Un ejemplo de los procesadores digitales de señales los presenta Motorola en la familia DSP56800. Esta tiene tecnología CMOS de 16 "bits", presenta una CPU de propósito general diseñada para el eficiente procesamiento digital de una señal y el control de operaciones. El repertorio de instrucciones que presenta es amplio facilitando la programación al usuario. El dispositivo que se utilizó para implementar este equipo en hardware fue el DSP56F807. [2]

A continuación se mencionarán algunas características generales de la familia DSP56800:

- * Es capaz de procesar hasta 35 millones de instrucciones por segundo (MIPS).
- * Requiere solo una fuente de alimentación entre 2,7 a 3,6 Voltios.
- * Presenta dos registros acumuladores de 36 "bits", incluyendo extensión de "bits".
- * Dos pines de interrupciones externas.
- * Tres buses de 16 "bits" de datos.
- * Tres buses de 16 "bits" de direcciones.
- * Repertorio de instrucciones que dan soporte al procesamiento digital de la señal y control de funciones.
- * Soporta compilador en C (*Codewarrior*).

Los DSP56F800 son un pequeño sistema encapsulado

²Throughput: Cantidad de datos procesados en un tiempo determinado.

conformado por subsistemas principales. Estos subsistemas son: memoria, Phase Lock Loop (PLL), Unidad Central de Procesos (CPU), periféricos, puerto de entrada salida, “bus” de interfaz externa [2].

La memoria de los DSP56F800 utiliza dos espacios de memoria independientes, los espacios de datos y los de programación.

PRUEBAS Y ANÁLISIS DE RESULTADOS

En esta sección se comprobó el funcionamiento de los diferentes módulos del DSP que se necesitaron en el desarrollo de la investigación.

PRUEBAS DEL CONVERSOR ANALÓGICO-DIGITAL.

Todo conversor analógico a digital presenta un error en el momento de cuantificar una muestra, en el caso del DSP, el fabricante proporciona un error de cuantificación de $V_{ref}/4096$.

Para verificar que la frecuencia de muestreo era la especificada por el fabricante, se configuró el puerto ADC de tal forma que tuviese la máxima frecuencia de reloj. Luego se tomaron 1442 muestras de una señal cuadrada de la cual se conocía previamente su frecuencia y su amplitud, ya que éstas se midieron con el generador de señales y un osciloscopio marca *Fluke*. Se realizaron varias pruebas que se visualizaron en *Codewarrior 6.1*, y algunos de los resultados fueron los siguientes: [8]

* Onda cuadrada de 4kHz: Tuvo su primer cambio de signo en la muestra 20 y su tercer cambio de signo en la muestra 169, al hacer la resta se obtiene el número de muestras existentes en un periodo, y por lo tanto como se sabe la frecuencia de la señal, el cálculo de la frecuencia de muestreo está dada a continuación (10).

$$f_s = (N + 1) * f_m \quad (10)$$

Donde f_s es la frecuencia de muestreo, $N+1$ el número de muestras y f_m la frecuencia de la señal de entrada.

Aplicando la ecuación (10) se obtiene que la frecuencia de muestreo del ADC es 600kHz.

* Onda cuadrada de 4,6kHz: Con esta onda se procedió igual que la anterior, donde el número de muestras contenidas en un periodo fue de 128. Reemplazando nuevamente en (10), $f_s=588\text{kHz}$.

* Onda cuadrada de 5kHz: Se realizó el mismo procedimiento, $N+1=122$, por lo tanto $f_s=610\text{kHz}$.

Para comprobar el error de cuantificación del ADC del DSP

se tomaron muchas pruebas de las cuales se obtuvieron los resultados en donde el error en porcentaje se calculó utilizando el error promedio, el cual arrojó un resultado del 9,84 %. Este resultado se obtuvo de la siguiente manera:

$$\left(\frac{0,325}{3,3}\right) * 100\%$$

PRUEBAS DE LOS ALGORITMOS

En esta sección se realizaron pruebas a los códigos implementados en el DSP (*Codewarrior*) contrastándose con funciones ya implementadas en el “software” *Matlab 6.5*. Las funciones probadas en esta sección son las correspondientes al filtrado digital de la señal, enventanado y FFT.

1. El filtrado digital se realizó como se mencionó anteriormente. El resultado arrojado se puede visualizar con la función *freqz* (respuesta en frecuencia de filtro digital) de *Matlab*. Este comando permite observar gráficas donde la frecuencia está normalizada de 0 a π (0 a 300 kHz en este caso).

2. Para la etapa de enventanado también se probó la respuesta en frecuencia. La **Figura 3** ilustra la respuesta en frecuencia del enventanado rectangular y la **Figura 4** el enventanado de Hanning (implementados en el DSP). En el equipo se utilizó la ventana de Hanning porque los resultados arrojados fueron mejores que los de la ventana rectangular teniendo en cuenta, que se presenta mayor atenuación de pequeñas componentes no deseadas en todo el ancho de banda. En la ventana rectangular por no tener un periodo entero de la señal al calcular la FFT si se presentan estas componentes no deseadas.

Otra característica por la cual se escogió ésta ventana es porque representa menor carga computacional en el DSP comparándola con la ventana de Blackman.

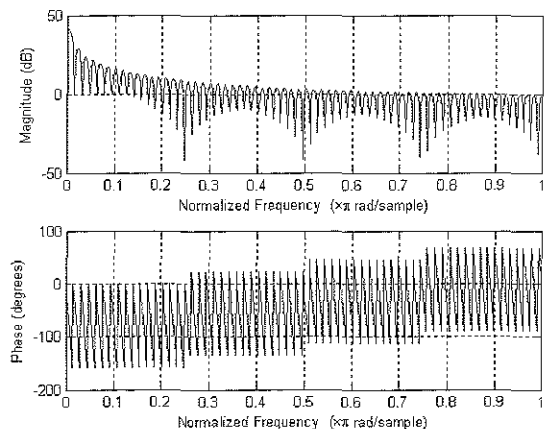


Figura 3. Respuesta en frecuencia del enventanado de rectangular

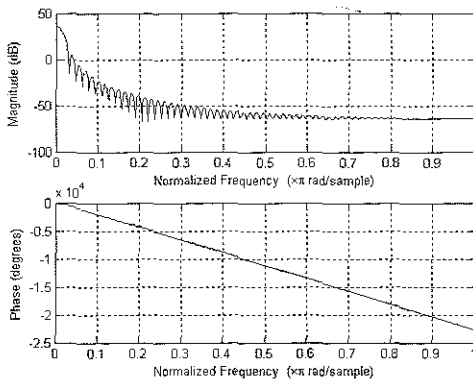


Figura 4. Respuesta en frecuencia del enventanado de Hanning

3. Para la prueba de la FFT se simularon varias señales, de las cuales se tomaron 128 muestras en un periodo. De éstas se obtuvo que su magnitud presentó las siguientes características:

- * Onda senoidal $[100 \cdot \sin(2 \cdot \pi / 128)]$.
- *Matlab*: En la posición 2 y en la 128 del vector aparecía una magnitud de 64.
- *Codewarrior*: En la posición 2 y en la 128 del vector aparecía una magnitud de 63,619.
El error de esta prueba fue de 0,595%.
- * Onda DC [10].
- *Matlab*: En la posición 1 del vector aparecía una magnitud de 16384.
- *Codewarrior*: En la posición 1 del vector aparecía una magnitud de 16384.
El error en DC fue de 0%.
- * Onda cuadrada [Amplitud 10, T=64muestras].
- *Matlab*: En la posición 2 y en la 128 del vector aparecía una magnitud de 814,955.
- *Codewarrior*: En la posición 2 y en la 128 del vector aparecía una magnitud de 810,246.

El error de esta prueba fue de 0,5%.

- *Matlab*: En la posición 4 y en la 126 del vector aparecía una magnitud de 271,87.
- *Codewarrior*: En la posición 4 y en la 126 del vector aparecía una magnitud de 271,73.
El error de esta prueba fue de 0,04%.
- *Matlab*: En la posición 64 del vector aparecía una magnitud de 20,006.
- *Codewarrior*: En la posición 4 y en la 126 del vector aparecía una magnitud de 20.
El error de esta prueba fue de 0,03%

Comparación de Desempeño: Para el desarrollo de estas pruebas se diseñó un algoritmo utilizando la técnica de **Decimación en Tiempo** la cual divide la secuencia de datos de entrada $x[n]$ en dos grupos, uno de índices par y el otro de índices impar. Con estas sub-secuencias se realiza el

DFT de $N/2$ puntos y sus resultados se combinan para formar el DFT de N puntos.

Para estas pruebas se usaron algunos macros predefinidos (*Beans*) que ofrece el software de programación para el cálculo de las funciones trigonométricas Seno y Coseno. El *Bean* a incluir es: **TFR1:DSP_Func_TFR**.

Cabe resaltar que el uso de estos macros obliga el manejo de un formato de variable llamado **Frac16**, el cual restringe el manejo de números entre -1 y 1 .

A continuación se describen las pruebas realizadas para verificar el funcionamiento del algoritmo implementado y evaluar la eficiencia de cálculo de la FFT con respecto al predefinido que ofrece *Code Warrior*.

En primera instancia para validar el funcionamiento del algoritmo se presentan los errores relativos y porcentuales de los valores obtenidos con el procesador (**Tablas 1, 2 y 3**), comparados con los valores teóricos calculados con *Matlab* y los resultados gráficos de la FFT (**Figura 5 y 6**). En segunda instancia para evaluar la eficiencia del algoritmo se midieron con osciloscopio los tiempos de ejecución (**Tabla 4**) de la FFT, Tanto del programa desarrollado como del *Bean* predefinido **DFR1:DSP_Func_DFR** que contiene la función **FFTC_Init**. Para ello se hace uso de un puerto de salida del DSP a Través del *Bean* **Bit1:BitIO** que maneja uno de los leds de la tarjeta de desarrollo. De igual forma se presentan los tamaños de los respectivos códigos (**Tabla 5**) a evaluar con el fin de mirar el espacio de memoria de grama utilizado.

Tabla 1. Valores de la señal de entrada.

REF	SEÑAL DE ENTRADA			
	DSP	MATLAB	%E	E. RELATIVO
0	0,00009155	0		-9,155273E-05
1	0,7070097	0,7071067	0,0137293	9,708119E-05
2	0,9998779	1	0,01221	1,221000E-04
3	0,7060092	0,7071067	0,15522142	1,097581E-03
4	-0,00009155	-2,068E-13		9,155272E-05
5	-0,7070312	-0,70710678	0,0106887	-7,558119E-05
6	-1	-1	0	0
7	-0,7069397	-0,70710678	0,0236288	-1,670812E-04

Tabla 2. Valores de la parte real de la FFT.

PARTE REAL DE LA FFT			
REF	DSP	MATLAB	E. RELATIVO
0	-0,0000610115	0	6,101145E-05
1	-0,0008670222	0	8,670222E-04
2	0,0001220144	0	-1,220144E-04
3	-0,0003726091	0	3,726091E-04
4	-0,0001831292	0	1,831292E-04
5	0,0002640355	0	-2,640355E-04
6	0,0001221262	0	-1,221262E-04
7	0,0019235110	0	-1,923511E-03

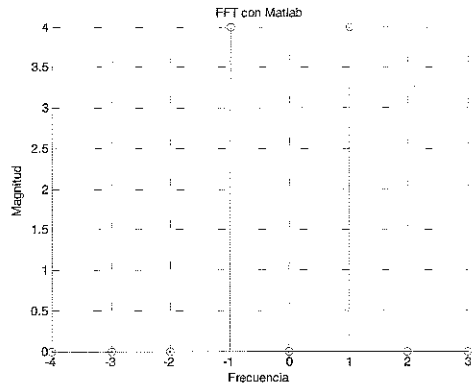


Figura 6. Magnitud de la FFT con Matlab

Tabla 3. Valores de la parte imaginaria de la FFT.

PARTE IMAGINARIA DE LA FFT				
REF	DSP	MATLAB	%E	E. RELATIVO
0	0	0		0
1	-3,9989070000	-4	0,027325	-1,093000E-03
2	-0,0001221167	0		1,221167E-04
3	0,0002138615	0		-2,138615E-04
4	0,0000000447	0		-4,471804E-08
5	-0,0003535748	0		3,535748E-04
6	0,0001220720	0		-1,220720E-04
7	3,9990890000	4	0,022775	9,110000E-04

En segunda instancia para evaluar la eficiencia del algoritmo se midieron con osciloscopio los tiempos de ejecución (Tabla 4) de la FFT, tanto del programa desarrollado como del Bean predefinido DFR1:DSP_Func_DFR que contiene la función FFTC_Init. Para ello se hace uso de un puerto de salida del DSP a través del Bean Bit1:BitIO que maneja uno de los Leds de la tarjeta de desarrollo. De igual forma se presentan los tamaños de los respectivos códigos (Tabla 5) a evaluar con el fin de mirar el espacio de memoria de programa utilizado.

Tabla 4. Tiempo de ejecución de la FFT.

# MUESTRAS	TIEMPO DE EJECUCIÓN	
	ALGORITMO	BEAN
8	428 [nS]	332 [nS]

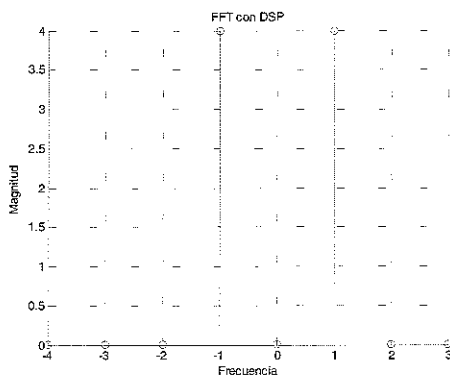


Figura 5. Magnitud de la FFT con DSP

Tabla 5. Espacio en memoria.

# MUESTRAS	TAMAÑO DEL CÓDIGO	
	ALGORITMO	BEAN
8	7860 [Bytes]	9622 [Bytes]

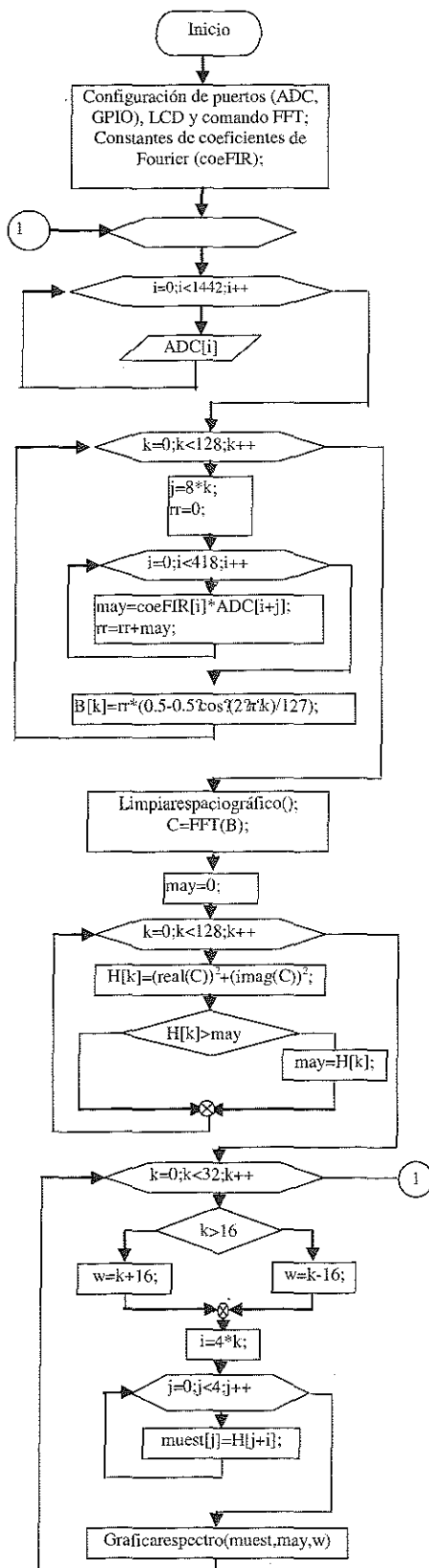


Figura 7. Diagrama de flujo implementado.

El diagrama de flujo implementado en el DSP que describe las operaciones realizadas se encuentra en la **Figura 7**. En éste se observa una estructura tipo *for* que no tiene variables en su interior como tampoco valores iniciales ni finales, este fenómeno se debe a la programación de este tipo de dispositivos los cuales tienen un ciclo “infinito” al cual siempre retorna el algoritmo. El círculo encerrado con el número uno (1) se usa como estructura indicadora de retorno al ciclo “infinito”.

CONCLUSIONES Y OBSERVACIONES

- Algo que es importante resaltar del equipo analizador de espectro portátil, es que este tendrá el puerto JTAG habilitado en la tarjeta, para ser programado de nuevo si así se desea, también incluye los puertos para expansión de memoria externa (bytes de dirección y datos) y dos canales de ADC.

- Por medio de dispositivos como los DSP se comprobó en esta investigación que se pueden elaborar aplicaciones en donde sea requisito una considerable carga computacional para realizar algoritmos de procesamiento de señal sin la utilización de una máquina tan potente como un computador, y a partir de una debida programación, construir equipos con visualización de eventos en pantalla LCD gráfica.

- Las aplicaciones presentadas por *Codewarrior* a través de la herramienta *Processor Expert* y en específico los “Beans”, sirven de mucha ayuda para aprender a utilizar el DSP como tal, es decir, con los debidos pasos para realizar una correcta configuración de un módulo y las funciones que emplea, además el entorno gráfico para la configuración de cada *Bean*, hacen ver fácil la programación del DSP.

- En este equipo se presentaron limitaciones de “hardware” tales como velocidad de conversión del ADC del DSP (600 kHz aproximadamente), alimentación de entrada al mismo (3,3 V); las cuales arrojaron características para el analizador de espectros como: la señal de entrada debe tener una frecuencia no mayor a 20kHz y su tensión no debe exceder de ± 1 V, para garantizar su correcto funcionamiento. Las limitaciones de “software” se dieron principalmente por la licencia entregada por *Codewarrior* para programación del dispositivo (16 KB).

- Los resultados obtenidos mediante el algoritmo desarrollado fueron satisfactorios. La desviación máxima mostró un error porcentual de 0.027%, obtenida en las componentes espectrales fuertes de la señal de entrada propuesta (Señal Senoidal), la cual se encontraba presente en la componente imaginaria del vector de salida; también se estableció una diferencia máxima o error relativo máximo

presente en los valores teóricamente cercanos al cero. En términos generales se logró obtener resultados aceptables y fiables para el cálculo de la FFT, donde los errores además de ser atribuidos a los problemas de cuantificación típicos de todo procesamiento digital con resolución finita, se debe sumar las desviaciones atribuidas a la solución planteada para evitar las incongruencias generadas en el calculo de las funciones trigonométricas cuando el argumento era cero y/o cercano a 0.5, donde EL Bean presentaba falencias, razón por la cual se optó por sumar al argumento el valor $1/32768$, cantidad que representa la mínima cuantía posible en el formato Frac16.

- En términos generales no se puede resaltar una diferencia sobresaliente entre los tiempos de ejecución del algoritmo para diferentes números de muestras, los resultados muestran que se obtuvo un tiempo estándar de procesamiento de [556nS] con una desviación máxima del 0.9%.

- De las pruebas finales realizadas al algoritmo desarrollado se puede extraer que el Bean realiza de una forma más rápida el calculo de la FFT, con seguridad se puede afirmar que es 1.29 veces más rapido comprobado con el programa en evaluación, lo cual permite concluir que los macros propuestos para funciones específicas dentro del DSP se pueden considerar óptimos desde el punto de vista de desempeño y aprovechamiento de recursos del procesador.

- Con referencia a trabajos realizados en el país en arquitecturas DSP [10], es importante que la ubicación de aplicaciones en tiempo real estén de la mano del diseño de hardware, lo cual fue una mejoría a muchos trabajos investigativos realizados para optar el título de ingenieros en distintas universidades de la nación donde no se implementaban diseños compactos.

AGRADECIMIENTOS

Se agradece por la colaboración a los diferentes docentes de la Universidad Industrial de Santander que aportaron para la elaboración de este artículo. Ellos son el Dr. Jorge Luis Chacón de la Escuela de Ingeniería Mecánica, y a los diferentes docentes de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

BIBLIOGRAFIA

[1] Tijaro O. Javier, Ruidíaz Yair. Analizador de espectros portátil utilizando la familia de DSPs 56F80X de Freescale, 2005. Tesis de pregrado. Escuela de Ingenierías eléctrica, electrónica y telecomunicaciones. Universidad Industrial de Santander UIS. Pág. 18-40.

[2] Motorola digital dna, DSP56F800 16-Bit Digital Signal Processor (PDF), [DSP56F800FM/D](#), 03, 2003.

[3] Freescale semiconductor, DSP56F800 User Manual, 56800 Documentation(PDF), DSP56F801-7UM, 06, 2004. Pág.9-5.

[4] Freescale Semiconductor, 56F807 16 bit Hybrid Processor. Technical Data, Data Sheets DSP56F807, DSP56F807, 12, 2004

[5] POSADAS Juan Luis. Transformada Rápida de Fourier (FFT) e Interpolación en Tiempo Real. Valencia, 1998. Universidad Politécnica de Valencia. Departamento de informática de sistemas y computadores. Pág. 4-11.

[6] OPPENHEIM, Alan. SCHAFFER, Ronald. Tratamiento de Señales en Tiempo Discreto. Prentice Hall. España 2000. Pág. 470.

[7] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DSP56F807&nodeId=01624686366292 Fabricante DSP56F807. (Junio 2005).

[8] <http://www.metrowerks.com/mw/download/> Proveedor de "software". (Junio 2005).

[9] <http://www.powertipusa.com/>. Fabricantes pantalla gráfica. (Junio 2005).

[10] R. ALZATE; G. CASTELLANOS. Realización para un DSP de aplicaciones en tiempo real. Ingeniería y competitividad. Volumen 6 n°1 –Septiembre de 2004; p 7-16.

AUTORES

Jaime Barrero P. Ingeniero Electricista UIS 1988. Magíster en Potencia Eléctrica UIS 2000. Profesor asociado escuela de ingeniería electrónica. Bucaramanga Colombia Universidad Industrial de Santander Tel 6344000 ext 2708 jbarrero@uis.edu.co Bucaramanga, Colombia.



Jorge Eduardo Higuera Portilla. Ingeniero electrónico UIS 2002. Magíster en Ingeniería áea Ingeniería Electrónica 2006. maepe3@uis.edu.co. Bucaramanga, Colombia



Omar Javier Tijero Rojas. Ingeniero electrónico UIS 2005. Tel 6324480. Candidato a Magíster en Ingeniería área Ingeniería Electrónica UIS 2005. Investigador Grupo CEMOS E3T, GOTS Escuela de Física. oma_ti@lycos.com Bucaramanga, Colombia.



Yair de Jesús Ruidíaz Palomino. Ingeniero electrónico UIS 2005. Tel 6833112. Investigador Grupo CEMOS E3T. yair0414@hotmail.com

