UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ANALIZADOR DE LINEA ELECTRICA BASADO EN PC

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRONICO

PRESENTADO POR:

FERNANDO DEMETRIO BRICEÑO GONZALES

PROMOCIÓN
1990 - I

LIMA – PERÚ
2006

ANALIZADOR DE LINEA ELECTRICA BASADO EN PC

A mi familia, por su apoyo incondicional a quienes me debo y en especial a mi Padre, que vive en mi corazón.

SUMARIO

El presente informe trata sobe el diseño e implementación del Hardware y Software de un Sistema de Adquisición de Datos, que utiliza el puerto paralelo de las PCs.

El resultado en un instrumento de medición económico y sencillo basado en PC, que permita determinar en forma básica, sin necesidad de cumplir con rigidez los estándares de medición, los factores básicos de la señal eléctrica domiciliaria.

INDICE

PROLOGO		1
CAPITULO I		
CONCEPTOS Y ELEMENTOS		**2**
1.1	Análisis de Señales	2
1.1.1	Señales y Análisis Discreto	2
1.1.2	Análisis de Fourier	2
1.2	Digitalización de señales	4
1.2.1	Muestreo Ideal	4
1.2.2	Muestreo Natural	6
1.2.3	Frecuencia de Muestreo	7
1.2.4	Esquema de la Digitalización	7
1.3	La Transformada Rápida de Fourier	8
1.4	El ADC 0804	11
1.5	D.O.S. Sistema Operativo de Tiempo Real	11
1.6	El Lenguaje C++ y el Compilador Open Watcom	12
1.7	La Programación orientada a Objeto	12
1.8	El Puerto Paralelo	12
1.8.1	Registro de Datos	13
1.8.2	Registro de Control	13
1.8.3	Registro de Estado	14
1.9	Temporización en la PC	14
CAPITULO II		
REQUERIMIENTOS		**16**
2.1	Planteamiento del Problema	16
2.2	Frecuencia de Muestreo	16
2.3	El Rango nominal de voltajes	16
2.4	Resolución de Frecuencia	16
2.5	Portabilidad	16
2.6	Precisión de Reloj	17
2.7	Interfase con otros programas	17

2.8 Procesamiento en Línea 17
2.9 Interfaz Amigable 17
CAPITULO III
DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN 18
3.1 Soluciones al problema planteado 18
3.1.1 Frecuencia de Muestreo 18
3.1.2 El Rango nominal de voltajes 18
3.1.3 Resolución de Frecuencia 18
3.1.4 Portabilidad 18
3.1.5 Precisión de Reloj 19
3.1.6 Interfase con otros programas 19
3.1.7 Procesamiento en Línea 19
3.1.8 Programa Amigable 19
3.2 Estructura del Sistema de Adquisición de Datos 19
3.2.1 Implementación de la Temporización 21
3.2.2 Voltaje de Referencia del ADC 22
3.2.3 Nivel 0 de la Señal de Entrada 22
3.2.4 Toma de Señal 23
3.2.5 Filtro Antialiasing 24
3.2.6 Digitalización 24
3.2.7 Frecuencia de Muestreo 26
3.2.8 Resolución y Precisión 27
3.2.9 Remoción de la componente DC 27
3.2.10 Sincronización para el Muestreo 27
3.2.11 Ventanas 28
3.2.12 Clases de Objetos 30
3.2.13 Manejo de Memoria 31
3.2.14 Archivos de Configuración 31
3.2.15 Archivos de Entrada/Salida de Datos 32
3.2.16 Programa ANALIZA 32
CAPITULO IV
PRUEBAS Y CALIBRACIONES 34
4.1 Contrastación del Algoritmo FFT con el MatLab 34
4.2 Medida de Señal Generada 36
4.3 Medida de Señal de Línea 37

CONCLUSIONES 38
ANEXO A 39
ANEXO B 41
BIBLIOGRAFÍA 45

PROLOGO

En el presente trabajo se muestra el diseño e implementación de un sistema de adquisición de datos cuyo objetivo es realizar análisis de señales de baja frecuencia, especialmente las de línea eléctrica monofásica. Se muestran los límites impuestos para el sistema y la consiguiente solución resultante en el diseño del Hardware y Software necesarios para cumplir los requisitos.

La estructura del informe es la siguiente:

En el Primer Capitulo se describen brevemente los conceptos y elementos utilizados como sustento teórico del trabajo realizado, se explican los conceptos teóricos de Análisis de Señales y los elementos de Software y Hardware de PCs a tener en cuenta.

En el Segundo Capitulo se enumeran los requerimientos que se tomarán en cuenta como problemática a resolver en el sistema por implementar.

En el Tercer Capitulo se presenta el diseño del sistema resaltando los alcances hacia cada problemática enfrentada, los criterios utilizados y algunos detalles de la implementación del Hardware y Software resultante.

En el Cuarto Capitulo se presentan resultados de pruebas y procedimientos realizados para calibración y contrastación del sistema diseñado.. Se hace una comparación de resultados entre el algoritmo FFT del sistema implementado con el del MatLab.

CAPITULO I
CONCEPTOS Y ELEMENTOS

1.1 Análisis de Señales

1.1.1 Señales y Análisis Discreto

Una señal puede ser continua o discreta puede ser también periódica o a periódica la combinación de estas características genera los siguientes tipos de señales:

- Señales continuas y aperiódicas
- Señales continuas y periódicas
- Señales aperiódicas discretas
- Señales periódicas discretas.

Todas estas señales extienden desde el infinito negativo hasta el infinito positivo. En nuestro caso las señales que se estudiarán en este proyecto son continuas y básicamente periódicas, pero debido a la digitalización se convierten en discretas y por el lapso de muestreo ya no son continuas sino de duración finita.

1.1.2 Análisis de Fourier

Es una familia de técnicas matemáticas todas basadas en la descomposición de señales en sinusoides. Esto permite cambiar el dominio del estudio pasando del tiempo a la frecuencia, para ello se utilizan la Transformada de Fourier, cuando el análisis es de señales continuas, o la transformada Discreta de Fourier cuando se trata de señales discretas.

Las señales sinusoidales utilizadas en el análisis de Fourier no son adecuadas para el caso de señales de duración finita ya que estas son de duración infinita por lo tanto es necesario adecuar las herramientas para este caso.

Para poder aplicar las técnicas de Fourier en estos casos es necesario simular que la señal de duración finita es periódica creando una nueva señal en la cual la parte de la señal de duración finita es repetida hasta el infinito. Teniendo como base esto entonces es posible

aplicar las técnicas de Fourier y específicamente la Transformada Discreta de Fourier (DFT por sus siglas en inglés).

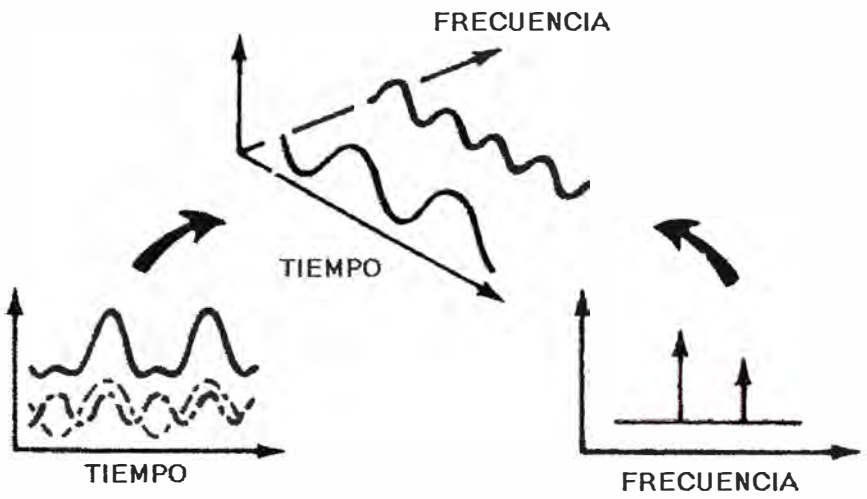


Fig. 1.1 Dominio del Tiempo y la Frecuencia

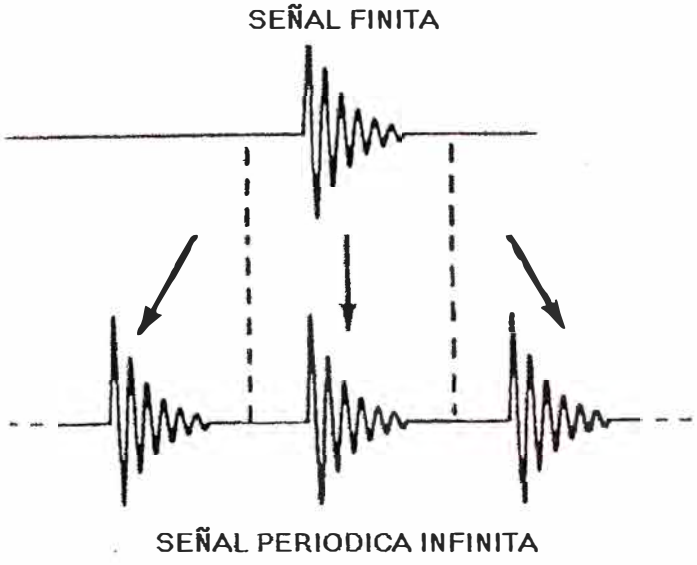


Fig. 1.2 Simulación de señal continua

La DFT es una secuencia no una función de una variable continua y corresponde a muestras equiespaciadas en frecuencia de la transformada de Fourier de la señal. Además de su importancia teórica como representación de Fourier en secuencias la presente tiene un papel crucial en la realización de una gran variedad de algoritmos de tratamiento digital

de señales. Básicamente la DFT tiene las mismas propiedades que la transformada de Fourier.

A su vez la DFT tiene variantes la DFT real y la DFT complejas, la primera solo utiliza valores reales en cambio la segunda se utilizan variables complejas para los cálculos, siendo esta ultima más genérica.

1.2. Digitalización de Señales

Para el proceso de los datos es necesario digitalizar las señales a las frecuencias adecuadas para poder captar todo el espectro necesario para el análisis.

1.2.1 Muestreo Ideal

Sea f(t) una señal continua en el tiempo y con ancho de banda finito W. Sea la onda muestreada:

$$m(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{1.1}$$

donde Ts es el periodo de muestreo

Cuando la entrada del muestreador es la señal f(t) su salida s(t)=m(t)*f(t) será:

$$s(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s)\delta(t - nT_s) \tag{1.2}$$

Puesto que la función de muestreo m(t) es periódica, la podemos representar mediante su serie de Fourier

$$c_n = \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{+\frac{T_s}{2}} \delta(t) e^{-j2\pi ft} dt = \frac{1}{T_s} \tag{1.3}$$

$$m(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{j2\frac{n\pi}{T_s}t} \tag{1.4}$$

por lo tanto la salida se expresará mediante:

$$s(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} f(t) e^{j2\frac{n\pi}{T_s}t} \tag{1.5}$$

La transformada de Fourier aplicada a s(t) nos da el espectro de salida de la señal muestreada:

$$S(f)=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}f(t)e^{-j2\pi(f-fn_S)t}=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}F(f-nfs) \qquad (1.6)$$

Conocido el espectro de f(t) que es F(f) el espectro de S(f) consiste en repetir el espectro de f(t) en unas frecuencias que son múltiplos de fs, aquí podemos tener dos posibilidades: Para fs,<2*W, o sea la frecuencia de muestreo es menor del doble del ancho de banda de f(t), en cuyo caso existe solapamiento de bandas. En este caso no se puede recuperar la señal original f(t) por haber solapamiento de las diferentes partes del espectro (aliasing).

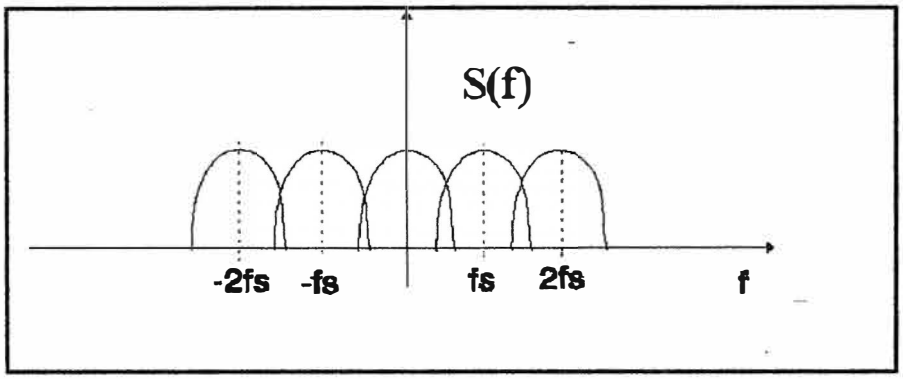


Fig 1.3 Filtrado con Solapamiento

Para fs,>=2*W, la frecuencia de muestreo es mayor del doble del ancho de banda de f(t), en cuyo caso no existe solapamiento de bandas. Este es el caso más interesante, puesto que a partir de una señal muestreada podemos volver a reconstruir la señal original utilizando un filtro pasabajos.

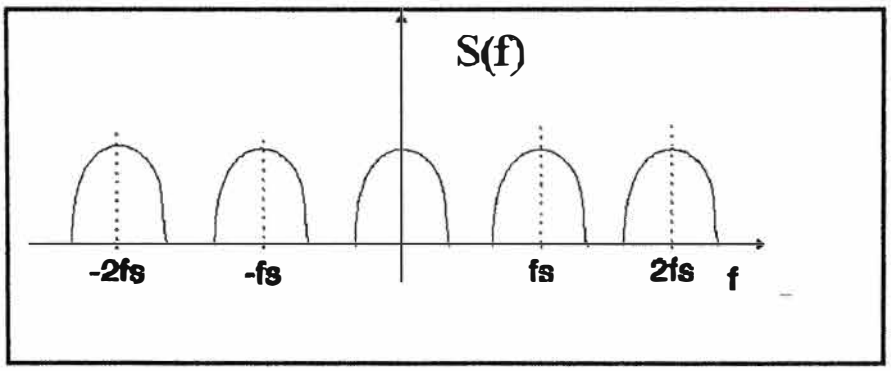


Fig 1.4 Filtrado sin Solapamiento

1.2.2 Muestreo Natural

En la práctica es imposible obtener un muestreador ideal. Técnicamente la función de muestreo es una señal cuadrada con un duty cycle muy pequeño y una amplitud finita

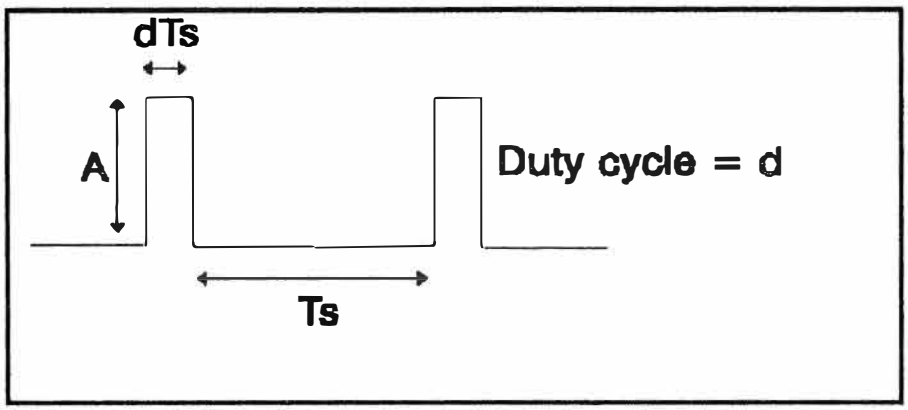


Fig 1.5 Señal de Muestreo Real

Esta función de muestreo periódica puede expresarse mediante su serie de Fourier:

$$m(t) = \sum_{n=-\infty}^{\infty} AdS_a(n\pi d_s)e^{-j2\pi n f_s t} \tag{1.7}$$

siendo la señal muestreada s(t)=m(t)*f(t) y su transformada de Fourier S(f):

$$S(f) = \sum_{n=-\infty}^{\infty} AdS_a(n\pi d)F(f - nf_s) \tag{1.8}$$

Si representamos el espectro para el caso en que fs>=2*W, obtenemos una gráfica similar a la obtenida en el muestreo ideal, pero modificada en amplitud por una función Simple. No obstante se siguen manteniendo las condiciones de recuperabilidad de la señal mediante un filtro paso bajo.

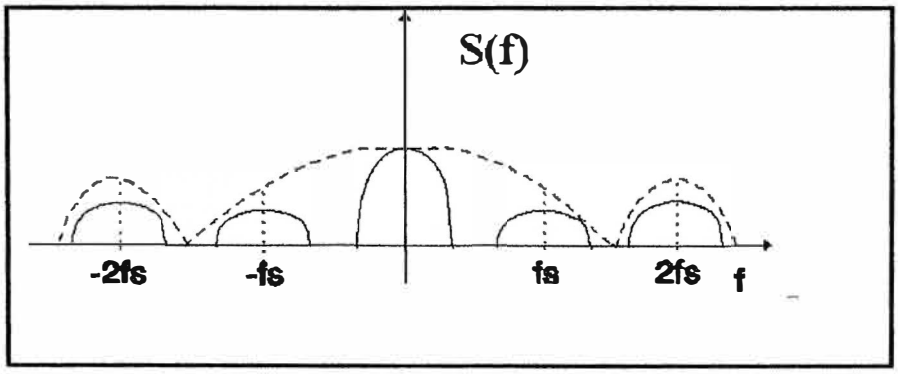


Fig 1.6 Espectro de Señal Muestreada

1.2.3 Frecuencia de Muestreo

De lo expuesto anteriormente se deduce que para poder digitalizar las señales sin perder información es necesario evitar el solapamiento puesto que este efecto no permite la recuperación de la señal ya que el espectro de frecuencia de la señal original se ha modificado. Por lo tanto como regla practica se tiene:

$$f_s >= 2 * W \tag{1.9}$$

donde fs es la frecuencia de muestreo y W el ancho de banda de la señal.

1.2.4 Esquema de la Digitalización

Para la digitalización práctica se utiliza el siguiente esquema:

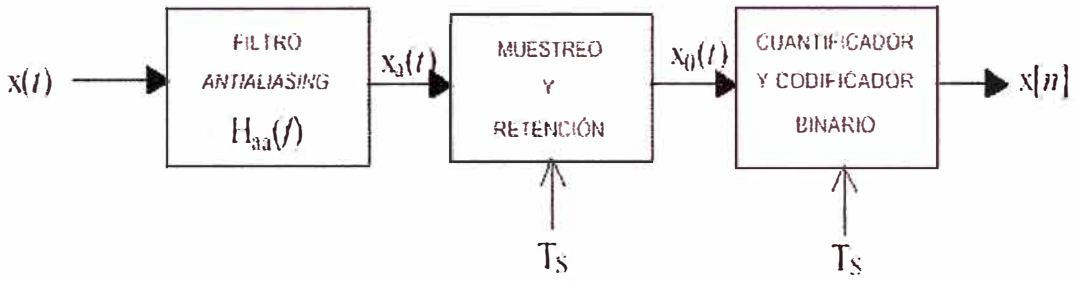


Fig 1.7 Filtrado con Solapamiento

Donde:

x(t) es la señal de entrada

Ts , el periodo de muestreo

x[n], la secuencia de valores digitalizador

La primera etapa, el filtro Antialiasing, permite eliminar los componentes de frecuencias altas de la señal de entrada obteniéndose entonces la señal $x_a(t)$ la cual ya cumple con el criterio de la frecuencia de muestreo, es decir sus componentes de frecuencia son inferiores al mitad de la frecuencia de muestreo.

La segunda etapa la función de muestreo y retención permiten que la señal sea muestreada, y conservada hasta que se complete el ciclo de digitalización, sino las variaciones de la señal de entrada producirán errores de digitalización.

La última el cuantificador y codificador binario en si corresponden con la técnica propia de la digitalización propiamente dicha, en esta etapa una magnitud analógica es convertida en información, en una serie de bits.

1.3 La Transformada Rápida de Fourier

La transformada Rápida de Fourier (FFT) es uno de los métodos más utilizados para la determinación computacional de Transformada Discreta de Fourier (DFT), produce el mismo resultado otros algoritmo pero es increíblemente más eficiente produciendo el tiempo de cómputo en cientos de veces comparativamente con los otros métodos ya que utiliza (N/2)log2(N) multiplicaciones en vez de N*N multiplicaciones, lo que para el caso de N=1024 representa un factor de 200 en el ahorro de tiempo computacional.

La FFT está basada en la de DFT compleja una versión más sofisticada de la DFT real. Es importante entender cómo se transfieren los datos reales al algoritmo o cómo recibir los del formato complejos.

En la DFT real se transforman en N puntos en el dominio del tiempo en dos señales en el dominio de la frecuencia, con N/ 2 + 1 puntos, la señal en el dominio del tiempo se llama justo así señal en el dominio del tiempo y las dos señales en el dominio de la frecuencia se le denomina parte real y parte imaginaria manteniendo las amplitudes de las ondas seno y coseno respectivamente

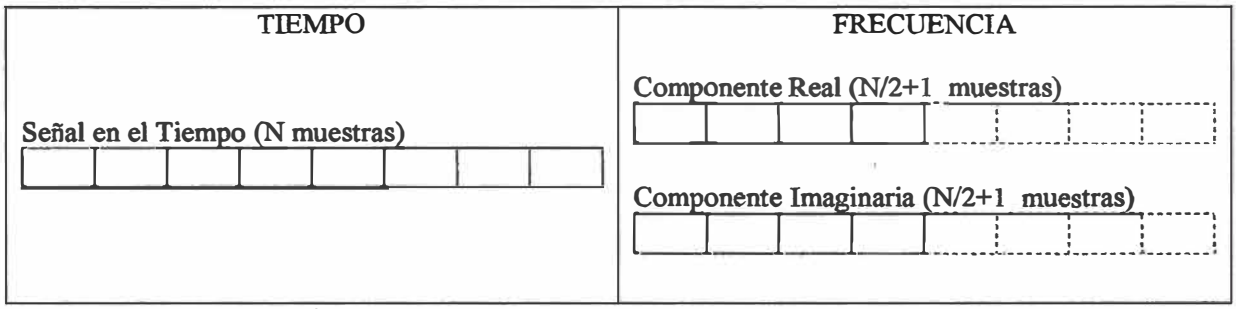


Fig 1.8 Dominio del Tiempo y la Frecuencia de la DFT Real

En la DFT compleja se transforma dos señales en el dominio del tiempo de N. puntos en dos señales en el dominio de la frecuencia con N. puntos. Las dos señales en el dominio del tiempo se le conoce como parte real y parte imaginaria tal como se le denomina a la señales en el dominio de la frecuencia todos los valores de estos arreglos son reales.

Si se tienen una señal de N. puntos y se necesita calcular la DFT real primero debemos mover la señal de N. puntos en la parte real del dominio del tiempo y luego colocar todos los valores de la parte imaginaria en 0. El cálculo de la DFT compleja resulta en una señal en el dominio de la frecuencia con parte real y parte imaginaria cada una de ellas compuestas por N. puntos. Las muestras desde el 0 hasta N/2 de estas señales corresponden al espectro real del DFT.

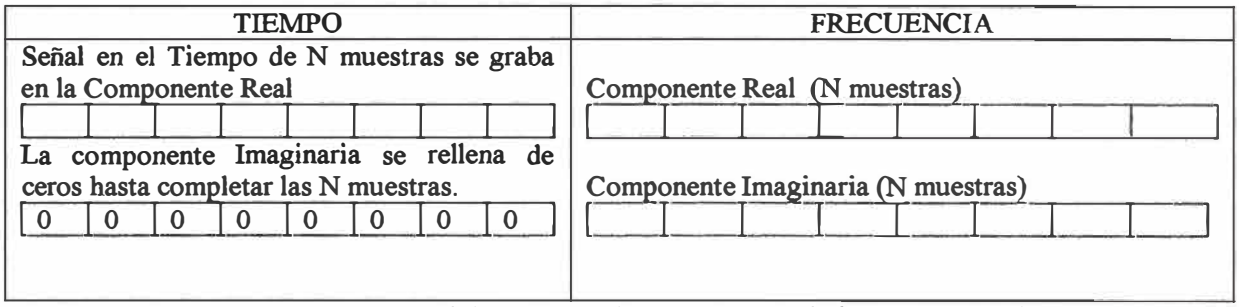


Fig 1.9 Dominio del Tiempo y la Frecuencia de la DFT Compleja

El algoritmo de la FFT se compone básicamente de tres pasos:

Primero: la descomposición de una señal de N. puntos en el dominio del tiempo en otras N. señales en el dominio del tiempo cada una de ellas compuestas por una sola muestra.

Segundo: El calculo del espectro de frecuencia de cada una de la N. señales en el dominio del tiempo.

Tercero: La sintetización de los N. espectros obtenidos en un solo espectro de frecuencia.

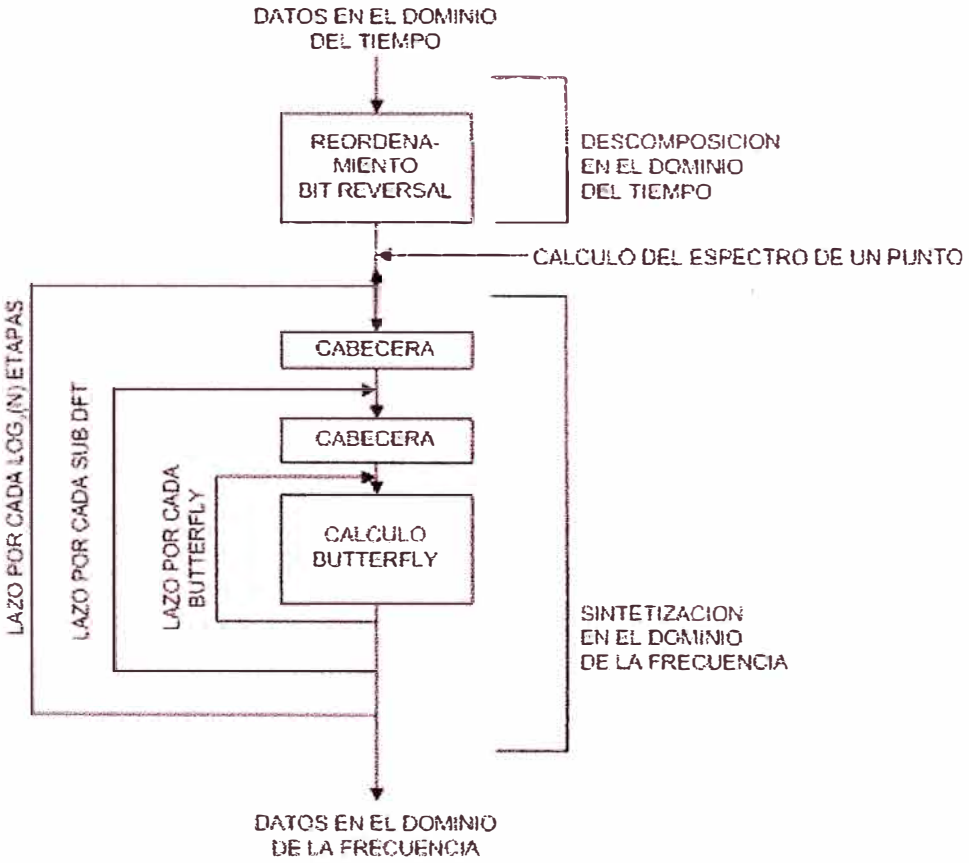


Fig 1.10 Algoritmo FFT

El primer paso, el de la descomposición, se realiza mediante un algoritmo especial por ejemplo una señal de 16 puntos se divide en dos señales de 8 luego cada una de esta señales se divide en dos de 4. Luego cada una de estas en dos señales de 2 puntos para

finalizar con la división de estas últimas en dos señales de un punto cada una. Se utiliza un patrón de descomposición interlazada cada vez que una señal se divide en dos señales con los componentes pares e impares respectivamente. Se requieren Log2N etapas para lograr la descomposición. Este descomposición se simplifica éste se tiene en cuenta que no es más que un reordenamiento de las muestras en la señal comúnmente se utiliza un algoritmo de ordenamiento bit reverso.

La siguiente etapa del algoritmo es hallar el espectro de frecuencia de una señal de un punto en el dominio del tiempo. Esto se simplifica ya que el espectro de una señal de un punto en el dominio del tiempo es igual a sí misma pero en el dominio de la frecuencia esto implica que ya se tiene los componentes espectrales de cada uno de los N puntos de la señal, ya no existen en el dominio del tiempo del tiempo.

La siguiente etapa es combinar los N. espectros de frecuencia en el orden inverso exacto a la descomposición en el dominio del tiempo, esto se logra sintetizando grupos de frecuencia hasta llegar a la unidad por ejemplo 16 espectro de un elemento se combinan logrando ocho espectros de dos elementos cada uno luego se combinarán en pares formando espectros de cuatro elementos luego se combinarán nuevamente formando dos espectros de 8 elementos para finalmente lograr el espectro de 16 puntos. Puesto que la combinación obliga a desplazar muestras para dejar espacio en lugares pares para poder recibir un grupo de muestras ahí, es necesario entonces realizar cálculos que compensen esto, la multiplicación por un sinusoide produce desplazamientos, entonces se tiene que realizar operaciones combinadas con sinusoides. La operación básica que especifica el algoritmo para esta parte se le conoce como Butterfly (mariposa) que es una operación compleja en la cual entran dos puntos complejos y salen dos puntos complejos. Iterando esta operación se puede lograr el efecto de sintetización. Esta es la parte compleja del algoritmo.

Hay que recalcar que el algoritmo básico optimiza el uso de memoria reutilizando los mismos arreglos datos como las salidas, por lo tanto los datos originales ya no existen más luego de ejecutar el algoritmo.

El algoritmo básico de la FFT se ha optimizado para situaciones específicas con el objetivo de reducir el tiempo de cálculo.

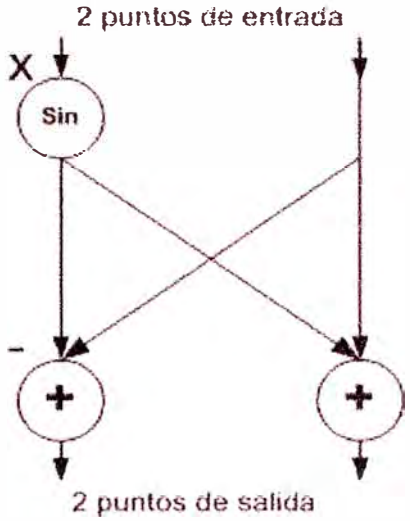


Fig 1.11 Representación del Butterfly

1.4 El ADC 0804

Es un circuito integrado ADC que trabaja con niveles lógicos TTL. Tiene una resolución de 8bits y un tiempo de muestro cercano a 100us. Tiene entradas que permiten interfacear con el micoroproceador. Su salida es de 8 bits.

1.5 D.O.S. Sistema Operativo De Tiempo Real

El DOS es un sistema operativo monolítico por lo tanto las aplicaciones pueden acceder directamente a muchos de los componentes internos de las PC y controlar los efectivamente este es el caso de nuestra aplicación la cual necesita controlar y temporizar acciones en el puerto paralelo.

El DOS, como sistema Monotarea, permite que sea fácilmente utilizable como plataforma de tiempo real debido a que básicamente solo una aplicación se ejecuta a la vez y no hay probabilidad de que sea interrumpida por otra aplicación o proceso, lo que si puede ocurrir en sistemas multitareas (como el Windows) donde se ejecutan concurrentemente varios procesos y se tienen asignadas prioridades de atención del Procesador, generando incertidumbre en cuanto se refiere al momento exacto de ejecutar algún proceso.

Otra gran ventaja del DOS es que no requiere grandes espacios de almacenamiento para ejecutar aplicaciones en muchos casos basta solo disquetes y no los discos duros, además que es bastante eficiente en PC tecnológicamente obsoletas como PC-386 y PC-486.

La gran desventaja del DOS es la relativa dificultad existente para manejar grandes cantidades de memoria.

1.6 El Lenguaje C++ y el Compilador OPEN WATCOM

El Lenguaje C++, evolución del Lenguaje C, permite la creación de programas utilizando técnicas orientadas a objeto. La sintaxis es básicamente la misma que el C tradicional sumándose las extensiones necesarias para la implementación de objetos.

Este lenguaje esta ampliamente difundido y existen una gran variedad de compiladores para ser utilizados en casi todas las plataformas de hardware de PCs existentes.

El compilador OPEN Watcom, permite compilar programas en C++ en entornos de 16 y 32 bits. Además permite crear programas que pueden utilizar los extensores DOS como el DOS4GW, con el cual permite entre otras cosas eliminar las restricciones del DOS en cuanto a memoria además de mejorar el rendimiento de las aplicaciones graficas. Este compilador es OPEN por lo cual es libre de utilizarse sin preocuparse por las licencias.

1.7 La Programación Orientada a Objeto

La programación Orientada a Objetos, permitirá la abstracción de los componentes del sistema creando una clase de objeto para cada componente.

La abstracción permite que la programación se enfoque a problemas específicos, como por ejemplo dibujar una señal en la pantalla sin preocuparnos del origen de los datos de la señal ni de las temporizaciones, puesto que esos problemas lo solucionarán otros objetos del sistema a implementar.

El diseño adecuado de los objetos y su adecuada interrelación nos permitirá crear un software mucho mas estable, de fácil evolución y altamente portable.

1.8 El Puerto Paralelo

El puerto paralelo de la PC, especialmente diseñado para manejo de impresoras, permite la salida y el ingreso de data mediante un conjunto de registros, los cuales pueden usarse como puertos de entrada o salida de información en cualquier entorno de programación.

La interfaz física consta de un conector DB25 hembra ubicado en la parte posterior de la PC.

La forma física y los pines que corresponden a cada registro lo podemos apreciar en el gráfico siguiente

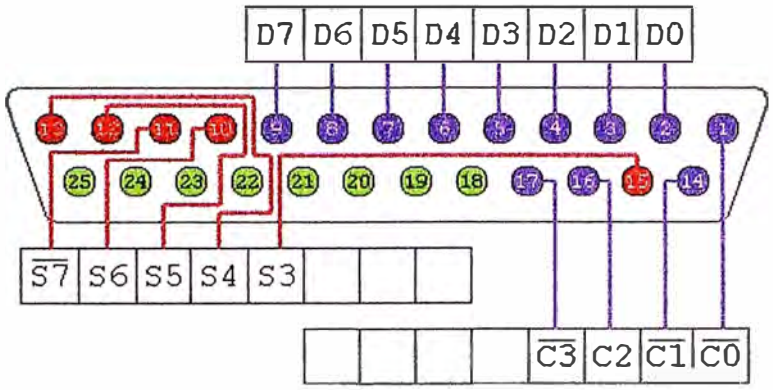


Fig 1.12 Interfaz Paralela de la PC

1.8.1 Registro de Datos

Puerto de salida, está compuesto por 8 bits, y tiene la dirección 378h ó 278h ó 3BCh, la siguiente tabla muestra la relación de señales del puerto y bits del registro y el respectivo pin en el puerto físico..

Tabla N° 1.1 Registro de Datos

BIT	SEÑAL	PIN
0	D0	2
1	D1	3
2	D2	4
3	D3	5
4	D4	6
5	D5	7
6	D6	8
7	D7	9

1.8.2 Registro de Control

Puerto de salida, permite enviar información de control, está compuesto por 4 bits y tiene la dirección 37Ah ó 27Ah ó 3BEh, la siguiente tabla muestra la relación de señales del puerto y bits del registro y el respectivo pin en el puerto físico.

Tabla N° 1.2 Registro de Control

BIT	SEÑAL	PIN
0	C0 negado	1
1	C1 negado	14
2	C2	16
3	C3 negado	17

1.8.3 Registro de Estado

Puerto de entrada, permite el ingreso de data, esta compuesto por 5 bits y tiene la dirección 379h, 279h, 3BDh. la siguiente tabla muestra la relación de señales del puerto y bits del registro y el respectivo pin en el puerto físico

Tabla N° 1.3 Registro de Estado

BIT	SEÑAL	PIN
0	--	--
1	--	--
2	--	--
3	S3	15
4	S4	13
5	S5	12
6	S6	10
7	S7 negado	11

1.9 Temporización en la PC

La PC tiene como parte de su arquitectura un Temporizador de Intervalo Programable (PIT) 8254 el cual es utilizado para temporizar eventos del sistema como el reloj y el refresco de la memoria. Este PIT tiene internamente tres contadores independientes que se alimentan con un reloj de 1.19318MHz y trabajan dividiendo esta frecuencia entre el numero que se le proporciona como entrada en cada uno de sus registros de 2 bytes del respectivo contador.

En forma normal el contador 0 del PIT trabaja en modo tres y divide la frecuencia de entrada entre el máximo numero permisible que es 65536 logrando una frecuencia de 18.207 Hz, la cual es utilizada para sincronizar el reloj del sistema y llevar la secuencia de la hora y fecha de la PC. El contador 1 se utiliza para sincronizar los ciclos de refresco de los bancos de memoria DRAM. El contador 2 se utiliza para el manejo de tonos en el parlante interno de la PC, mediante la reprogramación de este contador es que se lograban los primeros sonidos en las PCs. Este PIT se puede manipular mediante el uso de los puertos asignados a sus componentes: En la tabla siguiente se muestra la dirección de los puertos, a que contador están asignados.

Tabla 1.4 Puertos del PIT, Componentes

Puerto	Componente	Utilidad principal
40h	Contador 0	Reloj del sistema (Modo 3)
41h	Contador 1	Refresco de memoria (Modo 2)
42h	Contador 2	Función de Parlante y Cassete
43h	Registro de Control	Programación del PIT

La reprogramación del PIT se lleva a cabo enviando la configuración adecuada hacia el puerto del registro de control (43h)y los datos necesarios hacia el puerto por configurar
En la tabla siguiente se muestra los valores que pueden tomar los bits de control para seleccionar tipo de contador, modos, las lecturas y que contador utilizar.

Tabla 1.5 Uso del Registro de Contro l

Bit	Valor	Efecto/Control
0	0	Contador Binario
	1	Contador BCD
3,2,1	0,0,0	Modo 0
	0,0,1	Modo 1
	X,1,0	Modo 2
	X,1,1	Modo 3
	1,0,0	Modo 4
	X,1,1	Modo 5
7,6	0,0	Latch del Contador
	0,1	Byte Inferior
	1,0	Byte Superior
	1,1	2 Bytes
7,6	0,0	Timer 0
	0,1	Timer 1
	1,0	Timer 2
	1,1	Bit 6, Bit5, Bit4: 1,0,1→ Latch general de Contadores 1,1,0 → Latch Estatus de los Timers

CAPITULO II
REQUERIMIENTOS

2.1 Planteamiento del Problema

El problema que va solucionar el presente proyecto se puede plantear especificando los límites o parámetros y requerimientos mínimos a cumplirse, los cuales se especifican a continuación.

2.2 Frecuencia de Muestreo

El objetivo es analizar señales provenientes de la red eléctrica la cual tiene nominalmente una frecuencia de 60Hz, si se requiere obtener hasta el 40avo armónico se tiene entonces que la frecuencia límite es de 2400Hz, por lo tanto considerando el teorema de muestreo debemos tener como frecuencia de muestreo un valor superior a 4800Hz.

2.3 El Rango Nominal de Voltajes

El rango nominal de los voltajes de línea es aproximadamente 623 voltios (voltaje pico a pico ideal), pero considerando la presencia de sobre tensiones y picos se puede establecer un rango cercano a los 1000 voltios.

Si considerando una resolución de uno por ciento se tendrá que el error máximo en la medida de 6.23 voltios.

2.4 Resolución de Frecuencia

La normas eléctricas nacionales indican medidas con variaciones de 0.6Hz máximo por lo tanto consideraremos como resolución mínima ese valor que representa al 10% de la frecuencia fundamental.

2.5 Portabilidad

El sistema a implementar debe ser portátil y poder ser activado en cualquier PC, es decir no debe estar limitado por la velocidad de la PC..

2.6 Precisión de Reloj

La temporización del sistema debe ser lo suficiente para que el error de la muestra en frecuencia sea mínimo. La norma establece una precisión de 1E-7s.

2.7 Interface con otros Programas

Los datos que se procesen, e adquieran o generan deben poder ser exportados o importados hacia otros programas (como el MatLab) para contrastar resultados o realizar otros procesamientos.

2.8 Procesamiento en Línea

El sistema debe poder procesar datos en línea es decir muestrear y procesar simultáneamente.

2.9 Interfaz Amigable

El programa debe poder ser fácilmente manejado, así como también deben ser claros los resultados obtenidos.

CAPITULO III
DISEÑO E IMPLEMENTACION DE LA SOLUCION

3.1 Soluciones al Problema Planteado

3.1.1 Frecuencia de Muestreo

Se ha escogido una frecuencia de muestreo de 5Khz, suficiente para cubrir el límite del problema.

3.1.2 El Rango Nominal de Voltajes

Se va a muestrear a 256 sobre un rango de 1000 voltios, lo cual produce un error de 4 voltios, lo cual representa a un error aproximado de 0.65%.

3.1.3 Resolución de Frecuencia

La cantidad de muestras para cumplir con el problema debe ser: 5000/0.6 es decir 8333.3 muestras. Debido a que se va a usar el algoritmo de la FFT, se tendrá lo siguiente:

N=1024, Fm=5Khz, resolución = 4.88Hz

N=2048, Fm=5Khz, resolución = 2.44Hz

N=4096, Fm=5Khz, resolución = 1.22Hz

N=8192, Fm=5Khz, resolución = 0.61Hz

N=16384, Fm=5Khz, resolución = 0.31Hz

La cantidad de muestras debe ser un parámetro del sistema para poder acomodarse a las diversas situaciones y también a la memoria de la PC.

3.1.4 Portabilidad

Se ha escogido que la interfaz sea controlada por el puerto paralelo de la PC y además la temporización sea independiente de la velocidad de la PC, por lo tanto se logra el objetivo de que el sistema no dependa de una PC en particular y por lo tanto puede ser portátil, además la temporización también tratará de ser independiente de la velocidad de la PC..

3.1.5 Precisión de Reloj

La temporización implementada en el sistema solo alcanza la resolución de 1E-6 seg., ya que el reloj interno de la PC a utilizarse tiene un frecuencia de 1.22Mhz.y además es controlada por el programa.

3.1.6 Interfase con otros Programas

El programa tendrá opción de grabar y leer datos en formato ASCII de todas las señales utilizadas, esto permitirá importar o exportar datos a Hojas de Calculo como Excel, o mediante pequeños programas al Matlab.

3.1.7 Procesamiento en Linea

El sistema se basará en captura y procesamiento de datos en forma alternada, en ciclo que pueden programarse o durar en forma indetermina. Primero se captura la información, luego se procesa y luego se visualiza.

3.1.8 Programa Amigable

El uso de menús, y un entorno grafico básico permitirá al usuario escoger adecuadamente las acciones a realizar así como también visualizar los gráficos y datos necesarios para que el usuario sepa en todo momento en que estado esta el programa. Además se tratará en lo posible de parametrizar las variables para que los datos de configuración del sistema se ingresen vía archivo de textos.

Para la implementación se ha el uso del OPEN WATCOM como entorno de desarrollo (en Windows) y el producto de las compilaciones se harán para DOS con extensor DOS4GW. Esto nos permitirá aplicar la programación orientada a objeto al mismo tiempo que generará aplicaciones que puedan funcionar eficientemente en DOS.

3.2 Estructura del Sistema de Adquisición de Datos

En base a la solución planteada en punto anterior se ha diseñado un Interfase y un Programa al cual se le ha denominado ANALIZA, los cuales cumplirán con los objetivos planteados.

El hardware consiste en un sistema digitalizador de señales de ocho bits el cual se comunica con la PC mediante el puerto paralelo por el cual enviará datos y al mismo tiempo recibirá señales de control.

El diseño del software se ha hecho tomando en cuenta como criterio fundamental el ser portátil por lo cual sus parámetros no depende en gran medida de hardware por lo tanto la sincronización por ejemplo funcionará con igual precisión de una PC 486 como en una PC Pentium 4. La velocidad de la PC influye básicamente en la rapidez de los procesos de visualización y análisis.

En el diagrama de bloques se puede observar que el sistema utiliza los registros del puerto de impresora para recibir datos desde la interfaz así como controlar el proceso de digitalización.

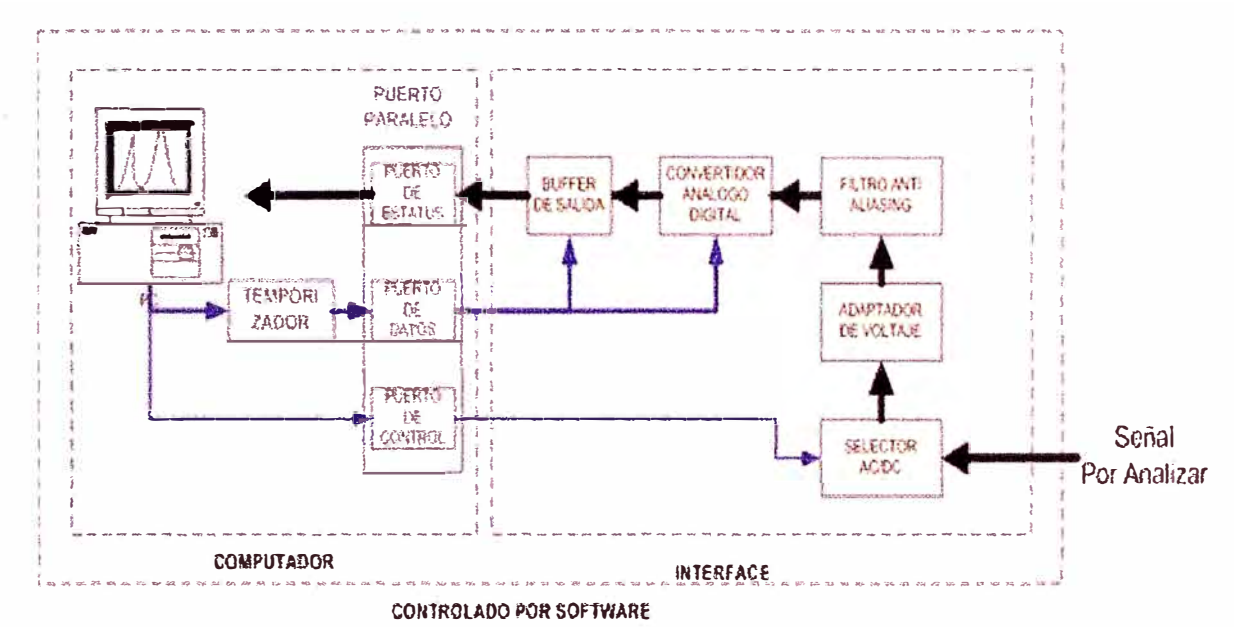


Fig 3.1 Diagrama de Bloques del Sistema de Adquisición de Datos

La sincronización se lleva a cabo mediante reprogramación del Timer de la PC del cual funciona a la misma velocidad en cualquier PC.

La señal tiene que ser adaptada para su digitalización debido a que el rango de digitalización es positivo es necesario adaptar las señales AC para este dentro del rango de digitalización de no ser así solo se digitalizará la parte positiva de la señal.

Asimismo es necesario la presencia de filtros en la señal de entrada para evitar el solapamiento de la señales muestreados con los componentes de alta frecuencia de la señal.

También se ha implementado un generador de señales que utiliza la misma técnica de sincronización y una red resistiva R-2R para obtener el voltaje analógico. Este generador se ha implementado para realizar pruebas mediante el uso de dos PCs, una funcionando como generador y el otro realizando la adquisición de datos.

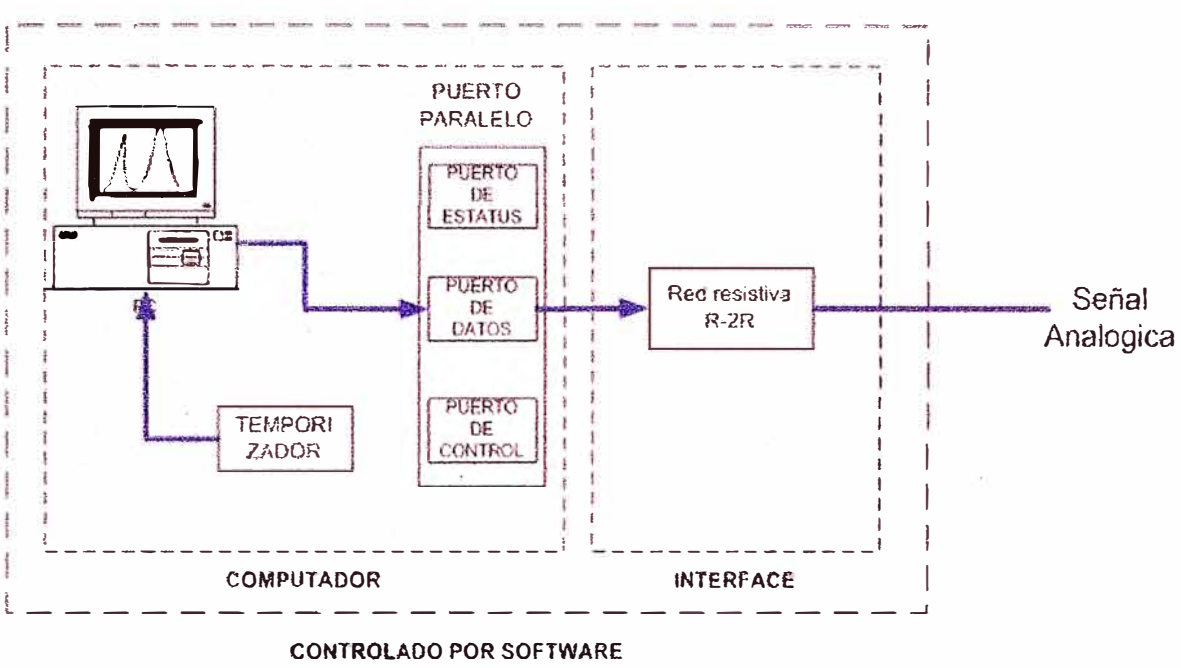


Fig 3.2 Diagrama de Bloques del Generador

A continuación detallamos las soluciones a los problemas puntuales que se han presentado en el desarrollo de la implementación.

3.2.1 Implementación de la Temporización

El temporizador de la PC el PIT 8254 (Programmble Interval Timer) de la PC está programado a una frecuencia aproximada de 17 Hz, la cual es inadecuada para nuestro proyecto por lo tanto se ha requerido reprogramar el temporizador a su máxima frecuencia (1.19318Mhz) para de esta manera usarlo como base de tiempo para la digitalización. Esta frecuencia nos permite lograr una resolución aproximada de un microsegundo..

Se ha escogido el contador 0 para la reprogramación. Con las siguientes instrucciones en lenguaje C, se le coloca en modo 2, se habilita la lectura sólo del Byte inferior del registro de datos y se inicia la lectura "latcheada".

```
//Inicializa el contador 0 en modo 2 y lectura del byte inferior
 outportb(0x43,0x14);
 outportb(0x40,0x00);
 //Inicia la lectura latcheada
outportb(0x43,0xD2);
```

Se ha escogido sólo capturar el dato del byte inferior del contador puesto en nuestra frecuencia de muestreo de 5 khz requieren sólo de 200us por lo cual el contador no

sobrepasará el conteo máximo que puede hacer el byte inferior que es 255 que multiplicado por 0.8381 nos da 213us como tiempo máximo.

Cada dato que es digitalizado se hace dentro de un lazo en el cual se va controlando el contador 0 del temporizador. Se realiza primero la digitalización y luego se espera mediante lazos sin operaciones hasta que se llegue a completar los 200us del ciclo de digitalización.

Los siguientes extractos del programa muestran la lectura del byte inferior del contador y la forma como se determina el tiempo transcurrido en unidades del Timer.

```
t_inicio=inportb(0x40);    //Lee tiempo inicial

t_actual=inportb(0x40);  //Lee tiempo actual
if (t_actual>=t_inicio)
   return t_inicio+(256-t_actual);
else
   return t_inicio-t_actual;
```

3.2.2 Voltaje de Referencia del ADC

La nivel del voltaje de referencia para la digitalización es necesaria que sea altamente estable, aunque el circuito integrado digitalizador 0804 proporciona un voltaje de referencia bastante precisa es necesario mejorarla, para ello se ha utilizado un CI generador de Voltaje de Referencia de alta precisión a 2.50 voltios, que es el KA 336 2.5. Su polarización se realiza R4 y el voltaje de referencia producido ingresa al pin 9 del ADC0804.y representa a Vrango/2

3.2.3 Nivel 0 de la Señal de Entrada

Para el caso del presente sistema es necesario referenciar el voltaje alterno que se obtiene de la linea eléctrica puesto que la red de resistencias producirán un voltaje diferencial prácticamente flotante.

Tomando los 2.5V del voltaje de referencia de la digitalización y pasándolo por un amplificador de ganancia unitaria se tiene un voltaje 2.5 estable y de baja resistividad, lo cual nos permite simular una tierra "virtual" a 2,5v por encima de los 0 voltios del digitalizador. De esta manera al colocar uno de los extremos del divisor de resistencia en esta tierra "virtual" se fija a 2,5v y el otro extremo tendrá con respecto a este nivel valores positivos y negativos siendo en realidad todos positivos.

Si se tiene una señal referenciada a la misma tierra que el digitalizador entonces no será posible colocar el nivel de tierra de la señal a la tierra virtual porque se producirá un cortocircuito en la salida del amplificador operacional que maneja la referencia el cual intentará mantener los 2.5v mientras que al conectar la tierra está forzándose a 0 v. Para evitar este problema se ha utilizado un relé el K1, el cual es comandado por el transistor Q1 y que a su vez recibe ordenes de la salida C0, el relé conectará el nivel de tierra de la señal a la tierra del digitalizador o al nivel de tierra virtual de 2.5v para señales flotantes, según la orden que se le dé.

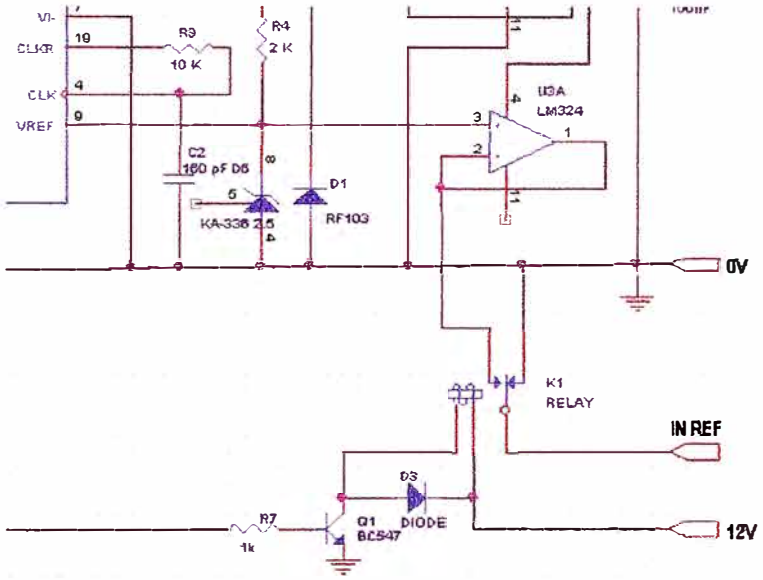


Fig 3.3 Circuito Adaptador de Referencia

3.2.4 Toma de Señal

Se han considerado un divisor resistivo para el muestreo del voltaje de la línea AC de la siguiente forma:

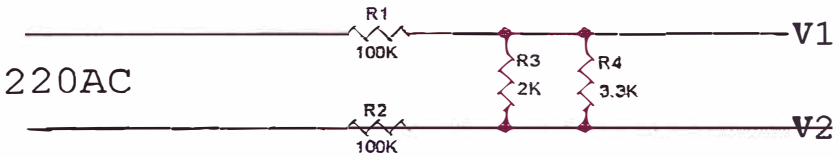


Fig 3.4 Sonda para AC de Linea

Se ha determinado experimentalmente que V2-V1 es 0.0063356 del voltaje de entrada. Esto se tomará en cuenta para el factor de conversión en el programa procesador de las señales digitales.

El voltaje obtenido por este arreglo resistivo representa prácticamente un voltaje flotante ya que representa la diferencia de voltajes existe en los extremos de las resistencias de salida.

Esta entrada es conectada a un amplificador operación de ganancia unitaria lo cual significa que la impedancia de entrada del sistema digitalizador es bastante alta y no influenciará en el factor de proporción.

3.2.5 Filtro Antialiasing

Colocando un condensador en la entrada de señal se tendrá el efecto de filtrado necesario para evitar el solapamiento por componentes de alta frecuencia. Aplicando el equivalente Thevenin al circuito de la sonda de voltaje se tiene el siguiente circuito RC el cual actua como un filtro pasabajos. El amplificador operacional de entrada no presenta carga a la entrada por lo cual se le obviará para el cálculo del condensador.

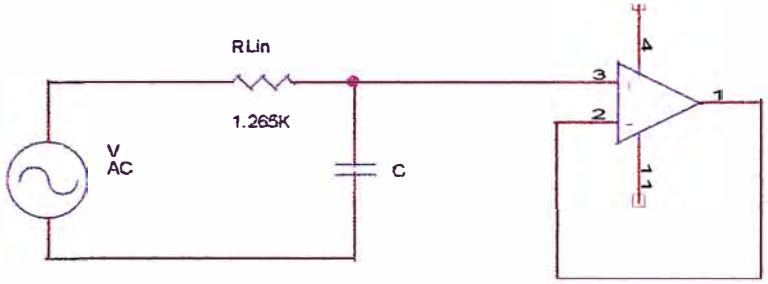


Fig 3.5 Filtro de Entrada

El cálculo del condensador de filtro se hará con la siguiente fórmula:

$$C = \frac{1}{2\pi R_m f_c} \qquad (3.1)$$

Donde fc es la frecuencia de corte del filtro y Rm en la resistencia en serie con el condensador. Si consideramos: fc =10Khz y Rm=1.2K, entonces se tiene que C=13nF

3.2.6 Digitalización

Para la digitalización se ha utilizado el integrado ADC-0804 conectado en modo libre esto significa que el término de inicio de cada ciclo de digitalización está controlado por sí mismo. El control sobre este convertidor se fundamenta únicamente en la deshabilitación o habilitación de la digitalización.

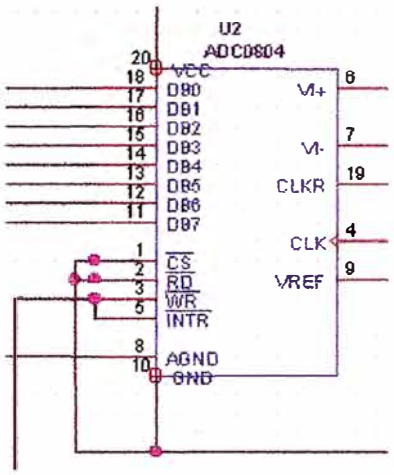


Fig 3.6 C.I. Digitalizador

Los pines 1 (Chip Select) y 2 (Read) están conectados a tierra lo cual indica que la lectura y la selección del integrado están habilitados los pines 3 y 5 están unidos lo cual permite el funcionamiento en modo libre. Cuando una digitalización está completa INTR cambiar de estado lo cual produce que WR habilite la salida del digitalizador. Para asegurar el funcionamiento es necesario que WR esté momentáneamente a tierra, esto se logra mediante el bit D4 del registro de datos del puerto paralelo. Al mismo tiempo sirve de control para la digitalización puesto que mientras WR esté conectado a tierra el digitalizador no trabajara.

Para poder ingresar los datos digitalizados al computador ser utilizada del registro de estado (S7 al S3) del puerto paralelo por ser de entrada. Puesto que sólo hay cinco bits en este registro es necesario entonces ingresar el byte obtenido en dos partes para ello se utiliza el driver de línea el 74244 el cual permitirán el ingreso de cuatro bits alternadamente mediante dos bits de control los cuales se asignarán a D0 y D1 del registro de datos.

En resumen para obtener una muestra es necesario primero activar el ADC mediante el bit D4 luego que la digitalización está completa se activan los bits D0 y D1que habilitan las entradas 1G y 2G respectivamente en forma alternada para poder llevar a cabo la lectura de las partes del byte que proporciona el ADC. El programa se encargará de acomodar los bits mediante corrimientos y luego los sumará para obtener el byte que corresponde a la muestra.

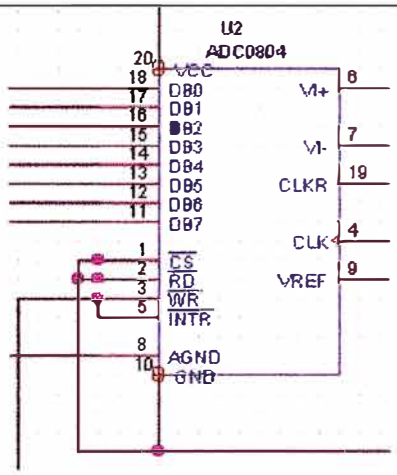


Fig 3.7 Interfase con el Puerto Paralelo

A continuación un extracto del código que permite obtener una muestra:

```
#define iniciar_adc     16    // 010000
#define detener_adc      0    // 000000
#define leer_nibble1    14    // 001110
#define leer_nibble2    13    // 001101
. . .
{  outp(LPT1,leer_nibble1+iniciar_adc);
  nibble1=15&(inp(LPT1+1)>>3);
  outp(LPT1,leer_nibble2+iniciar_adc);
  nibble2=240&(inp(LPT1+1)<<1);
  outp(LPT1,leer_nibble1+detener_adc);
  retar.retardo(t_adc_off);
  outp(LPT1,leer_nibble1+iniciar_adc);
  tic.continuar();
  return nibble1+nibble2;
 while (tic.transcurrido()<tictac);
}
```

3.2.7 Frecuencia de Muestreo

El ADC 0804 digitaliza, en promedio, en 100us lo cual da un límite de 10Khz para la frecuencia de muestreo. Se ha considerado sólo 5Khz como frecuencia de muestreo lo cual es superior al doble de 2400Hz que implica la frecuencia del 40vo armónico de la señal de línea de 60Hz, con lo cual aseguramos el muestreo hasta este armónico.

3.2.8 Resolución y Precisión

La resolución de ocho bits nos permite manejar precisiones de 4 voltios si consideramos que la amplitud de la señal a muestreo corresponde a un rango de +/ -500 voltios.

3.2.9 Remoción de la Componente DC

Para el caso de la señal de línea es necesario que no haya variaciones debido a la tierra virtual empleada para hecho se deben de remover en forma permanente los niveles de componente DC y de esta manera el espectro de frecuencias será mucho mas estable y limpio.

El algoritmo básico consiste en restar el valor medio de la señal a los valores muestreados en cada ciclo de muestreo, de esta manera los valores obtenidos solo reflejaran los componentes AC ya que el DC que corresponde al valor medio ha sido removido.

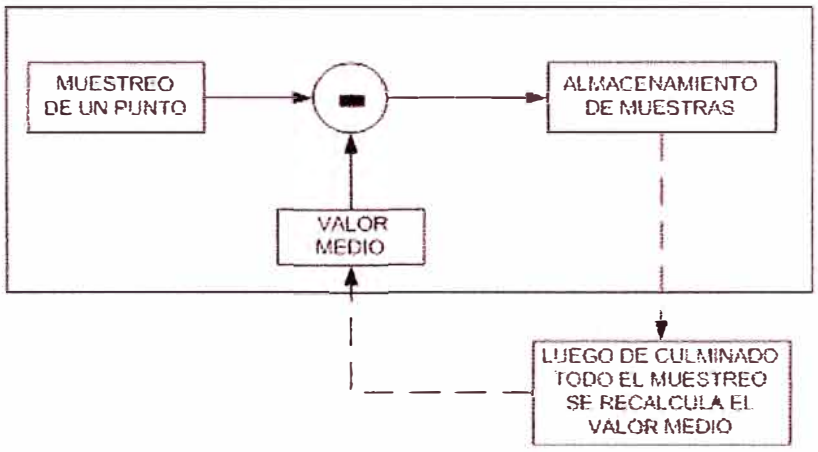


Fig 3.8 Removedor de DC

3.2.10 Sincronización para el Muestreo

Para sincronizar la muestra y presentar una imagen estable de las señales digitalizadas en el visor se ha seguido el siguiente algoritmo:

Se realiza un primer muestreo normal de la señal y se visualiza normalmente. Al finalizar esta primera muestra. se tendrá definido el valor medio de la señal lo cual se usará como valor umbral para el disparo del sincronismo.

En los siguientes muestreos lo primero que se realiza es el procedimiento de sincronismo que consiste en realizar muestreos sin almacenamiento hasta que se encuentre el valor umbral. Para asegurar el disparo en una sola dirección, es decir solo cuando la señal pasa de abajo para arriba, por ejemplo, es necesario realizar muestreos sin almacenamiento

mientras las señal sea mayor que el valor de disparo, luego se asume que la señal tiene valores menores al valor medio se sigue tomando muestras hasta que llegue a superar el valor de disparo, es en ese instante cuando se produce el disparo y se inicia el proceso de digitalización con almacenamiento de datos.

3.2.11 Ventanas

para disminuir el efecto del muestreo y el uso de la ventana rectangular implica la muestra tomada de un determinado intervalo de tiempo es a veces necesario el uso de la ventanas las cuales son funciones que al multiplicar por la señal en análisis permite minimizar ruidos y maximizar la respuesta en frecuencia de la señal.

Se han considerado la ventanas de:Hanning, Flattop, Blackman y Kaiser, las cuales se podrán escoger para aplicarlas a la señal muestreada.

A continuación las graficas de las ventanas generadas por el programa diseñado los algoritmos de formación han sido implementados en el módulo correspondiente al generador de señales.

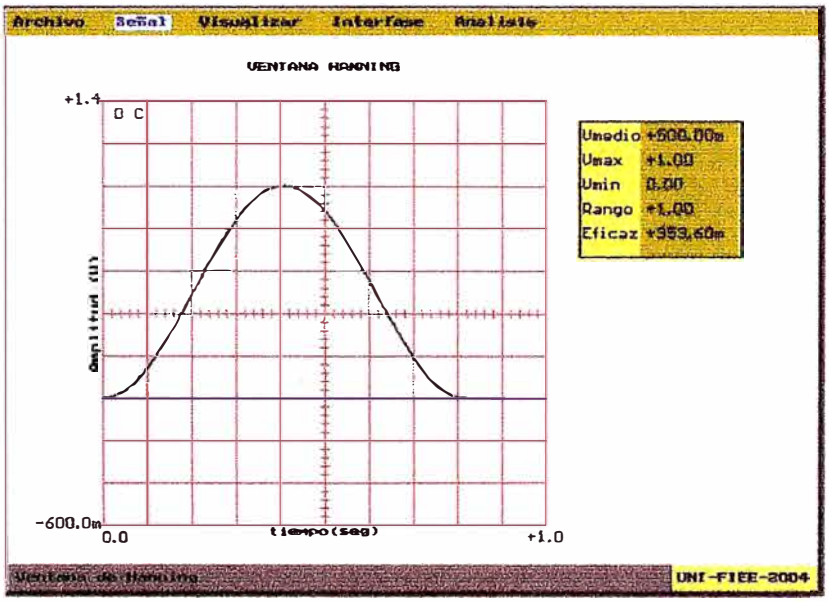


Fig 3.9 Ventana de Hanning

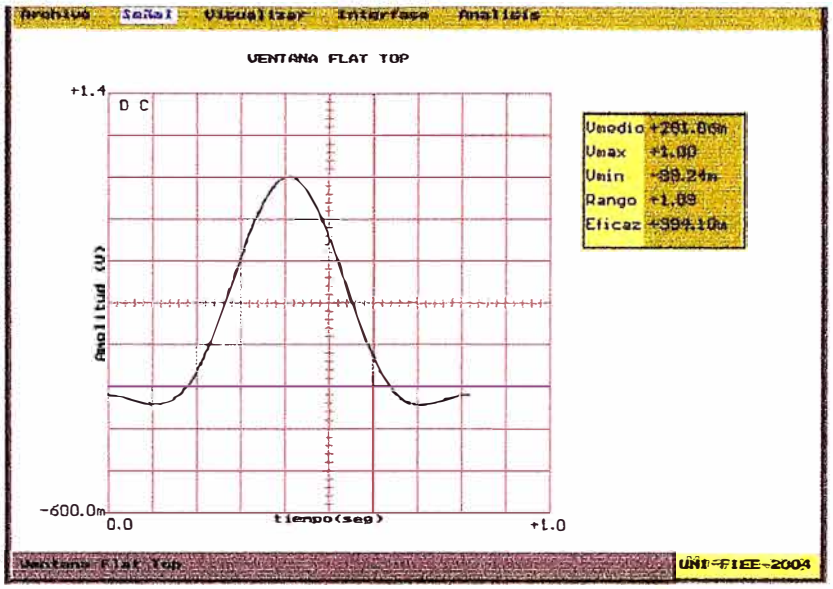


Fig 3.10 Ventana Flat Top

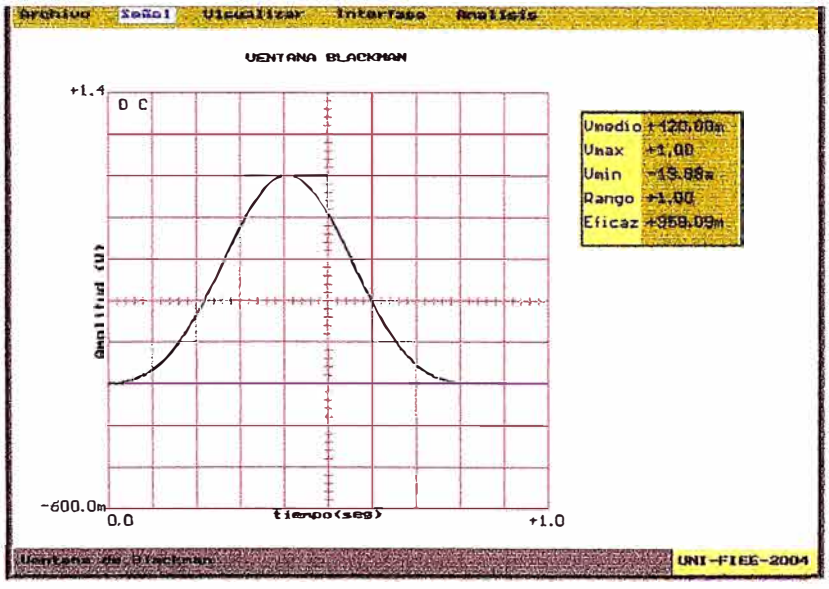


Fig 3.11 Ventana Blackman

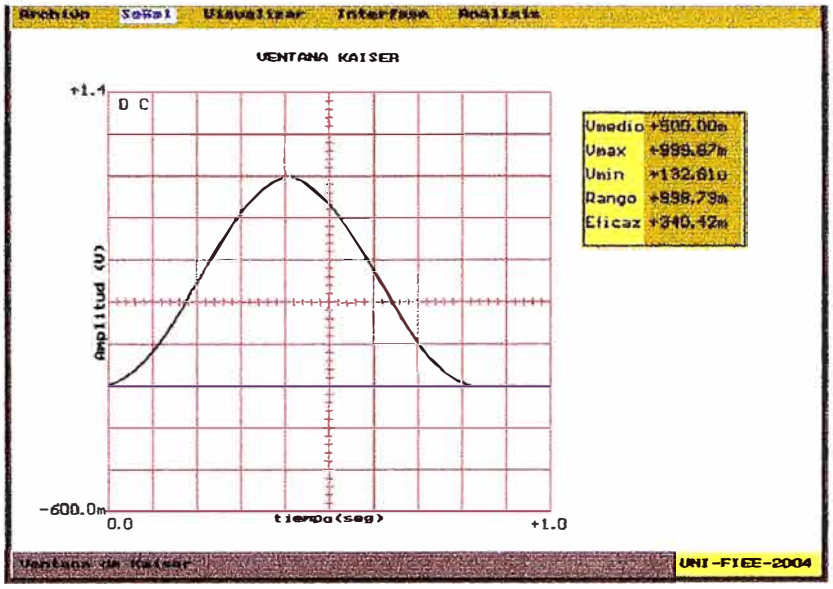


Fig 3.12 Ventana Blackman

3.2.12 Clases De Objetos

El diseño del sistema se ha hecho pensando en las partes de un osciloscopio digital, la abstracción de las funciones permitió el diseño de cada una de las clases a utilizarse

A continuación una lista de las principales clases del sistema:

INTERFASE

Objeto que se encarga de controlar el interfase que muestrea la señal objeto del análisis así como también se encarga de controlar el Generador de ondas.

Archivos: INTERFAZ.H , INTERFAZ.CPP

ARCHIVO

Objeto que se encarga de manejar los archivos tanto de datos para los objetos a crearse como de salida para el interfaz del sistema con el MatLab.

Sus métodos permiten abrir, cerrar, reaperturar archivos y realizar la lectura de diversos tipos de datos desde archivos texto.

Archivos: ARCHIVO.H , ARCHIVO.CPP

SENHAL

Objeto que contiene una secuencia de números correspondiente a una señal capturada o generada por software tanto para el convertidor digital analógico como para las ventanas. Sus métodos comprenden los algoritmos de operaciones con señales como: resta, suma, división y multiplicación de señales, multiplicación de señal por escalar y los algoritmos fundamentales del FFT e IFFT.

Archivos: SENHAL.H , SENHAL.CPP

TIEMPO

Objeto que se encarga de manejar el temporizador de la PC. Es el temporizador del sistema. Sus métodos permiten iniciar y detener ciclos de conteo del temporizador.

Archivos: TIEMPO.H , TIEMPO.CPP

VISOR

Objeto que permite la visualización en pantalla de un visor tipo osciloscopio y e el cual se dibujarán señales digitalizadas, generadas.y procesadas. Sus métodos permiten dibujar el visor, almacernarlos en memoria, dibujar señales encima del visor.

Archivos: VISOR.H , VISOR.CPP

PANEL

Objeto que se encarga de configurar la presentación de las señales en el visor.

Archivos: PANEL.H , PANEL.CPP

INTERFAZ

Objeto que se encarga de controlar el ADC para el analizador y también el DAC para el generador de señales.

Adicionalmente se tiene clases accesorias como MENU, CADENA, INGREDAT, SISTEMA, GRAFICO, que son utilizadas para el manejo interactivo del programa.

3.2.13 Manejo de Memoria

El manejo de la memoria ha sido uno de los principales problemas en la implementación del programa. La limitación que impone el DOS al uso de memoria extendida limita el tamaño de los objetos se almacenan datos de las señales por ello es tiempo de almacenamiento logrado es relativamente bajo.

Se ha hecho uso de punteros y reserva dinámica de memoria para optimizar la memoria existente.

Puesto que se esta compilando con el Open Watcom configurado para crear aplicaciones para DOS con el extensor DOS4GW, es posible el uso de mas de 1Mby de memoria, lo cual permite grabar grandes cantidades de muestras en memoria,

3.2.14 Archivos de Configuración

El programa considera para lograr cierto grado de adaptación personalizada el ingreso de los principales datos o parámetros del programa en archivos que se le ha momento de iniciar la ejecución. Estos son:

MENÚ.CFG, que guarda las opciones del menú del sistema.
VISOR.CFG, que guarda los parámetros del visor quie simula la apariencia de osciloscopio.
SENHAL.CFG que guarda los parámetros de las señales como cantidad de muestras, periodo de muestro.
ADC.CFG que guarda los parámetros para la digitalización así como del DAC.

3.2.15 Archivos de Entrada/Salida de Datos

Para el interfaz con otros programas como el MatLab se ha pensado en la capacidad de leer y escribir datos de las señales en archivos en formato ASCII
Se tienen los siguientes archivos de entrada:
- SEÑAL.IN, - REAL.IN, - IMAGIN.IN, - VENTANA.IN, - LINEA.IN
se tienen los siguientes archivos de salida:
- SENHAL.OUT, - REAL.OUT, - IMAGIN.OUT, - LINEA.OUT
El formato de los archivos es como sigue en las cuatro primeras líneas están los datos de la cantidad de muestras y el periodo de muestreo y en es restantes los datos correspondientes a cada muestra empezando desde la muestra 0, por ejemplo:

```
Numero de Elementos::
4096
Periodo de Muestreo:
0.000200
2.500000
2.656976
2.813333
```

3.2.16 Programa ANALIZA

Como resultado de todo el proceso de programación se tiene el archivo PROYECTO.CPP es el programa principal del sistema maneja todos los objetos creados para nuestro objetivo.. El ejecutable resultante es llamado ANALIZA, utiliza un menú principal a partir del cual se realizan todos los procesos. No usa mouse.
Las funciones implementadas en ANALIZA son:

Leer Archivos de Datos de Señales
Escribir Archivos con Datos de Señales
Generar señales con el modulo DAC
Capturar señales con el modulo ADC (DC, AC-linea, AC-In)

Generar internamente señales para simulación.

Efectuar operaciones con señales: Sumas, Restas, multiplicación y división por escalar, Multiplicación de señales (usado para aplicar ventanas)

Visualizar en diferentes escalas las señales, manejado por teclado.

Aplicar FFT e IFT a señales.

CAPITULO IV
PRUEBAS Y CALIBRACIONES

4.1 Contrastación de Algoritmo FFT con el MatLab

Puesto que el componente principal del sistema es el Algoritmo FFT se realizaron contrastaciones con el MATLAB para verificar la eficiencia y precisión de nuestra implementación

La prueba se realizó como sigue:

Primero, en el MATLAB se muestreó una señal senoidal de 50Hz con 5 de amplitud y 0 en componente DC, teniendo como datos del muestreo 5Khz y N=4096.

A continuación un detalle del el grafico de la FFT de la señal muestreada en MATLAB

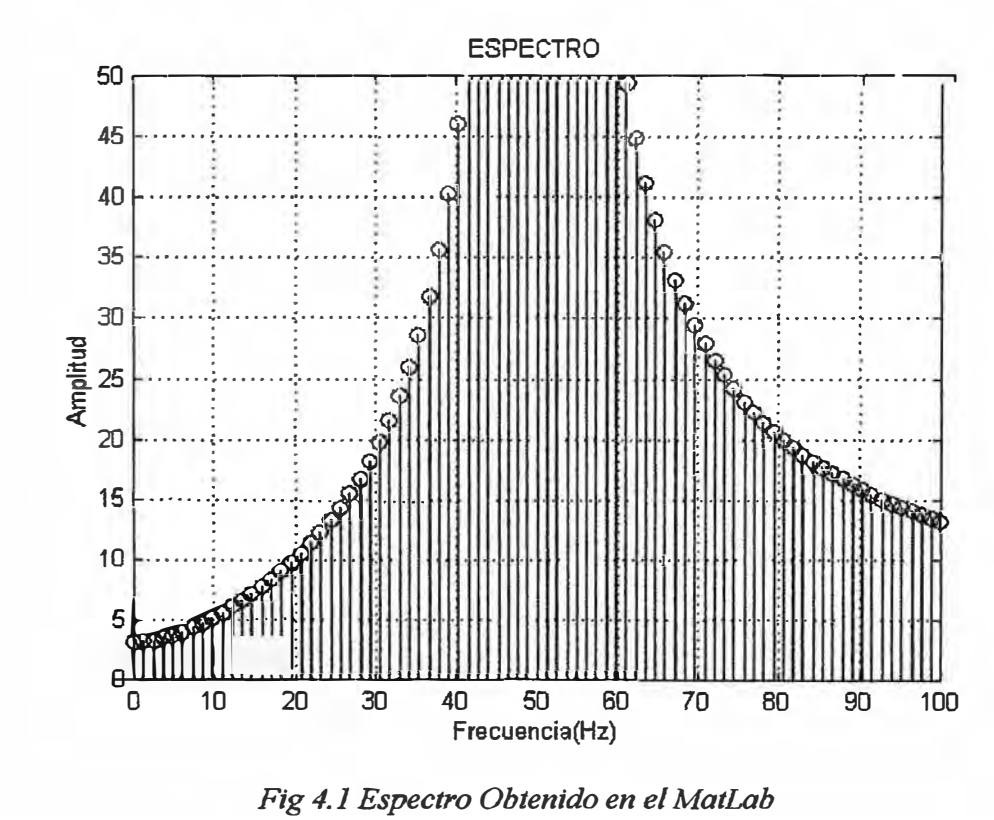


Fig 4.1 Espectro Obtenido en el MatLab

Segundo, en el Sistema ANALIZA se ingresaron los datos correspondientes para generar una señal senoidal de las mismas características, esto se hizo procesando la señal generada internamente. A continuación el resultado grafico del FFT obtenido por ANALIZA.

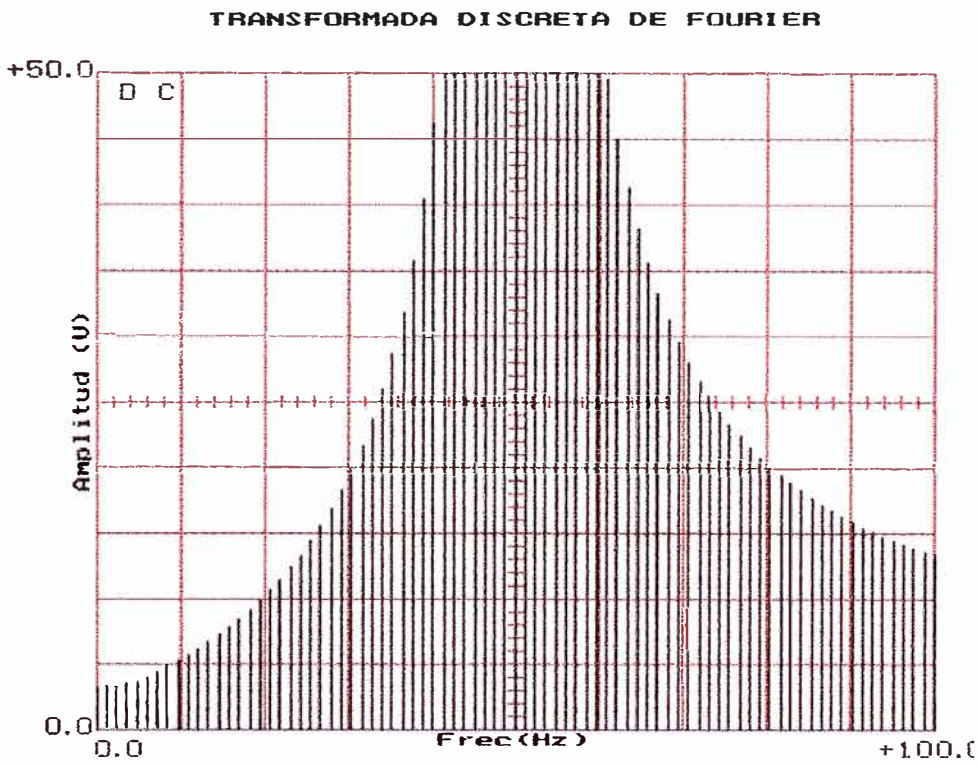


Fig 4.2 Espectro Obtenido en ANALIZA

Tercero, se exportaron los datos del espectro obtenido desde ANALIZA al MATLAB y se procesaron en el MATLAB. Se obtuvieron los siguientes datos estadísticos mostrados en la siguiente tabla:

Tabla N° 4.1 Datos Estadísticos por cada Curva

PARAMETRO	ANALIZA	MATLAB	ERROR %
Promedio FFT	8.076173244	8.07616419	-1.1211E-04
Desviación Estándar	226.2617045	226.261705	2.1231E-05

PARAMETRO	VALOR %
Media de la diferencia punto a punto	-1.1211E-04
Desviación Estándar de la diferencia punto a punto	2.1231E-05

Tabla N° 4.2 Datos Estadísticos de la Diferencia punto a punto de ambas curvas

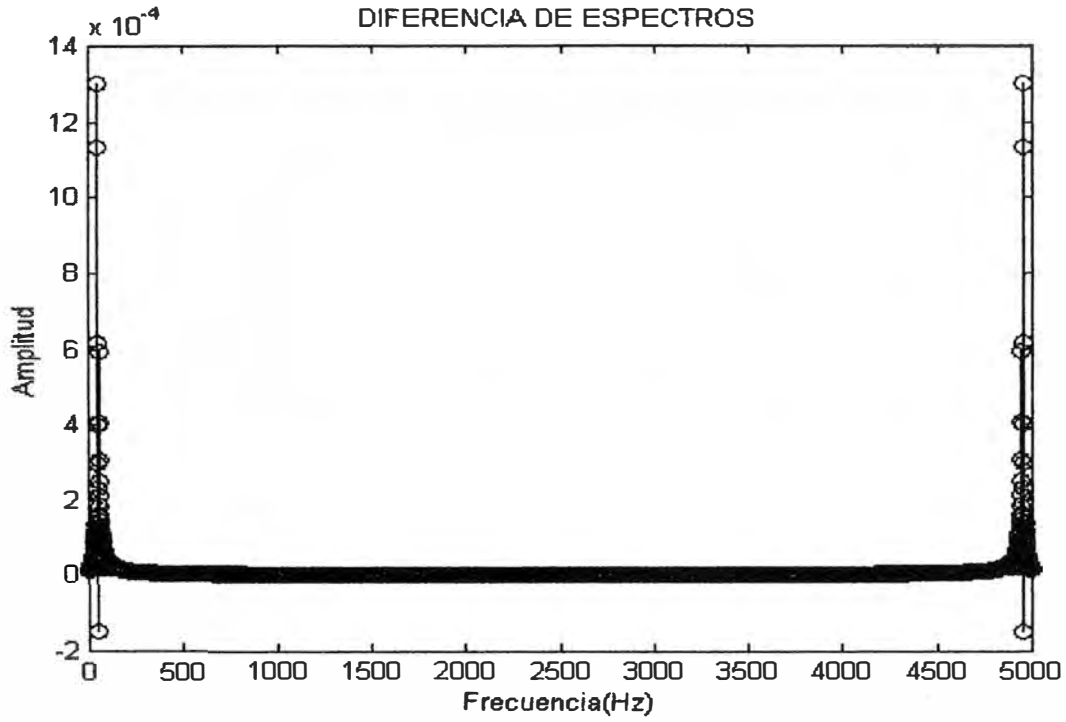


Fig 4.3 Diferencia de Espectros en MatLab

Visualmente se aprecia gran similitud y exactitud de ANALIZA, los resultados estadísticos obtenidos demuestran también que existe una gran precisión.

Puesto que las diferencias porcentuales entre los parámetros están en el rango de 0.00012% a menos, consideraremos el error del algoritmo de FFT de ANALIZA como 0.00012%.

4.2 Medida de Señal Generada

Usando dos PCs, una con el modulo generador DAC a la cual llamaremos PC GENERADORA y la otra con el ADC a la cual llamaremos PC ANALIZADORA, ambas utilizando el programa ANALIZA se realizó la siguiente prueba.

En la PC GENERADORA se generó una señal senoidal interna de 4V, la cual fue derivada al modulo DAC usando la opción ACTIVAR GENERADOR DE ONDAS del programa, la señal obtenida es ingresada a la PC ANALIZADORA y capturada usando la opción AC-IN.

A nivel visual se obtuvieron resultados bastante precisos. A nivel de datos se utilizó la diferencia obtenida para calibrar los valores tanto de ADC como del DAC.

Usando la opción de ESPECTOGRAFO en ambas PC se pudo obtener el espectro de ambas señales, la teórica de la PC GENERADORA y la practica de la PC ANALIZADORA.

Ambas señales se exportaron para realizar procesamiento estadístico similares a la hecha para verificar la exactitud del algoritmo FFT, contrastando el espectro del GENERADOR con el ANALIZADOR obteniéndose en resumen errores menores al 1%.

4.3 Medida de Señal de Línea

Para esta prueba se realizaron solo medidas básicas, para ello se conectó la PC ANALIZADORA, usando la sonda de línea, a una toma domiciliario monofásica de 220v AC, y usando la opción AC-LINEA en el programa, se pudo digitalizar, la alimentación de línea. El programa permite ver en tiempo real los parámetros básicos como son el valor eficaz y la frecuencia fundamental, estos valores que fueron contrastados por varios multímetros usados en diferentes periodos de prueba, esto sirvió para calibrar el factor de la sonda de línea y verificar el error de nivel del sistema que supera el 1% .

La prueba final de contrastar con una Analizador de Línea durante un periodo estandarizado de prueba aun no se pudo realizar por la dificultad de conseguir un analizador como los utilizados en las empresas eléctricas para la medida de la calidad de línea.

CONCLUSIONES

1. Se ha conseguido la implementación de un interfase de bajo costo (aprox. $15), el cual puede ser utilizado como un Analizador de Red Básico, aunque no cumple con la mayoría de las normas oficiales esta muy cerca a los valores establecidos y se puede utilizar como un instrumento auxiliar o de pre-evaluación de la calidad del producto eléctrico.

2. La resolución del algoritmo de FFT implementado comparada con los resultados del MatLab es bastante alta (1E-4 %) por lo cual se puede considerar exacto.

3. El programa o parte de el puede usarse también para la simulación de señales y el análisis, teórico, ya que cuenta con la herramienta fundamental del FFT y un modulo de señales.

4. La velocidad de muestreo puede mejorarse si se utiliza otros ADC como el 0820 con el cual se podría llegar a la velocidad límite del puerto paralelo en modo estándar que esta en alrededor de 100Khz según la velocidad de la PC.

5. El sistema implementado funciona también como osciloscopio de baja frecuencia (2.5KHZ) y se puede utilizar en aplicaciones de baja frecuencia como es el caso de Monitoreo de magnitudes eléctricas como corriente, o potencia, bastará con diseñar el transductor adecuado y calibrar el ADC.

ANEXO A

FOTOS DE LA IMPLEMENTACION

Fig A.1 El Interfaz Completo

Fig. A.2 ANALIZA en 386 Portatil

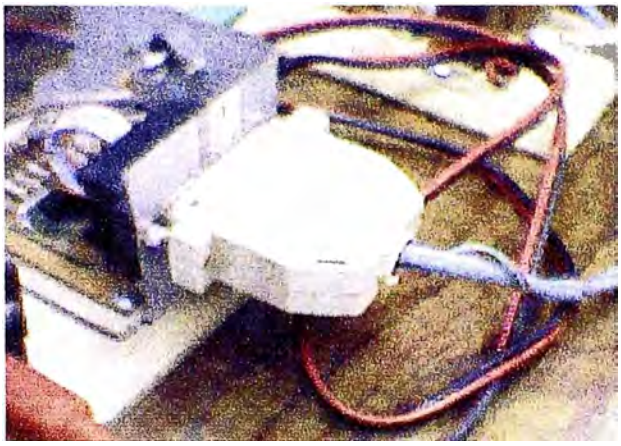


Fig. A.3 Conexión del DAC

Fig. A.4 Conexión de la PC al DAC

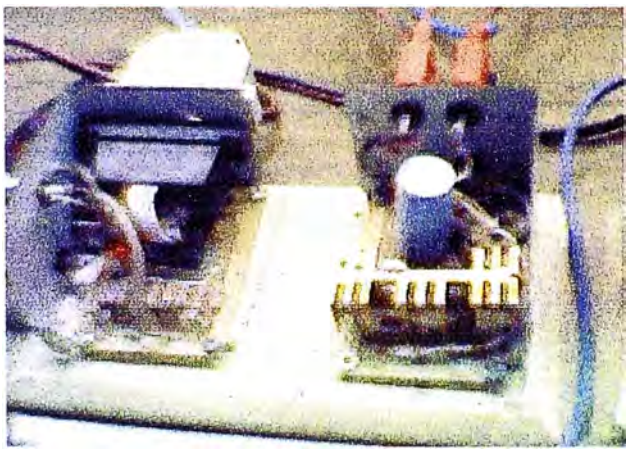


Fig. A.5 Modulo ADC

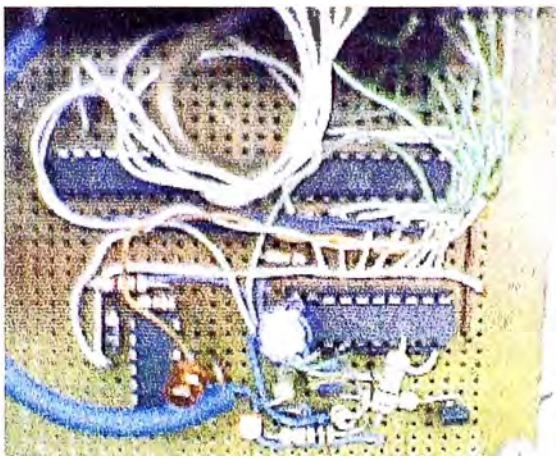


Fig. A.6 Detalle Módulo ADC

ANEXO B

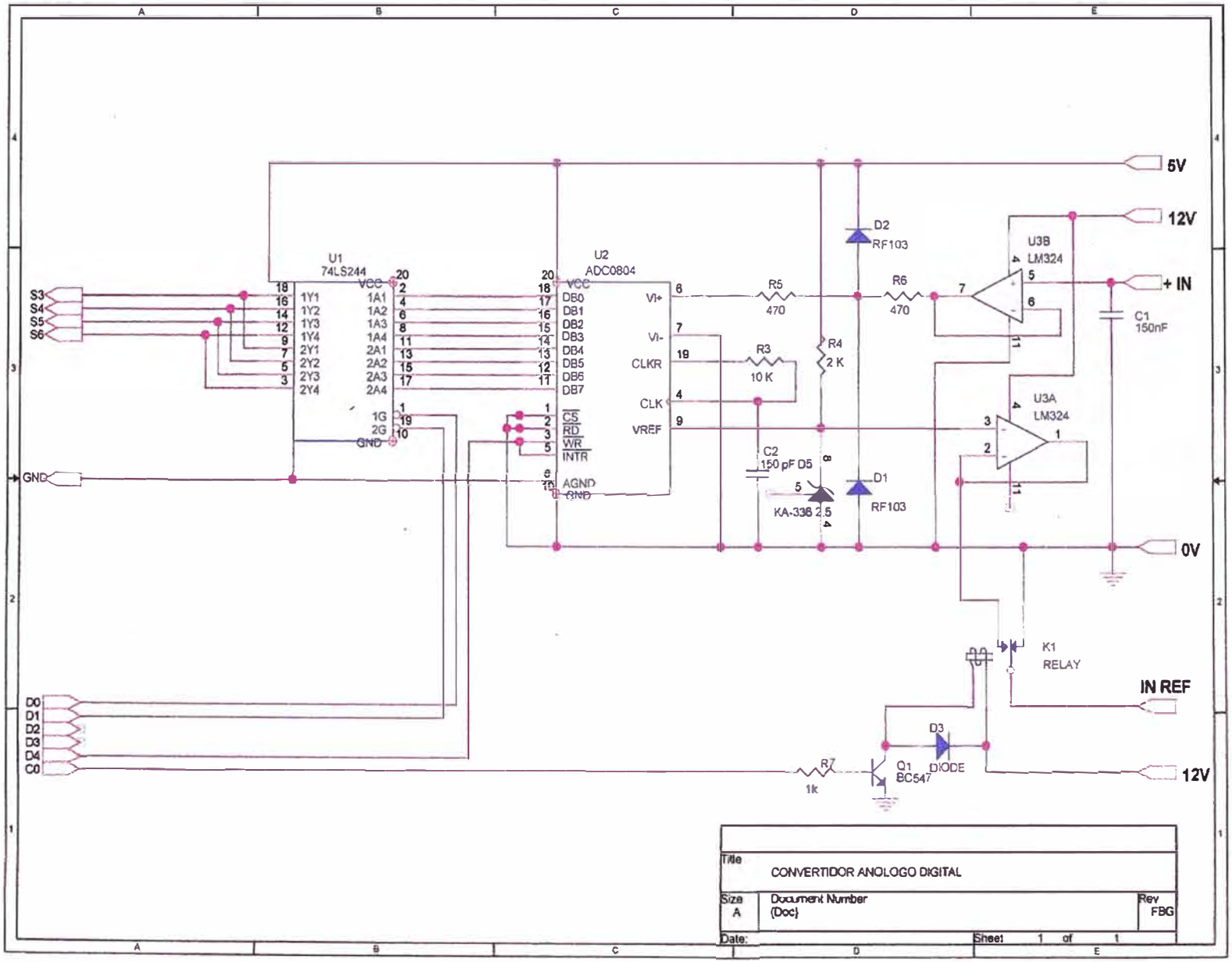


Fig B.1 Diagrama del Digitalizador de Señales

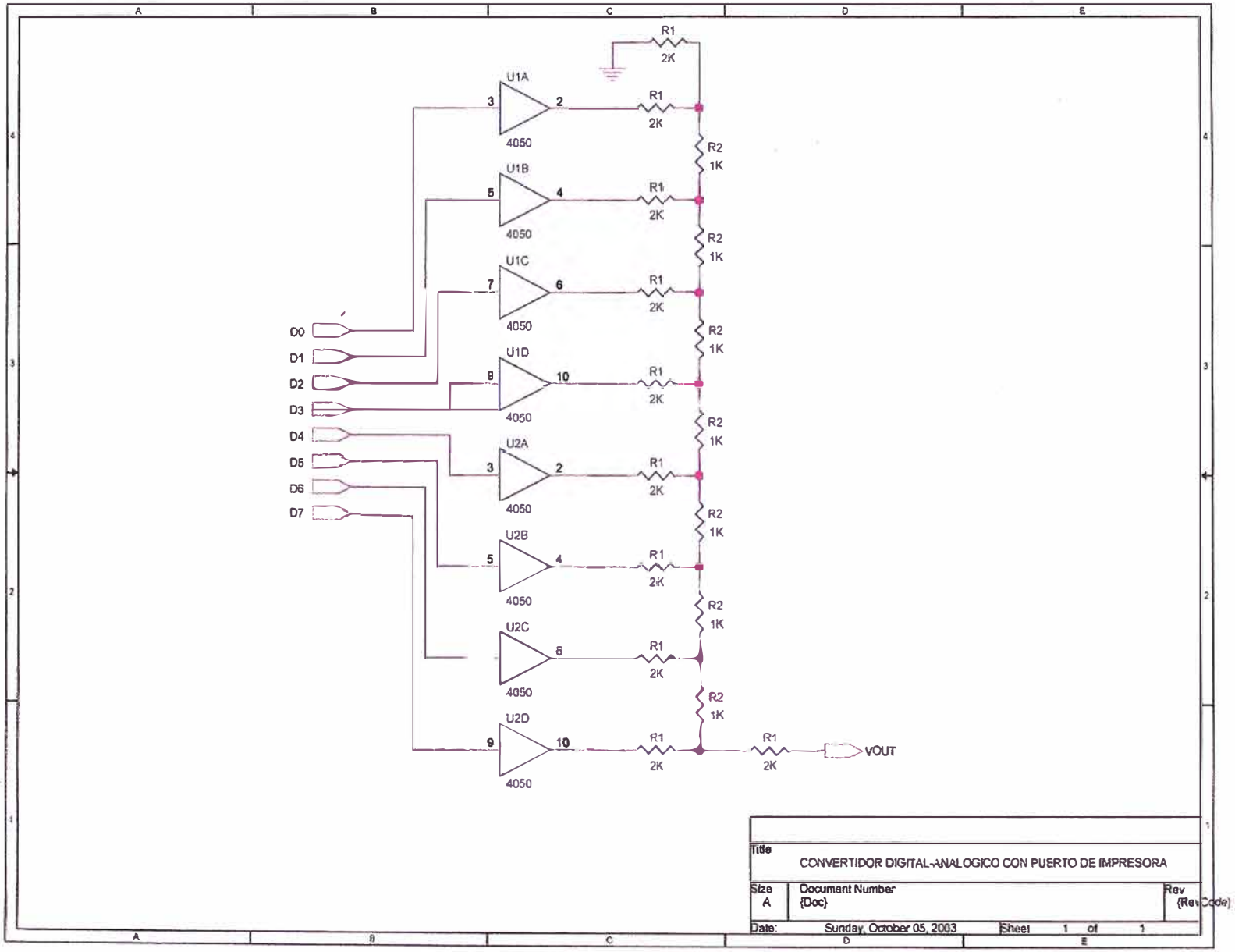


Fig B.2 Diagrama del Generador de Señales

National Semiconductor

November 1999

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit µP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Features

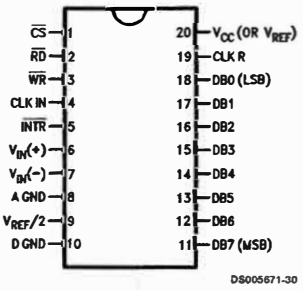
- Compatible with 8080 µP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

- Resolution 8 bits
- Total error ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time 100 µs

Connection Diagram

ADC080X
Dual-In-Line and Small Outline (SO) Packages



See Ordering Information

Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	-40°C TO +85°C
ERROR	±¼ Bit Adjusted	ADC0802LCWM		ADC0801LCN
	±½ Bit Unadjusted			ADC0802LCN
	±½ Bit Adjusted			ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCN	ADC0805LCN/ADC0804LCJ
PACKAGE OUTLINE		M20B—Small Outline	N20A—Molded DIP	

TRI-STATE® is a registered trademark of National Semiconductor Corp.
Z-80® is a registered trademark of Zilog Corp.

© 1999 National Semiconductor Corporation DS005671

Fig B.3 Hoja Técnica del ADC 0804

BIBLIOGRAFIA

1. Steven W. Smith, "The Scientist an Engineer´s Guide to Digital Signal Processing". Second Edition., USA 1999.

2. Alan V. Oppenheim, Ronald W.Schafer, "Tratamiento de Señales en Tiempo Discreto". Editorial Prentice Hall, Madrid 2000

3. Nacional Semiconductor "Nacional Analog and Interface Products Databook", Edición 2001.

4. Richard R. Eckerd, "The Intel 8253/8254 Programmable Interval Timer and Sound on a PC" www.cs.binghamton.edu/~reckert/220/8254_timer.html

5. Herbert Schildt, "C++ Para programadores", Editorial McGraw Hill. Mexico 1996.

6. Open Watcom "C/C++ Programmer´s Guide", "www.openwatcom.org" 2006.

7. Osinerg, "Base Metodológica para la Aplicación de la Norma Técnica de Calidad de los Servicios Eléctricos". Lima 2001