



Universidad Autónoma Metropolitana
Unidad Iztapalapa

División de Ciencias Básicas e Ingeniería

Departamento de Ingeniería Eléctrica

Licenciatura de Ingeniería Electrónica

**ANALIZADOR DE ESTADOS LÓGICOS
BÁSICO AL-UAMI-2004.VI**

Desarrollado por:

López López Mario Alberto 200215576
Quintero Zamudio José 98319116

Asesor: M. en C. Agustín Suárez Fernández

México D.F. a Mayo del 2004



Universidad Autónoma Metropolitana
Unidad I, Atoyac

División de Ciencias Básicas e Ingeniería

Departamento de Ingeniería Eléctrica

Licenciatura de Ingeniería Electrónica

**ANALIZADOR DE ESTADOS LÓGICOS
BÁSICO AL-UAMI-2004.VI**

Desarrollado por:

López López Mario Alberto 200215576

Quintero Zamudio José 98319116

Asesor: M. en C. Agustín Suárez Fernández

México D. F. a Mayo de 2004

ÍNDICE

Objetivos.	2
• Objetivo General.	2
• Objetivos Particulares.	2
INTRODUCCIÓN AL ANALIZADOR LÓGICO.	3
DESARROLLO.	6
• Diseño de un Analizador Lógico Básico.	6
• La Tarjeta PCI-DIO-32HS de National Instruments.	7
• Elección del Software para la programación.	8
• ¿Por qué LabVIEW?	8
• Hardware.	12
• Entrada/Salida.	13
METODOLOGÍA.	14
COMPARACIÓN DE UN ANALIZADOR LÓGICO COMERCIAL CON EL QUE SE DESARROLLARÁ	17
RESULTADOS	20
• Dilucidación del Diagrama de Programación del AL-UAMI-2004.	21
• Descripción del Panel Frontal.	25
CONCLUSIONES	36
APÉNDICE A	37
BIBLIOGRAFÍA.	39

OBJETIVOS.

Objetivo General:

- Aprender los conceptos básicos del lenguaje de programación LabVIEW®.
- Aprender el manejo del analizador de estados lógicos Prisma de Tektronix®.
- Estudiar la programación y manejo del analizador lógico básico desarrollado.
- Determinar las posibilidades de incorporar características de desensamble para los microprocesadores: 80188®, 68000® y para el microcontrolador 68HC11®.

Objetivos Particulares:

- Conocer a fondo el funcionamiento de un analizador lógico.
- Conocer las partes funcionales fundamentales de estos.
- Ampliar un diseño viable para un analizador lógico utilizando una tarjeta de adquisición y como herramienta de programación LabVIEW®.

INTRODUCCIÓN AL ANALIZADOR LÓGICO.

Una parte esencial de los circuitos digitales es la operación simultánea de muchas trayectorias de señales. Cuando un circuito digital presenta fallas, existen serias dificultades para la depuración. Las herramientas tradicionales de diagnóstico, tales como el osciloscopio, pueden monitorear hasta 4 canales (que no alcanza ni para medio bus de direcciones). Por ejemplo, si necesitamos ver simultáneamente las entradas y las salidas de un contador de 16 bits, no es posible determinar un error en el tiempo con el osciloscopio de 2 canales. Las puntas lógicas son otra herramienta pero que solamente pueden obtener el estado de una señal y tampoco es posible analizar la variación de las señales en el tiempo.

El analizador lógico es la herramienta adecuada para esos problemas ya se puede ver al analizador como un osciloscopio con muchos canales, excepto por el hecho de que el estado alto-bajo de una señal se puede ver mejor que una señal analógica continua. El analizador lógico sirve para una depuración compleja de circuitos digitales. No solamente puede mostrar y almacenar los estados de un número grande de señales digitales, sino que también puede hacer un análisis complejo sobre las señales, determinar sus tiempos y sus relaciones entre ellas. De hecho, existen dos tipos de analizadores lógicos: de estados y de tiempos.



Figura 1. Analizador Lógico Tektronix® 3002 Prism Series

Un análisis de mercado mostró una clara necesidad de un analizador que cubriera una velocidad asíncrona de 400MHz y con una resolución de 2.5ns, 100MHz para todos los canales y 500MHz síncronos para una frecuencia compleja con un rango de paso de 20ns para 8, 16 y 32 bits Pendientes de la velocidad del procesador. Otra especificación es una captura de glitch de 5ns, el uso de una tarjeta de memoria, un amplio espacio para interfaces. Para desarrollar el analizador con todos los requerimientos, es necesario tomar en cuenta los costos de desarrollo, así como los de los componentes que se requieren.

Todas las líneas contenidas en la sección de Adquisición de Datos son entradas, el analizador lógico nunca envía una señal al circuito que está conectado. Dicha sección también contiene una entrada de reloj, la cual puede usarse para el muestreo a tiempo del reloj del sistema que se está probando, la entrada de tierra y las líneas de entrada de datos.

Las señales entran a comparadores de voltaje en la sección de lógica de Disparo donde periódicamente se hace un muestreo determinando su nivel lógico alto y bajo. Algunos analizadores cuentan con un voltaje de umbral variable, el cual define los niveles altos y bajos dependiendo de la aplicación particular.

Después de la comparación de voltajes, la señal de entrada al analizador lógico es tomada y almacenada usando la técnica de muestreo.

Con un reloj muy rápido se puede observar la operación de un circuito en detalle con un periodo de tiempo muy corto. Si tiene un reloj lento se puede observar la operación de un circuito en un tiempo mayor, pero con menos precisión. Si lo que se está depurando es un circuito digital de alta velocidad, como un microprocesador, se deberá usar una base de tiempo rápida.

La entrada de reloj externo se usa cuando se desea hacer un muestreo a una razón de tiempo base del circuito externo que proporciona la sincronía. Un buen ejemplo de esto es cuando se coloca al analizador lógico con las líneas de datos y direcciones de un microprocesador para seguir la ejecución de un programa.

Después de empezar la captura en los latches de entrada las señales de entrada fluyen a dos lugares: a la RAM de alta velocidad y a la Unidad de Lógica de Disparo. Esta última toma la decisión del momento en que empiece a ser llenada dicha memoria.

La mayor parte de los sistemas digitales se interesan en una operación específica de un circuito. La acción de la Unidad de Lógica de disparo permite obtener solamente los datos que nos interesan.

Para cada señal que está siendo monitoreada, se puede especificar un patrón de disparo alto, bajo o "no importa". Cuando el analizador lógico esta habilitado continuamente hará un muestreo de las líneas de datos de entrada hasta que el patrón de disparo es reconocido. En este punto, el muestreo se detiene o continúa hasta que se obtienen un número determinado de muestras específicas. Esto permite que se pueda observar el estado de la señal antes, o después del punto de disparo.

La memoria de alta velocidad almacena el valor de los canales que son monitoreados.

Hay que tomar en cuenta que el analizador lógico, a diferencia de un osciloscopio, no trabaja a "tiempo real". Un osciloscopio puede continuamente mostrar el voltaje que se prueba. El analizador lógico por otro lado, almacena la señal de entrada hasta reconocer un patrón de disparo la señal de datos es mostrada después de que ocurrió. A continuación se presenta una breve introducción del diseño de una tarjeta de adquisición sencilla.

DESARROLLO

DISEÑO DE UN ANALIZADOR LÓGICO BASICO.

En la actualidad es importante, en el desarrollo de cualquier sistema electrónico o de computo, el tratar de enfocar los esfuerzos en la meta que uno se ha fijado en este caso el desarrollo de un analizador de estados lógicos digital basado en PC cuando uno se tarda y consume mucho del tiempo de desarrollo en otras cosas como el diseño de la tarjeta de adquisición, se llega a perder esta meta y se entra en complicaciones innecesarias, ya que se cuenta en el mercado con una variedad de tarjetas de adquisición digital que hacen que uno pueda dedicarle tiempo a la tarea o meta que se tiene fijado sin preocuparse por otras cosas. Esto es un tema importante que como próximos ingenieros tendremos que considerar ya que el diseño electrónico no solo consiste en diseñar por completo un sistema electrónico, sino ver que opciones se nos presentan en el mercado y considerar alguna de ellas que cumpla con todas nuestras demandas y necesidades.

Por tal motivo, presentamos el uso de una tarjeta de propósito general para el desarrollo de un analizador de estados lógicos digital por PC, bajo Windows®.

LA TARJETA PCI-DIO-32HS® DE NATIONAL INSTRUMENTS®.

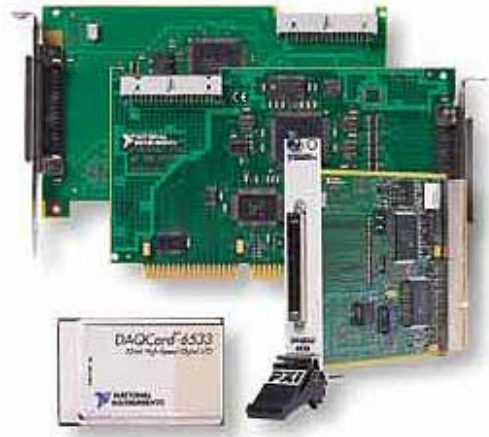


Figura 2. Tarjeta de Adquisición PCI-DIO-32HS®

La 6533 ® (DIO-32HS) es una interfase paralela de E/S de alta velocidad de 32 bits para computadoras PCI®, PXI®, Compac® PCI®, PCMCIA®, e ISA®. El 6533 incorpora ASIC DAQ-DIQ® de National Instruments®, una interfase de 32 bits de propósito general, específicamente diseñada para dar un alto rendimiento en la medición de sistemas digitales. La PCI-DIO-32HS® es una tarjeta completamente sin switches o jumpers, es decir se configura automáticamente al conectarla a la PC.

El chip DAQ-DIO® provee de dos canales independientes de salidas y entradas digitales, generación de modelos y comunicaciones. Cada canal ofrece las siguientes funciones:

- Selección de varios anchos de palabra (8, 16 o 32 bits).
- FIFO con capacidad para 16 muestras.
- Contadores de 16 y 32 bits para el tiempo base y generación de intervalos, con una resolución máxima de 50 ns.
- Un controlador de comunicaciones que implementa seis flexibles protocolos.

- Detección por disparo y detección digital de modelos.
- 24 mA de salida estándar.
- Histéresis y terminación de líneas basadas en diodos para entradas.

Con la DIO-32HS®, uno puede usar su computadora como un probador de E/S, un analizador de estados lógicos digital, o un controlador de sistema para pruebas de laboratorios, producción, y monitoreo y control de procesos industriales.

La PCI-DIO-32HS® contiene la interfase llamada MITE PCI® desarrollada por National Instruments®. La interfase MITE PCI 6533® desarrolla mediciones sobre uno o varios grupos de canales, hace generación de señales y transferencia de datos a altas velocidades gracias a un gran rango de protocolos de comunicación a velocidades cerca de los 76 Mbytes/s (PCI). Uno puede operar las 32 líneas individualmente en una entrada simple, o como puertos de 8, 16, o 32 bits para la generación de modelos y comunicaciones.

ELECCIÓN DE SOFTWARE PARA LA PROGRAMACIÓN.

Existen muchas opciones de donde escoger cuando se programa una tarjeta de adquisición como esta. Uno tiene a su disposición vía Internet paquetes como LabVIEW®, LabWindows/CVI®, Components Works®, o NI-DAQ®.

¿POR QUÉ LabVIEW®?

Después de asistir a algunos seminarios teóricos y prácticos nos dimos cuenta de que LabVIEW® es un revolucionario ambiente de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de mediciones y presentaciones de datos. LabVIEW® le da la flexibilidad de un poderoso ambiente de programación sin la complejidad de los ambientes tradicionales como se puede apreciar en la Figura 3.

Lenguaje Desarrollado para Medición, Control y Automatización

A diferencia de los lenguajes de propósito general, LabVIEW® provee funcionalidad específica para que se pueda acelerar el desarrollo de aplicaciones de medición, control y automatización de bloques.

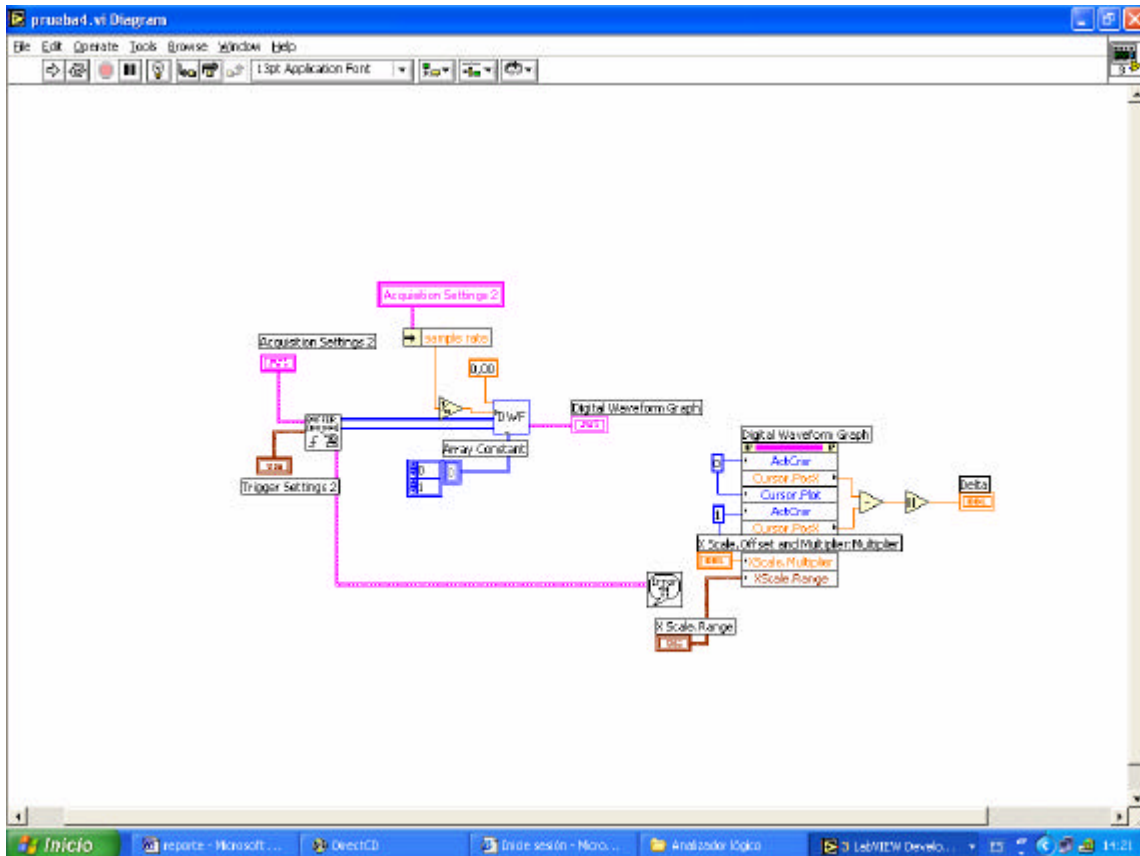


Figura 3. Ambiente de Programación Gráfico.

Fácil Integración con Miles de Instrumentos y Dispositivos de Medición

LabVIEW® se puede conectar de manera transparente y virtualmente todo tipo de hardware incluyendo instrumentos de escritorio, tarjetas insertables, controladores de movimiento y controladores lógicos programables (PLCs).

Compilado para Optimizar el Desempeño del Sistema

En muchas aplicaciones, la velocidad de ejecución es vital. Con un compilador incluido que genera código optimizado, sus aplicaciones en LabVIEW® entregan velocidades de ejecución comparables con programas C compilados. Con LabVIEW® puede desarrollar sistemas que cumplan con sus requerimientos de desempeño a través de las plataformas incluyendo Windows®, Macintosh®, UNIX® o sistemas de tiempo real.

Beneficios

LabVIEW® está altamente integrado con el hardware de medida, con lo que se puede configurar y usar rápidamente cualquier dispositivo de medida que se tenga. Con LabVIEW® puede conectarse a miles de instrumentos de medida para construir sistemas de medida completos, incluyendo desde cualquier tipo de instrumento autónomo hasta dispositivos de adquisición de datos, controladores de movimiento y sistemas de adquisición de imagen. Además LabVIEW® trabaja con más de 1000 librerías de instrumentos de cientos de fabricantes, y muchos fabricantes de dispositivos de medida incluyen también herramientas de LabVIEW® con sus productos.

NI-DAQ®.

NI-DAQ® es la única paquetería que está incluido sin ningún cargo con el hardware de National Instruments®. Este posee una extensa librería de funciones que uno puede llamar desde un ambiente de programación. Estas funciones incluyen rutinas para entrada analógica (Conversión A/D), adquisición de datos al buffer (conversión A/D de alta velocidad), salida analógica (conversión D/A), generación de ondas (conversión D/A síncrona), entrada y salida digital, operaciones con contadores y temporizadores, SCXI (Tecnología desarrollada por National Instruments®), RTSI® (Real Time System Integration), calibración, comunicación, y adquisición de datos a memoria extendida.

NI-DAQ® posee funciones de E/S de alto nivel para mayor facilidad de programación y funciones de E/S de bajo nivel para obtener mejor ejecución y flexibilidad. Ejemplos de funciones de alto nivel escribir datos al disco o adquirir un cierto número de

datos. Un ejemplo de una función de bajo nivel es escribir directamente en los registros de la tarjeta de una adquisición.

NI-DAQ® también direcciona internamente muchas de las tareas entre la computadora y el hardware de adquisición como la programación de interrupciones y controladores de DMA.

NI-DAQ® mantiene una constante interfase de software de tal manera que uno puede cambiar de plataforma con solo hacer unos pequeños cambios en el código. Usando lenguajes de programación convencionales o el software de National Instruments®, la aplicación siempre usará el software manejador NI-DAQ®.

Librerías de NI-DAQ®.

Las librerías de funciones para Windows de NI-DAQ® son DLL'S, lo que significa que las rutinas de NI-DAQ® no son ligadas dentro de los ejecutables de la aplicación. Solo la información acerca de las rutinas es almacenada en los ejecutables.

Las librerías importadas contienen información acerca de las funciones de la DLL exportadas. Estas indican la presencia y localización de las rutinas DLL. Dependiendo de las herramientas de desarrollo que se utilicen, uno puede mandar información a las rutinas DLL a través de librerías importadas o de declaración de funciones.

Usar prototipos de funciones es una buena práctica en la programación. Este es el motivo por el cual NI-DAQ® está empaquetado con archivos de funciones prototipo para diferentes herramientas de desarrollo para Windows®.

HARDWARE

DAQ-DIO® ASIC.

El DAQ-DIO® ASIC es una interfase de propósito general de 32 bits de E/S que tiene características avanzadas a un costo muy bajo. Entre estas están:

- Configuración de las líneas de E/S de forma individual.
- Salidas OR alambradas.
- Detección de Modelos.
- Disparo a la detección de un cierto patrón.
- Protocolos de comunicaciones de alta velocidad.
- Buffers (FIFO) de 16 palabras.

La Figura 4 muestra un diagrama a bloques de la PCI-DIO-32HS®.

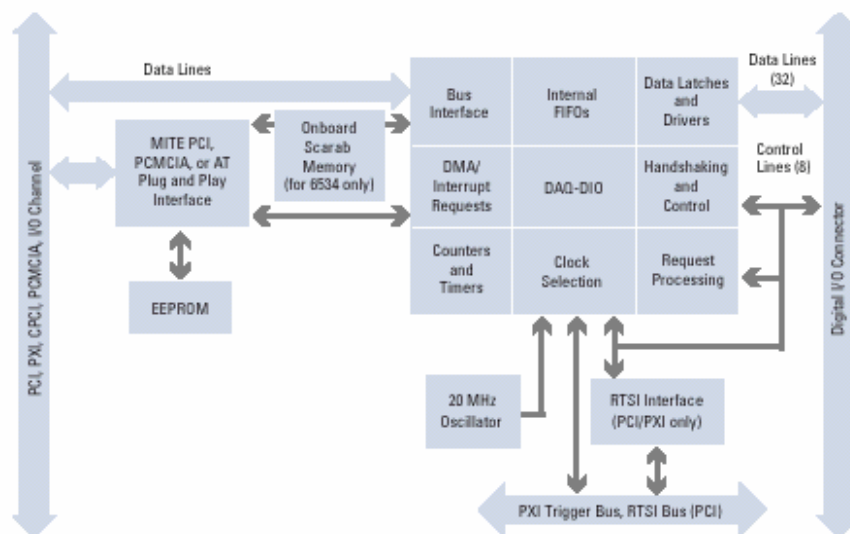


Figura 4. Diagrama de bloques de la tarjeta de adquisición

ENTRADA/SALIDA.

La DIO-32HS® puede realizar E/S sin reconocimiento, que es básicamente E/S digital que no utiliza comunicación o sincronización con algún equipo. Uno puede escribir o leer datos directamente hacia los cuatro diferentes puertos E/S digitales de la tarjeta. Los puertos contienen ocho líneas cada uno y están etiquetados como A, B, C, y D. Uno puede configurar cada línea para funcionar como entrada o salida.

Cuando uno realiza una E/S sin reconocimiento, la tarjeta no requiere ser configurada para control de comunicaciones o señales de temporización. Sin embargo, se pueden usar las líneas de REQ (reconocimiento) y STOPTRIG (paro del disparo) como entradas extras, y las líneas ACK y PCLK como líneas de salida extras.

La tarjeta también puede realizar E/S con reconocimiento (sincrónamente), por medio del cual se regula las operaciones ya sea temporizándolas o utilizando alguno de los protocolos de comunicaciones que ofrece.

La DIO-32HS® tiene dos controladores de comunicaciones y puede realizar arriba de dos operaciones de reconocimiento simultáneamente. Uno selecciona el ancho de cada transferencia separando los puertos de E/S digital en dos grupos para los dos diferentes controladores.

METODOLOGÍA

Como se pudo observar en las características y funcionamiento de la tarjeta PCI-DIO-32HS® es la herramienta perfecta para desarrollar una aplicación como esta, ya que dicha tarjeta posee todas las capacidades que necesita un analizador de estados lógicos para realizar las funciones principales que debe ofrecer, como son:

- a) Adquisición de datos a altas velocidades (cerca de los 76 Mbytes/s)
- b) División de sus 32 canales en puertos de 16 y 8 bits.
- c) Se pueden usar algunos pines como entrada.
- d) Monitoreo de los canales de E/S para buscar algún patrón o suceso en el sistema que se analice.
- e) Disparo para comenzar la adquisición de datos de las siguientes formas:
 - 1. Al presentarse un cierto patrón de bits dados por el usuario
 - 2. Adquisición de datos mientras no se presente un cierto patrón de bits.
 - 3. Adquisición de un cierto número de muestras.
 - 4. En una condición no importa hasta que se llene el buffer de recepción.
 - 5. Adquirir datos mientras los valores tengan un cierto rango o valor.
- f) Unidad de DMA propia, por lo que no necesita que se le configure en un canal de la computadora.
- g) Adquisición de datos sincrónamente, temporizando ya sea interna o externamente (reloj del sistema bajo análisis).

De acuerdo a estas características, lo que pensamos sería alcanzable como proyecto para un trimestre sería:

- a) Que el analizador realice todas las funciones anteriores.
- b) Que trabaje en un ambiente Windows®, bajo una interfase de usuario sencilla, con ayuda y ejemplos.
- c) No se necesitará un amplio conocimiento de cómo funciona un analizador de estados lógicos comercial para poder usar el programa.

d) Solo se necesitará de la tarjeta y el paquete de software para poder tener un analizador de estados lógicos controlado por PC.

Toda la información necesaria, manuales de la tarjeta, el software NI-DAQ®, de instalación y configuración, de programación bajo Windows®, LabVIEW® etc., ya se poseen por lo que, para cumplir con estos objetivos se planea que, al inicio, se harían pruebas sencillas iniciales para ver hasta donde se puede llegar con la tarjeta, y que aspectos además de los mencionados podrán agregarse a las funciones del analizador de estados lógicos. Esto es:

- 1) Juntar toda la información que ya poseemos de la tarjeta para separar y organizar el trabajo de diseño que haga falta.
- 2) Revisar la configuración necesaria para que la tarjeta funcione correctamente, y tratar de hacerlo sistemático y reproducible (esto es, hacer la instalación del software y hardware lo mas sencilla posible).
- 3) Revisar todas y cada una de las funciones que nos ofrece la tarjeta, ya sea para, agregar nuevas funciones a nuestro analizador, adicionar funcionalidad a las mencionadas, y observar los posibles pros y contras de cada una de ellas.

Ya que la velocidad a la que se puedan adquirir los datos depende gran parte de la velocidad de la computadora que se este usando, sería importante que se pudiese contar con una PC Pentium II®, a 200Mhz mínimo, con ranuras tipo PCI, ya que como el sistema requiere trabajar a velocidades muy altas, la rapidez es un factor importante. En cuanto a memoria, ya que Windows® necesita para un funcionamiento aceptable por lo menos de 12 MB, y que el software que se planea desarrollar consumirá muchos de los recursos de la PC, se desea que la PC cuente a lo menos con 32 MB de memoria en RAM.

Como segunda parte se desarrollarían módulos que corresponderían, cada uno, a una función del analizador de estados lógicos, esto es, se hará por separado cada una de las funciones del analizador utilizando para ello la herramienta de programación grafica que ofrece LabVIEW®, basándonos principalmente en las librerías de National Instruments® NI-DAQ®, para esto se piensa:

1) Realizar y programar los módulos para:

- a) La adquisición de datos en 32, 16, y 8 bits.
- b) El disparo para el comienzo de la adquisición.
- c) La división de los 32 canales en grupos.
- d) La detección de patrones.
- e) Adquisición de datos síncrona y asincrónamente
- f) Temporización Interna y externa.
- g) Velocidad de adquisición.
- h) Tamaño del buffer de adquisición.
- i) Programación adicional que pueda requerir la tarjeta.

Por último, ya con los módulos principales desarrollados, se integran estos junto con una interfase fácil para el usuario, que tendrá que ser lo mas sencilla y presentable posible, bajo un ambiente Windows, para ello se planea utilizar como herramienta de programación el software de LabVIEW®, teniendo en cuenta:

1. Decidir en que forma se presentaran los datos adquiridos gráficamente hexadecimalmente o decimalmente.
2. Decidir si se puede implementar el desensamble de instrucciones cuando se analiza un microprocesador o microcontrolador de Intel®.
3. Decidir que tipo de interfase será la mejor para nuestra aplicación y que elementos tendrá (íconos, barras de trabajo, menús, etc.).
4. Decidir si se podría crear un analizador de estados lógicos virtual, con botones y perillas controlados por el usuario.

COMPARACIÓN DE UN ANALIZADOR LÓGICO COMERCIAL CON EL QUE SE DESARROLLARÁ.

Para tener un buen punto de referencia, sobre que características debe cumplir un analizador de estados lógicos actual, se tomo como referencia a uno de fabricantes y diseñadores de equipos de laboratorio, TEKTRONIX®. Durante muchos años Tektronix® se ha destacado por crear los aparatos de medición mas innovadores, con lo ultimo en tecnología y diseño.

La serie de analizadores lógicos que se tomaron como base fue la serie AS/400 que posee una gran variedad de analizadores de estados lógicos para distintos tipos de aplicaciones. A pesar de que todos son diferentes y que cada uno resuelve una necesidad especifica, tienen similitudes, como los son:

1. Son capaces de manejar frecuencias arriba de los 400 Mhz. Para esto utilizan una tecnología desarrollada por ellos mismos llamada MagniVu®, la cual consiste en no tomar los datos completos de todos los canales de golpe, sino que se hace una especie de retardo sobre cada bit, cuando se tiene completo el grupo de bits se carga en la memoria, con esto se provoca que la frecuencia a la que trabaja el analizador en realidad sea mucho menor que la de los datos que recibe.
2. Todos poseen la característica de la detección de picos (glitch detection) la cual consiste en la detección de los posibles picos o alzas de voltaje que se pudieran presentar en el sistema, dichas alzas son importantes ya que debido a este tipo de fallas un sistema es capaz de tomar un cero como un uno digital y quedar congelado.
3. Se puede determinar el tipo de disparo que uno quiere hacer, bajo un cierto patrón, un cierto numero de condiciones, o de no importa. También se puede programar para que se dispare cuando detecte un pico de voltaje.
4. Posee 80 canales de entrada, los cuales pueden ser agrupados como uno desee, sin restricciones.
5. Algunos trabajan si se desea junto con una PC bajo un ambiente con una interfase de usuario excelente.

6. Debido a la increíble frecuencia a la que trabaja, tiene una característica única en el mercado de los analizadores lógicos, es capaz de presentar una señal analógicamente tal como si se viera esta en un osciloscopio.

Esta característica le da una increíble flexibilidad y habilidad para detectar los mínimos cambios más significativos en una señal digital que pudieran estar afectando al sistema bajo prueba.

7. Posee la capacidad de desensamblar instrucciones, esto es dado un cierto código maquina presenta en pantalla el código en ensamblador para las familias de procesadores mas importantes que existen (La familia Z80® de Zilog®, La familia 8086-88® de Intel®, La familia 68HC11® de Motorola® mas algunos otros).

En general estas son las características que hacen especial a este analizador de estados lógicos digital, cuyo precio es de alrededor de 30,000 USD, claro que lo vale, pero muchas de las funciones por lo cual tiene este precio nunca serán utilizadas en un laboratorio convencional.

El analizador lógico que proponemos, contará con las características generales con las que debe cumplir cualquier analizador de estados lógicos comercial. Por ejemplo la velocidad de trabajo de 400 Mhz es una velocidad de adquisición demasiado alta para la clase de diseños que se realizan en un laboratorio como el nuestro, y se desaprovecharía esta característica; nuestra tarjeta será capaz de adquirir datos hasta 20 Mhz, un valor que cubre fácilmente las necesidades.

La detección de picos es una característica que se deriva de la capacidad de adquisición de 400 Mhz, por lo cual no se podrá tener en nuestro analizador una característica similar.

La forma de disparo, con excepción del disparo al detectar un pico será exactamente igual, con los mismos tipos de disparo y características de cada uno.

Se tendrán solo 32 canales de entrada/salida, que pueden configurar grupos de 8, 16 ó 32 bits. Este número de canales es el necesario para conectar 8 bits de datos, 20 de direcciones y algunas otras señales que se quieren monitorear como alguna señal de habilitación.

Creemos que podemos llegar a presentar una interfase de usuario para Windows® lo suficientemente aceptable, que podrá tener todas las necesidades de usuario y a su vez será capaz de explotar lo mejor posible el software de la tarjeta y las características de la misma.

Con esta comparación queremos dar a entender que no se quiere ni se intenta construir un analizador capaz de ganar el mercado a uno comercial sino se quiere crear uno que sea capaz de resolver todas las necesidades que se tienen en un laboratorio, a un costo mucho mas bajo.

RESULTADOS.

En la primera parte del proyecto lo importante fue comprender el funcionamiento de un analizador lógico, por lo que realizamos las actividades del manual del usuario “Quick Start” del analizador de estados lógicos 3002 Prism Series de Tektronix® que se encuentra en el laboratorio de Instrumentación Médica Electrónica del área de Ingeniería Biomédica en el edificio T cubículo 224 de esta unidad.

También fue importante relacionarnos con el lenguaje de programación LabVIEW®, así que asistimos a algunos seminarios teóricos y prácticos, además de realizar el Tutorial de LabVIEW 6.1®.

Para la segunda parte se retomo el analizador lógico con que se contaba, el cual solo leía pulsos desde la tarjeta de adquisición, el diagrama de programación se muestra en la Figura 5, así como realizar las actividades propias del analizador TLA700 Series de Tektronix® que se encuentra en los laboratorios del área de Ingeniería Electrónica ubicados en el primer piso del anexo del edificio T de esta unidad, esto con el fin de ver otro tipo de analizador y compararlo con el anterior para poder tener en mente la presentación que requería nuestro proyecto.

Cabe mencionar que nuestra mayor limitación para el desarrollo de este proyecto fue aprender lo básico sobre el Lenguaje de programación de LabVIEW 6.1®, así también tuvimos dificultades para realizar las actividades correspondientes a los analizadores de Tektronix®.

Consideramos que los avances realizados a este proyecto son de gran utilidad y dejan la puerta abierta para quien se interese en seguir con este trabajo.

A continuación se explica de una manera concisa los avances que se realizaron al AL-UAMI-2004.

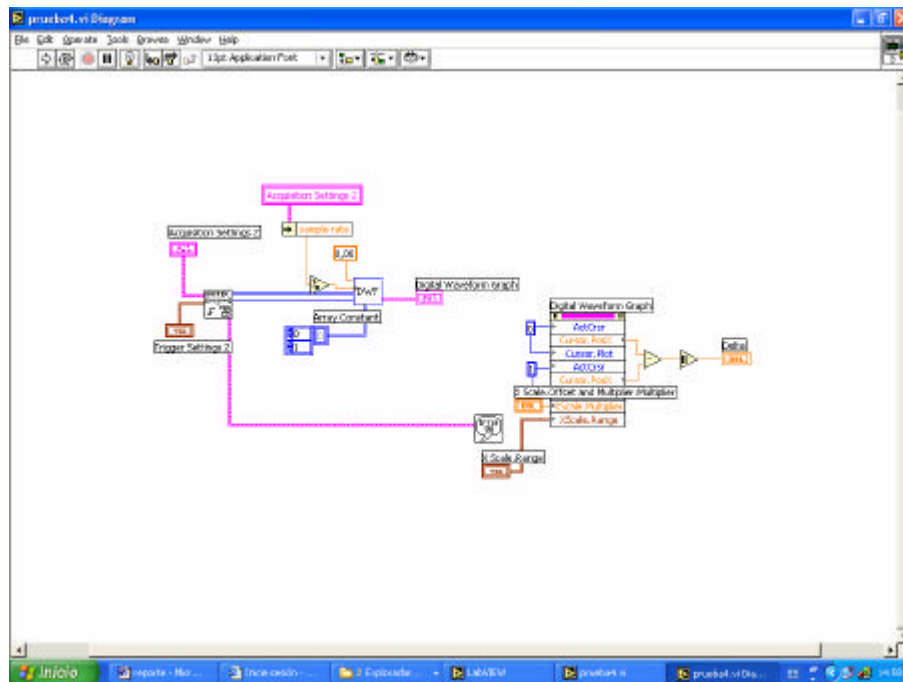


Figura 5. Diagrama de programación del Analizador de estados lógico inicial.

DILUCIDACIÓN DEL DIAGRAMA DE PROGAMACIÓN DEL AL-UAMI-2004

Por nuestra parte se intento que se pareciera mas al analizador lógico comercial, así que se agregaron los siguientes bloques:

Bits de Actividad.

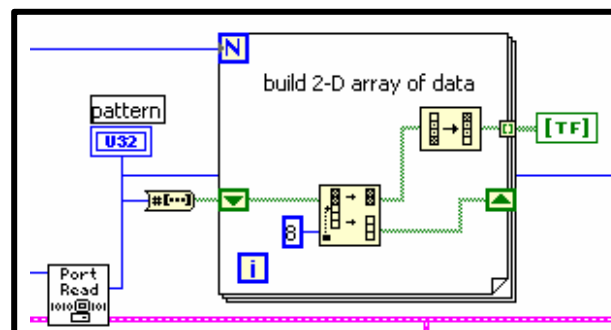


Figura 6. Diagrama de bits de actividad

En este bloque tenemos una librería que lee del puerto del cual obtenemos un patrón, el cual se convierte a un arreglo binario, así este arreglo es representado por leds dentro del panel (Fig. 6).

Condiciones de disparo (Trigger)

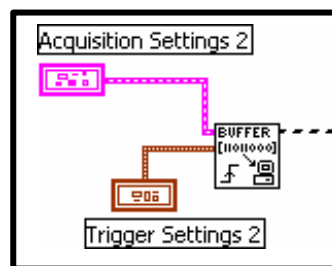


Figura 7. Buffer

Dentro del cluster *Trigger Settings 2* (Fig. 7) tenemos condiciones de disparo tales como:

- Muestras antes del disparo. Aquí el usuario decide cuantas muestras requiere que se tomen antes del disparo.
- Tiempo limite p/ disparo. Es el tiempo en unidad de segundos que se va a ejecutar el programa, es decir, si en el tiempo deseado no se encuentra palabra donde se empezara a muestrear, entonces el programa se detendrá.
- Patrón de disparo. Es la palabra que el programa buscara para dentro de los datos adquiridos, en el momento que se encuentre el programa empezará a almacenar los datos en el buffer.

Muestras y Lista de Puertos

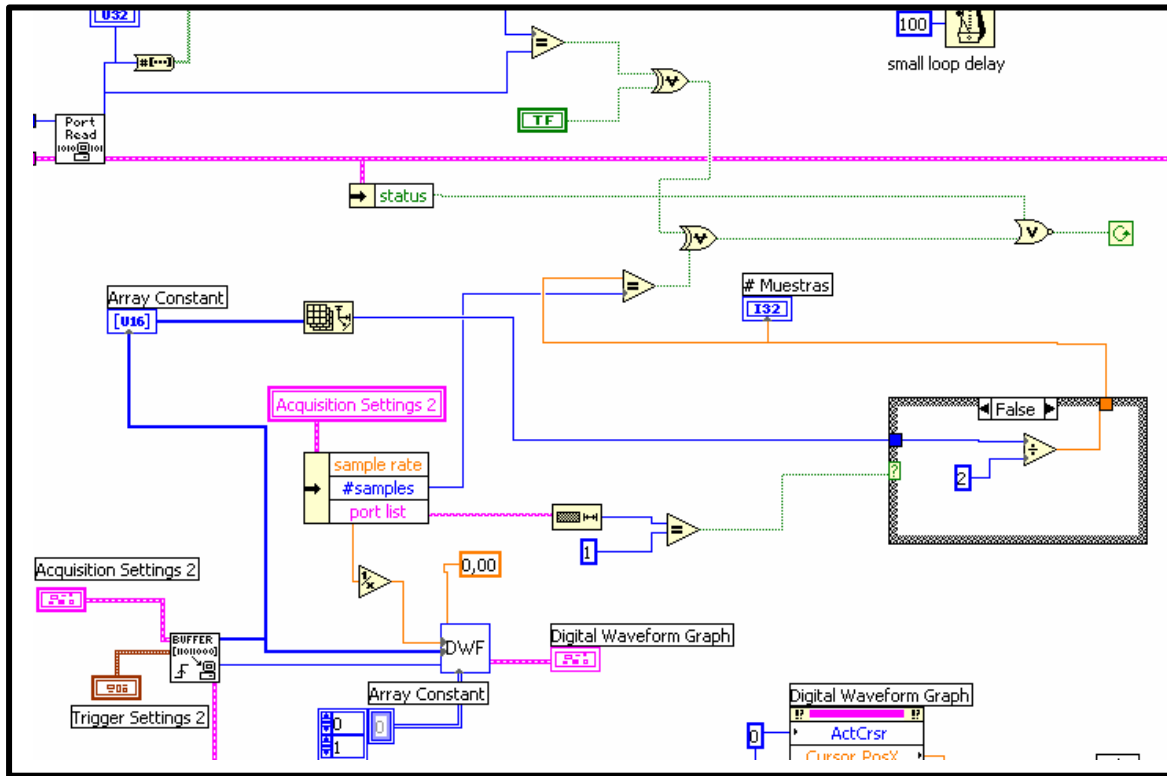


Figura 8. Bloque del número de muestras y *port list*.

Como se muestra en la figura 8 del buffer, que estamos leyendo pulsos directamente de la tarjeta de adquisición, los cuales son vistos en una grafica digital, pero también se pueden obtener en palabras de 8 u 16 bits, ya sea en binario, decimal o hexadecimal. Los datos leídos del buffer son almacenados en un arreglo de constantes, para que el programa se detenga cuando se aya cubierto el numero de muestras deseadas, necesitamos ir viendo el tamaño del arreglo. El tamaño del arreglo se introduce dentro de una estructura case, porque si estamos leyendo 8 bits estos se desplegarán en una casilla del arreglo, pero si leemos 16 bits necesitamos dos casillas para poder observarlos. Al salir de la estructura case, las muestras obtenidas son comparadas con el numero de muestras deseadas, hasta ser iguales el programa no se detiene.

La estructura de *case* es controlado por la lista de puertos (que puede ser {0} ó {0,1}), la lista de puertos es una cadena de caracteres que representa los puertos que quiero leer. La cadena la envío a una librería que me regresa el numero de componentes, así puedo yo saber se deseo 8 o 16 bits, de esta forma tengo dos casos cuando la cadena es de uno o de tres.

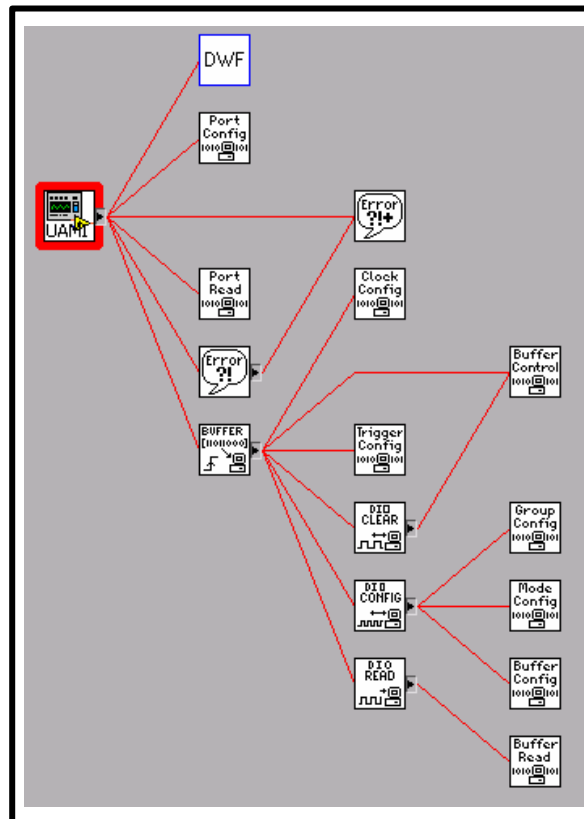


Figura 9. Diagrama de Jerarquía

En la figura 9 se muestra el diagrama de jerarquía, en el cual se puede observar las librerías y subVI's que se utilizaron dentro del programa AL-UAMI-2004; para tener acceso a esta ventana selecciona **Browse»Show VI Hierarchy** a partir del panel o del diagrama de dicho programa.

DESCRIPCIÓN DEL PANEL FRONTAL DEL AL-UAMI-2004

En esta sección se explicara el manejo del analizador de estados lógicos básico con el objetivo de que cualquier usuario pueda hacer uso del mismo.

En la figura 10. se muestra el panel completo del AL-UAMI-2004.

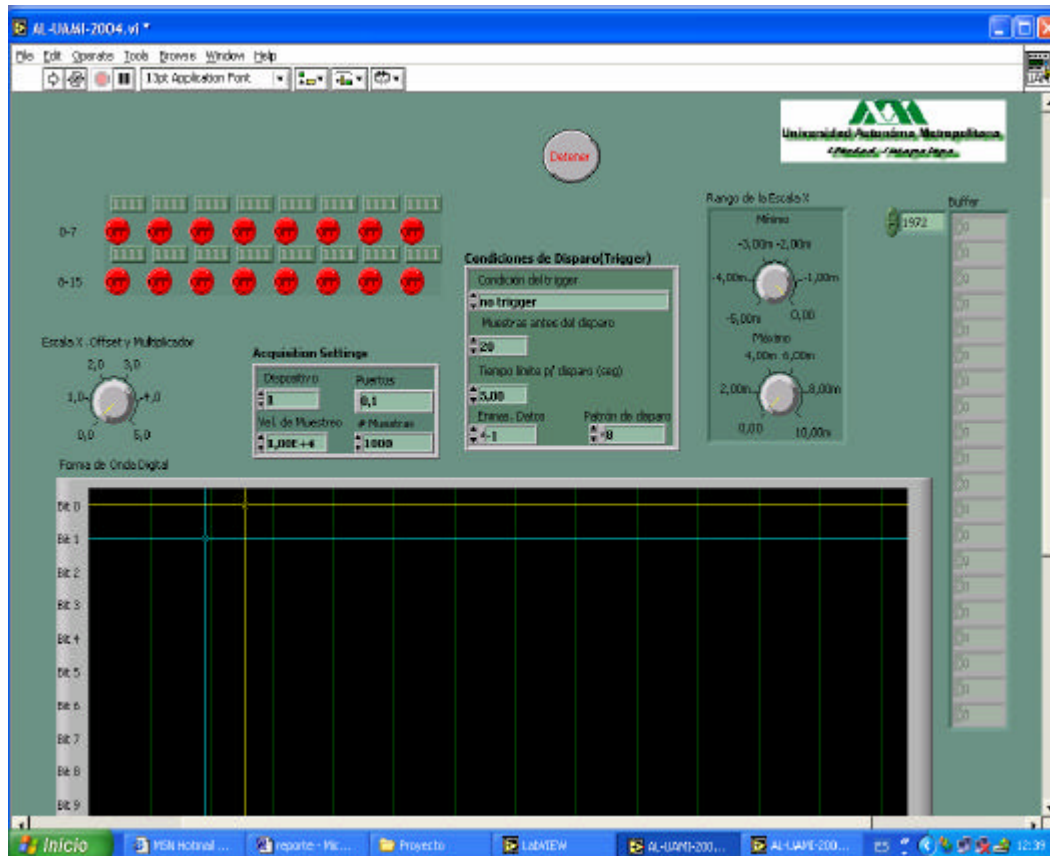


Figura 10. Panel de control del AL-UAMI-2004

Dentro de nuestro panel de control contamos con controles e indicadores, los cuales se explican a continuación.

Como controles tenemos los siguientes objetos:

Condición del Trigger.

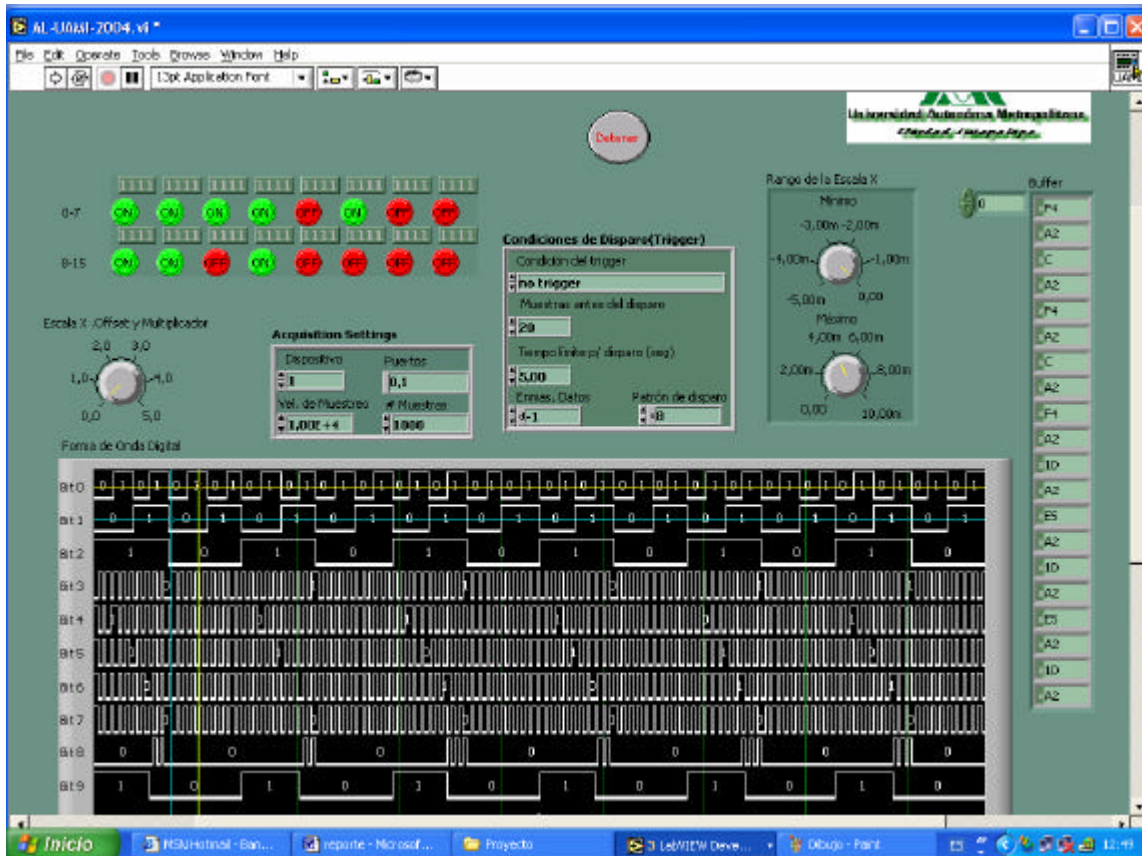


Figura 11. Condición del Trigger (no trigger)

En la condición del trigger tenemos dos opciones las cuales son *no trigger* y *trigger on pattern match*, el primero no toma condiciones de disparo simplemente registra los estados en el buffer además de desplegar los datos leídos en el display de forma de onda digital en el momento en que el usuario decida iniciar el muestreo (Fig. 11).

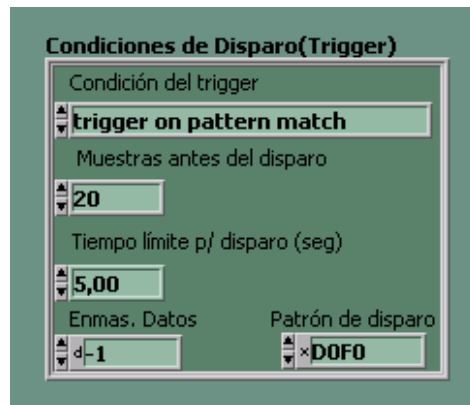


Figura 12. Condiciones de disparo (trigger on pattern match).

En el caso de que se requiera que el analizador empiece a monitorear según un patrón de disparo entonces se debe seleccionar *trigger on pattern match* dentro de *Condición del trigger* (Fig. 12), en este caso debemos considerar cuatro condiciones que son las siguientes.

Muestras antes del disparo. Es el número de muestras que el usuario requiere antes del patrón de disparo, por default son 20 muestras pero el mínimo de muestras que se pueden tomar son 8, el máximo depende de el número de muestras en total que se requieren.

Tiempo límite p/ disparo. Cuando el programa busca un patrón de disparo, puede ser que este no lo encuentre, entonces necesitamos darle un tiempo límite al analizador, para que en caso que no encontrar el patrón dentro de este tiempo, entonces el analizador se detenga.

Enmascaramiento de Datos (Enmas. Datos). Es el valor con el cual los datos obtenidos se enmascararán para ser comparados con el patrón de disparo. Para cualquier requerimiento por default siempre debe ser -1.

Patrón de Disparo. Aquí el usuario podrá buscar un valor dentro de los datos monitoreados para que el analizador empiece a almacenarlos en el buffer y podrán ser vistos en el display de forma de onda digital.

En la Figura 13. podemos ver el analizador con una condición de disparo *trigger on pattern match*, donde se busca el patrón D0F0, con 20 muestras de pre-disparo, podemos verificar en el buffer que después de 20 muestras se encuentra el valor deseado (como estamos leyendo 16 bits, el buffer almacena 8 bits en cada casilla, por lo tanto necesita dos casillas para una palabra de 16 bits).

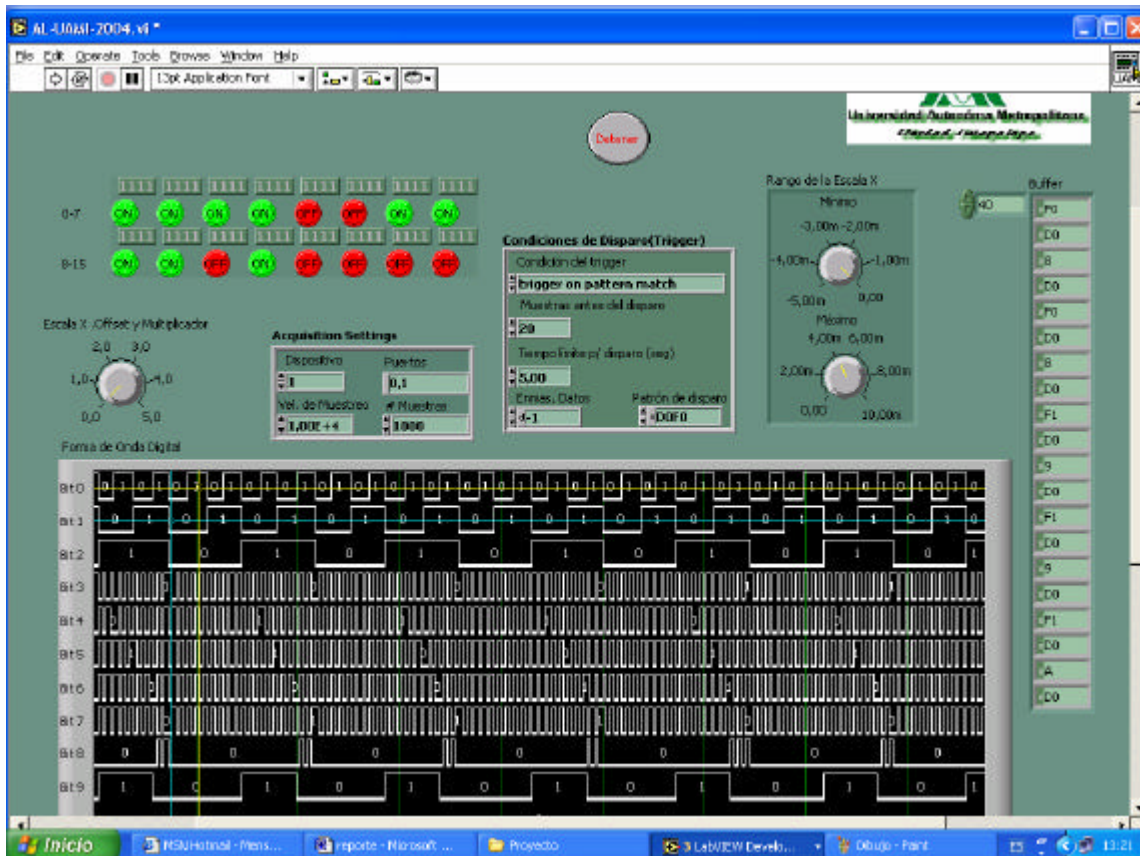


Figura 13. Panel de control con patrón de disparo

Otro de los cluster de control es Acquisition Settings. El cual contiene cuatro condiciones a considerar (Fig.14):



Figura 14. Acquisition Settings.

Dispositivo. En caso de tener mas de un dispositivo de I/O seleccionamos con cual se hará el análisis, como solo tenemos uno entonces siempre será 1(Fig. 14).

Velocidad de Muestreo (Vel. de Muestreo). Es la velocidad con que se hará el muestreo, por default son 10k (1,00E+4), pero puede trabajar con cualquier frecuencia menor a 20M.

Puertos. Para nuestra tarjeta de adquisición utilizamos dos puertos el puerto 0 muestrea del bit 0 al bit 7 y el puerto 1 del bit 7 al bit 15, así que los valores que puede admitir son 0 ó 0,1.

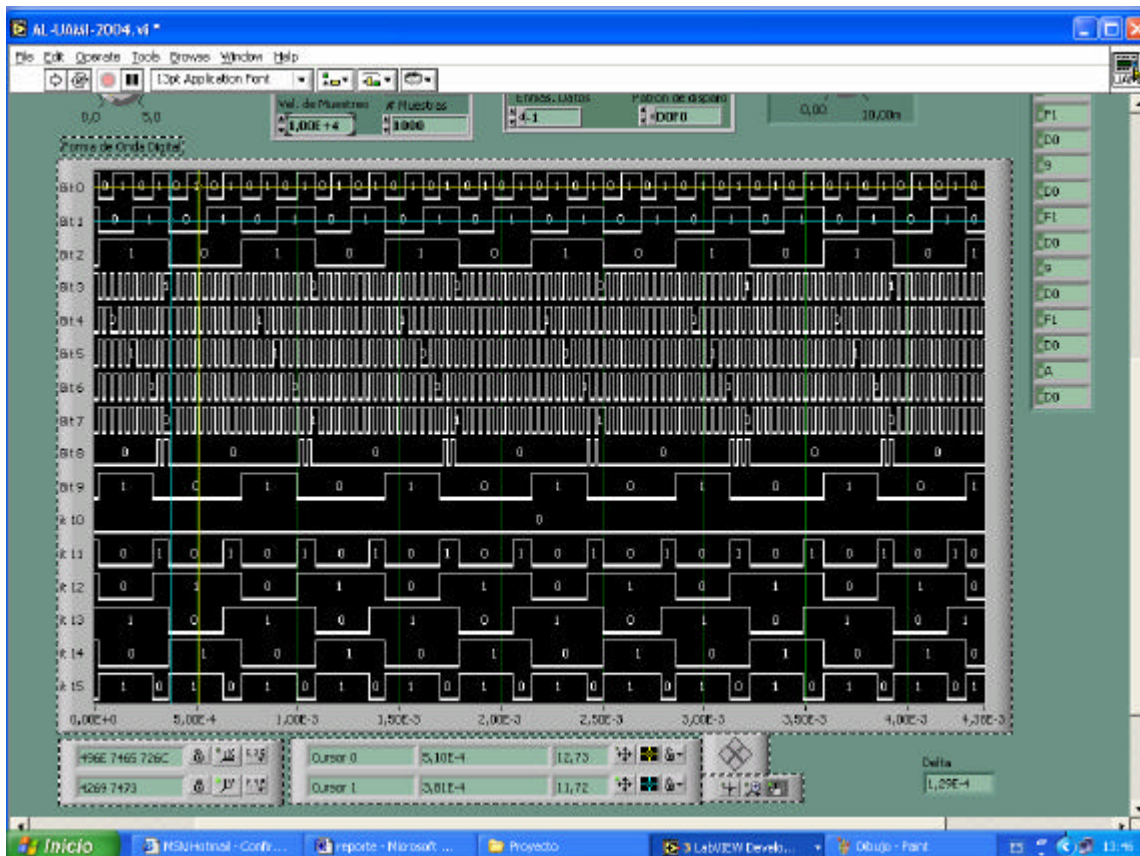


Figura 15. Display Digital con una lectura de 16 bits.

Los datos serán desplegados en el display digital (Fig. 15) y serán almacenados en el buffer, como el buffer almacena 8 bits por celda, necesitará el doble de celdas para almacenar todos los estados (Fig. 16).



Figura 16. Buffer para 16 bits.

Muestras. El usuario podrá determinar cuántas muestras requiere, ya sean de 8 y 16 bits, se almacenarán en el buffer (Fig. 16) y podrán ser visualizadas en el display (Fig. 15). (Si se hacen muestras de 16 bits, el buffer tendrá el doble de espacio ya que necesita dos celdas para una palabra de 16 bits).

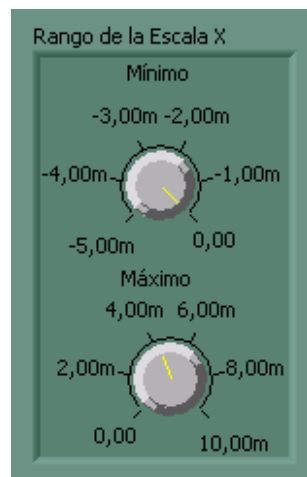


Figura 17. Rango de la Escala X.

El último cluster con que cuenta el analizador es el de *Rango de Escala X*, el cual contiene dos perillas, este cluster controla el despliegue de las líneas digitales dentro del display.

Mínimo. Es el valor donde se requiere que empiece a desplegar muestras por default es cero pero puede ser hasta -5m, en la Figura 17 podemos ver el display con un valor mínimo de -4,00m.

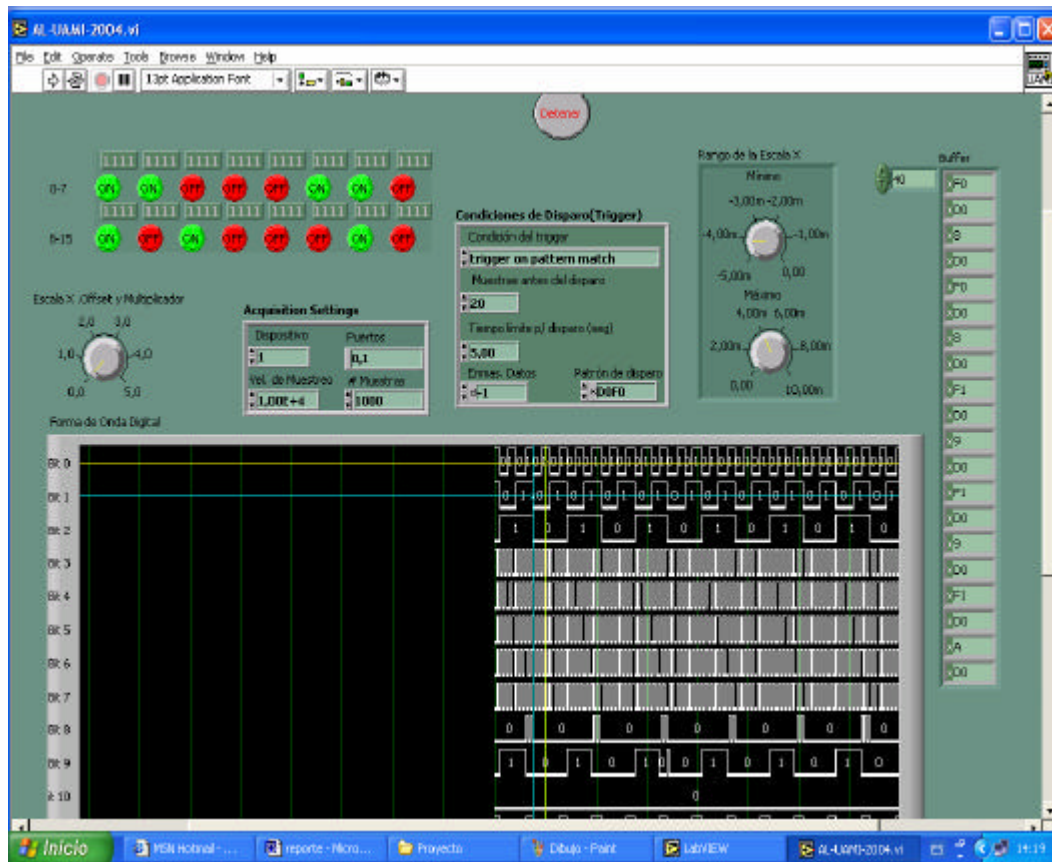


Figura 17. Digital con un valor mínimo de despliegue de -4m

Máximo. Es el valor máximo en X que va a abarcar el display, por default es 0 pero para poder visualizar los pulsos es necesario ajustarlo, en la figura 18 se ilustra un ejemplo con un valor máximo de 4m.

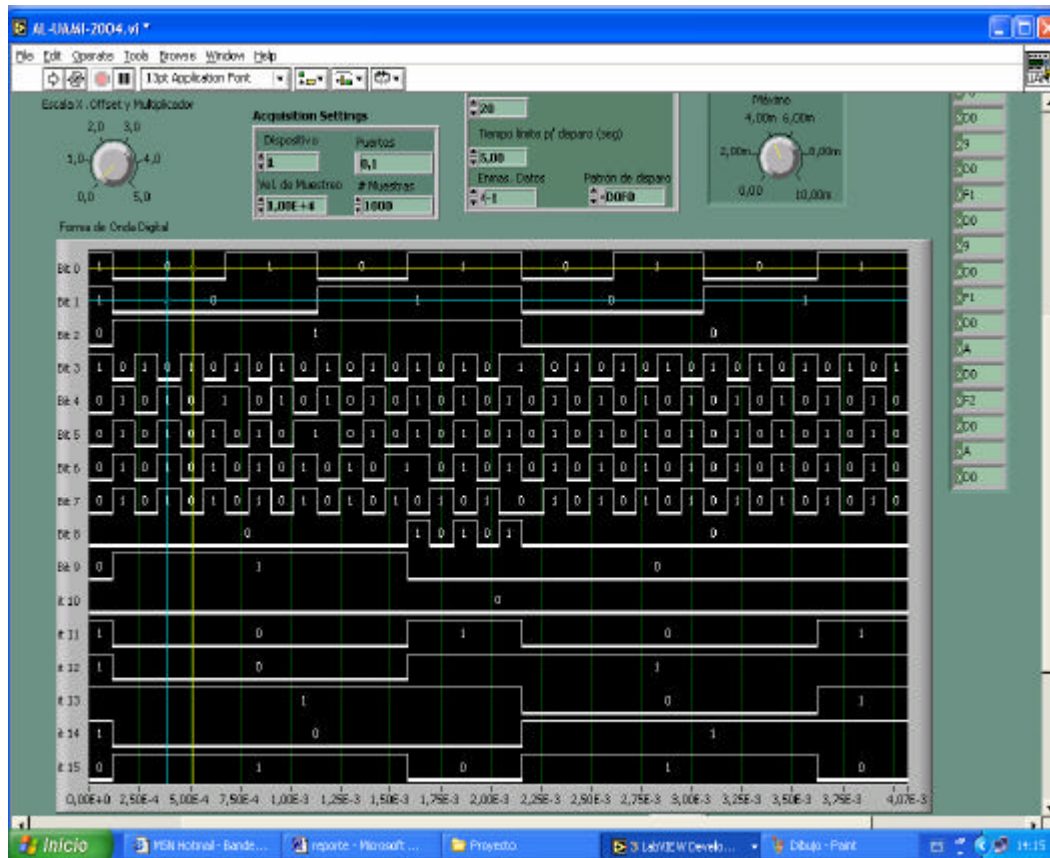


Figura 18. Display Digital con un valor máximo de despliegue de 4m.

Dentro del analizador de estados lógicos básico, tenemos dos maneras de de visualizar los estados obtenidos a través de la tarjeta de adquisición son:

Buffer de Datos.

El buffer almacena los estados lógicos los cuales pueden ser desplegados de en binario, decimal, octal o hexadecimal, para cambiar la forma de despliegue se hace click izquierdo en el pequeño botón que se encuentra a la izquierda de las celdas (Fig. 19).

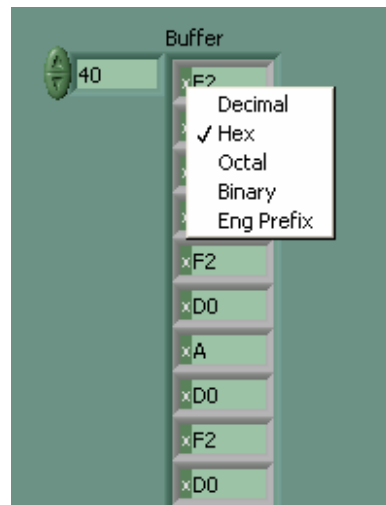


Figura 19. Buffer de Datos (Forma de despliegue).

El tamaño del buffer va en función de el control *# Muestras*, cada celda muestra 8 bits, así que para un palabra de 16 bits ocupara dos celdas, por ejemplo si se requiere 64 muestras, usando los puertos *0,1*, los datos de desplegaran en 128 celdas (Fig. 20).

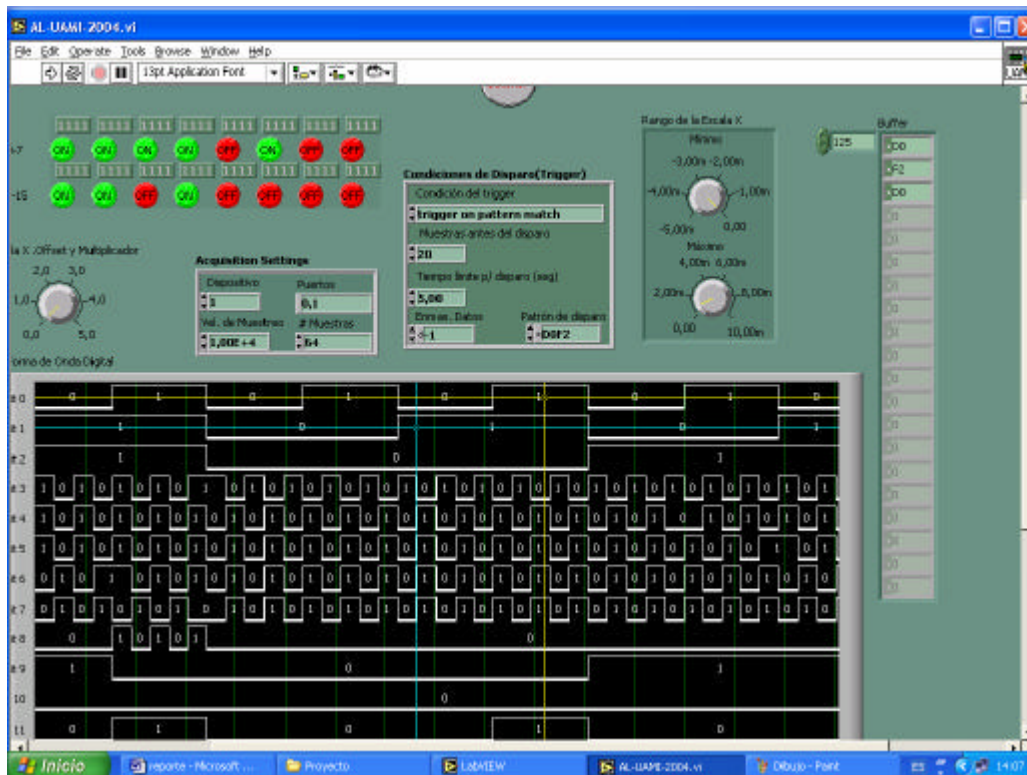


Fig. 20. Panel de control (64 muestras).

Para recorrer el buffer, se cuenta con un control que nos indica el número de la celda, se puede meter directamente el número o irse moviendo con las flechas que se encuentran a la derecha del indicador (Fig. 19).

Gráfica de Forma de Onda Digital.

La gráfica digital cuenta con algunas herramientas que se encuentran en la parte inferior de la gráfica y que se describen a continuación:

Cursores. Los cursores nos permiten medir el tamaño de los pulsos en unidades de segundos, en la Figura 21, podemos ver los controles de los cursores.

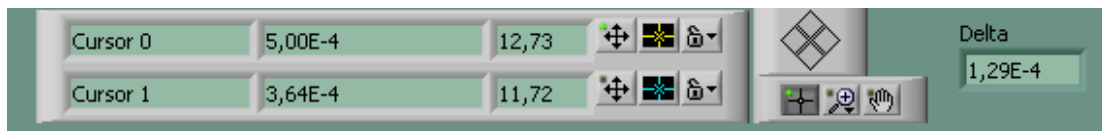


Fig.21 Controles del cursor.

Tenemos dos cursores (Cursor 0 y Cursor 1), La segunda ventanilla de izquierda a derecha muestra la posición del cursor, la cual puede ser modificada directamente o usando el control del cursor (Fig. 21.B), seleccionando antes el cursor requerido (Fig. 21.A)



Fig.21.A. Selección del Cursor 0



Fig. 21. B. Control del Cursor

También cuenta con un menú de formato donde puedes cambiar configuraciones tales como el color el estilo (Fig. 21.C)

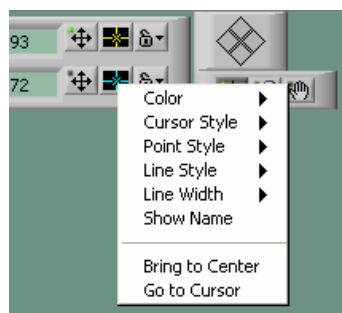


Fig. 21. C. Formato de cursor

La medición requerida es la diferencia entre los cursores y es desplegada en el indicador delta, (Fig. 21.D).

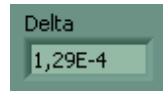


Fig. 21. D. Medición entre cursores

El cursor se debe de ajustar antes de correr el analizador, de otra forma la medición no será tomada, así que sugerimos que se ejecute dos veces, en la primera ajustamos los cursores para que en la segunda podamos ver la delta.



Fig. 21. E Controles

Los últimos tres controles son para ser aplicados directamente en la grafica, al seleccionar el primero (de izq. a der. de la Fig. 21.E), nos permite mover los cursores.

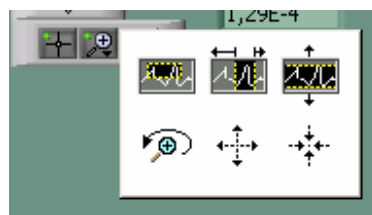


Fig.22. Controles de la grafica

El segundo control (Fig. 22), pertenece a la grafica con el podemos seleccionar diferentes maneras de hacer zoom a las lecturas.

El último del los controles es una manita que nos sirve para tomar la grafica y desplazarla de un lado a otro para observar las lecturas.

CONCLUSIONES.

Durante el tiempo de nuestra estancia en la Universidad surgieron varias inquietudes; una de ellas, tal vez la más importante, fue el deseo de trabajar en un proyecto en el que aplicáramos los conocimientos adquiridos durante la misma, al tener la oportunidad y la invitación del M. en C. Agustín Suárez Fernández, profesor titular de la unidad, decidimos aceptar este proyecto. Cabe mencionar que este proyecto fue empezado por Luis Bernardo González Burgos y Ricardo Gutiérrez Malacat, así que nosotros decidimos darle continuidad a dicho proyecto.

Hay que mencionar que durante los trimestres en los que desarrollamos este trabajo tuvimos complicaciones, la mayor fue el adaptarnos al estilo de programación de LabVIEW®, aunque es un software basado en programación gráfica tiene un número extenso de aplicaciones. Agradecemos la oportunidad y el interés de nuestro asesor por enviarnos a conferencias sobre este software, impartidas por personal de National Instruments®, esto fue de gran utilidad para el desarrollo del AL-UAMI_2004.

Consideramos que el avance que se tuvo en este proyecto por parte de nosotros fue productivo y que esperamos que este reporte sirva para quien se interese por continuar con el desarrollo de este analizador de estados lógicos, para poder así, darle a nuestra Universidad una opción para aplicar este proyecto en la docencia.

Por medio de este escrito agradecemos las facilidades otorgadas por la Universidad Autónoma Metropolitana, el Departamento de Ingeniería Eléctrica, el Área de Ingeniería Biomédica y a nuestro asesor para el desarrollo de este proyecto; esperamos que nuestro trabajo sea de utilidad para la Institución que nos forjó como Ingenieros.

López López Mario Alberto.

Quintero Zamudio José.

APÉNDICE A.

“PROCEDIMIENTO DE INSTALACIÓN DEL SISTEMA.”

Para la instalación de analizador de estados lógicos AL-UAMI-2004 es necesario que se cuente con el siguiente equipo:

- Tarjeta DAQ PCI-DIO-32HS® de National Instruments®.
- El software NI-DAQ V.6® o superior. Generalmente viene con la tarjeta.
- Windows® 95/98 ó superior.
- Computadora con un procesador Pentium® o superior.
- Una ranura PCI libre.
- 16 MB de memoria RAM.

Para la instalación completa del sistema se tienen que seguir los siguientes pasos:

1.- Instalación del software NI-DAQ® de National Instruments®.

- Inserte el CD-ROM de NI-DAQ®.
- El sistema arrancara automáticamente el programa de instalación, solo siga los pasos que se le indican.
- Solo instale los controladores de la tarjeta y el software NI-DAQ®.
- Reinicie su computadora.

2.- Instalación del software LabVIEW V6.1® de National Instruments®.

- Inserte el CD-ROM de LabVIEW®.
- El sistema arrancara automáticamente el programa de instalación, solo siga los pasos que se le indican.
- Reinicie su computadora.

3. Instalación de la tarjeta PCI-DIO-32HS® de National Instruments®.

- Asegúrese de tener apagada su computadora.
- Retire la cubierta de la computadora.
- Extraiga cuidadosamente la tarjeta de su empaque antiestático e insértela en la ranura PCI libre.
- Coloque de nuevo la cubierta de su computadora.
- Encienda la computadora y siga los pasos que le indique el sistema operativo cuando detecte la tarjeta.

BIBLIOGRAFÍA.

- DATA ADQUISICION TUTORIAL. National Instruments Corporation.
- FEELING COMFORRTABLE EHIT LOGIC ANALIZERS. Introduction to Logic Analizers, Tektronix.
- DAQ, AT/PCI-DIO-32HS USER MANUAL. HIGH DPEED DIGITAL I/O BOARS. AT, EISA, O PCI BUS COMPUTERS, National Instruments Corporation, Número de Serie 321464A-01, Febrero de 1997.
- NI- DAQ USERS MANUAL. National Instruments Corporation, Junio de 1998.
- DATA ADQUSIION (DAQ) FUNDAMENTTALS. Aplication Note 007, National Instruments.
- <http://www.ni.com>