



**BENEMÉRITA UNIVERSIDAD
AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

**“REGISTRO DE SEÑALES BIOLÓGICAS A
TRAVÉS DE UN CONVERTOR
ANALÓGICO-DIGITAL DE 12 BITS”**

TESIS

QUE PARA OBTENER EL GRADO DE:

**LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

PRESENTA:

ALICIA VERA LIMÓN

ASESOR:

DR. EDUARDO MARCOS SALINAS STEFANÓN
INSTITUTO DE FISIOLÓGÍA, BUAP

COASESOR:

DR. MANUEL MARTÍN ORTIZ
FAC. CIENCIAS DE LA COMPUTACIÓN, BUAP

PUEBLA, PUE. MÉXICO 2008

INDICE

INDICE	2
1 PLANTEAMIENTO DEL PROBLEMA	6
1.1 PLANTEAMIENTO DEL PROBLEMA	6
1.2 OBJETIVO GENERAL	6
1.3 OBJETIVOS ESPECIFICOS	6
1.4 HIPÓTESIS DE TRABAJO	7
1.5 REQUERIMIENTOS DEL PROGRAMA	7
2 GENERALIDADES DE LOS CONVERTORES ANALÓGICO-DIGITAL	8
2.1 MARCO TEORICO	8
2.2 ADQUISICIÓN DEL HARDWARE	9
2.3 SEÑALES, SISTEMAS Y PROCESADO DE LA SEÑAL	9
2.3.1 Elementos Básicos de un Sistema de Procesado Digital de Señales	9
2.3.2 Ventajas del procesado digital de señales frente al analógico	10
2.4 CLASIFICACIÓN DE LAS SEÑALES	11
2.5 EL CONCEPTO DE FRECUENCIA	12
2.6 CONVERSIÓN ANALÓGICO-DIGITAL Y DIGITAL-ANALÓGICA	12
2.6.1 Muestreo de señales analógicas	13
2.6.2 Teorema del muestreo (Teorema de Nyquist)	13
2.6.3 Cuantificación de señales de amplitud continua	14
2.6.4 Codificación de muestras cuantificadas	14
3 BIOELECTRICIDAD	15
3.1 POTENCIALES ELÉCTRICOS	15
3.2 CORRIENTES ELÉCTRICAS	15
3.3 CONDICIONANDO LA SEÑAL Y CONDICIONADORES DE SEÑAL	15
3.3.1 ¿Por qué las señales deberían ser filtradas?	16
3.4 PREPARANDO LAS SEÑALES PARA LA CONVERSIÓN A/D	16
3.4.1 Error de Cuantización	17
3.4.2 Eligiendo la frecuencia de muestreo	17
3.4.3 Formato de los datos	18
3.5 DMA, MEMORY BUFFERED, I/O DRIVEN	18
3.6 INTERRUPCIONES	19
3.7 TIMERS	19
3.8 DIGITAL I/O	20
3.9 OPERANDO BAJO SISTEMAS OPERATIVOS MULTITAREA	20
3.9.1 Soporte del Software	21

4	TARJETA DE ADQUISICIÓN DE DATOS LABMASTER ADVANCED DESIGN	22
4.1	INTRODUCCION	22
4.2	CARACTERISTICAS	22
4.3	CONSIDERACIONES DE PROGRAMACION	22
4.3.1	Localidades E/S	22
4.3.2	Dirección Base	22
4.3.3	Localidades de Registros	23
4.3.4	Registros Especiales	24
4.3.4.1	Registro que habilita la tarjeta	24
4.3.4.2	Estado habilitado de la tarjeta	24
4.4	CONVERSION ANALOGICO-A-DIGITAL (CAD)	24
4.5	CONVERSION DIGITAL-A-ANALOGICO (CDA)	25
4.6	SISTEMA DE REGULACION DE TIEMPO Y CONTEO (STC)	25
4.6.1	Señales de entrada	25
4.6.2	Entradas de habilitación	25
4.6.3	Frecuencias de entrada	25
4.6.4	Salidas del contador	26
4.7	E/S DIGITAL (DIO)	26
4.8	INTERRUPCIONES DE HARDWARE	26
4.9	ACCESO DIRECTO A MEMORIA (DMA)	27
4.10	MANEJO DE LA LABMASTER AD (CONTROLADOR LABPAC32)	27
4.10.1	Código Ejemplo de Inicialización	28
5	ANALISIS, DISEÑO E IMPLEMENTACIÓN	29
5.1	ANÁLISIS	29
5.1.1	Identificar la necesidad del cliente	29
5.1.2	Estudio de la viabilidad	30
5.2	DISEÑO	31
5.2.1	Programación Modular	31
5.2.1.1	Módulo Tarjeta LabMaster AD	33
5.2.1.2	Módulo Archivos ABF	33
5.2.1.3	Módulo Captura DMA	34
5.2.1.4	Módulo Análisis	34
5.2.1.5	Módulo Graficado	37
5.3	IMPLEMENTACIÓN	38
5.3.1	Prototipado Evolutivo	38
5.3.2	ALISE (Asistente para laboratorio en experimentos científicos)	39
	EJEMPLOS DE SEÑALES CAPTURADAS	41
	CONCLUSIONES Y RESULTADOS	46
	APÉNDICE A	48
	REGISTROS DE CONTROL LABMASTER AD, REFERENCIA RAPIDA	48
	Registro de Control ADC	48
	Registro de Estado ADC	48
	Arreglo de Canal/Ganancia	50
	Registro de Control DAC	51

Registro de Estado DAC	52
Controlador 8237 DMA	53
Registro Mascara	53
Registro Modo	54
Registro De Control IRQ	55
Registro De Estado IRQ	55
<i>APENDICE B</i>	56
ARCHIVOS ABF (AXON BINARY FILES)	56
Estructura del Archivo ABF	56
La sección ABF Header	56
La sección ABF Scope Config	57
La sección ABF Data	57
La sección ABF Synch	57
La sección ABF Tag	57
La sección ABF Deltas	57
La sección DAC Data	57
MODOS DE ADQUISICIÓN	58
1. Gap-Free	58
2. Longitud Variable	58
3. Longitud Fija	58
4. Osciloscopio De Alta Velocidad	58
5. Estimulación Episódica	59
<i>BIBLIOGRAFIA</i>	60

RESUMEN

La electrofisiología estudia el funcionamiento de los organismos considerando las señales eléctricas presentes en ellos. Los procesos fisiológicos son inherentes a pequeñas señales de corriente y voltaje. Los fisiólogos analizan estas señales para entender el funcionamiento de los sistemas orgánicos, que se comunican por este medio; desafortunadamente, estas señales no pueden ser analizadas directamente por una computadora. Para que pueda utilizarse una computadora en el registro y análisis de estas señales electrofisiológicas, es necesario que la computadora cuente con equipo especializado, genéricamente conocido como Sistemas de Adquisición Analógico-Digital. Este equipo se encarga de convertir estas señales analógicas a un formato digital que pueda ser manipulado por la computadora. Por extensión, este equipo también permite generar señales analógicas y digitales para tareas de control. La teoría del manejo y procesamiento de señales analógicas en sistemas digitales se conoce como Procesamiento Digital de Señales.

Los dos parámetros principales involucrados en el proceso de la Conversión Analógico/Digital son la frecuencia de muestreo y la resolución en bits. Para que una señal analógica pueda ser representada en un sistema digital (por ejemplo, una computadora), tiene que ser medida y debe registrarse su valor de la forma más fiel que sea posible. La frecuencia de muestreo indica el número de veces por segundo en que se mide la señal analógica; y cuanto más grande sea la resolución en bits, mayor será el número de posibles valores que pueden utilizarse para representar esa señal.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Existe una técnica (Fijación de voltaje)^[1] que permite desplazar el potencial de membrana al valor deseado, manteniéndolo constante, mientras se mide la intensidad de corriente a través de la membrana. En un experimento de fijación de voltaje uno controla el voltaje de la membrana y mide la corriente de la membrana interior requerida para mantener ese voltaje. A pesar del hecho de que la fijación de ese voltaje no imita un proceso encontrado en la naturaleza, hay 3 razones para este tipo de experimento:

- (1) Que al fijar el voltaje se elimina la corriente capacitiva, excepto para un breve intervalo de tiempo (proceso transitorio) dando paso a un nuevo voltaje;
- (2) Excepto por una breve carga de tiempo, las corrientes que fluyen son proporcionales solamente a la conductancia de la membrana, es decir al número de canales abiertos;
- (3) Si el canal de disparo es determinado sólo por el voltaje de la membrana interior (y es insensible a otros parámetros tales como la corriente y el historial de voltaje), la fijación de voltaje ofrece control sobre la variable clave que determina la apertura y cierre de canales de ion.

Para poder llevar a cabo esta técnica es indispensable contar con el equipo adecuado así como de un software que nos permita la captura de señales eléctricas para su registro y análisis. Nuestro proyecto consistirá en llevar a cabo la parte (etapa) de monitoreo de la señal, la cual constará de la captura de señales eléctricas a través de un conversor analógico digital, y el registro de las mismas guardándose en un formato de archivos ABF (Axon Binary File) para un mejor manejo de la señal. (Ver APENDICE B)

1.2 OBJETIVO GENERAL

Crear una herramienta de software para el registro y análisis de señales electrofisiológicas, haciendo uso de la tarjeta digitalizadora LabMaster Advanced Design (AD) de Scientific Solutions, Inc.

1.3 OBJETIVOS ESPECIFICOS

Desarrollar los métodos pertinentes para la captura de señales.

Hacer uso de las funciones necesarias para la creación de archivos ABF.

1.4 HIPÓTESIS DE TRABAJO

El desarrollo de un programa de captura de señales para la tarjeta digitalizadora LabMaster AD de Scientific Solutions, Inc. a través del cual se pueda hacer uso de ella bajo ambiente Windows debido a que no hay un software que le de soporte.

1.5 REQUERIMIENTOS DEL PROGRAMA

Se requiere que el sistema haga una captura de datos analógica-digital con cuatro canales de entrada. Antes de hacer la captura es necesario establecer los parámetros de la misma que especifiquen el modo de adquisición, el número y el nombre de los canales a ser utilizados, el intervalo de muestreo y el tiempo de captura.

Para evitar que se presenten interrupciones en el sistema los datos capturados durante la adquisición serán almacenados en memoria y posteriormente se guardarán a disco. El formato de archivo requerido para el almacenamiento de los datos es con extensión ABF.

Este tipo de archivos nos permitirá generar archivos de datos similares a otros programas comerciales lo cual admitirá la apertura de nuestros archivos en dichos programas, tal como es el caso de pCLAMP de Molecular Devices que nos permite la apertura y el manejo de este tipo de archivos.

2 GENERALIDADES DE LOS CONVERTORES ANALÓGICO-DIGITAL

2.1 MARCO TEORICO

Los procesos del mundo real producen señales analógicas que varían constantemente. La velocidad puede ser muy lenta, como la variación de la temperatura ambiente, o muy rápida, como sucede en un sistema de audio. Los procesos analógicos se describen mejor por medio de números decimales y letras del alfabeto. En cambio, en los microprocesadores y las computadoras utilizan patrones binarios para representar números, letras o símbolos.

Cuando se recurre a la tecnología analógica no es fácil almacenar, manipular, comparar, calcular o recuperar información con exactitud. En cambio, las computadoras pueden efectuar estas tareas rápidamente y hacerlo con precisión en una cantidad casi ilimitada de datos, empleando técnicas digitales. Así, nació la necesidad de los convertidores para interrelacionar el mundo analógico con el digital. Los Convertidores Analógico a Digital (CAD) permiten que el mundo analógico se comunique con las computadoras. Y estas a su vez se comunican con el hombre y con los procesos físicos a través de sus Convertidores Digital a Analógico (CDA).^[2]

El procesamiento digital de señales es un área de la ciencia y la ingeniería que se ha desarrollado rápidamente. Este rápido desarrollo es el resultado de los avances tecnológicos tanto en las computadoras como en la fabricación de circuitos integrados. Las computadoras y el hardware asociado eran relativamente grandes y caros, su uso se limitaba a aplicaciones de propósito general en tiempo no real, tanto científicas como comerciales. El rápido desarrollo de la tecnología de circuitos integrados ha estimulado el desarrollo de computadoras más potentes, pequeñas, rápidas y baratas y de hardware digital de propósito general. Estos circuitos digitales han hecho posible construir sistemas digitales altamente sofisticados, capaces de realizar funciones y tareas del procesamiento de señal digital que normalmente eran demasiado difíciles y/o caras con circuitería o sistemas de procesamiento de señales analógicas. De aquí que muchas de las tareas del procesamiento de señal que convencionalmente se realizaban analógicamente se realicen hoy mediante hardware digital.

Para muchas señales de gran ancho de banda, se requiere procesamiento en tiempo real. Para tales señales, el procesamiento analógico o, quizás, óptico, son las únicas soluciones válidas. Sin embargo, cuando los circuitos digitales se encuentran disponibles y tienen la velocidad suficiente, son normalmente preferibles. El procesamiento de señal digital permite operaciones programables. Por medio de software se pueden modificar fácilmente las funciones de procesamiento de señal para que sean realizadas por el hardware. Por tanto, el hardware digital y el software asociado proporcionan un mayor grado de flexibilidad en el diseño de sistemas.

El procesamiento digital de señales analógicas tiene algunos inconvenientes. El primero, la conversión de una señal analógica en digital, obtenida haciendo un muestreo de la señal y cuantificando las muestras, produce una distorsión que nos impide la reconstrucción de la señal analógica original a partir de las muestras cuantificadas. El control de esta distorsión se logra con la elección apropiada de la velocidad de muestreo y la precisión del proceso de cuantificación. En segundo, existen efectos debidos a la precisión finita que deben ser considerados en el procesamiento digital de las muestras cuantificadas.^[3]

2.2 ADQUISICIÓN DEL HARDWARE

Los científicos están siempre buscando mejores métodos para registrar y almacenar el crecimiento de la suma de los datos generados en los experimentos. En los experimentos electrofisiológicos, los datos están la mayoría de las veces en forma de ondas de voltaje cuyas magnitudes varían con el tiempo. Los datos en esta forma son adecuados para visualizarse en un osciloscopio o en una gráfica de registros, pero completamente inadecuados para guardarlos en el disco de una computadora. Puesto que las computadoras pueden solamente guardar números discretos, un proceso de conversión “analógico-a-digital” tiene que encargarse de convertir el dato analógico a un formato compatible para la computadora.

2.3 SEÑALES, SISTEMAS Y PROCESADO DE LA SEÑAL

Una *señal* se define como una cantidad física que varía con el tiempo, el espacio o cualquier otra variable(es) independiente(es).

Las señales de voz, los electrocardiogramas (ECG) y los electroencefalogramas (EEG) son ejemplos de señales que llevan información y que varían como funciones de una única variable independiente, el tiempo. Una imagen constituye un ejemplo de una señal que varía con dos variables independientes que en este caso son las coordenadas espaciales.

La forma en que se generan las señales se encuentra asociada con un *sistema* que responde ante un estímulo o fuerza. El estímulo en combinación con el sistema se llama *fente de señal*.

Un sistema se puede definir también como un dispositivo físico que realiza una operación sobre una señal. Cuando pasamos una señal a través de un sistema decimos que hemos procesado la señal.

En el procesado digital de señales en una computadora, las operaciones realizadas sobre una señal constan de varias operaciones matemáticas especificadas por un programa de software. Un sistema digital puede se puede implementar como una combinación de hardware digital y software, cada uno de los cuales desempeña su propio conjunto de funciones.

2.3.1 Elementos Básicos de un Sistema de Procesado Digital de Señales

La mayor parte de las señales que aparecen en los ámbitos de la ciencia y la ingeniería son de naturaleza analógica, es decir, las señales son funciones de una variable continua, como el tiempo o el espacio y normalmente toman valores en un rango continuo. Tales señales pueden ser procesadas directamente por sistemas analógicos adecuados (como filtros o analizadores de frecuencia) o multiplicadores de frecuencia con el propósito de cambiar sus características o extraer cualquier información deseada. (Ver Figura 1)

El procesado digital de señales proporciona un método alternativo para procesar una señal analógica. Para realizar el procesado digitalmente, se necesita una interfaz entre la señal analógica y el procesador digital. Esta interfaz se denomina *conversor analógico-digital (A/D)*. La salida del conversor analógico-digital es una señal adecuada como entrada al procesador digital.

El procesador digital de señales puede ser una gran computadora o un pequeño microprocesador programado para efectuar un conjunto de operaciones sobre la señal de entrada. Las máquinas programables proporcionan la flexibilidad de cambiar las operaciones de procesado de señales mediante un cambio del software. En aplicaciones donde la salida digital del procesador digital de señales se ha de entregar en forma analógica, debemos proporcionar

otra interfaz desde el dominio digital al analógico. Tal interfaz se denomina *conversor digital-analógico* (D/A). (Ver Figura 2)



Figura 1 Procesado de señal analógica

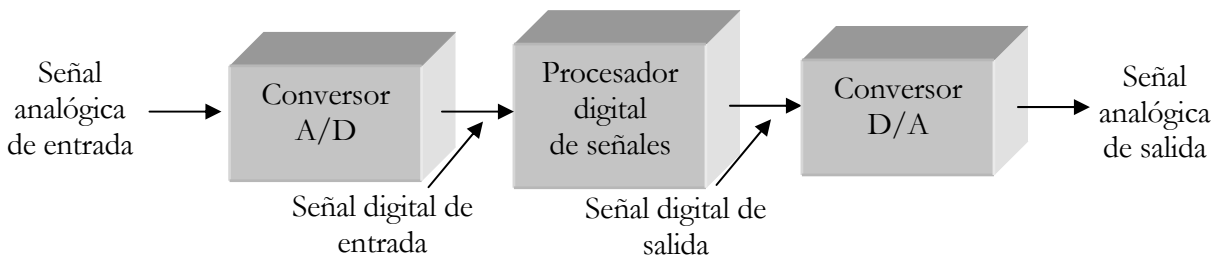


Figura 2 Diagrama de bloques de un sistema digital de procesamiento de señales

2.3.2 Ventajas del procesamiento digital de señales frente al analógico

Un sistema digital programable permite flexibilidad a la hora de reconfigurar las operaciones de procesamiento digital de señales sin más que cambiar de programa. La reconfiguración de un sistema analógico implica habitualmente el rediseño del hardware, seguido de la comprobación y verificación para ver que opera correctamente.

También desempeña un papel importante al elegir el formato del procesador de señales la consideración de la precisión. Las tolerancias en los componentes de los circuitos analógicos hacen que para el diseñador del sistema sea extremadamente difícil controlar la precisión de un sistema de procesamiento analógico de señales. En cambio, un sistema digital permite un mejor control de los requisitos de precisión del convertor A/D y del procesador digital de señales.

Las señales digitales se almacenan fácilmente en soporte magnético sin deterioro o pérdida en la fidelidad de la señal. Las señales se hacen transportables y pueden procesarse en tiempo no real en un laboratorio remoto. El método de procesamiento digital de señales también posibilita la implementación de algoritmos de procesamiento de señal más sofisticados. Generalmente es muy difícil realizar operaciones matemáticas precisas sobre señales en formato analógico, pero esas mismas operaciones pueden efectuarse de modo rutinario sobre una computadora utilizando software.

En algunos casos, la implementación digital del sistema de procesamiento de señales es más barato que su equivalente analógica.

Sin embargo, la implementación digital tiene sus limitaciones. Una limitación práctica es la velocidad de operación de los conversores A/D y de los procesadores digitales de señales. Veremos que las señales con anchos de banda extremadamente grandes precisan conversores A/D con una velocidad de muestreo alta y procesadores digitales de señales rápidos.

2.4 CLASIFICACIÓN DE LAS SEÑALES

Los métodos que usamos en el procesamiento de una señal o en el análisis de la respuesta de un sistema a una señal dependen fuertemente de las características de la señal en particular.

Las señales se pueden clasificar dependiendo de las características de la variable (independiente) tiempo y los valores que esta puede tomar. *Señales en tiempo continuo* o *señales analógicas* están definidas para todos los valores del tiempo y pueden tomar cualquier valor en el intervalo continuo (a, b) , donde a puede ser $-\infty$ y b puede ser ∞ . Las *señales en tiempo discreto* están definidas sólo para ciertos valores del tiempo. Estos instantes del tiempo no necesitan estar equidistantes, aunque en la práctica se toman normalmente instantes equiespaciados conforme a intereses computacionales y matemáticos.

En la práctica las señales en tiempo discreto pueden originarse de dos maneras:

1. Eligiendo valores de una señal analógica en determinados instantes de tiempo. Este proceso se denomina *muestreo*.
2. Acumulando una variable a lo largo de un determinado periodo.

El valor de una señal, en tiempo continuo o discreto, puede ser continuo o discreto. Si una señal toma todos los valores posibles en un intervalo tanto finito como infinito, se dice que es continua. Por el contrario, si toma valores de un conjunto finito de valores se dice que es discreta. Una señal en tiempo discreto, que toma valores discretos (es decir, ha de ser una señal digital).

Para que una señal pueda ser procesada digitalmente ha de ser en tiempo discreto y tomar valores discretos. Si la señal a procesar es analógica, se convierte a digital muestreándola en el tiempo y obteniendo por tanto una señal en tiempo discreto y posteriormente *cuantificando* sus valores en un conjunto discreto. El proceso de convertir una señal continua en discreta, denominado *cuantificación*, es básicamente un proceso de aproximación. Puede lograrse por redondeo o truncamiento.

El análisis matemático y el procesamiento de señales requieren que la señal sea descrita matemáticamente. Esta descripción matemática, normalmente denominada *modelo matemático*, conduce a otra importante clasificación de las señales. Cualquier señal que pueda ser definida por una forma matemática explícita, un conjunto de datos o una regla bien definida se denomina *determinista*. Este término se usa para resaltar el hecho de que valores de la señal, tanto presentes como pasados como futuros, se conocen exactamente, sin incertidumbre.

En muchas situaciones prácticas, sin embargo, existen señales que no se pueden describir con un grado de precisión razonable mediante fórmulas matemáticas explícitas, o cuya descripción es demasiado complicada para ser de utilidad práctica. La falta de tal relación supone que dichas señales evolucionan con el tiempo de forma impredecible. Nos referimos a estas señales como *señales aleatorias*. Esto motiva el análisis y descripción de las señales aleatorias mediante técnicas *estadísticas* en vez de mediante fórmulas explícitas. El marco matemático para el análisis de señales aleatorias lo constituye la teoría de la probabilidad y los procesos estocásticos.

2.5 EL CONCEPTO DE FRECUENCIA

De la Física sabemos que la frecuencia está íntimamente relacionada con un tipo específico de movimiento periódico llamado oscilación armónica, que se describe mediante funciones sinusoidales. El concepto de frecuencia está directamente relacionado con el del tiempo. De hecho, sus dimensiones son las inversas del tiempo. Por tanto, de acuerdo con esto, deberíamos esperar que la naturaleza del tiempo (continuo o discreto) afectase la frecuencia.

Por definición, la frecuencia es una cantidad física inherentemente positiva. Esto es obvio si interpretamos la frecuencia como el número de ciclos por unidad de tiempo de una señal periódica. Sin embargo, en muchos casos, únicamente por conveniencia matemática, necesitamos introducir frecuencias negativas.

Dado que una *frecuencia positiva* se corresponde con un movimiento angular uniforme contrario a las agujas del reloj, una *frecuencia negativa* simplemente se corresponde con un movimiento angular uniforme en el sentido de las agujas del reloj.

2.6 CONVERSIÓN ANALÓGICO-DIGITAL Y DIGITAL-ANALÓGICA

La mayoría de las señales de interés práctico, señales de voz, biológicas, sísmicas, radar, sonar y de distintos tipos de comunicación, como las señales de audio y video, son analógicas. Para procesar señales analógicas por medios digitales es necesario convertirlas a formato digital, esto es, transformarlas en una secuencia de números de precisión finita. Este procedimiento se denomina *conversión analógico-digital* (A/D) y los dispositivos correspondientes *convertidores* A/D (CAD).

Conceptualmente, podemos ver la conversión A/D como un proceso en tres pasos. (Ver Figura 3)

1. *Muestreo*. Ésta es la conversión de una señal en tiempo continuo a una señal en tiempo discreto obtenida tomando “muestras” de la señal en tiempo continuo en instantes de tiempo discreto.
2. *Cuantificación*. Ésta es la conversión de una señal en tiempo discreto con valores continuos a una señal en tiempo discreto con valores discretos (señal digital). El valor de cada muestra de la señal se representa mediante un valor seleccionado de un conjunto finito de valores. La diferencia entre la muestra sin cuantificar $x(n)$ y la salida cuantificada $x_q(n)$ se denomina error de cuantificación.
3. *Codificación*. En el proceso de codificación, cada valor discreto $x_q(n)$ se representa mediante una secuencia binaria de b bits.

El proceso de conversión de una señal digital en una analógica se conoce como *conversión digital-analógica* (D/A). Todos los convertidores D/A “conectan los puntos” de una señal digital efectuando cierto tipo de interpolación, cuya precisión depende de la calidad del proceso de conversión D/A. Una forma sencilla de conversión D/A es la denominada mantenedor de orden cero o aproximación por escalones. También son posibles otras aproximaciones, como la que conecta linealmente dos puntos (interpolación lineal), la que ajusta una función cuadrática a tres muestras sucesivas (interpolación cuadrática), etc.

El muestreo no da lugar a una pérdida de información ni introduce distorsión en la señal si su ancho de banda es finito. En principio, la señal analógica puede reconstruirse a partir de sus muestras, siempre que la tasa de muestreo sea lo suficientemente alta como para evitar el problema comúnmente denominado aliasing (solapamiento). Por otra parte, la cuantificación es

un proceso no invertible o irreversible que resulta en la distorsión de la señal. La cantidad de distorsión depende de la precisión, medida en función del número de bits del proceso de conversión A/D.

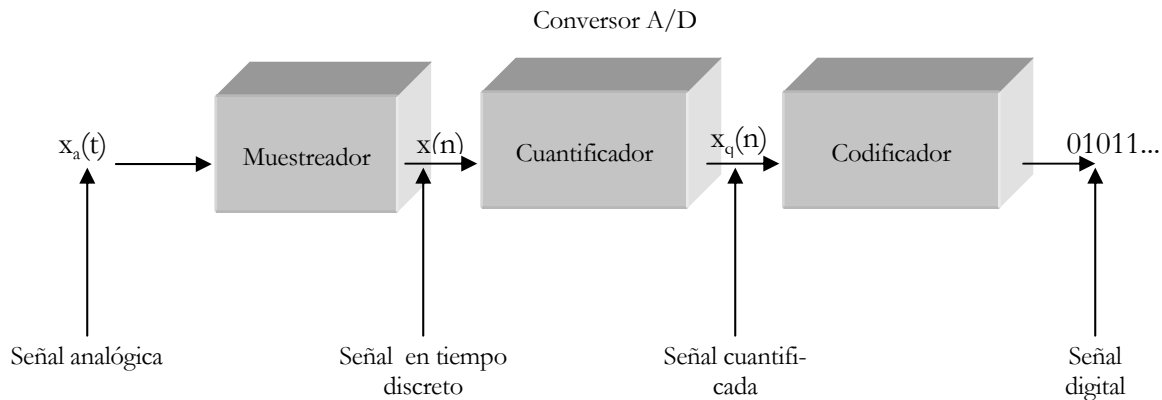


Figura 3 Partes básicas de un conversor analógico-digital (A/D)

2.6.1 Muestreo de señales analógicas

El *muestreo periódico o uniforme* es usado más a menudo en la práctica. Éste se escribe mediante la relación

$$x(n) = x_a(nT), \quad -\infty < n < \infty$$

donde $x(n)$ es la señal en tiempo discreto obtenida tomando muestras de la señal analógica $x_a(t)$ cada T segundos. El intervalo de tiempo T entre dos muestras sucesivas se denomina *periodo de muestreo* o *intervalo de muestreo*, y su recíproco $1/T = F_s$ se llama *velocidad de muestreo* (muestras por segundo) o *frecuencia de muestreo* (hertzios).

2.6.2 Teorema del muestreo (Teorema de Nyquist)

Para poder elegir el periodo de muestreo T ó, lo que es lo mismo, la velocidad de muestreo F_s dada una señal analógica cualquiera es necesaria cierta información sobre las características de la señal que va a ser muestreada. En particular, debemos tener cierta información general sobre el *contenido frecuencial* de la señal. Generalmente, dicha información se encuentra disponible en las amplitudes, frecuencias y fases de las distintas componentes de frecuencia. El propósito del procesado de la señal es normalmente la extracción de dichas características. Sin embargo, si conocemos la máxima frecuencia de una determinada clase de señales (por ejemplo, señales de voz, de vídeo, etc.), podemos especificar la velocidad de muestreo necesaria para convertir las señales analógicas en señales digitales.

Supongamos que las frecuencias de una determinada señal no exceden una frecuencia máxima conocida F_{max} . Dado que la máxima frecuencia puede variar ligeramente (por ejemplo, de un orador a otro) podemos querer asegurar que F_{max} no sobrepase determinado valor y para ello pasaremos la señal analógica por un filtro que atenúe fuertemente las componentes de frecuencia por encima de F_{max} . Así, estaremos seguros de que ninguna señal de la clase que nos interesa tendrá componentes de frecuencia (con amplitud o potencia significativa) por encima de F_{max} . En la práctica, este filtrado se realiza antes del muestreo. El conocimiento de F_{max} nos permite seleccionar la velocidad de muestreo apropiada. Sabemos que la frecuencia más alta de la señal

analógica que puede reconstruirse sin ambigüedad cuando la señal se muestrea a una velocidad $F_s = 1/T$ es $F_s/2$. Cualquier frecuencia por encima de $F_s/2$ o por debajo de $-F_s/2$ produce muestras que son idénticas a las correspondientes a frecuencias dentro del intervalo $-F_s/2 \leq F \leq F_s/2$. Para evitar las ambigüedades que resultan del aliasing, se debe seleccionar una velocidad de muestreo lo suficientemente alta, esto es, debemos escoger $F_s/2$ mayor que F_{\max} .

En otras palabras, la condición $F_s > 2F_{\max}$ garantiza que todas las componentes sinusoidales de la señal analógica se correspondan con componentes en frecuencia de tiempo discreto en el intervalo fundamental. Por lo tanto, todas las componentes en frecuencia de la señal analógica están representadas sin ambigüedad en la forma muestreada de la señal, y así la señal analógica puede ser reconstruida sin distorsión a partir de las muestras usando un método de interpolación “apropiado” (conversión digital-analógica).

2.6.3 *Cuantificación de señales de amplitud continua*

Una señal digital es una secuencia de números (muestras) en la que cada número se presenta por un número finito de dígitos (precisión finita).

El proceso de convertir una señal en tiempo discreto de amplitud continua en una señal digital, expresando cada muestra por medio de un número finito (en vez de infinito) de dígitos, se denomina *cuantificación*. El error cometido al representar la señal de valor continuo por un conjunto finito de valores discretos se denomina *error de cuantificación* o *ruido de cuantificación*.

Los valores permitidos en la señal digital se denominan *niveles de cuantificación*, mientras que la distancia entre dos niveles de cuantificación sucesivos se denomina *escalón de cuantificación* o *resolución*.

2.6.4 *Codificación de muestras cuantificadas*

El proceso de cuantificación en un conversor A/D asigna un número binario único a cada nivel de cuantificación diferente. Si disponemos de L niveles, necesitaremos al menos L niveles binarios distintos. Con una longitud de palabra de b bits se pueden crear 2^b números binarios diferentes. Por tanto, tenemos $2^b \geq L$ ó, de forma equivalente, $b \geq \log_2 L$. De este modo, el número de bits necesarios en el codificador es el menor entero mayor o igual que $\log_2 L$. Generalmente, cuanto mayor es la velocidad de muestreo y más fina la cuantificación, más caro es el dispositivo.

3 BIOELECTRICIDAD

3.1 POTENCIALES ELÉCTRICOS

Las propiedades eléctricas de las células se deben a las propiedades de la membrana celular. Una membrana, a su vez, adquiere las propiedades de sus lípidos y proteínas, tales como canales de iones y transportes, de tal manera que existe una *diferencia de potencial eléctrico* entre el interior y exterior de las células.

Un objeto cargado (ion) gana o pierde energía conforme se mueve entre lugares de diferente potencial eléctrico, justo como un objeto con masa que se mueve “arriba” o “abajo” entre puntos de diferente potencial gravitacional. Las diferencias de potencial eléctrico se denotan usualmente por V o ΔV y es medido en *volts*; por lo tanto el potencial es también determinado *voltaje*. La diferencia de potencial relaciona, a través de una célula, el potencial del interior de las células con la solución externa, la cual, según la convención normalmente aceptada, es cero.

Las diferencias de potencial entre dos puntos que son separados por un aislador son más grandes que las diferencias entre estos puntos separados por un conductor. De esta manera, la membrana lipídica, la cual es un buen aislador, tiene una diferencia de potencial eléctrico a través de ella. Esta diferencia de potencial (transmembrane potential) equivale a menos de 0.1 V, típicamente 30 a 90 mV en la mayoría de las células animales, pero pueden ser tantas como 150 a 200 mV en células vegetales.

Por otro lado, las soluciones ricas en sales del citoplasma y sangre son muy buenas conductoras, y hay muy pocas diferencias en condiciones estables (raramente unos pocos mas de milivolts) entre dos puntos cualesquiera dentro del citoplasma de la célula o dentro de la solución extracelular. El equipo electrofisiológico permite investigar la medición de diferencias de potencial (voltaje) en sistemas biológicos.^[4]

3.2 CORRIENTES ELÉCTRICAS

El equipo electrofisiológico también puede medir corriente, la cual es el flujo del paso en un punto de una carga eléctrica por unidad de tiempo. La corriente (I) se mide en amperes (A). Normalmente, la corriente es medida por el equipo electrofisiológico en el rango de pico amperes a micro amperes.

Dos reglas acerca de las corrientes que a menudo nos ayudan a comprender fenómenos electrofisiológicos son:

- (1) La corriente se conserva en el punto de ramificación; y
- (2) La corriente siempre fluye en un circuito completo.

En mediciones electrofisiológicas, las corrientes pueden fluir por medio de capacitores, resistores, canales de iones, amplificadores, electrodos y otros, pero siempre fluyen en circuitos completos.

3.3 CONDICIONANDO LA SEÑAL Y CONDICIONADORES DE SEÑAL

Es raro que para biología, fisiología, química, las señales eléctricas o físicas sean medidas en el formato apropiado para registro e interpretación. Normalmente, una señal puede ser manipulada para optimizar ambas de estas funciones.

Por ejemplo, el almacenamiento del registro de datos es más preciso si los datos son amplificados antes de la digitalización así ocupan el rango dinámico completo del convertidor A/D, y la interpretación se mejora si el ruido extraño y las señales por encima del ancho de banda de interés son eliminadas por un filtro pasa-bajas.

3.3.1 ¿Por qué las señales deberían ser filtradas?

Un filtro es un circuito que elimina las frecuencias seleccionadas de la señal. Filtrando es más frecuente realizar la eliminación no deseada de señales y ruido del dato. La forma más común de filtrar es la filtración pasa-bajas, la cual limita el ancho de banda del dato para eliminar señales y ruido por encima del ángulo de frecuencia del filtro.

La filtración pasa-altas es requerida cuando la principal fuente de ruido está por debajo del rango de frecuencia de las señales de interés. Esto se encuentra más comúnmente cuando hacemos registros intracelulares.

Hay fluctuaciones de baja frecuencia en el potencial de la membrana debido a la variedad de mecanismos, incluyendo la suma de las entradas sinápticas.

Los pequeños potenciales sinápticos, excitadores en los que el usuario podría estar interesado son a menudo más pequeños que las fluctuaciones de baja frecuencia.

Ya que los potenciales sinápticos excitatorios son a menudo pequeños, las fluctuaciones de baja frecuencia se pueden eliminar usando un filtro pasa-altas. La filtración pasa-altas también se refiere al acoplamiento de AC.

Otro tipo de filtro que es frecuentemente usado en los registros biológicos es el notch filter. Este es un filtro especial diseñado para eliminar una frecuencia fundamental y muy pequeña. Los notch filters son más comúnmente usados de 50 ó 60 Hz para eliminar líneas de frecuencia halladas.

3.4 PREPARANDO LAS SEÑALES PARA LA CONVERSIÓN A/D

Los convertidores Analógico a Digital (A/D) tienen una resolución fija y las señales son medidas de una manera granulosa. Esto significa que todas las señales que quedan entre ciertos niveles se convierten como si tuvieran el mismo valor. Minimizar el impacto de este efecto indeseable de “cuantización”, es importante para amplificar la señal previa a presentarla al convertidor A/D.

Idealmente, la ganancia debe elegirse para que las señales más grandes de interés ocupen el rango completo del convertidor A/D $\pm 10V$, pero no excederlo. En la mayoría de los sistemas de laboratorio, el rango completo del convertidor A/D es $\pm 10V$, pero otros rangos como $\pm 5V$ y $\pm 1.25V$ son utilizados usualmente en aplicaciones industriales.

3.4.1 Error de Cuantización

El conversor A/D genera números binarios que tienen una resolución finita. Esto es, un rango pequeño de valores analógicos que producirían los mismos números binarios después de la conversión. El rango de los valores analógicos que son registrados como el mismo número digital es el error de cuantización.

En la mayoría de las aplicaciones fisiológicas, la resolución de un CAD de 12 bits es suficiente. Las mejores mediciones permitidas son del rango completo de 0.025% de resolución. La resolución estándar de un CAD de 12 bits con un rango de entrada de $\pm 10V$ es 4.88mV. Aunque esta resolución no presenta una dificultad numérica para una computadora basada en un sistema, algunos investigadores prefieren un número redondeado como 5.00mV. Esto se puede lograr fácilmente con un campo de espacio del CAD de $\pm 10.24V$ en lugar de $\pm 10V$.

3.4.2 Eligiendo la frecuencia de muestreo

Anteriormente mencionamos un teorema conocido como Teorema de Muestreo o de Nyquist (ver CONVERSIÓN ANALÓGICO-DIGITAL Y DIGITAL-ANALÓGICA) que dice que el dato debería de ser muestreado con una frecuencia igual al doble del ancho de banda de la señal o rápidamente en orden para prevenir un incremento artificial en el ruido, debido a un fenómeno conocido como aliasing, y garantiza que la señal analógica puede ser reconstruida inequívocamente de la muestra digital.

En la práctica, es común muestrear una frecuencia significativamente más rápida que la frecuencia mínima especificada por el teorema del muestreo. Esto es conocido como sobremuestreo (over sampling). El sobremuestreo debería ser usado dependiendo del tipo de experimento.

Para experimentos donde los datos son analizados en un dominio de frecuencia (p.e. análisis de ruido, análisis de impedancia), es común sobremuestrear modestamente. El principal interés es prevenir aliasing. Un filtro anti-aliasing se introduce entre la fuente de la señal y el convertidor analógico a digital para controlar el ancho de banda del dato. El factor del doble del ancho de banda requerido por el teorema de muestreo es solamente aplicable si el filtro anti-aliasing es ideal, i.e., la ganancia es uno en el pasa bandas y abruptamente cambia a cero en el tope de banda. Los filtros ideales no pueden realizarlo, aunque pueden aproximarse estrechamente. Para el análisis en un dominio de frecuencia es común usar filtros de corte pronunciado tales como lo realizan el Butterworth o Chebyshev. El muestreo es típicamente realizado a 2.5 veces el ancho de banda del filtro. Por ejemplo, si los datos son filtrados a 10 Khz. ellos deberían ser muestreados aproximadamente a 25 Khz. Las frecuencias más lentas de muestreo son inaceptables. Las frecuencias más rápidas de muestreo son aceptables, pero ofrecen pequeñas ventajas e incrementan los requerimientos de almacenamiento y análisis.

Para experimentos donde los datos son analizados en el dominio de tiempo (p.e. análisis de pulsos, curvas I-V), se requiere de muestreos más grandes porque la reconstrucción de la señal analógica necesita no solamente un filtro anti-aliasing sino de la reconstrucción de un filtro ideal. La reconstrucción más simple y más común de un filtro es unir cada muestra por una línea recta. Otras técnicas, como la interpolación cubic-spline pueden ser usadas, pero debido a los muchos requerimientos computacionales son usadas con poca frecuencia.

No hay ninguna regla normalmente aceptada con respecto a la frecuencia de muestreo de los datos para el análisis en un dominio de tiempo. En general, cinco veces el ancho de banda de los

datos es una frecuencia de muestreo común, y 10 veces es considerado bueno. El muestreo de 20 veces es excesivo y raramente usado.

3.4.3 *Formato de los datos*

Comprenderemos los términos usados para describir el formato de los datos para poder reforzar una apreciación del proceso de conversión D/A.

Offset Binary

Es el esquema de numeración binaria más simple. En este esquema el valor más pequeño es todos los ceros (i.e. 0000 0000) y el número más grande es todos los unos (i.e. 1111 1111).

Complemento a 2

Este es el esquema de numeración binaria usado por computadoras compatibles con IBM y muchas otras computadoras. En este esquema el bit más significativo (MSB) es invertido de manera que el valor más pequeño es 1000 000 y el número más grande 0111 1111.

Big Endian vs. Little Endian

Cada palabra de 16 bits consiste de 2 bytes de 8 bits. En los microprocesadores Intel 80x86 (i.e., los únicos usados en computadoras personales compatibles con IBM), el byte menos significativo es almacenado primero y el byte más significativo es almacenado en segundo lugar. Esto es llamado como “little endian”. En los microprocesadores 68000 (i.e., los únicos usados por las computadoras Macintosh) y la mayoría de los microprocesadores RISC (i.e., los únicos usados en la mayoría de las “estaciones de trabajo”), el byte más significativo es almacenado primero y el byte menos significativo es almacenado en segundo lugar. Esto es llamado como “big endian”. Esta simple inconsistencia hace complicada la transferencia de datos entre diferentes computadoras.

3.5 **DMA, MEMORY BUFFERED, I/O DRIVEN**

Es esencial para adquirir el dato ser transferido de la tarjeta CAD al disco de la computadora para un almacenamiento permanente. La forma más simple para lograr esto es bajo el control directo del microprocesador. Ya que esta técnica se hace fuerte en el uso de la entrada/salida de puertos del microprocesador, esto usualmente lo llamamos como “controladores I/O”. Un reloj de cristal controlado en la tarjeta CAD inicializa las conversiones A/D en intervalos regulares. Después de que cada conversión es completada, el estado del bit en la tarjeta es puesto en alto para indicar que la conversión ha sido terminada. El microprocesador lee el valor convertido de la tarjeta CAD y lo mueve a una localidad en memoria. Si es necesario, el microprocesador mueve un nuevo valor D/A a la tarjeta, entonces espera para convertir la siguiente palabra. Cuando un arreglo de datos convertidos se ha llenado, el microprocesador detiene las conversiones, despliega los datos y los guarda en el disco. En computadoras personales compatibles con IBM, no es posible continuar almacenando las conversiones durante la escritura a disco.

En muchos experimentos, en el modo gap-free (ver APENDICE B) la adquisición continua a disco es requerida. Para lograr esto, técnicas más sofisticadas para manejar la entrada de datos son requeridas, tales como el Acceso Directo a Memoria (DMA) o buffer de memoria.

La adquisición DMA requiere la presencia de un circuito controlador DMA en la motherboard de la computadora o en la tarjeta CAD. El propósito del controlador DMA es para mover el arreglo de los datos entre dos localidades sin involucrar al microprocesador. Para usar el

controlador DMA, el microprocesador se coloca arriba de antemano para decidir cuantos valores de datos mover, donde obtenerlos y donde ponerlos. Una vez que el proceso comienza, el microprocesador es libre de realizar otras tareas. Mientras el controlador esta cargándose adquiere los datos dentro del perímetro de los dos buffers de memoria, el procesador simultáneamente despliega los datos. Cuando el primer buffer de memoria esta lleno, el DMA comienza a llenar el segundo buffer sin una pausa. Una vez que esto comienza, el microprocesador toma la oportunidad de escribir el contenido del primer arreglo a disco. Cuando la transferencia a disco se ha completado, el microprocesador continúa desplegando en tiempo real los datos que están llenando el segundo buffer de memoria. Cuando el segundo buffer de memoria esta lleno, el DMA comienza a llenar el primer buffer nuevamente sin hacer una pausa. Este ciclo continua indefinidamente.

El buffer de memoria es otra manera de lograr el modo de adquisición gap-free. En este caso, la tarjeta CAD contiene su propia memoria dividida en dos mitades. Mientras una mitad es llenada con el dato recién convertido, el microprocesador es libre para desplegar el dato de la otra mitad y escribirlo a disco.

Una ventaja del buffer de memoria comparado con la transferencia de datos DMA es que pueden soportar potencial y extremadamente rápidas frecuencias de conversión de datos para adquisiciones de longitud fija. La memoria en la tarjeta es usualmente memoria estática de alta velocidad y puede ser llenada con una frecuencia de muestreo de unos pocos megahertz.

Por otro lado, los conversores A/D de 12 bits operan con una frecuencia mucho más lenta que 1 MHz, y el controlador DMA en más computadoras puede manejar el flujo de datos con facilidad. Una desventaja de la técnica del buffer de memoria es que es más caro porque necesita suministrar memoria y control de circuiteria en la tarjeta CAD.

3.6 INTERRUPCIONES

Muchas de las actividades durante los experimentos ocurren en instantes de tiempo impredecibles, p.e., cuando terminamos de cambiar la solución, cuando se forma un sello o cuando se encuentra una entrada presináptica. Es esencial tener un significado de la interrupción en la computadora durante la adquisición de datos para decir si un evento significativo ha ocurrido.

Líneas especiales en el bus de I/O de la computadora son dedicadas para transmitir “interrupciones”. Las interrupciones son generadas por un sistema de reloj, por el teclado y por el CAD de la tarjeta. Un uso típico de una interrupción por teclado es permitir al usuario abortar la adquisición presionando la tecla <Esc>. Un uso típico de la interrupción por CAD es disparar el comienzo de una adquisición. Otro es pegar etiquetas al dato durante la adquisición para indicar que ha ocurrido un evento específico.

3.7 TIMERS

La mayoría de los sistemas CAD tienen varios timers disponibles para una multitud de tareas. Lo más esencial es la provisión de una señal regular de reloj para iniciar cada conversión A/D.

Los timers adicionales son usados para implementar un “cambio de velocidad”. Esta es una técnica donde la frecuencia de adquisición se cambia rápidamente durante la adquisición de una frecuencia baja a alta o viceversa, sin detener la adquisición.

Raramente los timers son generalmente proporcionados para usarse en mediciones de frecuencia, conteo de eventos o intervalos de tiempo.

3.8 DIGITAL I/O

Las salidas digitales son usadas durante los experimentos para control externo del equipo o material. Por ejemplo, un osciloscopio puede ser disparado antes de que comience la adquisición, o un rayo de luz o un simulador de aislamiento podrían ser activados durante la adquisición. En otros experimentos, varios solenoides pueden ser secuencialmente activados antes, durante y después de la adquisición.

Las entradas digitales son usadas rutinariamente en aplicaciones de control industrial para monitorear el estado de solenoides y otros aparatos, pero raramente usadas en experimentos electrofisiológicos. La principal aplicación para entradas digitales en electrofisiología es para disparar o activar las adquisiciones y para indicar que una etiqueta debería ser vinculada al dato.

3.9 OPERANDO BAJO SISTEMAS OPERATIVOS MULTITAREA

Los sistemas de adquisición de datos no son naturalmente compatibles con los sistemas operativos multitarea. Como una regla general, los sistemas de adquisición de datos son diseñados para usar todos los recursos de la computadora (p.e., pantalla y disco) y usan estos recursos por petición, en lugar de que cuando la computadora los haga disponibles.

No obstante, por el diseño cuidadosamente del hardware y el software, se alcanza algún grado de compatibilidad. Sin embargo, los programas de adquisición de datos diseñados para trabajar bajo sistemas operativos multitarea han llegado a ser solo comercialmente disponibles.

Idealmente, el hardware de adquisición de datos no debería tener controladores de E/S ya que estos requieren la atención de tiempo completo de la computadora. Sin embargo, si el software de adquisición de datos puede tomar el control de la computadora, los controladores de E/S del hardware pueden ser usados. El control DMA o el buffer de memoria local están más preparados para funcionar bajo sistemas operativos multitarea ya que pueden ser usados sin tomar el control sobre la computadora.

Los manejadores de interrupciones representan un agudo problema para la adquisición de datos en los sistemas operativos multitarea. Usualmente, cuando una interrupción ocurre se requiere una respuesta inmediata. Si el sistema operativo multitarea esta ocupado actualizando una pantalla para un procesador de texto, una respuesta inmediata claramente no será próxima. Por lo tanto, un alto desempeño requiere claramente que la tarea de adquisición de datos se corra como aplicación principal. Otro problema es que los buses inteligentes usados en modernas computadoras, tales como NuBus para la Macintosh II y MicroChannel para la IBM PS/2, generalmente tienen largas latencias de interrupción de varios cientos de microsegundos o más. Una forma para minimizar el problema de las excesivas latencias de interrupción es asegurar que la rutina de adquisición de datos tiene la más alta prioridad y, en muchos casos, control exclusivo sobre la computadora. Otra forma de minimizar la excesiva latencia de interrupciones es proporcionar soporte al hardware en el sistema de adquisición. Por ejemplo, un buffer de memoria en el sistema de adquisición de datos puede ser configurado por el software de adquisición y dejarlo en estado de "preparado". Cuando vemos un disparo externo se puede generar una interrupción, entonces comienza la adquisición antes incluso de recibir una respuesta del host.

Otro problema significativo cuando se ejecuta una adquisición de datos bajo un sistema operativo multitarea resulta de la tendencia de estos sistemas a usar "ventanas" gráficas como interfaz de usuario. El sistema operativo controla todos los dibujos a la pantalla de video pero el software de adquisición de datos usualmente prefiere escribir directamente a la pantalla de video

por razones de desempeño. Para operar bajo una interfaz de ventanas el software de adquisición de datos tiene que cooperar en el uso de la pantalla, y por encima de esta cooperación puede retrasar significativamente el desempeño.

3.9.1 Soporte del Software

Hoy en día pocos investigadores están escribiendo su propio software de análisis y adquisición de datos usando lenguajes de bajo nivel y controladores de dispositivos. La mayoría de investigadores están usando uno de dos tipos de software comercialmente disponibles. El primer tipo es un entorno de desarrollo como AxoBASIC. Estos entornos permiten a los investigadores diseñar su propio software sin tener que llegar a expertos en el control de bajo nivel del hardware de adquisición de datos. El segundo tipo de software es un paquete (turn-key) con una llave o clave, como pCLAMP, AxoTape o AxoData. Estos paquetes proporcionan una sofisticada adquisición y análisis, pero con habilidades limitadas para personalizarlos.

4 TARJETA DE ADQUISICIÓN DE DATOS LABMASTER ADVANCED DESIGN

4.1 INTRODUCCION

La LabMaster AD, producto de Scientific Solutions es una tarjeta de adquisición de datos que proporciona la funcionalidad requerida para un amplio rango de pruebas y aplicaciones de medición. Proporciona el desempeño y flexibilidad necesarios para manipular difíciles tareas en el área de ingeniería e investigación científica.

Este equipo se encarga de convertir señales analógicas a un formato digital que pueda ser manipulado en la computadora. Por extensión, este equipo también permite generar señales analógicas y digitales para tareas de control.^[5]

4.2 CARACTERISTICAS

- Resolución de 12 bits para la Conversión Analógico-a-Digital y Digital-a-Analógico
- 16 canales de entrada analógica
- Tiempo de conversión 3 μ Sec. (330,000 muestras por segundo)
- 5 contadores de 16 bits
- 8 líneas de entrada digital, 8 líneas de salida digital y 8 de expansión
- Salida digital coordinada con CDA (pueden ser usadas con DMA)
- Capacidad para ganancias programables en software
- Alto desempeño del sistema CAD que tiene un arreglo canal/ganancia con auto incremento que permite el muestreo automático de entradas analógicas en cualquier orden con diferentes ganancias
- Coordinación del CAD y el CDA en canales simultáneos DMA
- Múltiples interrupciones de hardware activas

4.3 CONSIDERACIONES DE PROGRAMACION

Hay situaciones donde la programación directa de la interfaz de la LabMaster AD es necesaria. Para esto se requiere conocimiento sobre el uso de los registros y el espacio de E/S en el sistema ISA (Arquitectura Estándar Industrial).

4.3.1 Localidades E/S

El software accesa a las características de la LabMaster AD a través de las localidades de registros en el espacio de E/S. Estos registros son usados para escribir o leer información de la interfaz.

4.3.2 Dirección Base

Dieciséis localidades consecutivas a partir de una dirección base son usadas en el espacio de E/S ISA para proporcionar acceso a los registros. La base o la dirección inicial para la primera localidad E/S es determinada por la posición del interruptor SW1.

- La dirección por default para los sistemas ISA es 02C0 hexadecimal.

4.3.3 Localidades de Registros

Las siguientes dos tablas identifican los registros y sus direcciones de desplazamiento desde la dirección base. Cualquier registro listado para un byte de acceso es normalmente programado con una instrucción de E/S de 8 bits. Algunos pares de registros byte están posicionados en el espacio de E/S para su conveniente acceso usando instrucciones de 16 bits. Cualquier registro listado para una palabra de acceso debe ser programado con una instrucción de E/S de 16 bits. Los desplazamientos desde la dirección base están dados en hexadecimal.

LOCALIDADES DE SALIDA (ESCRITURA)		
Dirección	Función	Tipo
Base + 0	Dato DAC0	Word
Base + 2	Dato DAC1	Word
Base + 4	Habilitar Tarjeta	Byte
Base + 5	Control IRQ	Byte
Base + 6	Counter (9513) Data	Word
Base + 8	Counter (9513) Control	Byte
Base + 9	Seleccionar Canal Virtual ADC	Byte
Base + A	Arreglo de Datos Canal/Ganancia	Byte
Base + B	Salida Digital	Byte
Base + C	Expansión Digital	Byte
Base + D	Ultimo Canal Virtual ADC	Byte
Base + E	Control ADC	Byte
Base + F	Control DAC	Byte

LOCALIDADES DE ENTRADA (LECTURA)		
Dirección	Función	Tipo
Base + 0	ID Byte #1	Word
Base + 1	ID Byte #2	Word
Base + 2	# ID del Producto	Byte
Base + 3	# Versión del Producto	Byte
Base + 4	Estado Habilitar Tarjeta	Word
Base + 5	Estado IRQ	Byte
Base + 6	Counter (9513) Data	Byte
Base + 8	Counter (9513) Status	Byte
Base + A	Arreglo de Datos Canal/Ganancia	Byte
Base + B	Entrada Digital	Byte
Base + C	Dato ADC	Byte
Base + E	Estado ADC	Byte
Base + F	Estado DAC	Byte

Tabla 1 Localidades de Entrada/Salida

4.3.4 Registros Especiales

4.3.4.1 Registro que habilita la tarjeta

Un interruptor de software que pone en marcha la operación de la LabMaster AD es localizado en el registro que habilita la tarjeta en la localidad de salida base+4. Cuando el bit 0 es igual a cero, la interfaz es deshabilitada y no responderá a la programación de los registros de E/S. Cuando el bit 0 es igual a uno, la interfaz es habilitada y responderá a la lectura y escritura.

4.3.4.2 Estado habilitado de la tarjeta

El estado del registro que habilita la tarjeta se puede leer del registro de estado de la localidad de entrada base+4. Cuando el bit 0 es igual a cero; la LabMaster AD es deshabilitada. Cuando el bit 0 es igual a uno, la interfaz es habilitada. (Ver apéndice A).

4.4 CONVERSION ANALOGICO-A-DIGITAL (CAD)

El CAD de 12 bits usado en la LabMaster AD produce una escala de 4096 intervalos iguales dentro de un rango de $\pm 10V$. Cada intervalo en la escala equivale a 4.88mV.

En la LabMaster AD hay 16 líneas de entrada dentro del sistema A/D. Las 16 líneas pueden ser configuradas para el modo de una sola punta (single-ended SE) (16 canales de entrada) o para el modo diferencial (DI) (8 canales de entrada).

- En el modo de una sola punta, una línea lleva la señal de voltaje y todas las fuentes de voltaje comparten una tierra.
- En el modo diferencial, la entrada de voltaje es transmitida como una diferencia de voltaje entre dos líneas.

La conversión A/D toma 3 microsegundos (μsec) para completarse. Bajo condiciones óptimas las conversiones A/D pueden alcanzar un rendimiento total de 330, 000 muestras por segundo. Las entradas analógicas pasan por un amplificador de ganancia programable previo a la conversión analógico-a-digital. Ejemplos de entradas analógicas son: temperatura, corriente, voltaje, desplazamiento, aceleración, velocidad, nivel, presión y humedad.

El dato convertido es transferido a un buffer (memoria de amortiguación). Los buffers permiten velocidad de operación máxima bajo sistemas operativos sin capacidad de respuesta en tiempo real. La alta velocidad de transferencia del dato a la memoria es posible usando la capacidad del Acceso Directo a Memoria (DMA) de la computadora.

Ya que las entradas físicas son multiplexadas a un solo CAD, el rendimiento total por canal depende del total de canales usados. Para determinar el máximo rendimiento total por canal divídase la frecuencia de muestreo del CAD por el número de canales. El rendimiento total real por canal puede variar dependiendo sobre los requerimientos de la velocidad de muestreo de la aplicación y la flexibilidad del paquete de software para establecer las velocidades.

4.5 CONVERSION DIGITAL-A-ANALOGICO (CDA)

Los CDA's de 12 bits usados en la LabMaster AD toman una escala de 4096 puntos para representar intervalos iguales sobre el rango de $\pm 10V$. Cada intervalo es igual a 4.88mV.

La LabMaster AD tiene dos CDA's independientes. Cada DAC tiene doble buffer para eliminar interferencia (trasientes). Los CDA's pueden ser actualizados (i.e. cambiar el voltaje) uno a la vez o simultáneamente.

Cada valor de salida es guardado y permanece constante hasta que el siguiente valor digital es presentado al CDA. El dato binario es transferido desde la computadora hacia dentro de una FIFO de 1024 palabras. El dato es cronométricamente tomado de la FIFO dentro del CDA(s) usando el contador 0. Esta señal puede proporcionar pulsos de reloj con una velocidad arriba de 2MHz. Esto proporciona un sistema extremadamente flexible y preciso para las señales del DAC. Adicionalmente, ya que el CDA es controlado por el contador 5, la salida del CDA puede ser coordinada con la entrada del CAD o la entrada del dato repetido a través de los CDA's a la misma velocidad que fue colectado.

4.6 SISTEMA DE REGULACION DE TIEMPO Y CONTEO (STC)

Está basado en el circuito integrado 9513, que usa 11 pines de entrada y 6 de salida. Tiene 5 contadores de 16 bits de propósito general y un contador de 4 bits para control del CDA. Los contadores son muy flexibles. Cada contador puede ser usado individualmente o en tandem para:

- Intervalos de tiempo entre eventos
- Generar interrupciones de hardware
- Contar eventos discretos
- Generar señales digitales simples o complejas
- Comenzar conversiones A/D para intervalos exactos
- Comenzar conversiones D/A para intervalos exactos

4.6.1 Señales de entrada

Por medio de las 5 líneas de entrada de la señal, señales externas son conducidas a cualquiera de los 6 contadores.

4.6.2 Entradas de habilitación

Las 5 entradas de habilitación desempeñan dos funciones. La primera es controlar (i.e., habilitar) la operación del contador. El contador puede ser inicializado, detenido, reinicializado (i.e., redisp parado), descarga del contador actual, y cambiar el estado de la salida por el nivel o el límite de las señales disparadas. Un modo de habilitación especial es proporcionado por el hardware redisp arando el proceso de conteo. Las señales de habilitación pueden ser usadas como señales de entrada para cualquiera de los 6 contadores.

4.6.3 Frecuencias de entrada

Junto con las diez entradas de señal externas para los contadores, hay 5 entradas internas que están conectadas a 4MHz en la LabMaster AD. Cada una de las 5 entradas tiene un circuito que divide los 4MHz de pulsos de corriente para que la frecuencia de los pulsos sea diferente para cada entrada. El divisor puede ser 10 (BCD) o 16 (Binario) entre cada una de las frecuencias.

4.6.4 Salidas del contador

Una señal de salida puede ser un pulso, una onda cuadrada, o un ciclo de onda compleja dependiendo de la programación de uno o más contadores. El estado de salida para los contadores puede ser programado para tener alta actividad, baja actividad, un nivel alternativo alto/bajo, o alta impedancia. La salida del contador 0 es siempre una onda cuadrada. Tres juegos de salida y líneas de habilitación son conectados junto al conector externo (Out 2/Gate 3, Out 3/Gate 4, y Out 4/Gate 5).

4.7 E/S DIGITAL (DIO)

La LabMaster AD tiene 8 líneas de entrada digital, 8 líneas de salida digital y 8 líneas de expansión digital (salida). Son señales de propósito general que pueden ser usadas para transferir información de control entre la interfaz y el ambiente exterior (i.e., identificar disparos, encender LEDs o transmisiones).

Cada línea de E/S (entrada/salida) digital es una señal estándar TTL. Una señal TTL es la conversión del estado lógico ON/OFF en un nivel de voltaje bajo/alto. El estado bajo u OFF es un voltaje entre 0 y .8V. El estado alto u ON esta entre 2.5V y 5.0V. Cualquier voltaje arriba de 5.0 puede dañar las líneas E/S digital en la LabMaster AD.

4.8 INTERRUPCIONES DE HARDWARE

En una computadora, las interrupciones generadas al hardware proporcionan una sincronización con periféricos relativamente lentos tales como una impresora, reloj o teclado. Donde el CPU puede procesar miles de instrucciones en un segundo, los periféricos lentos solamente manejan de 300 a 2400 eventos en un segundo. La computadora puede continuar con otras tareas mientras espera por un periférico lento o eventos discontinuos en cambio de tener que consumir tiempo en un “ciclo de espera” para verificar el estado de los eventos de los periféricos. Esto es muy importante en un sistema multitarea donde la computadora puede realizar otras funciones mientras el evento de interrupción no esta ocurriendo.

Usualmente, alguna señal de estado (tal como “dato listo” o “listo para transferir”) es conectada a la línea de interrupción (IRQ) de la computadora. Cuando la línea de interrupción es activada por la señal de estado, la computadora detiene el programa en ejecución (salvando la información vital para reiniciar el programa). El procesador entonces encuentra la siguiente instrucción y salta a esa localidad de memoria. Un pequeño programa llamado manejador de interrupciones ocupa esta área de memoria. El manejador de interrupciones hace lo que sea necesario para darle servicio a la interrupción. Por ejemplo, si la interrupción es generada por la señal “dato listo” del teclado, el manejador de interrupciones habría de leer del teclado el dato y almacenarlo en un buffer. El manejador de interrupciones regresa al programa principal.

Los sistemas ISA tienen varias líneas de interrupción. Aquí, cada manejador asume que fue invocado por una interrupción generada por un dispositivo en particular. Cuando una computadora tiene solamente una línea de interrupción, o tiene varios dispositivos adjuntos a una línea, el manejador de interrupciones tiene una votación de todos los dispositivos para determinar cual interrupción generar para la petición del servicio. Entonces, el manejador atiende a ese dispositivo.

Hasta tres canales de interrupción pueden estar simultáneamente activos en la LabMaster AD. El contador, el CDA, y las señales del CAD pueden conectarse en la misma línea IRQ o en líneas separadas.

La LabMaster AD esta diseñada de manera que los subsistemas de la tarjeta puedan ser directamente conectados a una de las interrupciones de hardware en el sistema ISA. El diseño usa la circuitería del contador para la interfaz y control de las entradas de interrupción de propósito general. El conteo terminado DMA del CAD y las banderas de desbordamiento de la FIFO del CAD pueden usarse para generar interrupciones. El conteo terminado DMA del CDA y las banderas de desbordamiento de la FIFO del CDA pueden usarse para generar interrupciones.

Un requerimiento común es coleccionar el dato y poner intervalos. Cuando el intervalo es grande (i.e., cada 200 microsegundos), uno de los contadores de 16-bit puede ser usado para cronometrar el intervalo. Las líneas de salida de los contadores 1-4 están ya conectadas internamente en la LabMaster AD al circuito de control IRQ.

Cuando usamos entradas externas digitales para interrupción del sistema ISA, la entrada externa puede ser conectada a una de las entradas del contador o a los pines de habilitación.

4.9 ACCESO DIRECTO A MEMORIA (DMA)

El Acceso Directo a Memoria (DMA) es la transferencia de información entre la Memoria de Acceso Aleatorio (RAM) y una localidad E/S. La transferencia puede hacerse de la RAM a E/S o de E/S a RAM, pero no en ambas direcciones al mismo tiempo en el mismo canal. Hay 8 canales para DMA en un sistema de bus ISA. Son usados por diferentes periféricos para mejorar la velocidad de transferencia de los datos entre el dispositivo y la RAM.

En la LabMaster AD, las transferencias DMA pueden hacerse de la FIFO del CAD o a la FIFO del CDA. La LabMaster AD puede usar los canales DMA 5, 6 o 7. Los canales DMA no pueden ser compartidos por dos dispositivos al mismo tiempo, de este modo las transferencias DMA del CAD y CDA deben ocurrir en diferentes canales DMA. Dos canales DMA pueden ser usados concurrentemente con ambos procesos simultáneamente activos.

4.10 MANEJO DE LA LABMASTER AD (CONTROLADOR LABPAC32)

Para tener un sistema operativo seguro, se restringe acceso al hardware y solo puede accederse a través de Controladores de Dispositivos. Un Controlador de Dispositivo es un software que es cargado y se vuelve parte del sistema operativo. Cada dispositivo de hardware requiere un único controlador. Y cada sistema operativo requiere un diferente controlador de dispositivo. Esto significa que cada dispositivo debe tener un controlador para Windows 95 y uno diferente para Windows NT y aun otro para Windows 98 y 2000.^[6]

Este controlador de dispositivo o API (Interfaz de Programación para Aplicaciones) proporciona un conjunto de funciones que pueden ser usadas en programas escritos para Windows 95/98/NT/2000. Estas funciones usan la Convención Estándar de Llamadas (STD) y puede ser usada por programas escritos en C, C++, C#, Java, VB, Delphi, Pascal y muchos otros.

La LabPac32 API proporciona un método consistente para acceder al diferente hardware (i.e. programar al hardware de adquisición de datos) entre diferentes sistemas operativos Windows. El DLL LabPac32 accesa al hardware usando el controlador de dispositivo.

Para usar la LabPac32 API, el archivo LabPac32 DLL debe ser cargado en el sistema junto con el controlador de dispositivo para el hardware.

La LabPac32 API proporciona un método consistente para programar el hardware de adquisición de datos.

Las funciones de LabPac32 API están contenidas en el archivo LabPac32.dll. Cada programa debe llamar a la función LabPac_AddDev antes de llamar a cualquier otra función de LabPac. Esta función “abre” el dispositivo específico de hardware y regresará un número de dispositivo. El número de dispositivo es asociado con el dispositivo de hardware y es usado por todas las demás llamadas para acceder al hardware correcto. Antes de finalizar tu aplicación, tú debes liberar el dispositivo a través de la llamada a LabPac_DelDev.

Hay hardware que es inicializado cuando el controlador del dispositivo es abierto o cuando se llama a la función LabPac_Reset. Use la función LabPac_Versión para determinar el controlador del dispositivo y la versión dll que es cargada en el sistema. Algún hardware tendrá un número de serie único. Este número puede ser determinado llamando a LabPac_GetSN.

La mayoría de las funciones regresan un código de estado. LabPac_GetLastError puede utilizarse también para determinar si ha ocurrido algún error.

4.10.1 Código Ejemplo de Inicialización

```
/* Segmento de código en C */
int main (argc, argv)
{
    ulong DevNum;
    long DllVer, DrvVer;
    char DeviceName[9] = "LMPro0";
    char SN[11];
    LabPac_AddDev(DeviceName, DevNum); /*Esta función debe ser llamada*/
    LabPac_Version(Dev, &DllVer, &DrvVer); /*Determina la versión del Driver y del Dll*/
    LabPac_Reset(DevNum); /*Coloca al hardware en las condiciones por default*/
    LabPac_GetSN(DevNum, SN); /*Obtiene el número de serie para el hardware*/
    Return(0);
}
/*Fin del ejemplo*/[7]
```

5 ANALISIS, DISEÑO E IMPLEMENTACIÓN

5.1 ANÁLISIS

El análisis del sistema se llevó a cabo con los siguientes objetivos en mente ^[8]:

1. identificar la necesidad del cliente
2. evaluar el concepto del sistema para establecer la viabilidad

5.1.1 *Identificar la necesidad del cliente*

El primer paso del proceso de análisis del sistema afecta a la identificación de la necesidad. Como en la ingeniería de la información, la intención es entender los objetivos del producto y definir las metas necesarias para alcanzar esos objetivos.

El trabajo de los laboratorios de investigación involucra el diseño y plantación de experimentos para los cuales se requieren ciertos procedimientos y herramientas de programación. Establecer una configuración con una computadora para que realice el trabajo deseado no es siempre simple y requiere de un conocimiento básico de las operaciones y accesorios de una computadora. Muchas compañías realizan ese trabajo por el usuario y es posible adquirir comercialmente productos que realizan la operación que se desea. A pesar de ello, es mas frecuente que el investigador requiera de alguna modificación especial del programa o instrumento que utiliza, a veces de tal magnitud, que desea diseñar un instrumento o programa enteramente acorde a sus necesidades.

Se sabe que algunas de estas herramientas llegan a ser muy costosas, sobre todo si son comercialmente disponibles. Ante esto los investigadores se dan a la tarea de la búsqueda de otras alternativas para desarrollar sus propias herramientas de trabajo. Así pues, este proyecto surge por la necesidad de proporcionar al laboratorio de Biofísica Cardíaca con la herramienta de programación para realizar alguno de sus trabajos de investigación.

Una computadora podría adquirir y/o generar señales analógicas. Para poder realizar estas tareas, necesita la asistencia de un conversor analógico-digital (CAD). Este componente se fabrica típicamente como una tarjeta que se inserta en el corredor de datos de la computadora (bus). Este instrumento es capaz de transformar magnitudes eléctricas que varían en forma continua, a valores digitales que varían en forma discontinua. Inversamente, el CAD puede transformar valores numéricos de voltaje en magnitudes eléctricas correspondientes.

El laboratorio de Biofísica Cardíaca cuenta con la tarjeta digitalizadora LabMaster AD pero no con un software que soporte la misma. Por consiguiente en este proyecto se pretendió crear el desarrollo de un software que nos permite la adquisición y el análisis de señales electrofisiológicas provenientes de los diversos experimentos.

Las 5 características más importantes de un CAD son: el número de canales digitales, el número de canales de conversión analógica a digital, el número de canales de conversión digital a analógica, la velocidad máxima de conversión dada por su reloj y finalmente, su resolución.

La velocidad máxima de conversión varía entre 100 KHz. (100 000 conversiones por segundo) hasta 1.25 Mhz. en las tarjetas más rápidas. El costo de estas tarjetas fluctúa entre los 400 y 2000 dólares respectivamente. Es importante notar que esta velocidad se reparte entre todos los canales sobre los cuales se opera. Este aspecto debe considerarse con cuidado porque la velocidad de conversión que se necesita depende del número de canales muestreados y de la resolución necesaria.

Respecto al almacenamiento de datos, el escoger el medio de almacenaje, depende de la cantidad de datos adquiridos y la necesidad de acceder a ellos con rapidez.

Se requiere que el sistema haga una captura de datos analógica-digital con cuatro canales de entrada. Antes de hacer la captura es necesario establecer los parámetros de la misma que especifiquen el modo de adquisición, el número y el nombre de los canales a ser utilizados, el intervalo de muestreo y el tiempo de captura.

Para evitar que se presenten interrupciones en el sistema los datos capturados durante la adquisición serán almacenados en memoria y posteriormente se guardarán a disco. El formato de archivo requerido para el almacenamiento de los datos es con extensión ABF (Ver apéndice B).

Este tipo de archivos nos permitirá generar archivos de datos similares a otros programas comerciales lo cual admitirá la apertura de nuestros archivos en dichos programas, tal como es el caso de pCLAMP que nos permite la apertura y el manejo de este tipo de archivos.

5.1.2 Estudio de la viabilidad

Los CAD se manipulan generalmente con programas computacionales que se instalan junto con la tarjeta. Los controladores de cada tarjeta son particulares de cada fabricante y en la gran mayoría de los casos no son portátiles entre una computadora y otra. En los últimos años, las compañías fabricantes de CAD han creado controladores y programas que facilitan de gran manera la manipulación y control de estos instrumentos.

La computadora es un instrumento multipropósito que permite realizar todas las operaciones de control del sistema, así como el almacenamiento y procesamiento de los datos obtenidos. La computadora emplea sus recursos para definir el orden de eventos que ocurren durante el experimento y controla el flujo de datos que manipula el CAD. El CAD posee una memoria limitada y una vez que los datos son obtenidos o generados deben transferirse a la memoria o disco duro de la computadora.

En una computadora el control y procesamiento se realiza a través de programas específicos para cada aplicación. Estos programas son de 2 tipos: de propósito general, que permite al usuario diseñar una aplicación específica para sus intereses, y aquellos de propósito específico. Un ejemplo del último caso es pCLAMP de Molecular Devices o Datawave de Datawave Technologies, que se utilizan para la adquisición de datos obtenidos en experimentos de fijación de voltaje y en la adquisición de señales neuronales, respectivamente. En cada caso, el fabricante recomienda alguna tarjeta CAD que el programa puede manipular. El usuario tiene acceso a una colección de funciones de adquisición, proceso y despliegue de los datos. Otro ejemplo de este tipo de programas es DIAdem de National Instruments con un costo aproximado de \$995 USD a \$2995 USD dependiendo de la versión del mismo.

Una segunda clase de programas permite al usuario desarrollar sus propias aplicaciones a partir de una interfase de programación. El uso de estos programas requiere un manejo básico de algún lenguaje computacional. Un ejemplo de tal programa es LabView de National Instruments, que utiliza una manipulación gráfica para el control de CAD así como del resto del procesamiento y manipulación de datos. Este programa tiene un costo de \$4095 UDS a \$4299 USD aproximadamente dependiendo de la versión. Otro ejemplo es LabWindows, de la misma compañía que utiliza el lenguaje C para fabricar aplicaciones en forma sencilla y rápida. Otro producto similar es HPView producto de Data Translation.

Programas de desarrollo de aplicaciones son deseables en situaciones donde se desea un control acabado de las situaciones experimentales especiales o en casos donde aparecen situaciones nuevas, no cubiertas por un programa comercial. También se da el caso de que los programas comerciales cuentan con un número de funciones que bien podrían ser no utilizadas o requeridas por el usuario.

En cuanto al controlador que requerimos para hacer uso de la tarjeta bajo un ambiente de sistema multitarea como lo es Windows no tuvo costo ya que nos fue proporcionado por el fabricante haciéndole mención que sería utilizado para desarrollo, de igual manera que las bibliotecas para poder generar y manipular archivos ABF.

Finalmente evaluando los diferentes tipos de programas disponibles, los costos de los mismos y que ninguno proporciona soporte para el tipo de tarjeta digitalizadora con la que contamos encontramos que es viable la realización de este proyecto ya que sin él no habría manera de hacer uso de la misma.

5.2 DISEÑO

Todos los sistemas basados en computadora pueden modelarse como la transformación de la información empleando una arquitectura del tipo entrada-proceso-salida. Se debe establecer el límite de información entre el sistema que se está implementado y el entorno en que va a operar. El diseño de la arquitectura del sistema se deriva de un proceso de partición en donde se establecen las relaciones entre los elementos de una solución de software con partes del mundo real.

Un principio importante que ayuda a tratar la complejidad de un sistema es la modularidad. La descomposición del problema se realiza a través de un diseño descendente que a través de niveles sucesivos de refinamiento se obtendrán diferentes módulos. Normalmente los módulos de alto nivel especifican que acciones han de realizarse mientras que los módulos de bajo nivel definen como se realizan las acciones.

5.2.1 Programación Modular

La programación modular tiene muchas ventajas. A medida que el tamaño de un programa crece, muchas tareas de programación se hacen más difíciles. La diferencia principal entre un programa modular pequeño y un programa modular grande es, simplemente, el número de módulos que cada uno contiene, ya que el trabajo con programas modulares es similar y solo se ha de tener presente el modo en que unos módulos interactúan con otros.

La modularidad tiene un impacto positivo en los siguientes aspectos de la programación:

- Construcción del programa. La descomposición de un programa en módulos permite que los diversos programadores trabajen en modo independiente en cada uno de sus módulos. El trabajo de módulos independientes convierte la tarea de escribir un programa grande en la tarea de escribir muchos programas pequeños.
- Depuración del programa. La depuración de programas grandes puede ser una tarea enorme, de modo que se facilitara esa tarea, al centrarse en la depuración de pequeños programas más fáciles de verificar.
- Legibilidad. Los programas grandes son muy difíciles de leer, mientras que los programas modulares son más fáciles de leer.
- Eliminación de código redundante. Otra ventaja del diseño modular es que se pueden identificar operaciones que suceden en muchas partes diferentes del programa y se implementan como subprogramas. Esto significa que el código de una operación aparecerá solo una vez, produciendo como resultado un aumento en la legibilidad y modificabilidad.

En base a los requerimientos el sistema esta formado por cinco módulos básicos, como se muestran en el diagrama a bloques de la Figura 4 y los cuales son:

- 1) **Tarjeta LabMaster AD**
- 2) **Captura DMA**
- 3) **Archivos ABF**
- 4) **Análisis**
- 5) **Graficado**

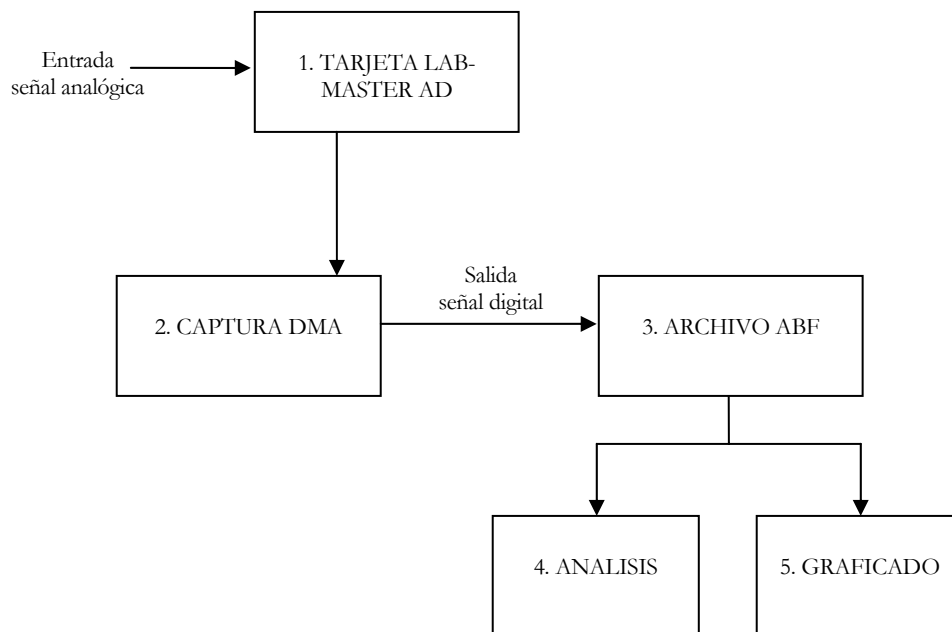


Figura 4 Diagrama a bloques del sistema

Explicaremos los procedimientos que conforman cada uno de estos módulos.

5.2.1.1 Módulo Tarjeta LabMaster AD

Los procedimientos asociados a este módulo son:

Procedure Initialize

Descripción:

Esta rutina se encarga de detectar si la tarjeta digitalizadora esta presente en el equipo y de ser así la inicializa en el sistema con los parámetros necesarios para su funcionamiento.

Primero agrega el dispositivo, lo resetea y detecta la presencia del hardware y si es así obtiene el identificador de la versión y del producto, la versión del Dll y del driver así como también el número de serie el cual nos servirá para saber el modelo de la tarjeta que ha sido detectada. Si la tarjeta no se encuentra presente entonces no habilitará la función de captura.

Procedure FileExit1Execute

Descripción:

En esta rutina además de cerrar nuestro archivo ABF creado por default al inicio del programa también se encarga de borrar el dispositivo que fue adicionado al inicio del programa en la rutina de Initialize.

5.2.1.2 Módulo Archivos ABF

Los procedimientos asociados a este módulo son:

Procedure IniProtocol

Descripción:

Esta rutina se encarga de crear e inicializar la cabecera de un archivo ABF con los parámetros necesarios para el funcionamiento y manejo del archivo.

Procedure OpenABF

Descripción:

La función de esta rutina es la apertura de un archivo, el cual debe ser de extensión valida por nuestro programa. Las extensiones validas son: ABF, DAT, PRO y TXT. Una vez abierto el archivo podemos leer todos los parámetros de la cabecera y obtenemos el número de datos que contiene nuestro archivo en caso de tenerlos y poderlos graficar.

Procedure SaveFile

Descripción:

Esta rutina se encarga de guardar los datos que hemos capturado así como los cambios que hemos hecho en los parámetros de la cabecera del archivo. Primero checa que el archivo que vamos a guardar aun no tenga un nombre para asignarle uno y guardar los cambios y de no ser así solamente guardar los cambios.

Procedure FileNew1Execute

Descripción:

Esta rutina cierra el archivo ABF que teníamos abierto e inmediatamente inicializa los valores de la cabecera del archivo creando de esta forma un archivo nuevo listo para recibir nuevos parámetros y guardar información de captura.

Procedure Close1Click

Descripción:

En esta rutina se cierra el archivo ABF en uso y al igual que en el procedimiento anterior se inicializan los valores de los parámetros de la cabecera del archivo así como también se limpian los componentes y las variables que son utilizadas para hacer la gráfica del archivo.

5.2.1.3 Módulo Captura DMA

Los procedimientos que forman parte de este módulo son los siguientes:

Procedure EditProtocolClick

Descripción:

En esta rutina se hace un llamado a una unidad aparte que se encarga del manejo del protocolo del archivo ABF, es decir de todos los parámetros relacionados con la cabecera del archivo requeridos para la creación, captura y graficado del mismo.

Procedure bDMAClick

Descripción:

En esta rutina se realiza el proceso de captura para lo cual es necesario obtener del protocolo los valores requeridos por el controlador de la tarjeta para verificar si son los adecuados y proceder a la captura dependiendo del número de datos a capturar. Los datos capturados son almacenados en un buffer temporal para después ser guardados en el archivo ABF.

5.2.1.4 Módulo Análisis

Este módulo solamente consta de un procedimiento que describimos a continuación:

Procedure FileProperties1Click

Descripción:

Nos muestra información del valor de los parámetros de la cabecera del archivo que es el primer bloque de datos del principio de un archivo de datos ABF. La cabecera contiene los parámetros que describen la estimulación, la adquisición y la jerarquía de los datos. Describe el contenido del archivo de datos cuando fue adquirido. Los parámetros principalmente utilizados por nuestro programa los describimos a continuación:

Información de identificación y tamaño del archivo

Nombre: nOperationMode

Tipo: short

Descripción: nos indica el modo de operación de captura: 1=manejador de eventos, longitud variable; 2=osciloscopio, manejador de eventos-longitud fija; 3=Gap-free; 4=osciloscopio de alta velocidad; 5= estimulación episódica (Clampex solamente).

Nombre: lActualLength

Tipo: long

Descripción: número actual de muestras CAD (agregadas) en el archivo de datos.

Nombre: lActualEpisodes

Tipo: long

Descripción: número actual de barridos (sweeps) en el archivo incluyendo barridos promediados. Si el modo de operación es el 3 (gap-free) el valor de este parámetro es 1.

Estructura del archivo

Nombre: lDataSectionPtr

Tipo: long

Descripción: número del bloque de inicio de la sección de datos.

Nombre: nDataFormat

Tipo: short

Descripción: representación del dato, 0= 2 bytes entero; 1= IEEE 4 bytes flotante

Información de la jerarquía del experimento

Nombre: nADCNumChannels

Tipo: short

Descripción: número de canales de entrada analógica muestreados.

Nombre: fADCSampleInterval

Tipo: float

Descripción: intervalo entre las muestras (A/D) multiplexadas (μ s).

Nombre: fSecondsPerRun

Tipo: float

Descripción: solicitud de la longitud de la adquisición en segundos. 0 significa usar el espacio disponible en disco. 1 significa ignorar este parámetro y referirse a lEpisodesPerRun.

Nombre: lNumSamplesPerEpisode

Tipo: long

Descripción: número de muestras multiplexadas CAD por barrido si nOperationMode es 4 ó 5. Indefinido si nOperationMode es 1 ó 3.

Nombre: lEpisodesPerRun

Tipo: long

Descripción: solicitud del número de barridos/corridas (run). 0= Corrida hasta que sea terminada por el usuario. Si nOperationMode= 3 (gap-free), este parámetro es 1 y la solicitud de la longitud de la adquisición se pone en fSecondsPerRun.

Nombre: lRunsPerTrial

Tipo: long

Descripción: solicitud del número de corridas/experimento (trial). 0= Corrida hasta que sea terminada por el usuario. Las corridas se promedian. Si nOperarionMode= 3 (gap-free), el valor de este parámetro es 1.

Información del Hardware

Nombre: fADCRange

Tipo: float

Descripción: escala de entrada completa CAD positiva en volts.

Nombre: fDACRange

Tipo: float

Descripción: escala completa CDA rango en volts.

Nombre: lADCResolution

Tipo: long

Descripción: número de conteos CAD correspondientes al voltaje de la escala positiva completa en ADCRange.

Nombre: lDACResolution

Tipo: long

Descripción: número de conteos CDA correspondientes al voltaje de la escala positiva completa en DACRange.

Información del ambiente

Nombre: sCreatorInfo

Tipo: 16 char

Descripción: nombre y versión del programa usado para crear el archivo.

Información Multi-canal

Nombre: nADCPtoLChannelMap

Tipo: short

Descripción: mapa del canal CAD físico-a-lógico. Las entradas están en el orden físico 0, 1, 2,..., 14, 15. Si hay menos de 16 canales lógicos en el sistema, el arreglo es rellenado con -1. Todos los canales soportados por el hardware están presentes, aun si solo un subconjunto es usado.

Nombre: nADCSamplingSeq (0-15)

Tipo: short

Descripción: secuencia de los canales de muestreo CAD. Este es el orden en el cual los canales físicos CAD son muestreados. Si menos que el número máximo de canales son muestreados, rellena con -1.

Nombre: sADCChannelName (0-15)

Tipo: 10 char

Descripción: nombre del canal CAD en orden del número de canal físico.

Nombre: sADCUnits (0-15)
Tipo: 8 char
Descripción: las unidades de usuario para los canales CAD en orden del número de canal físico.

Nombre: fADCProgrammableGain (0-15)
Tipo: float
Descripción: ganancia programable CAD en orden del número de canal físico. Su valor por default es 1.

Nombre: fADCDisplayAmplification (0-15)
Tipo: float
Descripción: despliega la amplificación del canal CAD en orden del número de canal físico. Su valor por default es 1.

Nombre: fADCDisplayOffset (0-15)
Tipo: float
Descripción: despliega el offset del canal CAD en orden del número de canal físico. Su valor por default es 0.

Nombre: fInstrumentScaleFactor (0-15)
Tipo: float
Descripción: instrumento del factor de escala en orden del número de canal físico CAD.

Nombre: fInstrumentOffset (0-15)
Tipo: float
Descripción: instrumento offset en orden del número de canal físico CAD (las unidades de usuario corresponden a 0 volts para el ADC). Su valor por default es cero.

Nombre: fSignalGain (0-15)
Tipo: float
Descripción: acondicionador de la ganancia de la señal en orden del número de canal físico CAD. Su valor por default es 1.

Nombre: fSignalOffset (0-15)
Tipo: float
Descripción: en orden del número de canal físico CAD (unidades de usuario). Su valor por default es 0.

5.2.1.5 *Módulo Graficado*

Los procedimientos que componen este módulo son los siguientes:

Procedure bChartClick

Descripción:

Esta rutina verifica el número de canales a graficar y prepara los componentes indicados para el graficado de los datos. Esta rutina solo nos graficará la primera pantalla de cada uno de los

canales a graficar (máximo 4) y después habilitará un componente que continuará con el graficado de los datos restantes contenidos en el archivo.

Procedure Timer2Timer

Descripción:

Esta rutina continúa con el graficado de los datos restantes de cada uno de los canales a graficar indicados en el procedimiento anterior.

Procedure BResetZoomClick

Descripción:

Esta rutina se encarga de restaurar la vista del componente de graficación seleccionado si es que ha sufrido alguna modificación en su tamaño.

Procedure BGrowupClick

Descripción:

La función de esta rutina consiste en maximizar el componente de graficado relacionado con el canal seleccionado.

Procedure BDecreaseClick

Descripción:

Al contrario del procedimiento anteriormente descrito este se encarga de mostrar todos los componentes de graficado en su tamaño inicial, es decir de minimizar el componente si sufrió algún cambio en su tamaño.

5.3 IMPLEMENTACIÓN

5.3.1 Prototipado Evolutivo

La etapa de implementación (codificación) traduce el diseño en un programa escrito en un lenguaje de programación. Los algoritmos y las estructuras de datos realizadas en pseudo código han de traducirse a un lenguaje que entiende la computadora.

La implementación de dicho programa se desarrolló basándose en el modelo de prototipado evolutivo (Figura 5). En este modelo el sistema se desarrolla en incrementos, de forma que puede modificarse de manera inmediata en respuesta a la realimentación de cliente y del usuario final. La mayoría de los esfuerzos de prototipado evolutivo comienzan con el prototipado de la interfaz de usuario, y luego desarrollan el sistema completo a partir de esta.^[9]

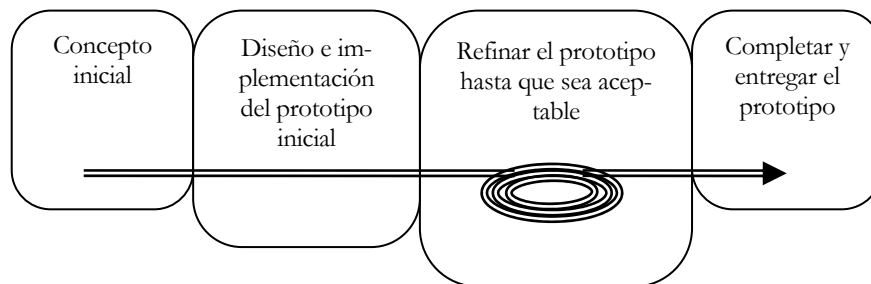


Figura 5 Modelo de Prototipado Evolutivo

Con el prototipado evolutivo comienza diseñando e implementando las partes más destacadas del programa en un prototipo. Luego continúa aumentando y refinando el prototipo hasta que termina. Normalmente el prototipo evoluciona y se transforma en el software que se entrega.

5.3.2 *ALISE (Asistente para laboratorio en experimentos científicos)*

El sistema se denominó Assistant for Laboratory In Scientific Experiments (ALISE) que registra señales analógicas las cuales nos permite guardarlas y visualizarlas posteriormente. Dicho sistema también nos permite crear archivos ABF.

Desde el punto de vista del programador, las aplicaciones que se ejecutan bajo Windows se caracterizan sobre todo porque utilizan ventanas como interfaz de comunicación con el usuario y porque se controlan a través de eventos. La programación de aplicaciones para interfaces gráficas multitarea como Windows, enfrenta al programador a otros problemas muy distintos a los habituales en la creación de programas sencillos en modo texto.

Por todos estos requerimientos el programa se realizó en Delphi^[10], que es un buen entorno de programación y facilitó el desarrollo de aplicaciones para Windows lo cual nos permitió desarrollar una interfaz amigable con el usuario, fácil de manejar y que realiza las funciones necesarias para la adquisición y análisis de señales. La comunicación con la tarjeta LabMaster AD para su programación se logró mediante el controlador LabPac 32 (proporcionado por el fabricante). Ver MANEJO DE LA LABMASTER AD (CONTROLADOR LABPAC32).

Para graficar las señales se hizo uso del componente TChart que contiene una variedad de tipos de gráficas así como diferentes propiedades que nos permiten controlar el número de series a graficar, ponerles título, fijar escalas automáticamente en los ejes o establecer valores, poner títulos, poner etiquetas, etc.

Para cada señal capturada se implementó un TChart a través del cual se grafican los datos capturados, para cada gráfica hay un botón que permite hacer un acercamiento a la señal permitiéndonos ver con más exactitud los valores de los ejes y un botón que muestra los valores de la señal.

Todo el protocolo y la información de los datos de las adquisiciones son almacenados en el formato ABF. Consecuentemente no solo los datos sino también la información del protocolo pueden ser extraídos del ABF. Estos archivos pueden ser creados y leídos por computadoras bajo ambiente Windows. Para guardar la señal capturada se utilizó la biblioteca Axabffio32.dll la cual contiene las rutinas para acceder a los datos del archivo ABF. (Ver apéndice B).

En la Figura 6 vemos la interfaz principal del programa mostrando las gráficas de una señal capturada a través de cuatro canales. La aplicación contiene botones de acceso rápido para hacer la captura de la señal, guardar, abrir, crear un nuevo archivo, mostrar las propiedades de un archivo ABF y graficar.

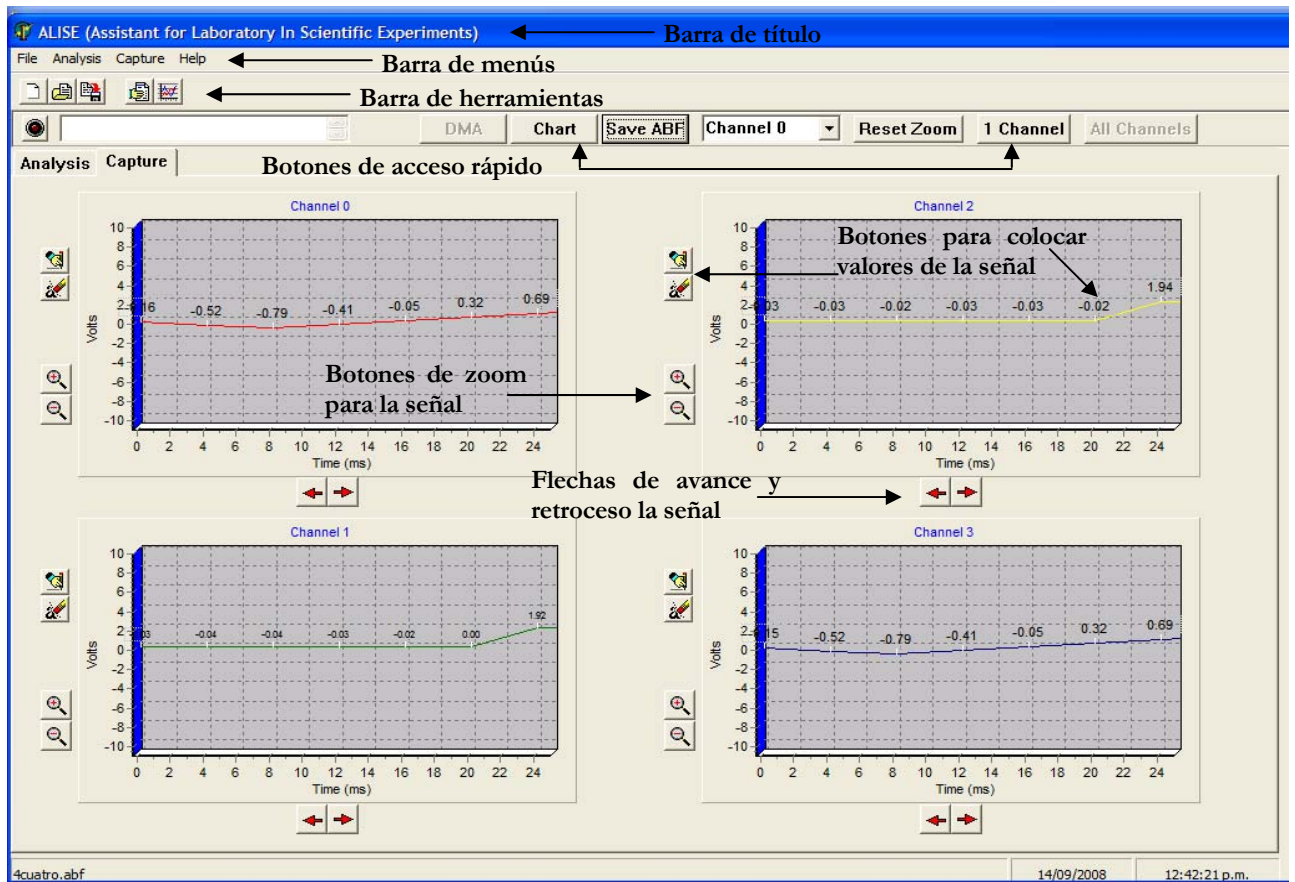


Figura 6 Pantalla Principal del Programa

EJEMPLOS DE SEÑALES CAPTURADAS

A través de un generador de funciones y con ayuda de un osciloscopio se generó una señal (onda cuadrada) la cual fue capturada bajo el modo gap-free con un intervalo de muestreo de 1 khz en un tiempo de 1 segundo y que contiene 1000 muestras. En la Figura 7 se muestra la gráfica de esta señal con el programa ALISE (Asistente para laboratorio en experimentos científicos) y como un comparativo la observamos en la Figura 8 graficada con el programa comercial Clampex de Axon Instruments (ahora Molecular Devices). Y en la Figura 9 se muestran datos de la cabecera del archivo capturado por nuestro sistema.

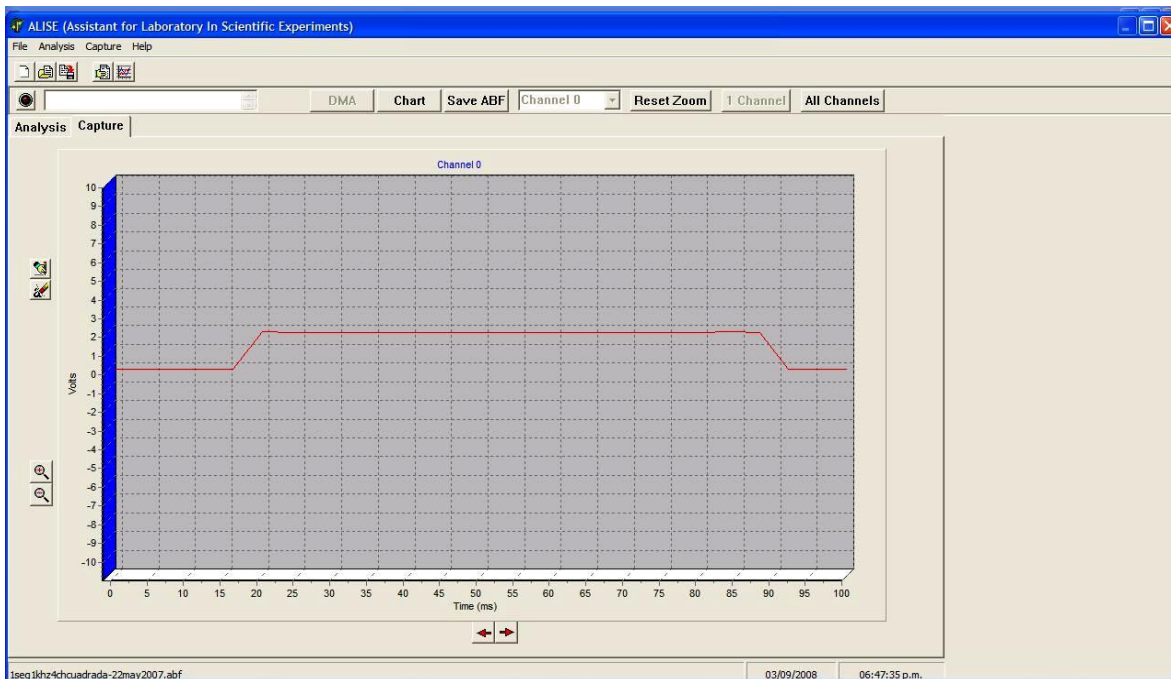


Figura 7 Gráfica de la señal con ALISE

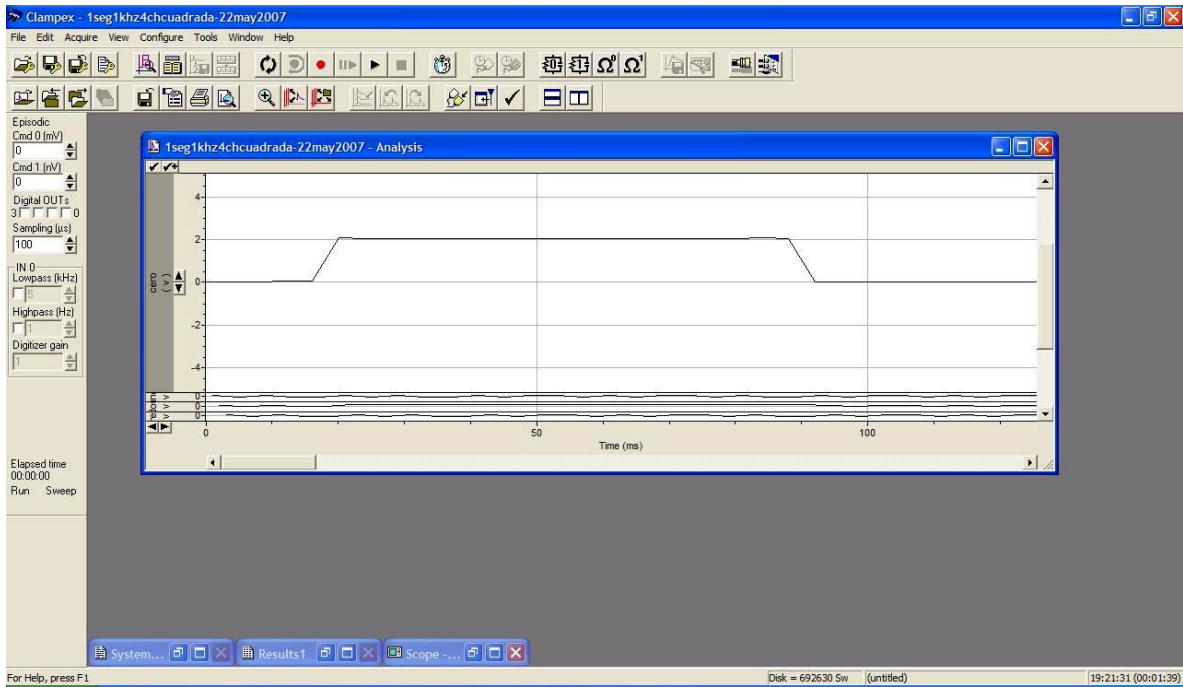


Figura 8 Gráfica de la señal con Clampex

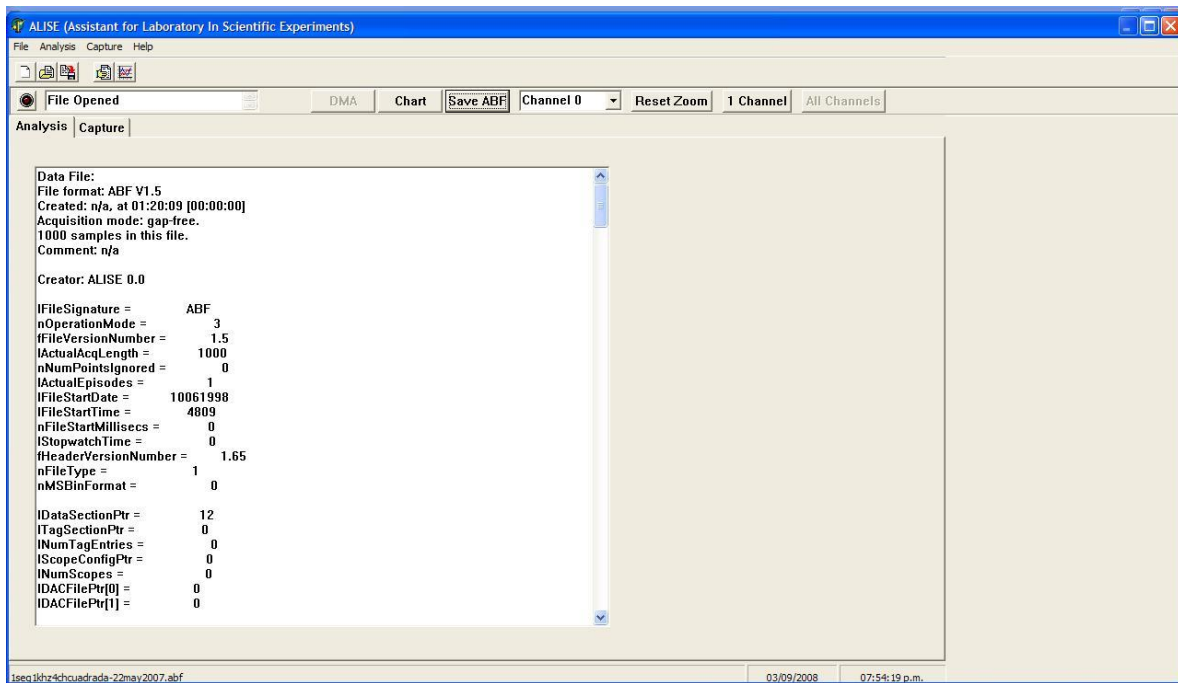


Figura 9 Datos del archivo con ALISE

La Figura 10 muestra una onda cuadrada capturada con un tiempo de un segundo y una frecuencia de muestreo de 5 khz. Los valores de la señal oscilan de 2 volts a -2 volts. El archivo contiene 5 000 muestras.

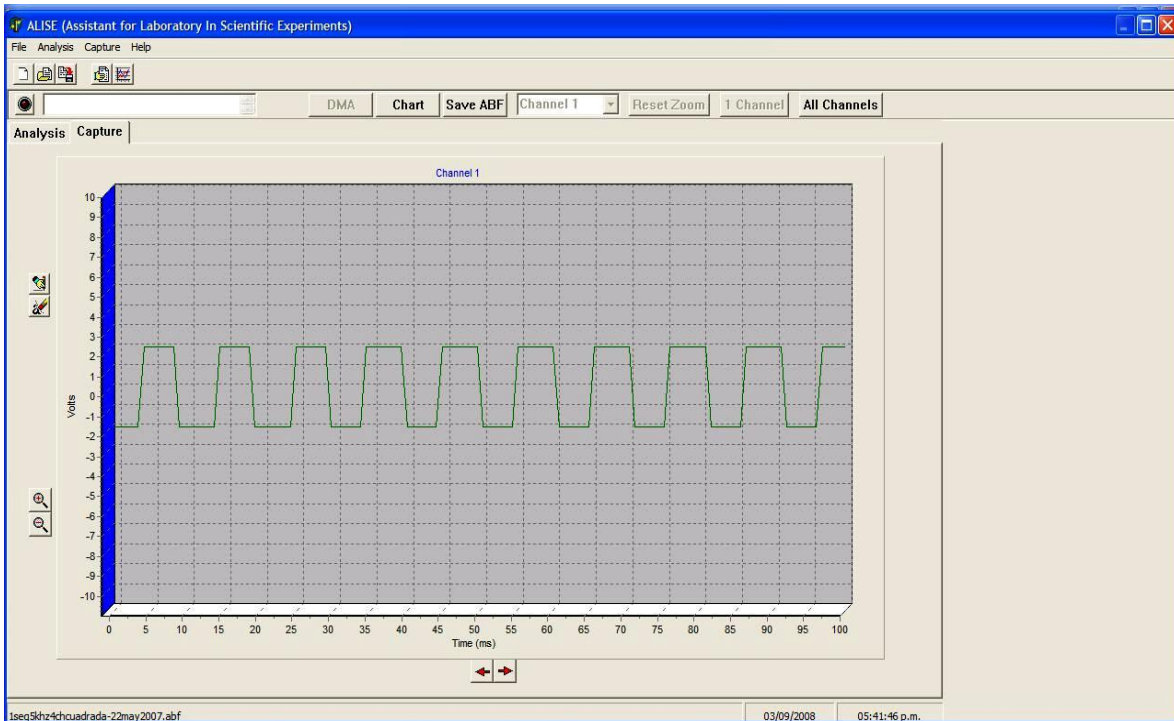


Figura 10 Señal onda cuadrada

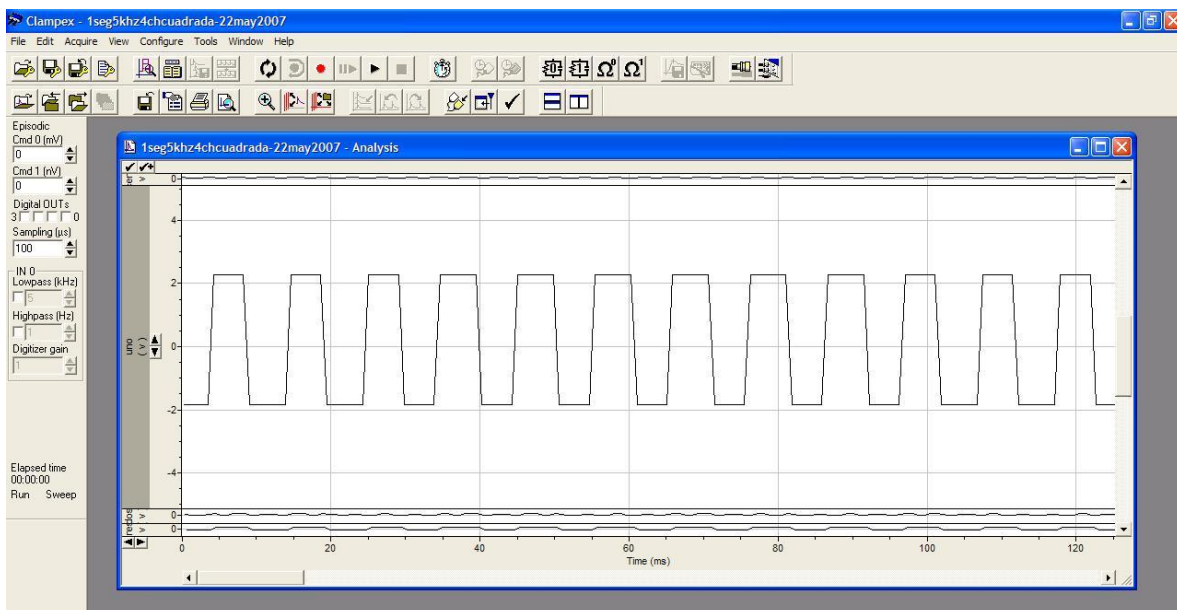


Figura 11 Señal onda cuadrada registrada en Clampex

En la Figura 12 tenemos una señal diente de sierra capturada con una frecuencia de muestreo de 10 khz y un tiempo de 10 seg. El archivo contiene 100 000 muestras. Los valores oscilan de 2 a -2 volts.

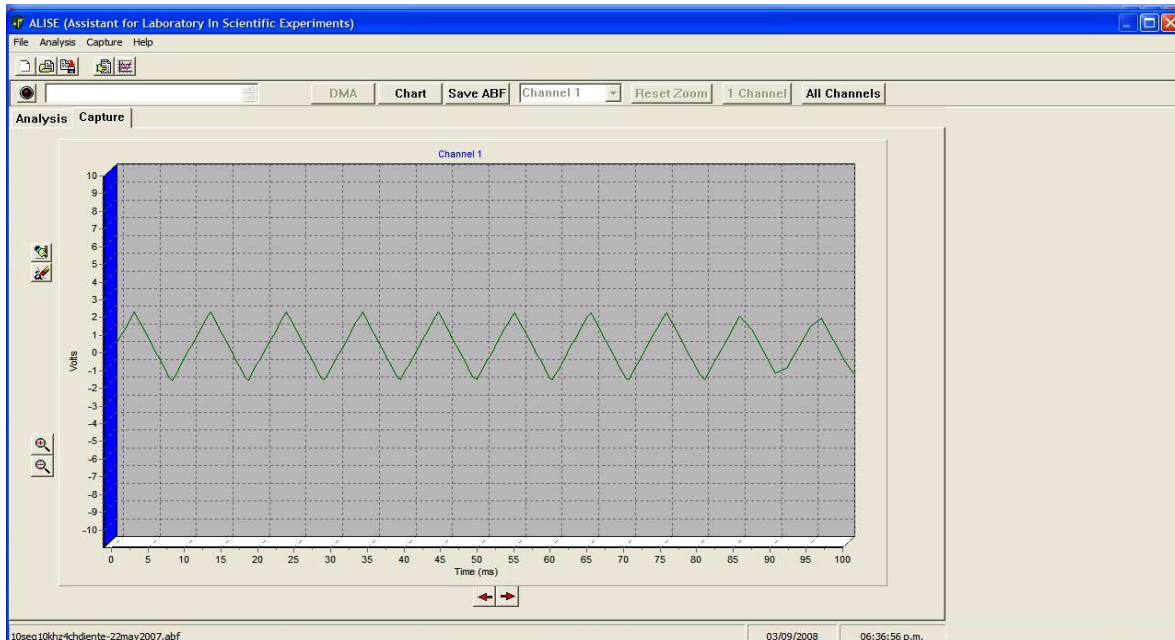


Figura 12 Diente de sierra

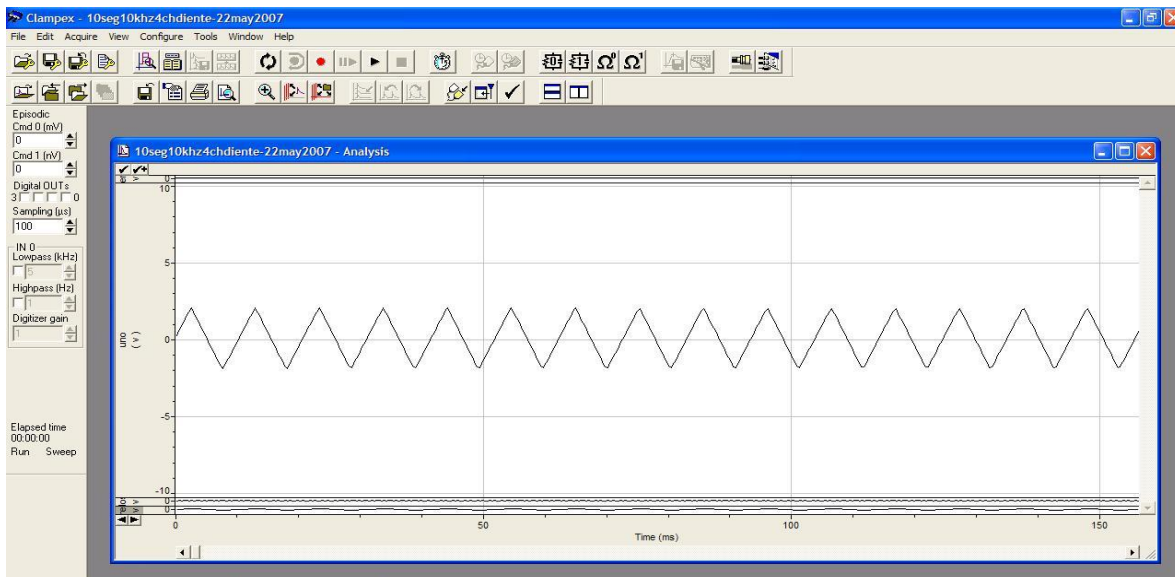


Figura 13 Diente de sierra en Clampex

La Figura 14 muestra una onda senoidal capturada 5 segundos con una frecuencia de muestreo de 10 khz. El total de muestras es de 50 000.

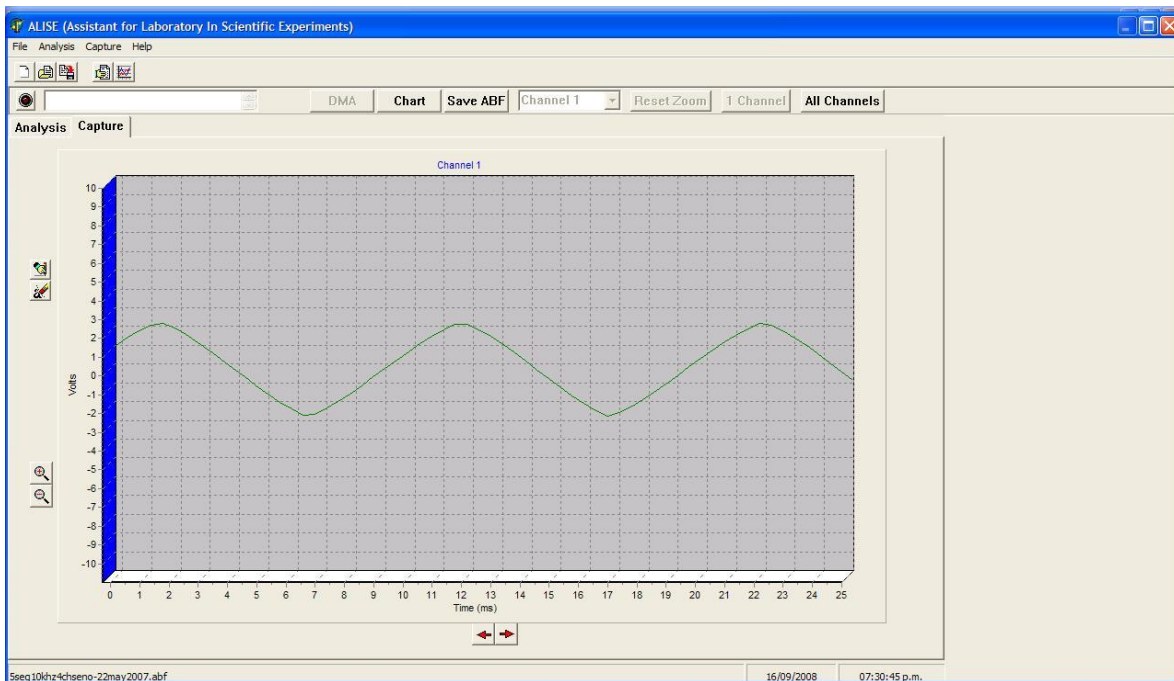


Figura 14 Señal onda senoidal

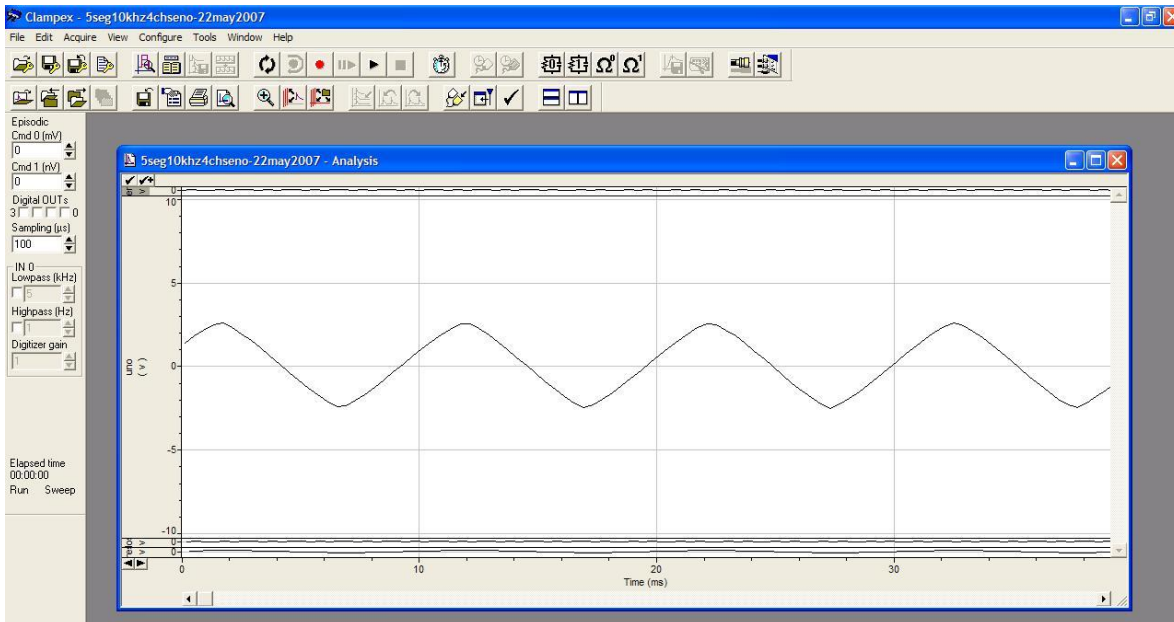


Figura 15 Señal onda senoidal desde Clampex

CONCLUSIONES Y RESULTADOS

Al término de este proyecto podemos concluir que el objetivo general fue alcanzado, es decir se creó una herramienta de software que nos permite hacer uso de la tarjeta digitalizadora LabMaster AD de Scientific Solutions, Inc. Los pasos que se siguieron fueron:

- ▣ Se programó la tarjeta a través del controlador LabPac 32 el cual nos permitió la interfaz con la tarjeta para desarrollar los módulos de captura de señales analógico-digitales.
- ▣ Se incorporó un módulo de captura, que permite la adquisición de señales analógico-digitales de 1 a 4 canales simultáneamente.
- ▣ Se desarrolló otro módulo para que las señales digitalizadas puedan guardarse como archivos ABF (Axon Binary Files)
- ▣ Se terminó con un módulo de despliegue de datos a pantalla.

Nuestra aplicación corre bajo sistemas Windows 95/98, Windows 2000 y también Windows XP; tiene una interfaz amigable la cual cuenta con botones de acceso rápido. Una de sus limitaciones es que el valor para la frecuencia de muestreo es hasta 20 000 (indicar aquí q las frecuencias de las células mas bien los voltajes q se registran son muy bajos, no pasan de los 5 volts)

La funcionalidad de las tarjetas de adquisición digital, que son tarjetas de expansión es la de poder capturar o generar señales analógicas para su manipulación dentro de la computadora. Las tarjetas son comunes en laboratorios dedicados a algún tipo de investigación ya que permiten que las computadoras se conecten con los instrumentos de medición y control para realizar tareas de forma automatizada. Es así como tenemos un proceso de monitoreo y registro en el cual la computadora es capaz de registrar la señal procedente de diversos instrumentos y reportar sus valores en forma de gráficos, que son mas útiles que una sola lectura en la carátula o pantalla de algún instrumento. Los datos también se pueden guardar para su posterior análisis.

Es así como nuestra aplicación cubre este proceso pero por otro lado también tenemos que la computadora puede usarse para controlar los instrumentos de manera inteligente dando lugar a que se tenga como perspectiva la implementación de un método de control de señales

para la tarjeta digitalizadora LabMaster AD de Scientific Solutions, Inc. Las señales de control pueden generarse como resultado del análisis de las señales monitoreadas. Otra prestación que pueden tener las tarjetas de adquisición digital es la inclusión de contadores que establecen la frecuencia a la que se realizan las conversiones. Por otro lado este trabajo también permitirá tanto al usuario como al investigador poder adecuarlo a sus necesidades lo cual no es posible con un programa comercial que en principio es costoso y que no permite modificaciones en cuanto a su programación.

APÉNDICE A

REGISTROS DE CONTROL LABMASTER AD, REFERENCIA RAPIDA

Registro de Control ADC

Algunas características de hardware son controladas por la posición de los registros de E/S. Los datos y la información del estado esta también disponible en los registros de E/S. El software que controla estas características incluye: selección de canal DMA, nivel de interrupción de hardware, y tipo de entrada solo una punta/diferencial. Las características son seleccionadas por la posición de los bits en el Registro de Control ADC. La siguiente tabla resume los bits en el Registro de Control ADC.

REGISTRO DE CONTROL ADC (Escritura, Base + E)			
BIT	FUNCION		
7	Habilitar FIFO		
			0 = Deshabilitar
			1 = Habilitar
6	Una sola punta/Diferencial		
			0 = Una sola punta
			1 = Diferencial
5	Sin uso		
4	Sin uso		
3	Seleccionar bit 1 DMA ADC		
2	Seleccionar bit 0 DMA ADC		
	Bit 3	Bit 2	Asignación
	0	0	No DMA
	0	1	Canal DMA 5
	1	0	Canal DMA 6
	1	1	Canal DMA 7
1	Seleccionar bit 1 Canal IRQ		
0	Seleccionar bit 0 Canal IRQ		
	Bit 1	Bit 0	Asignación
	0	0	Deshabilitar IRQ
	0	1	Canal IRQ 10
	1	0	Canal IRQ 11
	1	1	Canal IRQ 12

Registro de Control ADC

Registro de Estado ADC

La posición de los canales DMA e IRQ está disponible en el Registro de Estado ADC. Este registro también contiene 4 banderas de estado para monitorear el proceso de transferencia de datos al ADC. La posición para el Registro de Estado ADC es como sigue:

REGISTRO DE ESTADO ADC (Lectura, Base + E)			
BIT	FUNCION		
7	FIFO ADC tiene espacio		
		0 = Lleno	
		1 = No lleno	
6	FIFO ADC tiene muestras		
		0 = Vacío	
		1 = No vacío	
5	Desbordamiento FIFO ADC		
		0 = No Desbordamiento	
		1 = Desbordamiento	
4	Estado TC DMA ADC		
		0 = No TC	
		1 = TC	
3	Selecciona bit 1 DMA ADC		
2	Selecciona bit 0 DMA ADC		
	Bit 3	Bit 2	Asignación
	0	0	No DMA
	0	1	DMA Canal 5
	1	0	DMA Canal 6
	1	1	DMA Canal 7
1	Select bit 1 Canal IRQ		
0	Select bit 0 Canal IRQ		
	Bit 1	Bit 0	Asignación
	0	0	Deshabilitar IRQ
	0	1	IRQ Canal 10
	1	0	IRQ Canal 11
	1	1	IRQ Canal 12

Registro de Estado ADC

Arreglo de Canal/Ganancia

Para incrementar el desempeño del sistema ADC, la LabMaster AD tiene una tabla de interpretación que almacena el número del canal físico y pares del descriptor de ganancia. Esta tabla es referenciada como Arreglo de Canal/Ganancia y contiene 256 elementos. Cada entrada de la tabla puede contener cualquier combinación de números de canales físicos (0-15) y descriptor de ganancia (0-3). Esto significa que las localidades secuenciales en el arreglo pueden contener cualquier secuencia (o ninguna secuencia) de canales físicos emparejados con los descriptores de ganancia. La siguiente tabla indica como estos bits están ordenados en el Registro de Datos del Arreglo Canal/Ganancia.

REGISTRO DEL ARREGLO DE DATOS (Lectura/Escritura, Base + A)			
BIT	FUNCION		
7	Seleccionar bit 1 Ganancia		
6	Seleccionar bit 0 Ganancia		
	Bit 7	Bit 6	Asignación
	0	0	Ganancia = 1
	0	1	Ganancia Binaria = 2
			Ganancia Decimal = 10
	1	0	Ganancia Binaria = 4
			Ganancia Decimal = 100
	1	1	Ganancia Binaria = 8
		Ganancia Decimal = 1000	
5	Select bit 1 Banco		
4	Select bit 0 Banco		
	Bit 5	Bit 4	Asignación
	0	0	Usar Banco 0
	0	1	Usar Banco 1
	1	0	Usar Banco 2
	1	1	Usar Banco 3
3	Seleccionar bit 3 Canal		
2	Seleccionar bit 2 Canal		
1	Seleccionar bit 1 Canal		
0	Seleccionar bit 0 Canal		

Registro del Arreglo de Datos

Registro de Control DAC

Características del hardware son controladas por la posición del Registro de Control DAC. El software que controla estas características incluye: Canal de Selección DMA, nivel de interrupción de hardware, repetición del dato en la FIFO, actualización de los datos simultáneamente, y el uso superior de 4 bits de datos de 16 bits como salida de expansión digital. La siguiente tabla resume los bits en el Registro de Control DAC.

REGISTRO DE CONTROL DAC (Escritura, Base + F)			
BIT	FUNCION		
7	Habilitar FIFO		
		0 = Deshabilitar	
		1 = Habilitar	
6	Expansión Digital		
		0 = Expansion canal a conector	
		1 = Analog upper nibbles to connector	
5	Actualizar simultáneamente		
		0 = Actualizar DAC0 después de cargar	
		1 = Esperar para cargar DAC1 antes de actualizar ambos DACs	
4	Reciclar FIFO		
		0 = No reciclar	
		1 = Reciclar FIFO cuando sea llenada	
3	Seleccionar bit 1 DMA ADC		
2	Seleccionar bit 0 DMA ADC		
	Bit 3	Bit 2	Asignación
	0	0	No DMA
	0	1	DMA Canal 5
	1	0	DMA Canal 6
	1	1	DMA Canal 7
1	Seleccionar bit 1 Canal IRQ		
0	Seleccionar bit 0 Canal IRQ		
	Bit 1	Bit 0	Asignación
	0	0	Deshabilitar IRQ
	0	1	IRQ Canal 10
	1	0	IRQ Canal 11
	1	1	IRQ Canal 12

Registro de Control DAC

Registro de Estado DAC

Las posiciones de los canales DMA e IRQ están disponibles en el Registro de Estado DAC. Este registro también contiene 4 banderas de estado para monitorear el proceso de transferencia del dato del DAC. Las posiciones para el Registro de Estado DAC son como sigue:

REGISTRO DE ESTADO DAC (Lectura, Base + E)			
BIT	FUNCION		
7	FIFO DAC tiene espacio		
			0 = Lleno
			1 = No lleno
6	FIFO DAC tiene muestras		
			0 = Vacío
			1 = No vacío
5	Desbordamiento FIFO DAC		
			0 = No Desbordamiento
			1 = Desbordamiento
4	Estado TC DMA DAC		
			0 = No TC
			1 = TC
3	Selecciona bit 1 DMA DAC		
2	Selecciona bit 0 DMA DAC		
	Bit 3	Bit 2	Asignación
	0	0	Deshabilitar DMA
	0	1	DMA Canal 5
	1	0	DMA Canal 6
	1	1	DMA Canal 7
1	Seleccionar bit 1 Canal IRQ		
0	Seleccionar bit 0 Canal IRQ		
	Bit 1	Bit 0	Asignación
	0	0	Deshabilitar IRQ
	0	1	IRQ Canal 10
	1	0	IRQ Canal 11
	1	1	IRQ Canal 12

Registro de Estado DAC

Controlador 8237 DMA

El controlador 8237 DMA es usado para monitorear y controlar el proceso DMA: Los sistemas EISA/ISA asignan múltiples localidades de E/S para acceder a los registros en el Controlador 8237 DMA. Los registros clave que son usados en la inicialización del proceso DMA son: Byte Puntero Limpio, Modo, Mascara, Estado, Transferencia del contador, Dirección de desplazamiento de memoria, y Pagina.

Registro Mascara

Antes de que cualquier programación de un canal DMA pueda ocurrir, deshabilite al 8237 de responder a la solicitud DMA en el canal siendo inicializado. Esto es realizado por 3 bits en el Registro Mascara del 8237 en la localidad D4 hexadecimal. La asignación de los bits para el registro es como sigue.

8237 REGISTRO MASCARA (Escritura, D4)			
BIT	ASIGNACION		
7	Siempre igual a 0		
6	Siempre igual a 0		
5	Siempre igual a 0		
4	Siempre igual a 0		
3	Siempre igual a 0		
2	Solicita Habilitar/Deshabilitar DMA		
	0 = Enable selected channel		
	1 = Disable selected channel		
1	Seleccionar bit 1 Canal DMA		
0	Seleccionar bit 0 Canal DMA		
	Bit 1	Bit 0	Asignación
	0	0	NA
	0	1	DMA Canal 5
	1	0	DMA Canal 6
	1	1	DMA Canal 7

8237 Registro Mascara

Registro Modo

El siguiente paso es la programación del Registro Modo DMA en la localidad de E/S D6 hexadecimal. La siguiente tabla describe la asignación de los bits modelo del Registro Modo 8237.

8237 REGISTRO MODO (Escritura, D6)			
BIT	ASIGNACION		
7	Siempre igual a 0		
6	Demand Mode Select		
	0 = Demand Mode		
	1 = Single Mode		
5	Selecciona incrementar/decrementar dirección		
	0 = Incrementar		
	1 = Decrementar		
4	DMA Continuo		
	0 = Deshabilitar		
	1 = Habilitar		
3	Select bit 1 Transferir Tipo		
2	Select bit 0 Transferir Tipo		
	Bit 3	Bit 2	Asignación
	0	0	NA
	0	1	Transferir ADC
	1	0	Transferir DAC
	1	1	NA
1	Seleccionar bit 1 Canal DMA		
0	Seleccionar bit 0 Canal DMA		
	Bit 1	Bit 0	Asignación
	0	0	NA
	0	1	DMA Canal 5
	1	0	DMA Canal 6
	1	1	DMA Canal 7

8237 Registro Modo

Registro De Control IRQ

El Registro de Control IRQ determina cual línea IRQ (10, 11 o 12) es usada con uno de los contadores de tiempo para generar interrupciones. La siguiente tabla indica el bit modelo para seleccionar la línea IRQ.

BIT	ASIGNACION		
7	Seleccionar bit 1 Canal IRQ		
6	Seleccionar bit 0 Canal IRQ		
	BIT 7	BIT 6	Asignación
	0	0	Deshabilitar IRQ
	0	1	IRQ Canal 10
	1	0	IRQ Canal 11
	1	1	IRQ Canal 12
5	Sin uso		
4	Interrupción habilitada para el contador OUT 4		
3	Interrupción habilitada para el contador OUT 3		
2	Interrupción habilitada para el contador OUT 2		
1	Interrupción habilitada para el contador OUT 1		
0	Sin uso		

Registro De Estado IRQ

Determina cual señal del contador causo la interrupción por medio de la verificación de los bits 1-4 (ver tabla siguiente).

BIT	ASIGNACION		
7	Seleccionar bit 1 Canal IRQ		
6	Seleccionar bit 0 Canal IRQ		
	BIT 7	BIT 6	Asignación
	0	0	Deshabilitar IRQ
	0	1	IRQ Canal 10
	1	0	IRQ Canal 11
	1	1	IRQ Canal 12
5	Sin uso		
4	Interrupción ocurriendo para el contador OUT 4		
3	Interrupción ocurriendo para el contador OUT 3		
2	Interrupción ocurriendo para el contador OUT 2		
1	Interrupción ocurriendo para el contador OUT 1		
0	Hardware Interrupt Mode		
	0 = rising edge triggered (ISA)		
	1 = active low level triggered (EISA)		

APENDICE B

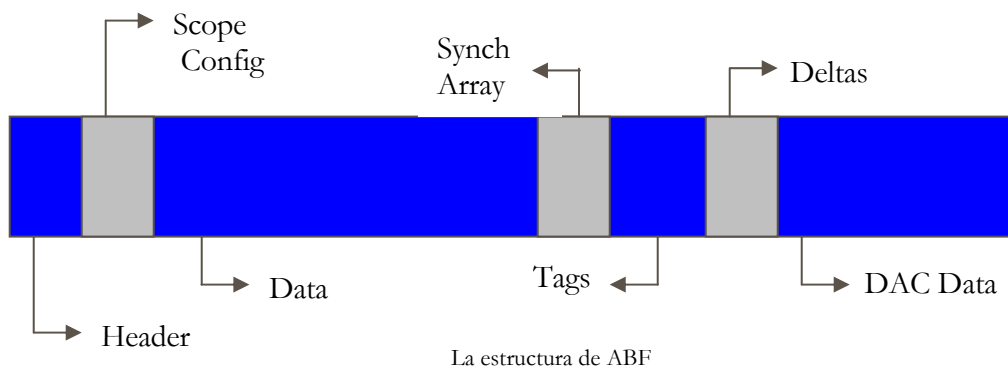
ARCHIVOS ABF (AXON BINARY FILES)

Estructura del Archivo ABF

El formato Archivos Binarios de Axon (ABF) fue creado para el almacenamiento de los datos binarios experimentales. Fue originado con el conjunto de programas de adquisición de datos y análisis de Axon pCLAMP pero también es soportado por otros softwares científicos de Axon.

El formato ABF es diseñado para adaptar todas las características de los programas de adquisición de datos de Axon. Por consiguiente, no solo los datos sino la información del protocolo pueden ser extraídos del ABF.

Estos archivos pueden ser creados y leídos en computadoras con sistemas Windows y Macintosh. Para un óptimo desempeño de la adquisición los datos binarios son escritos en la convención de la adquisición del orden de bytes de la computadora y las rutinas de lectura automáticamente implementan el intercambio de bytes si es requerido.



El archivo binario de Axon (ABF) está formado por las secciones que se muestran arriba. La cabecera y la sección de datos aparecen en el orden mostrado. Las otras secciones pueden aparecer en cualquier orden ya que son apuntados por parámetros en la cabecera. Todas las secciones son almacenadas en bloques de 512 bytes cada una. La localidad de inicio de una sección es dada como un número de bloque. El número de bloque 0 representa el principio del archivo. Si el número de bloque que apunta a una sección (otra además de la cabecera) es cero, la sección correspondiente no es escrita.

La sección ABF Header

La cabecera (header) es el primer bloque de datos en el inicio de un archivo de datos ABF. La cabecera contiene los parámetros que describen el estímulo, la adquisición y la jerarquía de los datos. Describe el contenido del archivo de datos y contiene entradas para describir la configuración, naturalmente cuando el archivo de datos fue adquirido.

La sección ABF Scope Config

Si presenta, la sección ABF Scope Config contendrá una o más estructuras ABFScopeConfig describiendo los atributos de la ventana scope usada para el despliegue de datos durante la adquisición.

La sección ABF Data

Las muestras de datos adquiridas son almacenadas como enteros de dos bytes binarios multiplexados. Una versión especial de punto flotante de cuatro bytes es usada por algún programa de análisis para el almacenamiento de los resultados del análisis. Si el archivo de datos contiene múltiples barridos o segmentos de datos, estos son almacenados de fin a fin sin un intervalo. Es decir, ningún parámetro se guarda entre los barridos de los datos; todos los parámetros son almacenados en el header o en la sección especializada que se describe abajo. Solo hay una sección de datos.

La sección ABF Synch

El arreglo ABF Synch es importante ya que almacena el tiempo de inicio y la longitud de cada porción del dato si el dato no es parte de una adquisición continua gap-free. La sección data podría contener igual longitud o longitud variable de barridos del dato. El Arreglo Synch contiene un registro para indicar el tiempo de inicio y la longitud de cada barrido o Evento en el archivo de datos. Las rutinas de lectura ABF automáticamente decodifican el Arreglo Synch cuando proporcionan información acerca del dato.

Un arreglo Synch es creado y usado en los siguientes modos de adquisición: modo de longitud variable, modo de longitud fija y modo osciloscopio de alta velocidad. Los modos de adquisición gap-free y estimulación episódica no siempre usan un arreglo Synch.

La sección ABF Tag

Durante una adquisición, algunos programas permiten al usuario etiquetar puntos de interés en el flujo de entrada del dato. Estas etiquetas son guardadas en la sección Tag. Cada etiqueta consiste de una impresión de tiempo, un comentario de texto, y una etiqueta tipo identificador. Si la etiqueta es una etiqueta de voz, el dato es retenido en una estructura ABFVoiceTagInfo.

La sección ABF Deltas

Cuando los parámetros de adquisición son cambiados durante una adquisición, los cambios son rastreados e introducidos en la sección ABF Deltas. Cada entrada es tiempo impreso en las unidades del fSynchTimeUnit, así que el valor del parámetro puede ser determinado en cualquier punto durante la adquisición.

La sección DAC Data

En algunos experimentos, en lugar de que la onda de estímulo de la salida analógica sea descrita paramétricamente en una tabla es descrita en base a muestra por muestra de un archivo. En experimentos donde los archivos son usados para describir el estímulo de la salida analógica, una copia del archivo de estimulación es adjuntado en la sección DAC Data para asegurar que la onda de estímulo este siempre disponible durante el análisis. Las rutinas ABF permiten la lectura del DAC data.

MODOS DE ADQUISICIÓN

1. *Gap-Free*

Los archivos ABF Gap-free contienen un solo barrido de hasta 4 GB de datos multiplexados. Es usado un intervalo de muestreo uniforme. Hasta la fecha, no hay ondas de estímulos asociadas con datos gap-free.

El modo Gap-free normalmente es usado para adquisición continua de datos en los cuales hay una actividad bastante uniforme en el tiempo.

2. *Longitud Variable*

Tanto en el modo de longitud variable como en el modo de longitud fija la adquisición de datos es iniciada en segmentos siempre que un evento cruzando el umbral es detectado. En el modo de longitud variable, el predisparo y las porciones de arrastre debajo del umbral también son adquiridos. La longitud del segmento de datos es determinado por la naturaleza del dato, siendo automáticamente extendido de acuerdo a la cantidad de tiempo que el dato excede el umbral. Si la porción de predisparo del siguiente evento apareciera traslapada de la porción de arrastre del evento actual, el segmento actual es extendido. No hay almacenamiento de datos traslapados. El tiempo preciso de inicio y longitud de cada segmento es almacenado en el arreglo Synch. Este modo de adquisición es usado para el registro continuo de “ráfagas” de datos en los cuales hay ráfagas de actividad separadas por largos períodos quietos.

3. *Longitud Fija*

En el modo de longitud fija, una porción de predisparo debajo del umbral es adquirida. A diferencia del modo de longitud variable, la longitud de cada segmento de datos es una constante preespecificada para todos los segmentos. Por esta razón, los segmentos son usualmente referenciados como barridos. En este modo, cada disparo que cruza el umbral es un barrido, por lo tanto el modo de longitud fija es también referenciado como modo osciloscopio de pérdida libre. Si un segundo evento ocurre antes de que el barrido actual finalizara, un segundo barrido es adquirido disparando el segundo evento. A esto se le llama traslape. En este caso, barridos consecutivos en el archivo de datos contiene datos redundantes.

El tiempo preciso de inicio y la longitud de cada barrido es almacenado en el arreglo Synch. Aunque la longitud de cada barrido es redundante en este modo, es almacenado para simplificar la lectura y escritura del arreglo Synch. De igual manera, el almacenamiento de datos redundantes durante el traslape no es estrictamente necesario, pero simplifica el análisis y despliegue por cada barrido que es regresado como un barrido de longitud fija con un tiempo conocido y una constante de disparo. Ya que los disparos no son perdidos, este modo de adquisición es ideal para el análisis estadístico de constantes con eventos tales como potenciales de acción.

4. *Osciloscopio De Alta Velocidad*

Como en el modo de longitud fija, en el modo de osciloscopio una porción de predisparo debajo del umbral es adquirida. Diferente al modo de longitud fija, en el modo de osciloscopio de alta velocidad cada disparo que cruza el umbral no es un barrido. El énfasis está en permitir al digitalizador ser usado para la frecuencia de muestreo más alta posible. Al igual que un osciloscopio real de alta velocidad, hay un “tiempo muerto” al final de cada barrido durante el cual el despliegue es actualizado y el circuito de disparo es rearmado. Los cruces del umbral que llegan durante este tiempo muerto son simplemente ignorados. Similarmente, el segundo y subsecuen-

tes cruces del umbral durante un barrido no comienzan un nuevo barrido. Así no hay almacenamiento de datos traslapados.

Aunque las condiciones de adquisición son diferentes para el modo de longitud fija y el modo osciloscopio de alta velocidad, en la práctica los archivos de datos son idénticos y los programas de análisis pueden tratarlos idénticamente. La única advertencia es que debido al almacenamiento de datos traslapados que es posible en el modo de longitud fija, el tiempo de salida de un barrido podría ocurrir antes del fin del barrido anterior.

5. Estimulación Episódica

En este modo, un número de barridos de igual longitud (también conocidos como episodios) son adquiridos. Al conjunto de barridos paraméricamente relacionados se les llama una corrida. Las corridas pueden repetirse un número específico de veces para formar un experimento. Si las corridas se repiten, los barridos correspondientes en cada corrida son automáticamente promediados y el experimento contiene solamente el promedio. El experimento es almacenado en un archivo. Solamente un experimento puede ser almacenado en un archivo ABF.

Dentro de cada barrido puede generarse una onda de estímulo compleja consistente de hasta diez períodos. Se genera una muestra de salida por cada conversión A/D. Note que esto se refiere a las conversiones multiplexadas A/D. Por ejemplo, si hay tres canales multiplexados A/D y los barridos contienen 500 muestras por cada canal, el conversor D/A genera 1500 muestras. De esta manera hay una muestra de la onda de estímulo correspondiente a cada muestra en el multiplexor de la onda A/D.

Las amplitudes y duraciones de los pasos, rampas y pulsos digitales (es decir TTL) comprendiendo los períodos pueden incrementarse automáticamente de barrido a barrido. En lugar de crear ondas basadas en períodos, el usuario puede elegir leer la onda de estímulo de un archivo. Cualquiera que sea el método usado, un arreglo completo que contiene la onda de estímulo es proporcionado por las rutinas de ABF cuando el programador de las aplicaciones requiere el arreglo de la onda de estímulo asociado con cualquier barrido.

El intervalo de muestreo puede cambiar una vez dentro del barrido a un segundo intervalo. Si el programador de las aplicaciones pide el arreglo X (i.e. tiempo) para el barrido, las rutinas de lectura ABF proporcionan un arreglo que contiene los intervalos de tiempo propiamente espaciados tomando en cuenta el cambio en el intervalo de muestreo. Note que independiente de si el programa de adquisición especifica el intervalo de muestreo sobre uno por-canal o una base multiplexada, el valor almacenado en el archivo ABF es el valor multiplexado. En el ejemplo de los tres canales usado previamente, si cada canal era muestreado a 21 μ s, el intervalo multiplexado de muestreo sería 7 μ s.

BIBLIOGRAFIA

- [1] Mora Francisco, *Diccionario de Neurociencias*, termino: Fijación de Voltaje, Alianza Editorial (1994).
- [2] Tocci Ronald J., *Sistemas Digitales: Principios y Aplicaciones*, Prentice-Hall, México (1993).
- [3] Soliman, Samir S. y Mandyan D. Srinath, Señales y Sistemas Continuos y Discretos, Prentice-Hall Madrid (1999).
- [4] Axon Instruments, Inc., The *Axon Guide for Electrophysiologic and Biophysics Laboratory Techniques*, USA
- [5] LabMaster Advanced Design Handbook, Scientific Solutions
- [6] Scientific Solutions, *LabPac32 Function Specification*, (2001)
- [7] Herbert Schildt; *Turbo C/C++ Manual de Referencia*; McGraw-Hill (1992)
- [8] Roger S. Pressman; *Ingeniería del Software, Un enfoque práctico 4ª. Edición*; McGraw-Hill (1998).
- [9] Gilb Tom; *Principles of software engineering management*; Workingham, Englandi Addison-Wesley (1998).
- [10] Victor Moral; *Delphi 4*; Prentice Hall Iberia, Madrid (1999).
- [10] Luis Joyanes Aguilar; *Programación en Turbo/Borland Pascal 7 3ª. Edición*; Osborne McGraw-Hill (1997).
- [10] Dirk Louis; *Gran Libro Delphi 5*; Marcombo (2000);