



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN

**“Sistema de adquisición y envío de datos biomédicos a través
de Internet utilizando el protocolo FTP”**

T E S I S

QUE PARA OBTENER EL TÍTULO
DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

JOSÉ LUIS JUÁREZ TREJO

ASESOR: ING. ARTURO OCAMPO
ÁLVAREZ

MÉXICO

2004



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN
DIRECCIÓN

JOSE LUIS JUAREZ TREJO
Presente

Con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobado su tema de tesis y asesor.

TÍTULO:
"SISTEMA DE ADQUISICIÓN Y ENVÍO DE DATOS BIOMÉDICOS A TRAVÉS DE INTERNET
UTILIZANDO EL PROTOCOLO FTP"

ASESOR: Ing. ARTURO OCAMPO ÁLVAREZ

Aprovecho la ocasión para reiterarle mi distinguida consideración.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
San Juan de Aragón, México, 5 de mayo de 2004.

LA DIRECTORA

ARQ. LILIA TURCOTT GONZÁLEZ



C p Secretaria Académica
C p Jefatura de Carrera de Ingeniería en Computación
C p Asesor de Tesis

LTG/AIR/cs

Dedicatoria.

Dedico este trabajo a todas aquellas personas que tuvieron algo que ver para que éste fuera posible. A mi asesor, el Ing. Arturo Ocampo Álvarez, a los profesores de la carrera Ing. En Computación generación: 1997-2001, gracias, no olvidaré las lecciones aprendidas.

Dedico este trabajo a mi madre, y quiero decirte que aún recuerdo con tanto cariño aquella frase que algún día me dijiste: “Querer es poder...” gracias Mamá. Al igual, dedico este trabajo a mi padre, que, aunque físicamente ya no está con nosotros, espero que desde allá arriba pueda sentirse orgulloso de mí...

A mi hijo “Dany”, porque sé que con este trabajo te darás muy bien cuenta de que todo lo que te propongas en la vida lo puedes lograr, sólo hay que ponerle mucho empeño a las cosas, ¡espero ser siempre tu mejor y más correcto ejemplo que elijas mi niño hermoso!...

Finalmente a ti Carmen ya que fuiste la persona que me impulsó a seguir estudiando, y no te importó que nos priváramos de ciertas cosas sólo con el fin de que culminara con este proyecto, pero sobretodo, porque siempre, desde que te conocí, has sido mi motor y mis ganas de superación en la vida.

Agradecimientos.

Con este trabajo, culmina otro ciclo muy importante en mi vida, con el paso de los años he aprendido lo valioso que es el tiempo, y lo mucho que puede significar para lograr, o no, nuestros propósitos, por ello, agradezco primeramente, a ti Señor, pues has sido mi guía inequívoca a través de todos estos años. Gracias por mandarme retos y lecciones, gracias por las cosas aprendidas y porque siempre has sabido guiarme en el gran sendero de la vida, gracias por enviarme tantas y tantas cosas buenas, simplemente, gracias por un nuevo día...

Gracias a ti Mamá, porque siempre que te he necesitado he contado con tu valioso apoyo y porque siempre me has ayudado desinteresadamente en los momentos más difíciles, este logro también es tuyo...No olvides que te quiero mucho...

Gracias a ti Papá, y aunque ya no estés aquí con nosotros físicamente, estoy seguro que desde allá arriba te sientes orgulloso de mi y velas por nuestro bienestar junto con Dios...

Muchas gracias Carmen, simplemente porque siempre has estado ahí, cuando más te necesito, creo que este logro en gran parte también es tuyo, tu te has encargado en todo momento de encarrilarme en el canal apropiado, gracias por compartir tu vida conmigo, eres la mejor mujer que dios me pudo haber enviado. ¡Te amo!

A ti Danielin, mi querido niño, pues tú más que nadie eres mi motor y mis ganas de superación y el deseo de ser para ti un ejemplo bueno en la vida, espero que con este trabajo contribuya en brindarte una buena aspiración en tu pequeña pero inteligente personita que eres ahora, ¡te quiero mucho!

Finalmente, deseo también expresar mi agradecimiento al Ing. Arturo Ocampo Álvarez, pues sin su valiosa ayuda, no hubiera sido posible el desarrollo de este trabajo de tesis. Gracias maestro por su sencillez y sus buenos consejos...

Índice de contenidos.

	PÁGINA
ÍNDICE.....	ix
OBJETIVOS.....	xiii
a) Objetivo general.....	xiii
b) Objetivos específicos.....	xiii
INTRODUCCIÓN.....	xv
 CAPÍTULO 1. CONCEPTOS FUNDAMENTALES REFERENTES AL HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS.	
1.1 ¿Qué es un sistema de adquisición de datos?.....	3
1.1.1 Elementos de un sistema de instrumentación ó de adquisición de datos.....	4
1.1.2 Aplicaciones de los sistemas de Adquisición de Datos.....	5
1.2 Introducción al Puerto Paralelo.....	7
1.3 Direccionamiento del Puerto Paralelo.....	9
1.4 Pines y buses del Puerto Paralelo.....	10
1.5 Evolución y tipos de puertos paralelos, la norma IEEE1284.....	12
1.5.1 El estándar IEE 1284-1994.....	13
1.5.2 Modo EPP.....	14
1.5.3 Modo ECP.....	15
1.6 El conector Centronics.....	15
1.7 Interrupción del Puerto paralelo.....	16
1.8 El Convertidor Analógico a Digital ADC0808.....	18
1.8.1 Líneas de dirección (A, B, C).....	19
1.8.2 Cerrojo habilitador de dirección (Address Latch Enable ALE).....	19
1.8.3 Pulso de Reloj (Clock).....	19
1.8.4 Inicio (Start).....	20
1.8.5 Habilitador de salida (Output Enable OE).....	20
1.8.6 Señal de fin de conversión (End Of Conversion EOC).....	20
1.8.7 Entradas analógicas IN0 – IN7.....	20
1.8.8 Salidas digitales.- $2^1 - 2^8$	20
1.8.9 Vcc.....	20
1.8.10 Vref (+). Voltaje de referencia positivo.....	20
1.8.11 GND.....	21
1.8.12 Vref (-) Voltaje de referencia negativo.....	21
1.9 Modo de funcionamiento del ADC0808.....	21
1.10 Voltaje de Referencia y pulso de Reloj del ADC0808.....	25

1.11 El LM385 variable.....	25
1.12 El temporizador LM555.....	26
1.13 El integrado DM74L573AN.....	29
1.14 El sensor de temperatura LM35.....	30
1.15 Aplicaciones del LM35.....	31
1.16 El buffer SN74LS244.....	32

CAPÍTULO 2. CONCEPTOS FUNDAMENTALES REFERENTES AL SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS.

2.1 El lenguaje de programación Visual Basic 6.0.....	37
2.2 Razones para utilizar el lenguaje Visual Basic 6.0.....	38
2.3 La programación orientada a objetos.....	38
2.4 Librerías de vínculos dinámicos DLL.....	39
2.5 El archivo de definiciones en una DLL.....	40
2.6 Nombres decorativos para invocar funciones en Visual Basic.....	41
2.7 Conexión de Bases de datos Access sobre Visual Basic utilizando el modelo de datos ADO.....	41
2.7.1 Comparativa de modelos de datos.....	41
2.7.2 Recordset.....	43
2.8 El protocolo FTP.....	43
2.9 Configuración de un servidor FTP.....	44
2.9.1 Terminología relacionada.....	45
2.9.2 Tipos de datos en la transferencia por FTP.....	46
2.9.3 Modos de transferencia en FTP.....	46
2.9.4 Comandos utilizados frecuentemente.....	47
2.10 El control Transfer de Visual Basic 6.0.....	50
2.10.1 Aplicaciones posibles.....	50
2.10.2 Funcionamiento básico.....	50
2.10.3 Transmisión síncrona y asíncrona.....	51
2.11 Derechos y restricciones de acceso al servidor FTP.....	54
2.11.1 Descargar archivos hacia el servidor.....	54
2.11.2 Cantidad de Kb que se pueden descargar hacia el servidor.....	54
2.11.3 Establecer el número de conexiones al servidor de cada cliente.....	54
2.11.4 Establecer un tiempo determinado de conexión para cada cliente.....	54
2.11.5 Atributo de sólo lectura.....	55
2.11.6 Atributo de sólo escritura.....	55
2.11.7 Atributo de lectura/escritura.....	55
2.11.8 Atributo para borrar archivos.....	55
2.12 Operaciones lógicas BITWARE (&, ^,).....	56
2.12.1 Operación AND (&).....	56

2.12.2 Operación Exclusivo-OR (^).....	56
2.12.3 Operación Inclusivo-OR ().....	57
2.13 La parte grafica de Visual Basic.....	57
2.13.1 Usar el control Image para presentar mapas de bits.....	58
2.13.2 Usando el método PaintPicture.....	59
2.13.3 Liberar la memoria utilizada por los gráficos.....	59
2.13.4 Usando mapas de bits en formato .rle o metarchivos.....	59
2.13.5 El control Picture Box.....	60
2.13.6 Formatos gráficos aceptados.....	60
2.13.7 Cargar un gráfico en el control PictureBox.....	60
2.13.8 Usar el Portapapeles.....	61
2.13.9 Ajustar el tamaño de una imagen.....	61
2.13.10 Métodos gráficos.....	61
2.13.11 Usar el método Print.....	62
2.14 El Objeto FileSystemObject (FSO) de Visual Basic 6.0.....	62
2.15 El objeto Internet Explorer de Microsoft Windows.....	64
2.15.1 El método Navigate.....	66
2.15.2 La propiedad LocationURL.....	66
2.15.3 El suceso NavigateComplete2.....	67
2.15.4 Propiedad AddressBar.....	67
2.15.5 El evento BeforeNavigate2.....	67
2.15.6 Propiedad Busy.....	67
2.15.7 Propiedad GoBack.....	67
2.15.8 Propiedad GoFoward.....	67
2.15.9 Propiedad GoHome.....	67
2.15.10 Propiedad Refresh.....	67
2.15.11 Propiedad Stop.....	67
2.16 Archivos de Ayuda en Visual Basic.....	68
2.16.1 La propiedad HelpFile.....	68
2.16.2 La propiedad HelpContextID.....	69

CAPÍTULO 3. IMPLEMENTACIÓN Y RESULTADOS DEL SISTEMA DE ADQUISICIÓN DE DATOS.

3.1 Visión general del sistema de adquisición de datos.....	73
3.2 Convertidor analógico a digital ADC0808 configurado en modo de corrida libre.....	73
3.3 El termómetro electrónico, rango elegido e interpretación de la salida de datos digitales obtenidos en el ADC0808.....	74
3.4 El 555 configurado como multivibrador astable.....	75
3.5 El flip-flop JK DM74LS73A para generar pulso de reloj cuadrado-exacto.....	78

3.6 El Voltaje de referencia del ADC0808.....	79
3.7 Determinación de rangos de trabajo.....	79
3.8 El buffer SN74LS244.....	80
3.9 Análisis del sistema de adquisición de datos.....	81
3.10 Puerto y tipo de bus a utilizar.....	82
3.11 Metodología de programación de la aplicación Cliente.exe.....	84
3.12 Redireccionamiento del puerto paralelo.	88
3.13 Código fuente de la librería de vínculos dinámicos “REDIRECC.DLL”.....	89
3.14 Proceso y creación del archivo a transferir por Internet.....	98
3.15 Creación del dictamen en modo gráfico.....	104
3.15.1 Control PictureBox.....	104
3.15.3 Valores.....	106
3.16 Visualización de documentos HTML con Internet Explorer.....	109
3.17 La aplicación CesarFTP.exe.....	110
3.18 Envío del dictamen médico (Derechos, login, password, control de transferencia de archivos, propiedades, métodos, etc.).....	111
3.19 Metodología de programación de la aplicación Servidor.exe.....	112
3.20 La base de datos termometro.mdb.....	115
3.20.1 DataSource.....	118
3.20.3 DataField.....	118
3.21 Manejo de archivos y carpetas utilizando el objeto FileSystemObject de Microsoft Visual Basic 6.0.....	119
3.21.1 CopyFile.....	119
3.21.2 CopyFolder.....	119
3.21.3 CreateFolder.....	119
3.21.4 DeleteFile.....	120
3.21.5 DeleteFolder.....	120
3.21.6 FileExists.....	120
3.21.7 FolderExists.....	120
3.21.8 MoveFile.....	120
3.21.9 MoveFolder.....	121
3.22 Creación del Archivo de Ayuda: el programa Salomón.exe.....	121
CONCLUSIONES.....	123
GLOSARIO.....	125
BIBLIOGRAFIA.....	129
ANEXOS.....	131

OBJETIVOS.

a) Objetivo general.

Diseñar un prototipo para un sistema de adquisición de datos biomédicos multicanal, desarrollando para éste, un software de programación propio en el lenguaje de programación Visual Basic, haciendo uso del protocolo de transmisión de archivos (File Transfer Protocol) FTP.

b) Objetivos específicos.

- Elaborar un mecanismo de software para llevar a cabo el redireccionamiento del puerto paralelo.
- Establecer el pulso de reloj adecuado para el convertidor analógico a digital.
- Establecer el voltaje de referencia correcto, así como el cálculo de los rangos de trabajo.
- Establecer la programación adecuada para obtener los datos en la computadora desde el sistema de adquisición y poder visualizarlos en tiempo real.
- Elaborar un dictamen gráfico con los valores obtenidos.
- Establecer el mecanismo y programación adecuados para el envío de la información a través de Internet.
- Configurar un servidor FTP para poder transmitir ó recibir información.
- Crear una base de datos y vincularla con software al servidor, para almacenar la información obtenida por parte de las máquinas cliente.

INTRODUCCIÓN.

Actualmente los sistemas de comunicaciones son elementos indispensables en nuestra vida cotidiana, y en especial la telemedicina, la cual está desempeñando un papel muy importante, por lo que los Ingenieros en computación se ven involucrados en todos estos aspectos, pues resulta sumamente necesario el tener un perfil en hardware para interactuar con señales del mundo real, tal es el caso de las señales biomédicas, pues, saber procesarlas y transmitir las a una PC a través de un software que sea amigable pero a la vez confiable, resulta altamente indispensable pero sobretodo muy interesante, gratificante y puede contribuir con la medicina y otros muchos campos más.

PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA.

La presente tesis surge a partir de trabajos anteriores realizados en la Enep Aragón, donde se involucran aspectos de procesamiento digital de señales, microcomputadoras y bioingeniería, en estos trabajos se ha realizado ya, por ejemplo, un sistema de monitoreo y adquisición de señales de ECG (electrocardiograma) a través de Internet.

Este trabajo tenía la limitante de que requería software comercial para enviar los datos biomédicos en tiempo real (Wincon) con la limitante de manejar un sólo canal analógico de entrada, por lo que, en el presente tema de tesis se han propuesto como hipótesis, el hecho de que es posible la ampliación en el número de canales receptores, además de diseñar un software propio para enviar los datos biomédicos obtenidos a través de Internet mediante el protocolo FTP.

El prototipo que proponemos es capaz de redireccionar el puerto paralelo con el fin de habilitarlo como receptor de datos y mostrarlos gráficamente en tiempo real en la pantalla de una PC, para que, de esta manera, puedan ser almacenados como registros dentro de una base de datos, teniendo, finalmente, la opción de transmitirlos por la Web,

Es de ésta manera, que el presente trabajo está destinado a coadyuvar en todas estas funciones, cabe recalcar, que el sistema implementado tiene la capacidad de adquirir hasta ocho señales analógicas de baja frecuencia y está orientado a ser utilizado por cualquier médico que se dedique a hacer este tipo de diagnósticos y necesite llevar un registro de sus pacientes, ya que está implementado con una interfaz lógica y amigable, por lo que no se requieren grandes conocimientos de cómputo o conceptos altamente electrónicos, más bien lo que se pretende es implementar una solución asequible de adquisición de datos biomédicos para cualquier tipo de persona.

Es además, muy necesario que, cuando un sistema de control está destinado a atender a varios usuarios, exista algún mecanismo práctico para determinar el usuario que está siendo atendido en el momento y la manera en que se le pueda otorgar un mejor servicio personal además de contar con un registro detallado de los datos obtenidos. Para ello se requiere de

una serie de registros dentro de una base de datos, la cual podrá identificar a los usuarios y llevar una especie de bitácora ó expediente de sus señales biomédicas obtenidas, con relación al sistema en general.

Es en éstas tareas donde el control y la automatización de carácter electrónico, entran en funcionamiento, realizando la toma de datos y según éstos, la activación de una base de datos ante determinadas variables obtenidas.

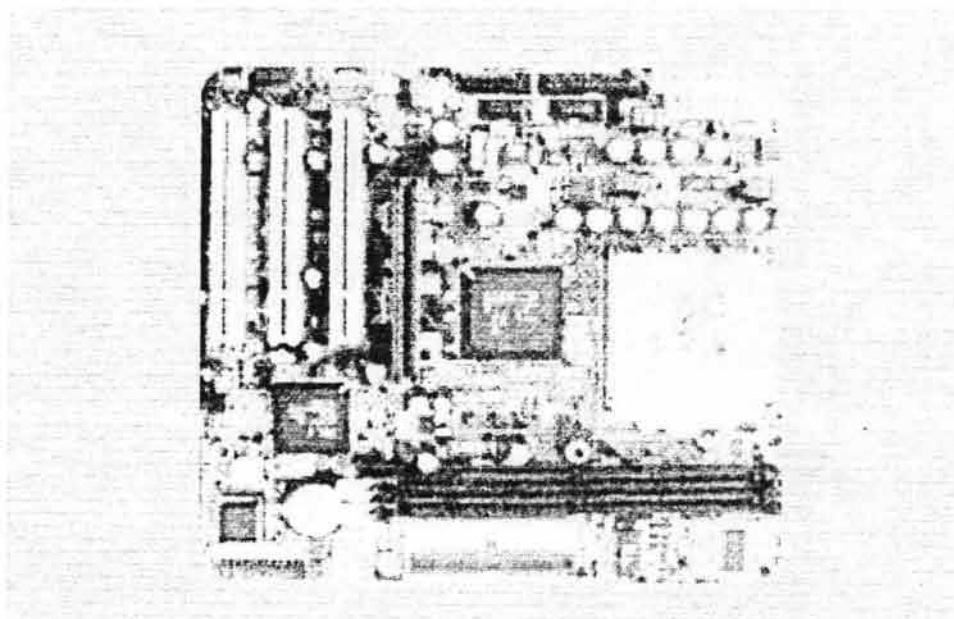
El trabajo está dividido en tres partes, en el capítulo 1, se describen los fundamentos necesarios para conocer el funcionamiento del hardware, haciendo hincapié en la creación de una librería de vínculos dinámicos (.dll) programada en Visual C que nos permite mandar y recibir datos desde el hardware hacia la PC y viceversa.

En el capítulo 2, se muestran los conceptos fundamentales de software involucrados en el diseño del sistema de adquisición, almacenamiento y envío de las señales biomédicas, a través de programación orientada a objetos haciendo uso del lenguaje de programación Visual Basic.

Por último, en el capítulo 3, se exponen los resultados obtenidos al acoplar el hardware, el software y el paquete de comunicación (Cesar.exe) el cual, es de distribución gratuita y de fácil uso.

CAPÍTULO UNO

Conceptos fundamentales referentes al hardware del sistema de adquisición de datos



By JLJT

CAPÍTULO 1.

CONCEPTOS FUNDAMENTALES REFERENTES AL HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS.

1.1 ¿Qué es un sistema de adquisición de datos?

Es un sistema que reúne información; por lo que puede considerarse igualmente como un instrumento. Los sistemas de adquisición de datos consisten en sensores o transductores, transmisores, selectores, convertidores y acondicionadores de señales para recibir variables, convertirlas, procesarlas y documentarlas para análisis y/o control. La realización de un sistema de adquisición de datos se hace a través de un sistema de instrumentación.

El transductor es un elemento que convierte la magnitud física que vamos a medir en una señal de salida (normalmente tensión o corriente) que puede ser procesada por nuestro sistema. Salvo que la señal de entrada sea eléctrica, podemos decir que el transductor es un elemento que convierte energía de un tipo en otro. Por tanto, el transductor debe tomar poca energía del sistema bajo observación, para no alterar la medida.

El acondicionamiento de señal es la etapa encargada de filtrar y adaptar la señal proveniente del transductor a la entrada del convertidor analógico / digital. Esta adaptación suele ser doble y se encarga de:

- Adaptar el rango de salida del transductor al rango de entrada del convertidor. (Normalmente en tensión).
- Acoplar la impedancia de salida de uno con la impedancia de entrada del otro.

La adaptación entre los rangos de salida del convertidor y el de entrada del convertidor tiene como objetivo el aprovechar el margen dinámico del convertidor, de modo que la máxima señal de entrada debe coincidir con la máxima que el convertidor (pero no con la máxima tensión admisible, ya que para ésta entran en funcionamiento las redes de protección que el convertidor lleva integrada).

Los sistemas de instrumentación pueden ser sistemas de medida o sistemas de control de procesos. El sistema de medida tiene como función primaria la presentación o despliegue de una variable física que se está censando. El sistema de control de proceso tiene como función, además de censar las variables críticas del proceso, manipular o accionar variables para el buen desempeño de una planta siguiendo un objetivo o estrategia prescrita.

A continuación observemos los elementos que pueden formar parte de un sistema de adquisición de datos:

1.1.1 Elementos de un sistema de instrumentación ó de adquisición de datos.

Los elementos más importantes en los sistemas de instrumentación ó de adquisición de datos (aunque a veces no todos se utilizan) son: transductores, amplificadores, convertidores, sistemas de despliegue, fuentes de poder, microcontroladores, microprocesadores o microcomputadoras, multicanalizadores (MUX), y demulticanalizadores (DE-MUX). La figura 1.1 muestra los diferentes elementos localizados dentro de un sistema de adquisición de datos:

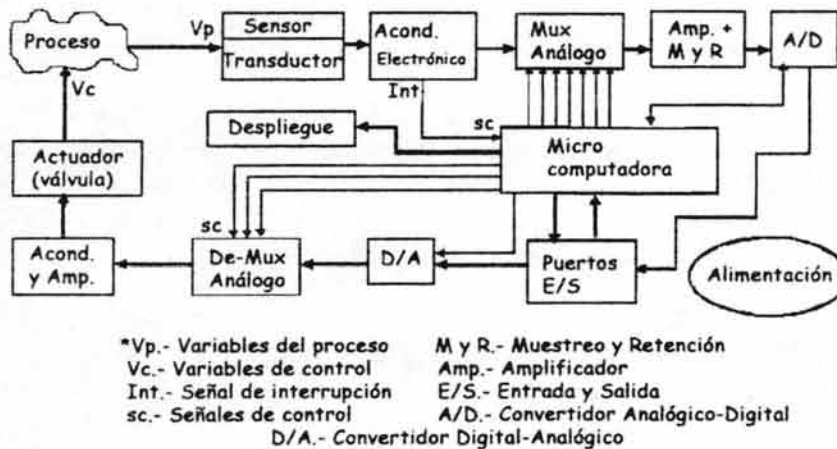


Figura 1.1 Elementos que pueden conformar un sistema de adquisición de datos

Los elementos ilustrados en la figura 1.1 se describen a continuación:

- **Transductores:** Convierten la magnitud a medir a una salida utilizable (por lo general esta salida es una señal eléctrica).
- **Acondicionador:** Convierte la salida del transductor a una magnitud eléctrica adecuada para el sistema de conversión de señales o para el sistema de despliegue de información.
- **Convertidor:** Transforma el modo de la señal a las necesidades específicas del sistema de despliegue o procesamiento.
- **Despliegue o Desplegador:** Ilustración de la información acerca de la magnitud que se mide.
- **Fuente de poder:** Proporciona la energía eléctrica para la operación del sistema completo.

- Microcontrolador, Microprocesador o Microcomputadora: Procesa la información capturada para desplegar las mediciones en forma consistente y fácil de interpretar.
- Multicanalizador(MUX): Sistema selector de entradas.
- DeMulticanalizador(DE-MUX): Sistema decodificador de una salida hacia un número predeterminado de canales.

Para el sistema de adquisición de datos biomédicos desarrollado, el diagrama de flujo es mostrado en la figura 1.2:

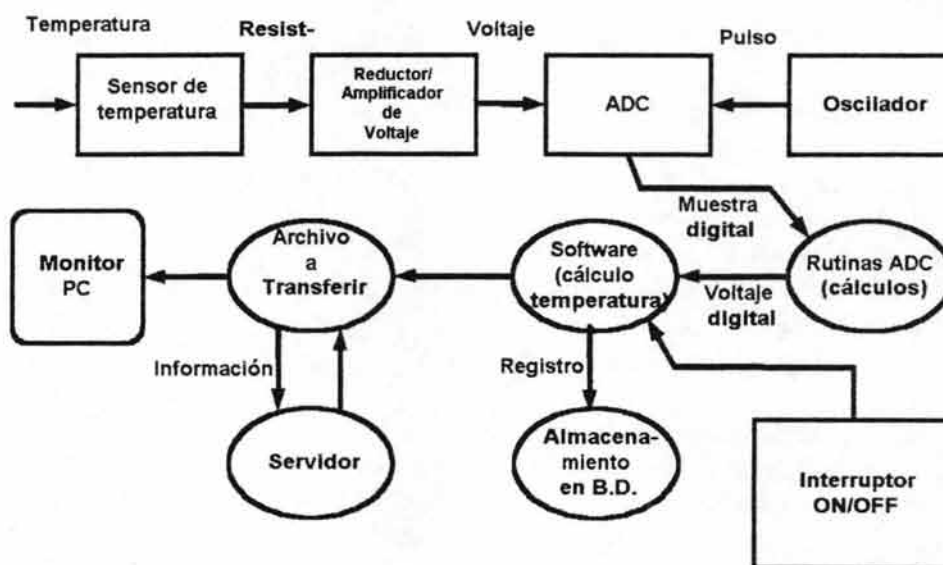


Figura 1.2 Diagrama de flujo del sistema de adquisición de datos desarrollado.

1.1.2 Aplicaciones de los sistemas de Adquisición de Datos.

Dentro de las aplicaciones importantes que se visualizan en la industria se tienen las siguientes:

- A. En control de procesos:
 - a. Control digital directo.
 - b. Control digital supervisorio.
 - c. Control distribuido por computadora.

En instrumentación y medición industrial y/o médica:

- d. Medición y despliegue de información en industrias, comercios y residencias.
- e. Instrumentos de medición para propósitos generales.
- f. Instrumentación médica y de diagnóstico clínico.
- g. Instrumentación y control ambiental de quirófanos en hospitales.

- h. Medición de consumos, abastecimientos y distribuciones de energía eléctrica.
- B. En Telemetría:
 - a. Medición, transmisión, recepción y acondicionamiento de señales provenientes de lugares remotos, lugares peligrosos, satélites y naves espaciales.
 - b. Mediciones en sistemas de transportación, aviones, trenes y embarcaciones.
- C. En Telecomunicaciones:
 - a. Administración de sistemas de comunicaciones.
 - b. Medición en redes locales de comunicación.
 - c. Instrumentación en sistemas de telefonía, microondas y redes computacionales.
- D. En Geofísica:
 - a. Instrumentación en Ingeniería Petrolera.
 - b. Mediciones en Ingeniería Naval.
 - c. Instrumentación sismológica.
 - d. Instrumentación climatológica.
 - e. Instrumentación georeferenciada.
- E. En sistemas Agro tecnológicos:
 - a. Automatización y control de invernaderos.
 - b. Instrumentación y control de riego.
 - c. Medición y control de granjas acuícolas.
- F. En sistemas de Reconocimiento de Formas:
 - a. Instrumentación opto electrónica para reconocimiento de huellas digitales y firmas.
 - b. Visión robótica usando RAM-óptica.
 - c. Medición óptica usando dispositivos CCD.
- G. En Procesamiento Digital de Señales:
 - a. Filtrado adaptativo para análisis de vibraciones y control activo de ruido.
 - b. Filtrado óptimo para estimación y predicción.
 - c. Instrumentación para estimación de espectros de frecuencia en señales de audio o video.

Dentro del estudio y desarrollo de sistemas tecnológicos en el campo electrónico y computacional, se hace necesario satisfacer las necesidades de la industria y la medicina, las mismas que requieren dispositivos que automaticen labores tediosas pero importantes como lo son los datos biomédicos, ayudando a tener una adquisición más rápida y óptima además de ciertas tareas, que de otro modo, se convertirían en actividades monótonas y susceptibles de errores.

En particular, se han desarrollado los sistemas de adquisición de datos para la medición y el procesamiento de variables en búsqueda de un conocimiento más exhaustivo, por parte del hombre, hacia el medio ambiente que le rodea. Además, los sistemas de control de procesos han evolucionado de tal manera que ahora la instrumentación electrónica y los sistemas de adquisición de datos se pueden utilizar de manera eficiente para sintetizar algoritmos y estrategias de control inteligente más complejos.

La instrumentación industrial ha evolucionado tremendamente ya que desde la revolución industrial de principios del siglo XX, pasando por las guerras mundiales, continuando con la guerra fría y hasta la destrucción del muro de Berlín, la tecnología avanzó desde los clásicos sistemas neumáticos y electromecánicos para censado, despliegamiento y control, hasta los modernos sistemas de sensores integrados basados en electrónica analógica, y los microcontroladores para procesamiento y despliegue de información. Los pasos que se dieron al inventar el transistor, el amplificador operacional, la computadora digital y ahora los sensores/transductores integrados con tecnología VLSI (Circuitos Integrados de Muy Alta Integración) marcan la dirección y la rapidez con que han cambiado los sistemas de instrumentación industrial.

1.2 Introducción al Puerto Paralelo.

El puerto paralelo fue desarrollado por la IBM como una manera de conectar la impresora a la computadora. Dado que la firma deseaba emplear impresoras construidas por Centronics, la IBM no quería emplear el mismo puerto que Centronics empleaba en sus impresoras, en vez de eso los ingenieros de la IBM acoplaron un conector de 25-pines DB-25, con uno de 36 de Centronics para crear un cable especial para conectar la impresora a la computadora. Otros fabricantes terminaron por emplear el mismo sistema del interfase de Centronics, de esta manera construyeron una especie de híbrido que se fue estandarizando poco a poco.

Originalmente la especificación del puerto paralelo era unidireccional, esto significa que los datos enviados viajan en una sola dirección (de la computadora hacia el periférico). Con la introducción del PS/2 en 1987, IBM ofreció un diseño de puerto paralelo bidireccional. Este es conocido como Standard Parallel Port (SPP) y ha desplazado completamente a la forma original. La comunicación bidireccional permite recibir datos y también transmitirlos. Muchos aparatos usan ocho pines (de las patas 2-9 del conector) originalmente diseñados para datos. La utilización de ocho pines limita la comunicación a half-duplex. Esto significa que la información sólo puede viajar en una dirección a la vez. Pero los pines del 18 al 25, usados como "tierra", pueden ser usados también como pines de datos.

Esto permite el desarrollo de comunicación full duplex, esto significa que en esta la información se da simultáneamente en dos direcciones.

Intel, Xircom y Zenith realizaron mejoras significativas en el Puerto Paralelo (EPP) en 1991. Entonces el EPP permite ahora transmitir más datos, puede transportar desde 500K a 2

MB por segundo. Este sistema está especialmente diseñado para aparatos que no sean impresoras y que deban pegarse a un puerto paralelo, particularmente se emplea en sistemas de almacenamiento con un alto tránsito de información.

Inmediatamente después de esta mejora, Microsoft y Hewlett Packard anuncian conjuntamente la salida del Extended Capabilities Port (ECP) en 1992. Mientras que el EPP estaba diseñado para otros aparatos, el ECP estaba diseñado para mejorar el trabajo de las impresoras.

En 1994 sale al mercado el IEEE 1284 Standard. Este incluye dos especificaciones para los aparatos del puerto paralelo, EPP y ECP. Para que estos funcionen es necesario un sistema operativo y el aditamento debe contener las especificaciones requeridas. Algunas veces ocurre un problema relativo a esto ya que la mayoría de las computadoras que se venden en la actualidad incluyen SPP, ECP y EPP e inclusive el modo Bidireccional, esta detecta y elige cuál modo es el indicado en determinado momento, esto depende del tipo de sistema que se vaya a utilizar. En su efecto, podemos pasar el modo de la computadora a "manual" y se puede realizar el cambio deseado en el momento en que lo deseemos, aunque claro, necesitamos conocer detalladamente la función específica de cada pin en el puerto.

El puerto paralelo de una típica PC utiliza un conector hembra de tipo D de 25 patitas (DB-25 S), éste es el caso más común, sin embargo es conveniente mencionar los tres tipos de conectores definidos por el estándar IEEE 1284, el primero, llamado 1284 tipo A es un conector hembra de 25 patitas de tipo D.

El segundo conector se llama 1284 tipo B que es un conector de 36 patitas de tipo centronics y lo encontramos en la mayoría de las impresoras; el tercero se denomina 1284 tipo C, se trata de un conector similar al 1284 tipo B pero más pequeño, además se dice que tiene mejores propiedades eléctricas y mecánicas, éste conector es el recomendado para nuevos diseños, aunque no ha sido del todo aprobado, por cuestiones técnicas.

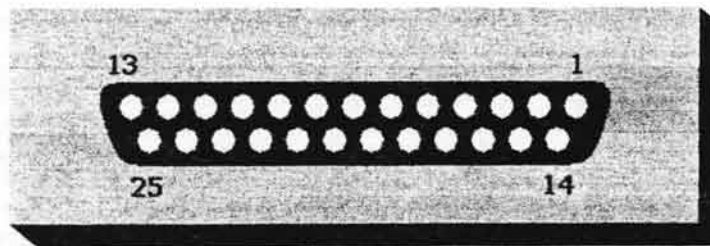


Figura 1.3 Este es el Puerto Paralelo comúnmente encontrado en la parte posterior de una PC comercial.

1.3 Direccionamiento del Puerto Paralelo.

Existen tres direcciones de E/S asociadas con un puerto paralelo de la PC, estas direcciones pertenecen al registro de datos, el registro de estado y el registro de control.

Tabla 1.1. Registro de datos

Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 0	Puerto de datos	Lectura/Escritura	Bit 7	Dato 7
			Bit 6	Dato 6
			Bit 5	Dato 5
			Bit 4	Dato 4
			Bit 3	Dato 3
			Bit 2	Dato 2
			Bit 1	Dato 1
			Bit 0	Dato 0

Tabla 1.2. Registro de estado

Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 1	Puerto de estado	Sólo Lectura	Bit 7	Busy
			Bit 6	Acknowledge*
			Bit 5	Falta de papel
			Bit 4	Select In
			Bit 3	Error*
			Bit 2	IRQ (Not)
			Bit 1	Reservado
			Bit 0	Reservado

Tabla 1.3. Registro de control

Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 2	Puerto de control	Lectura/Escritura	Bit 7	No usado
			Bit 6	No usado
			Bit 5	Permite bidireccional puerto
			Bit 4	Permite IRQ a través de la línea acknowledge
			Bit 3	Selecciona impresora (SELECT IN)*
			Bit 2	Inicializa impresora INIT
			Bit 1	Nueva línea automática(AUTOFEED)*
			Bit 0	Strobo (STROBE)*

El *registro de datos* es un puerto de lectura-escritura de ocho bits. Leer el registro de datos (en la modalidad unidireccional) retorna el último valor escrito en el registro de datos. Los registros de control y estado proveen la interfase a las otras líneas de E/S. La distribución de las diferentes señales para cada uno de los tres registros de un puerto paralelo se puede apreciar en las tablas anteriores.

Nota: los bits que aparecen en las tablas marcados con un asterisco, trabajan con lógica negativa.

Una PC soporta hasta tres puertos paralelo separados, por tanto puede haber hasta tres juegos de registros en un sistema en un momento dado. Existen tres **direcciones base** para el puerto paralelo asociadas con tres posibles puertos paralelo: 0x3BCh, 0x378h y 0x278h, (956, 888, 632 en decimal, respectivamente) nos referimos a éstas como las direcciones base para el puerto **LPT1**, **LPT2** y **LPT3**, respectivamente. El registro de datos se localiza siempre en la dirección base de un puerto paralelo, el registro de estado aparece en la dirección base + 1, y el registro de control aparece en la dirección base + 2. Por ejemplo, para un puerto LPT2 localizado en 0x378h, ésta es la dirección del registro de datos, al registro de estado le corresponde la dirección 0x379h y su respectivo registro de control está en la dirección 0x37Ah. Cuando la PC se enciende el BIOS ejecuta una rutina para determinar el número de puertos presentes en el sistema asignando la etiqueta LPT1 al primer puerto localizado, si existen más puertos entonces se asignarán consecutivamente las etiquetas LPT2 y LPT3.

1.4 Pines y buses del Puerto Paralelo.

Técnicamente el puerto paralelo dispone de 8 líneas digitales de salida (ó de entrada), las cuales llamaremos D0 a D7, cuyos niveles posibles son 0v (para bajo) y 5v (para alto), cumpliendo con el estándar de la familia lógica TTL. Estas líneas están conectadas a los pines 2 al 9 para los bits 0 al 7 respectivamente. Estas ocho líneas conforman el comúnmente llamado bus de datos del puerto. Como ya vimos, existen también dos buses más que permiten ingresar y salir desde y hacia la computadora llamados buses de estado y de control.

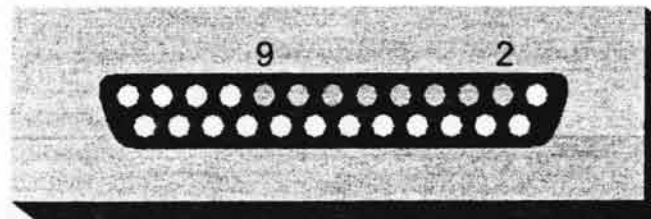


Figura 1.4 Bus de Datos del Puerto Paralelo.

El bus de estado se empleaba originalmente para censar justamente el estado en el que se encontraba la impresora para saber, entre otras cosas, si esta encendida, si esta en línea, si dispone de espacio para recibir datos, si hay papel, etc. Pero puede ser empleado para ingresar datos hacia la computadora desde el mundo exterior, por ejemplo, en nuestro sistema de adquisición de datos. Este bus, a diferencia del de datos, dispone de sólo 5 líneas digitales de entrada (mostrados físicamente) partiendo desde el bit 3 y hasta el bit 7. Por lo general a estas líneas se las denomina S3 a S7 y la asignación de pines se la puede ver en la figura 1.5. La dirección de memoria asignada para comunicarse con este bus es igual a la dirección base (la del bus de datos) mas 1. O sea, si el puerto base es 378, este puerto es 379. Si la base es 278, este es 279. Y, si el puerto base es 3BC, este será 3BD. Un dato a resaltar es que el bit 7 de este puerto (conectado al terminal 11 del conector) presenta un estado lógico inverso. Esto significa que, si en el pin 11 se colocan 5v (nivel lógico alto) el bit al ser leído presentará un cero. Y, por consiguiente, si el pin 11 es puesto a masa (nivel lógico bajo) al leer el bit se verá un uno.

Los nombres de cada pin según su numeración del bus de estado son los siguientes:

Pin 10. - Acknowledge

Pin 11. - Busy.

Pin 12. - Paper out.

Pin 13. - Select out.

Pin 15. - Error.

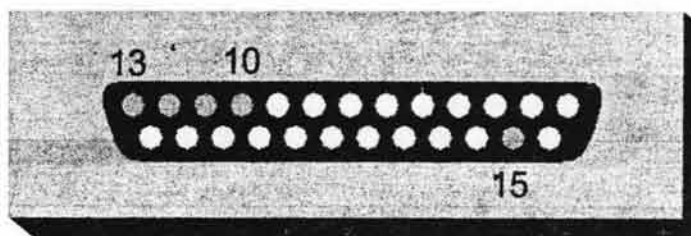


Figura 1.5 Bus de Estado del Puerto Paralelo.

El bus de control, al igual que el de datos, dispone de líneas de salida desde la computadora hacia el exterior. Las líneas también son digitales pero en este caso son solo visibles cuatro, las de menor pero que van del bit 0 al 3. Estas cuatro líneas son generalmente llamadas como C0 a C4 y se pueden controlar por medio de la dirección de memoria base (la del bus de datos) mas 2. De esta forma si la dirección base es 378, este bus es 37A. Si la dirección base es 278, la dirección de este bus será 27A. Y, si la dirección base es 3BC, la dirección de este bus será 3BE. En este bus hay tres bits cuyos niveles lógicos se encuentran invertidos (bit0, bit1 y bit 3) y sólo uno que presenta un estado lógico normal.

Resumiendo podemos decir que un puerto paralelo estándar o SPP permite salir desde la computadora hacia el exterior por medio de 12 líneas o bits y posibilita entrar a la misma por medio de 5 líneas o bits, con la llegada del ECP y EPP si se redirecciona el bus de datos mediante el bit 5 del bus de control (el cuál se manipula internamente ya que físicamente no se encuentra presente en el Puerto Paralelo) se podrá habilitar el bus de datos como de entrada.

En todos los casos cada terminal puede proporcionar un máximo de 1mA de corriente. Superado este miliamperio no se garantiza el nivel lógico TTL de la señal presente e incluso de sobrecargarlo demasiado de puede dañar físicamente el puerto. Una forma de utilizar en forma segura el puerto es conectando a cada terminal de salida un resistor de 100 ohms en serie y luego un buffer octal como el integrado 74LS244. Esto permite, por un lado, aislar el puerto del dispositivo y, por el otro, efectuar una adecuada carga de corriente al sistema.

Los nombres de cada pin según su numeración del bus de control son los siguientes:

Pin 1. - Strobe.

Pin 14. - Auto feed.

Pin 16. - Printed reset.

Pin 17. - Select in.

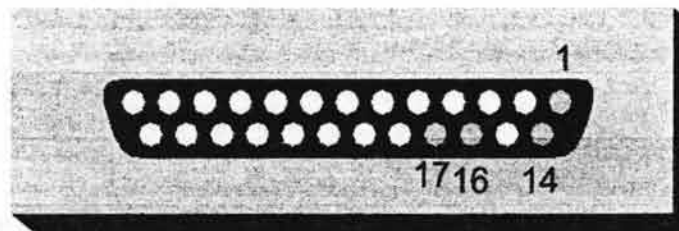


Figura 1.6 Bus de Control del Puerto Paralelo.

1.5 Evolución y tipos de puertos paralelos, la norma IEEE1284.

En 1981, la IBM (International Business Machines) introdujo la Computadora Personal (PC). El puerto paralelo (Standart Parallel Port SPP) estaba incluido en el primer PC y se agregó a éste como una alternativa al bajo rendimiento del puerto serial, para utilizarlo como controlador de las impresoras de matriz de punto de alto desempeño. Este puerto tenía la capacidad de transmitir 8 bits de datos a la vez (del PC a la impresora), mientras que el puerto serial lo hacía de uno en uno. En el momento que el puerto paralelo fue presentado, las impresoras de punto fueron el principal dispositivo externo que se conectó a éste. Al

hacerse extensamente utilizado, el puerto paralelo llegó a ser la respuesta para conectar dispositivos más rápidos.

Después de este inicio, tres grandes grupos de problemas aparecieron a los desarrolladores y usuarios de este puerto: Primero, aunque éste había aumentado su velocidad considerablemente, no había cambio en la arquitectura o desempeño. La máxima velocidad de transferencia alcanzable estaba por los 150 kbyte /seg. y era extremadamente dependiente del software. Segundo, no había un estándar para la interfase eléctrica. Esto causaba muchos problemas cuando se quería garantizar la operación en múltiples plataformas. Por último, la forma de diseño que le dieron, limitaba la distancia de los cables externos hasta un máximo de 1.8 metros. En 1991 hubo una reunión de fabricantes de modo que se pudiera desarrollar un nuevo estándar para el control inteligente de impresoras a través de una red. Estos fabricantes, donde estaban incluidos Lexmark, IBM, Texas Instruments y otros, formaron la Network Printing Alliance (NPA), como una respuesta a estas necesidades.

Desde la introducción del PC al mercado, el puerto paralelo ha sufrido varias modificaciones para hacerlo más veloz. Ya que el puerto original era unidireccional, se creó el puerto bidireccional. El puerto bidireccional fue introducido con el PS/2 compatible. Este permite una comunicación de 8 bits en ambas direcciones. Algo interesante de notar es que el puerto original tenía la posibilidad de ser bidireccional realizando una conexión entre dos pines de un componente electrónico que venía incluido en éste. (Dicho de otro modo, el puerto original es bidireccional en diseño básico, pero no en el diseño operacional). Finalmente se han creado el Enhanced Parallel Port (EPP) y el Extended Capability Port (ECP). Estos dos últimos son al puerto estándar como el Pentium al 286, además de ser bidireccionales.

1.5.1 El estándar IEE 1284-1994.

Este estándar proporciona una comunicación bidireccional de alta velocidad entre un PC y un periférico externo, estableciendo una comunicación entre 50 y 100 veces más rápida que el original puerto paralelo. Por supuesto es totalmente compatible con todos los periféricos existentes para puertos paralelos.

El estándar 1284 define 5 modos de transferencia de datos. Cada modo proporciona un método de transferencia de datos hacia el exterior (PC a periférico), hacia el interior (periférico a PC) o bidireccional (dúplex).

Los modos definidos son:

- Sólo hacia el exterior: modo de compatibilidad "centronics" o modo estándar.
- Sólo hacia el interior.
- Modo nibble, 4 bits a un tiempo empleando las líneas de estado para datos.

- Modo byte, 8 bits a un tiempo empleando líneas de datos, a veces referido como puerto bidireccional. Este modo sólo lo soportan los ordenadores de IBM (PS/2).
- Bidireccional.
- EPP (Enhanced Parallel Port), empleado por periféricos como CD-ROM, cintas, discos duros, adaptadores de red, etc.
- ECP (Extended Capability Port), empleado por la nueva generación de impresoras y scanners.

Todos los puertos paralelos pueden implementar un enlace bidireccional empleando los modos "compatible" y "nibble" para transferencia de datos. El modo byte puede ser empleado por el 25% de los puertos paralelos (aproximadamente). Estos tres modos hacen uso intensivo del software para la transferencia y limitan ésta a ratios de 50 a 100 Kbytes por segundo.

Los modos EPP y ECP están siendo implementados en la mayoría de los últimos controladores de E/S. Estos modos emplean hardware para asistir la transferencia de datos. Por ejemplo, en el modo EPP un byte de datos puede ser transferido al periférico con una simple instrucción OUT. El controlador de E/S se encarga de gestionar toda la transferencia.

En conjunto, el estándar 1284 proporciona lo siguiente:

- 5 modos de operación para transferencia de datos.
- Un método para determinar por parte del periférico y el controlador los modos soportados y negociar el modo requerido.
- Las interfaces físicas (cables y conectores).
- La interfaz eléctrica (conductores, receptores, terminaciones e impedancia).

1.5.2 Modo EPP.

El protocolo de puerto paralelo mejorado (EPP) fue originalmente desarrollado por Intel, Xircom y Zenith Data Systems, como una forma de proporcionar un enlace por puerto paralelo de alto rendimiento que pudiera seguir siendo compatible con el puerto paralelo estándar.

Este protocolo compatible fue implementado por Intel en el chipset 386SL (chip I/O 82360). Esto sucedió antes del establecimiento del comité IEEE 1284 y que los estándares asociados funcionasen.

El protocolo EPP ofrece muchas ventajas a los periféricos que lo utilicen y fue rápidamente adoptado por muchos como un método opcional de transferencia de datos. Una gran asociación de 80 empresas interesadas fue formada para desarrollar y promover el protocolo EPP. Esta asociación se denominó el comité EPP y fue el instrumento empleado para adoptar este protocolo como uno de los modos avanzados del IEEE 1284.

Desde que los primeros puertos con capacidad EPP estuvieron disponibles antes del lanzamiento del estándar 1284, hay una pequeña desviación entre las primeras versiones (pre-1284 EPP) y el protocolo definitivo.

1.5.3 Modo ECP.

Este modo es muy similar al EPP. La mayor diferencia es que emplea compresión de datos mediante algoritmo RLE, por lo que se hace ideal para la comunicación con impresoras láser y scanners.

No entraremos a detallar este modo, ya que no es muy empleado salvo en los mencionados periféricos.

En la tabla 1.4 se muestra información sintetizada de cada uno de estos tipos de puertos.

	SPP	PS/2	EPP	ECP
Fecha de Introducción	1981	1987	1994	1994
Fabricante	IBM	IBM	Intel, Xircom y Zenith Systems	Hewlett Packard y Microsoft
Bidireccional	No	Si	Si	Si
DMA	No	No	No	Si
Velocidad	150 Kbyte/seg.	150 Kbytes/seg.	2 Mbytes/seg.	2 Mbytes/seg.

Tabla 1.4 Evolución de los modos de funcionamiento del puerto paralelo

1.6 El conector Centronics.

En la actualidad, muchos circuitos robóticos y sistemas en general, usan el PC como medio sencillo y flexible para programarlos de manera rápida y segura o como control y procesamiento del sistema, evitando circuitería complementaria.

Dos medios para conseguir esta conexión de un PC con un dispositivo externo es mediante el puerto RS232 (comúnmente conocido como "serie") o el puerto paralelo Centronics.

Patillaje del conector Centronics.

La relación de los pines es la siguiente (* significa lógica negativa):

1 - STROBE *	10 - ACK*
2 - D0	11 - BUSY*
3 - D1	12 - PE
4 - D2	13 - SLCT
5 - D3	14 - AUTO FEED *
6 - D4	15 - ERROR *
7 - D5	16 - INIT *
8 - D6	17 - SLCT IN
9 - D7	18 a 25 - GND

Ahora bien, todos estos pines no se controlan con un mismo registro de datos sino que un mismo puerto Centronics, (LPT1, LPT2, etc...) como ya se observó, tiene 3 registros para poder ser controlado; a saber: Enviar Datos, Inicializar Dispositivo y Entrar Datos. Para acceder a los tres registros de cada puerto, nos referiremos a cada registro con una dirección de memoria implícita que fija la BIOS al arrancar el ordenador. Sería conveniente disponer de un algoritmo que nos informase de la dirección de memoria de cada uno de los tres registros, ya que cada máquina es distinta de otra.

1.7 Interrupción del Puerto paralelo.

Entre las diferentes interrupciones de las que dispone la BIOS para la llamada de sus servicios, la interrupción 17h está exclusivamente reservada para la comunicación con el puerto paralelo. A pesar de que a este puerto se pueden conectar teóricamente otros periféricos, que no sólo la impresora, en referencia a la interrupción 17h se habla frecuentemente de la interrupción BIOS de la impresora, ya que en la práctica no se pueden encontrar otros periféricos en este puerto. Los conceptos "Impresora" y "Puerto paralelo" en este sentido eran intercambiables.

En la PC se pueden conectar un máximo de tres puertos paralelos diferentes. Se pueden utilizar mediante tres funciones distintas, que la BIOS pone a nuestra disposición a través de la interrupción 17h. Sin embargo no se debe crear la impresión, de que cada una de estas funciones esté relacionada directamente con uno de los tres posibles puertos. Ya que en realidad las tres funciones se ocupan de tareas muy distintas y variadas en el marco de las comunicaciones con un puerto paralelo, y pueden utilizarse, según se desee, en cada uno de los tres puertos paralelos.

Se pueden conectar hasta tres puertos paralelos al PC sin ningún problema, ya que hay reservadas tres zonas en el espacio de direcciones de I/O (entrada/salida) del PC para puertos paralelos:

3BCh- 3BFh.- puerto paralelo en la tarjeta MDA

378h-:37Fh.- primer puerto paralelo

278h-27Fn.- segundo puerto paralelo

Las direcciones de Port de los puertos de la tabla anterior no se listan en orden numérico ascendente intencionadamente, sino que se indicaron en el orden en el que la BIOS busca los supuestos puertos al arrancar el sistema. Ya que la BIOS primero ha de analizar los supuestos (decimos supuestos, por que al principio la BIOS no sabe si dispone de ellos) puertos, para poder determinar cuales están presentes. Y del orden en que ocurre esto depende de qué puerto se convierte en LPT1, en LPT2 y en LPT3.

Primero, la BIOS se fija en la zona de direcciones 3BCh - 3BFh. Es una parte de una zona de direcciones mayores, que va desde 3B0h hasta 3BFh y que está reservada para una tarjeta de vídeo monocroma (MDA) o una tarjeta gráfica Hércules. Ya que hasta bien entrados los años ochenta la mayoría de PC se suministraba con este tipo de tarjetas, que aparte de la lógica correspondiente al vídeo también albergaban un puerto paralelo.

Si se descubre una de estas tarjetas con su puerto paralelo, el puerto es direccionado por la BIOS como LPT1. La siguiente tarjeta aparece automáticamente como LPT2. Si la búsqueda de una tarjeta gráfica monocroma con puerto paralelo fracasa, el siguiente puerto paralelo descubierto será direccionado con LPT1.

Las dos zonas de direcciones que hay aparte de la tarjeta de pantalla monocroma están reservadas para puertos paralelos independientes. Teniendo en cuenta la cada vez mayor miniaturización, hoy en día ya no se puede partir de que realmente se está tratando con dos tarjetas de expansión. Por que estos puertos de comunicación ya no sólo se pueden colocar en la placa madre, sino incluso dentro del chip del procesador, hacia donde derivan las tendencias en el ámbito de notebooks y palmtops.

Independientemente de cómo estén realizados los puertos, la BIOS analiza en cualquier caso durante su inicialización los Ports enumerados, para descubrir todos los puertos paralelos instalados. Esto sin embargo puede llevar a resultado inesperados: Si por ejemplo sólo se ha instalado un puerto paralelo, pero que ocupa la zona de direcciones que en realidad está reservado para el segundo, se direccionará como LPT1.

Selección de LPT1 hasta LPT3.

La selección de los diferentes puertos paralelos como LPT1, LPT2 y LPT3 se realiza colocando sus direcciones de base en las variables del segmento de variables de la BIOS.

Allí se encuentra, a partir de la dirección de Offset 0008h un array de cuatro palabras, en las que se guardan las direcciones de Port de los puertos paralelos.

0040:0008h.- Dirección Base LPT1

0040:000Ah.- Dirección Base LPT2

0040:000Ch.- Dirección Base LPT3

0040:000Eh.- Dirección Base LPT4

A pesar de que la BIOS al arrancar sólo busca tres puertos paralelos, en la zona de variables aparentemente queda espacio para cuatro puertos. Y realmente, las funciones de la BIOS también se pueden utilizar en relación a un cuarto puerto paralelo, poniendo simplemente su dirección de base (a mano) en el segmento de variables de la BIOS, y al llamar la función de la BIOS se especifica como número de puerto un 3.

Por cierto la terminología LPT1 no viene de la BIOS, que simplemente direcciona los puertos paralelos con los números 0 hasta 3. Esta denominación se introdujo con el DOS, que identifica la denominación de dispositivo LPT1, LPT2 y LPT3 con los números de Puerto paralelo 0 hasta 2. Desde el punto de vista del DOS, sin embargo no existe el LPT4.

1.8 El Convertidor Analógico a Digital ADC0808.

El convertidor Analógico a Digital ADC0808 es un dispositivo con un convertidor analógico a digital de 8 bits, está formado por un DAC (Convertidor Digital a Analógico), un registro de Aproximaciones Sucesivas y un comparador.

Algunas características importantes del convertidor son las siguientes:

- Fácil interfase con todos los microprocesadores.
- Opera con 5v. fijos de tensión.
- No requiere ajustes de calibración externos.
- 8 canales multiplexados con lógica de dirección.
- Rango analógico de entrada de 0v a 5v.
- Salidas digitales con las especificaciones de nivel TTL.
- Resolución de 8 bits.
- Tiempo por conversión 100µs.

Para que puedan ser transmitidas las entradas analógicas de un ADC y obtener salidas digitales es necesario introducir señales de control hacia éste. Esto es necesario porque los ADC's requieren de un pulso de reloj y en este caso, explícitamente, el ADC0808 contiene lógica de control (además de un multiplexor) así como comparador y registros.

Para conseguir que trabaje un ADC0808 éstas son las señales que se deben enviar. Para todas las señales de control es necesario tener un voltaje de entrada de -0.3v. hasta 1.5v. para un “bajo” y > 1.5v. hasta 15v. para un “alto” Los pines y funcionamiento de cada una de éstos son los siguientes:

1.8.1 Líneas de dirección (A, B, C)

Puesto que el dispositivo tiene un multiplexor de 8 canales existen tres líneas de dirección selectoras: A, B, C. C es el bit más significativo (MSB) y A es el bit menos significativo (LSB).

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Figura 1.7 Líneas de selección del ADC0808

1.8.2 Cerrojo habilitador de dirección (Address Latch Enable ALE)

ALE es requerido para cargar la línea de dirección seleccionada en el ADC. Una vez cargada el multiplexor envía el canal apropiado al convertor en el dispositivo. La ALE debe ser pulsada por al menos 100ns. para que las direcciones se carguen apropiadamente.

1.8.3 Pulso de Reloj (Clock)

La señal de reloj es requerida para el ciclo a través de la etapa de comparación para realizar la conversión. Hay 8 periodos de ciclo de reloj que se requieren para completar una conversión entera. Esto significa que una conversión entera toma 64 ciclos de reloj por lo menos. El reloj también deberá adaptarse al mismo rango como todas las demás señales de control. La máxima frecuencia de reloj a la que puede trabajar el convertor es de 1.28 Mhz y la mínima de 10 Khz., para poder operar el convertidor en modo de corrida libre es necesario que la frecuencia no sobrepase de los 500KHz, esto es porque el reloj acelera a una velocidad mayor y entonces no pasará suficiente tiempo para que la dirección elegida sea cargada correctamente antes de iniciar una conversión.

1.8.4 Inicio (Start)

El propósito de la señal Start es doble, en el lado creciente del pulso los registros internos son limpiados y en el lado descendiente del pulso la conversión es iniciada; el ancho del pulso ALE debe ser de por lo menos 100ns. para que el canal que se desea habilitar sea cargado apropiadamente.

1.8.5 Habilitador de salida (Output Enable OE)

La señal OE permite leer la salida actual de valores del ADC en las líneas de salida digitales. El ADC entrega los valores en las líneas de salida digitales hasta que otra conversión es iniciada.

El convertidor proporciona una señal en cuánto se completa una conversión:

1.8.6 Señal de fin de conversión (End Of Conversion EOC)

La señal EOC es enviada por el ADC, cuando nos envía un “alto” significa que la conversión se ha realizado satisfactoriamente.

1.8.7 Entradas analógicas IN0 – IN7

Son los 8 posibles datos analógicos de entrada. Si alguno no va a ser utilizado deberá ser conectado a tierra.

1.8.8 Salidas digitales.- $2^1 - 2^8$

Son los bits digitales obtenidos de la conversión. El bit más significativo (MSB) es el 2^8 y el menos significativo (LSB) es el 2^1 .

1.8.9 Vcc

Entrada de alimentación del dispositivo el rango permitido es desde 4.5v. hasta 6.0v DC.

1.8.10 Vref (+). Voltaje de referencia positivo

Es el máximo voltaje que se permitirá pensar en las entradas analógicas; si esta cantidad es recibida en cualquiera de los canales analógicos la salida obtenida digital será “11111111” el valor máximo permitido para Vref(+) es: $V_{cc} + 0.1v$.

1.8.11 GND

La patilla GND corresponde a la tierra del convertidor.

Es posible conectar juntas, ésta patilla con la patilla Vref (-), ya que con esto le estamos diciendo al convertidor, que el mínimo voltaje analógico de entrada permitido será de cero voltios.

1.8.12 Vref (-) Voltaje de referencia negativo

Es la cantidad más baja del voltaje de referencia, es el mínimo voltaje permitido de entrada en cualquiera de los canales analógicos, si esta cantidad es recibida la salida digital será de: "00000000", el valor mínimo es de -0.1v

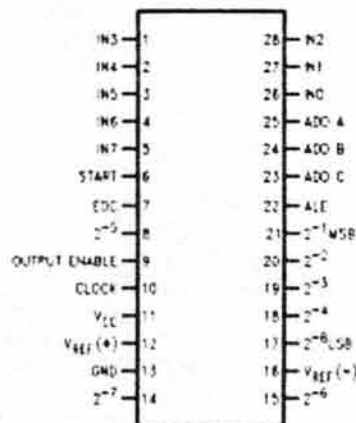


Figura 1.8 El Convertidor Analógico a Digital ADC0808.

1.9 Modo de funcionamiento del ADC0808.

El modo en que trabaja el convertidor Analógico a Digital ADC0808 es quizá el más comúnmente utilizado, se trata de una conversión basada en Aproximaciones sucesivas. La ventaja principal de éste método con respecto a los demás es precisamente el tiempo de conversión mucho menor que los demás métodos de conversión. Su funcionamiento básico es el siguiente:

Los bits del DAC se habilitan (se ponen a 1) uno por vez, comenzando por el bit menos significativo (MSB). Cada vez que se habilita un bit el comparador produce una salida que indica, si la tensión analógica de entrada es mayor ó menor que la salida del DAC.

Si la salida del DAC es mayor que la entrada analógica recibida, la salida del comparador estará a nivel bajo, pasando a bajo (cero) al bit del registro. Por el contrario, si la salida es menor que la entrada analógica recibida, se mantendrá un 1 (nivel alto) en el registro.

El convertidor realizará esta labor con el MSB, posteriormente con el siguiente bit más significativo y así sucesivamente. Al concluir con todos los bits del DAC, el ciclo de conversión estará completo.

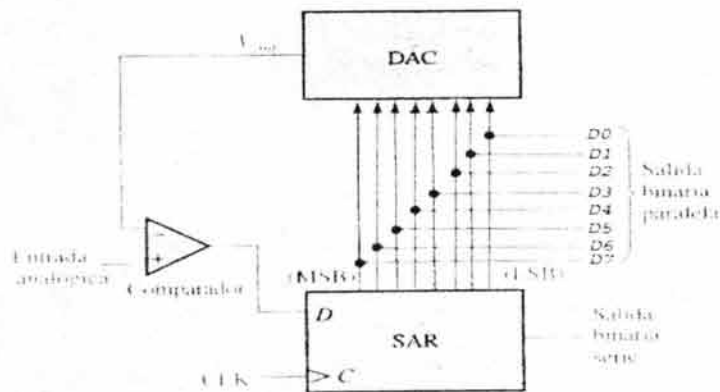


Figura 1.9 Diagrama a bloques del ADC0808.

Observemos con mayor detalle este tipo de funcionamiento:

El convertidor en su DAC tiene una entrega de voltaje dependiendo de la entrada del mismo, para ejemplificar el funcionamiento del convertidor, consideremos los siguientes datos supuestos:

Primeramente consideremos que el convertidor es de 4 bits (ya que se repite la misma operación con los otros 4 bits restantes). El convertidor recibe una tensión analógica de entrada constante de 5v., y supongamos que el DAC tiene la siguiente característica de salida: $V_{out} = 8v.$ para el bit 2^3 (MSB), $V_{out} = 4v.$ para el bit 2^2 , $V_{out} = 2v.$ para el bit 2^1 y $V_{out} = 1v.$ para el bit 2^0 (LSB).

La figura 1.10 muestra el primer paso del ciclo de conversión con el MSB = 1. La salida del DAC es de 8V. Puesto que es mayor que la entrada analógica de 5V., la salida del comparador estará a nivel bajo, lo que hace que el bit se mantenga en el SAR.

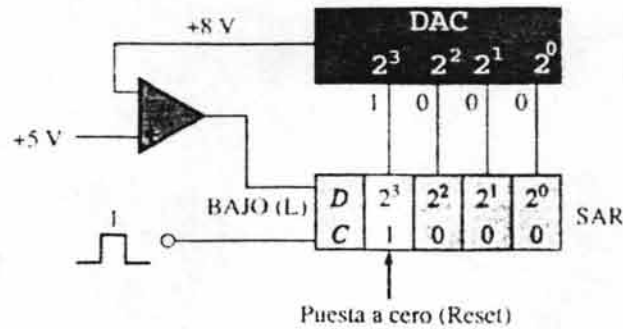


Figura 1.10 Funcionamiento del ADC0808 (primer paso)

La figura 1.11 muestra el segundo paso del ciclo de conversión, con el bit 2^2 igual a 1. La salida del DAC es 4V. Puesto que es menor que la entrada analógica de 5V, la salida del comparador conmuta a nivel alto, lo que hace que este bit se mantenga en el SAR.

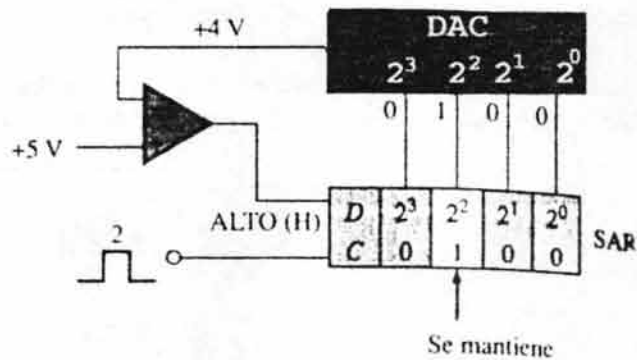


Figura 1.11 Funcionamiento del ADC0808 (segundo paso)

La figura 1.12 muestra el tercer paso del ciclo de conversión, con el bit 2^1 igual a 1. La salida del DAC es 6V, ya que los bits de entrada 2^2 y 2^1 están a 1; ($4V + 2V = 6V$). Puesto que es mayor que la entrada analógica de 5V, la salida del comparador conmuta a nivel bajo, lo que hace que este bit se ponga a cero.

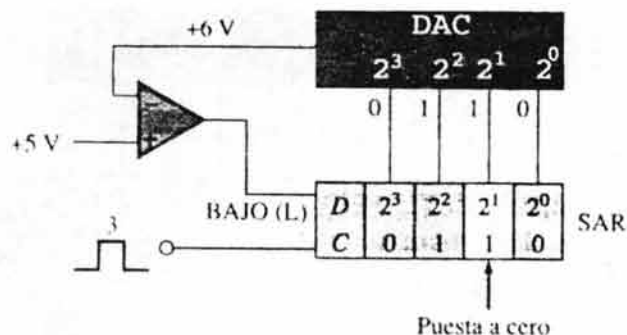


Figura 1.12 Funcionamiento del ADC0808 (tercer paso)

La figura 1.13 muestra el cuarto y último paso del ciclo de conversión, con el bit 2^0 igual a 1. La salida del DAC es igual a 5V, ya que los bits de entrada 2^1 y 2^0 están a 1; ($4V. + 1V = 5V.$), lo que hace que el LSB se mantenga en el SAR.

Este modo de trabajo del ADC0808 (aproximaciones sucesivas), es uno de los modos de operación más rápidos, pues sólo esta por debajo del método flash de operación, la desventaja principal de este último es que los componentes son relativamente más costosos y resulta un poco más difícil el conseguir este tipo de convertidores ya que actualmente no son muy comerciales en el mercado.

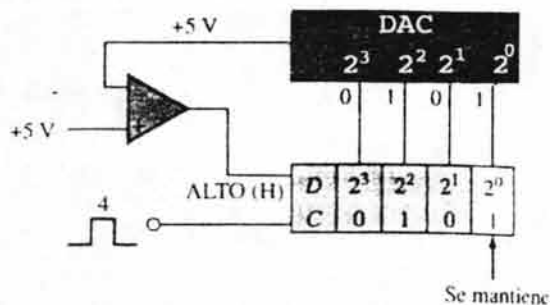


Figura 1.13 Funcionamiento del ADC0808 (último paso)

Se han probado los cuatro bits y el ciclo de conversión se ha completado. En este momento, el código binario almacenado en el registro es: 0101, que es el valor binario de la entrada analógica de 5V. A continuación, se inicia un nuevo ciclo de conversión y el proceso se repite. El SAR se borra al comienzo de cada nuevo ciclo.

1.10 Voltaje de Referencia y pulso de Reloj del ADC0808.

El convertidor analógico a digital ADC0808 puede ser fácilmente usado por un microprocesador en un ambiente controlado. Muchos medios sofisticados a través de aplicaciones pueden ser manipulados con un mínimo de hardware, y tal vez, adicionalmente, algún tipo de software para complementar este control exacto del dispositivo.

Todos y cada uno de los pines del ADC0808 son importantes, en cuánto a lo que se refiere a alambrado, ya que cada uno de ellos desempeñan alguna labor necesaria dentro de la aplicación que se este desarrollando. En especial el voltaje de referencia juega un papel muy importante dentro del dispositivo, pues es el que nos marcará la pauta del rango de trabajo a utilizar, en el ADC0808 tenemos dos tipos de voltaje de referencia, el voltaje de referencia (+) y el voltaje de referencia (-). El voltaje de referencia (+) se utiliza para determinar cuál es el valor máximo que esta en disponibilidad de recibir el convertidor analógico a digital, mientras que el voltaje de referencia (-) es el voltaje mínimo que podrá recibir el convertidor.

Como ya vimos, para que el convertidor analógico a digital trabaje correctamente es necesario un voltaje de referencia tanto positivo como negativo $V_{ref} (+)$ y $V_{ref} (-)$. Con lo que respecta al $V_{ref} (-)$ basta conectarlo a tierra (GND) para que el voltaje mínimo a recibir sea de 0v. Pero hablando del $V_{ref} (+)$ es necesario el uso del transistor LM385, ya que este dispositivo es capaz de manejar el voltaje y corriente específicos que vamos a necesitar para establecer algún rango determinado.

Con lo que respecta al pulso de reloj, nos podemos valer del timer LM555, el cuál tiene infinidad de usos, en este caso lo utilizaremos como oscilador para proporcionar el pulso de reloj que necesita el ADC0808.

1.11 El LM385 variable.

Este es un micro-dispositivo que proporciona voltajes de referencia a base de diodos. La mayor ventaja al utilizar este dispositivo es que su entrega de corriente es proporcional a la cantidad de tensión manejada, dos características muy importantes del LM385 variable, son las siguientes:

- Ajustable desde 1.24v. hasta 5.30 v.
- Corriente de operación de $10\mu\text{A}$ a 20mA.



Figura 1.14 El LM385 variable para suministrar pequeños voltajes de referencia

Utilizamos específicamente el LM385 variable en presentación plástica, para tener opción a elegir un voltaje de referencia distinto y por consecuencia otro rango de trabajo. El esquema de alambrado del funcionamiento de este dispositivo y demás elementos que utiliza, se verá en detalle en el capítulo 3 en el tema 3.9: “Análisis del sistema de adquisición de datos”, en donde se presenta el diagrama entero y una explicación detallada de cómo funciona cada uno de los elementos involucrados.

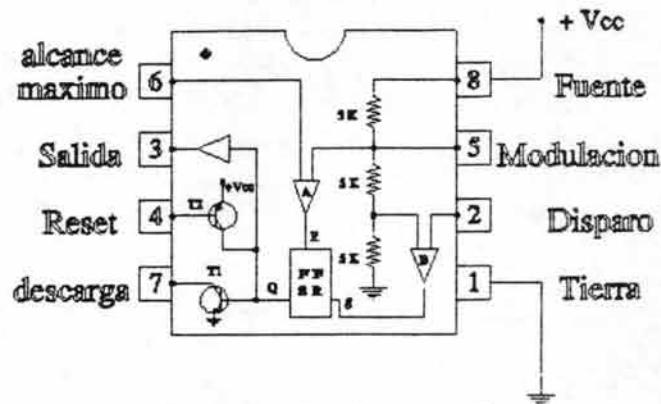
Para mayor información consultar el anexo 5.

1.12 El temporizador LM555.

Jack Kilby ingeniero de Texas Instrument en el año de 1950 se las ingenió para darle vida al primer circuito integrado, una compuerta lógica, desde entonces y hasta nuestros tiempos han aparecido innumerables circuitos integrados, en Julio de 1972, apareció en la fábrica de circuitos integrados SIGNETICS CORP., un microcircuito de tiempo el NE555V, inventado por el grupo que dirigió el Jefe de Producción en ese tiempo, Gene Hanateck, este integrado se puede aplicar a diversas aplicaciones, tales como:

- Control de sistemas secuenciales,
- Generación de tiempos de retraso,
- Divisor de frecuencias,
- Modulación por anchura de pulsos,
- Repetición de pulsos,
- Generación de pulsos controlados por tensión, etc.

Además de ser tan versátil contiene una precisión aceptable para la mayoría de los circuitos que requieren controlar el tiempo, su funcionamiento depende únicamente de los componentes pasivos externos que se le interconectan al microcircuito 555.



El Circuito de Tiempo 555

Figura 1.15 Diagrama interno del 555

Descripción del Timer 555.

Se alimenta de una fuente externa conectada entre sus terminales (8) positiva y (1) tierra; el valor de la fuente de alimentación se extiende desde 4.5 Volts hasta 18.0 Volts de corriente continua, la misma fuente exterior se conecta a un circuito pasivo RC exterior, que proporciona por medio de la descarga de su Capacitor una señal de voltaje que esta en función del tiempo, esta señal de tensión es de $1/3$ de V_{cc} y se compara contra el voltaje aplicado externamente sobre la terminal (2) que es la entrada de un comparador como se puede apreciar en la gráfica anterior.

La terminal (6) se ofrece como la entrada de otro comparador, en la cual se compara a $2/3$ de la V_{cc} contra la amplitud de señal externa que le sirve de disparo.

La terminal (5) se dispone para producir (PAM) modulación por anchura de pulsos, la descarga del condensador exterior se hace por medio de la terminal (7), se descarga cuando el transistor (NPN) T1, se encuentra en saturación, se puede descargar prematuramente el capacitor por medio de la polarización del transistor (PNP) T2.

Se dispone de la base de T2 en la terminal (4) del circuito integrado 555, si no se desea descargar antes de que se termine el periodo, esta terminal debe conectarse directamente a V_{cc} , con esto se logra mantener cortado al transistor T2 de otro modo se puede poner a cero la salida involuntariamente, aun cuando no se desee.

La salida esta provista en la terminal (3) del microcircuito y es además la salida de un amplificador de corriente (buffer), este hecho le da más versatilidad al circuito de tiempo 555, ya que la corriente máxima que se puede obtener cuando la terminal (3) sea conecta directamente al nivel de tierra es de 200 mA.

La salida del comparador "A" y la salida del comparador "B" están conectadas al Reset y Set del FF tipo SR respectivamente, la salida del FF-SR actúa como señal de entrada para el amplificador de corriente (Buffer), mientras que en la terminal (6) el nivel de tensión sea más pequeño que el nivel de voltaje contra el que se compara la entrada reset del FF-SR no se activará, por otra parte mientras que el nivel de tensión presente en la terminal 2 sea más grande que el nivel de tensión contra el que se compara la entrada Set del FF-SR no se activará.

El microcircuito 555 es un circuito de tiempo que tiene las siguientes características:

- La corriente máxima de salida es de 200 mA cuando la terminal (3) de salida se encuentra conectada directamente a tierra.
- Los retardos de tiempo de ascenso y descenso son idénticos y tienen un valor de 100 nseg.
- La fuente de alimentación puede tener un rango que va desde 4.5 Volts hasta 16 Volts de CD.
- Los valores de las resistencias R1 y R2 conectadas exteriormente van desde 1 ohm hasta 100 kohms para obtener un corrimiento de temperatura de 0.5% a 1% de error en la precisión, el valor máximo a utilizarse en la suma de las dos resistencias es de 20 Mohms.
- El valor del Capacitor externo contiene únicamente las limitaciones proporcionadas por su fabricante.
- La temperatura máxima que soporta cuando se están soldando sus terminales es de 330 centígrados durante 19 segundos.
- La disipación de potencia o transferencia de energía que se pierde en la terminal de salida por medio de calor es de 600 mW.

Las aplicaciones del 555 son tan numerosas que prácticamente no existe un sistema electrónico que no lo utilice de alguna forma. Su versatilidad, bajo costo y sencillez de uso lo hacen imprescindible en muchos casos. Además sobre el se ha escrito mucha literatura y existen cientos de libros, artículos y documentos sobre sus aplicaciones reales y potenciales.

El circuito integrado 555 en su presentación usual de cápsula plástica dispone de 8 pines. Puede estar etiquetado bajo distintos nombres o referencias dependiendo del fabricante (NE555, μ A555, LM555, SN72555, XR-555, CA555, HA1755, NC1455, TA7555P, ECG955, etc.). También se consigue en otras presentaciones incluyendo cápsulas metálicas para aplicaciones de montaje superficial (SMT).

El 555 convencional consta internamente de 23 transistores, 2 diodos y 12 resistencias de las cuales 3 son de 5 Kohms por eso tiene el nombre de 555. Este circuito tiene una

capacidad suficiente para impulsar directamente leds, zumbadores, bobinas de relé, parlantes, piezoeléctricos y otros componentes, además, es directamente compatible con circuitos integrados digitales estándares, que es otra de sus grandes ventajas.

El 555 puede ser configurado para operar en 2 modos básicos: como multivibrador Astable y como multivibrador Monoastable.

- En el modo astable actúa como un oscilador de relajación, generando un tren de pulsos simétricos o asimétricos.
- En el modo monoastable actúa como un temporizador, generando un pulso de determinada duración en respuesta a una orden de disparo.

En ambos casos, produce señales positivas con una amplitud próxima o igual al voltaje de alimentación. La frecuencia, el ciclo de trabajo y la duración o ancho de los pulsos se programan externamente mediante resistencias y condensadores de valor adecuado.

1.13 El integrado DM74LS73AN.

Este dispositivo contiene dos flip-flops independientes tipo J-K con salidas complementarias:

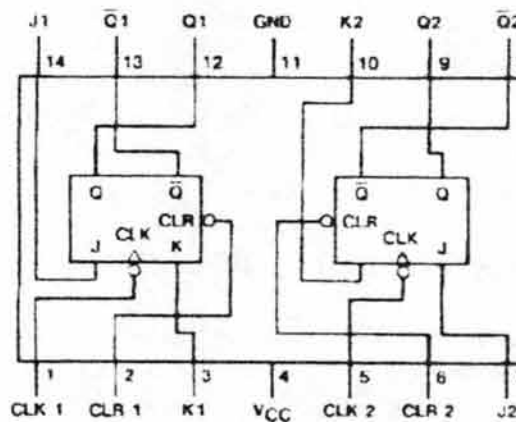


Figura 1.16 Diagrama de conexión del DM74LS73AN.

La finalidad de utilizar este dispositivo es la de proporcionar una señal de reloj al ADC0808 lo más exacta y cuadrada posible, configurado de la siguiente manera:

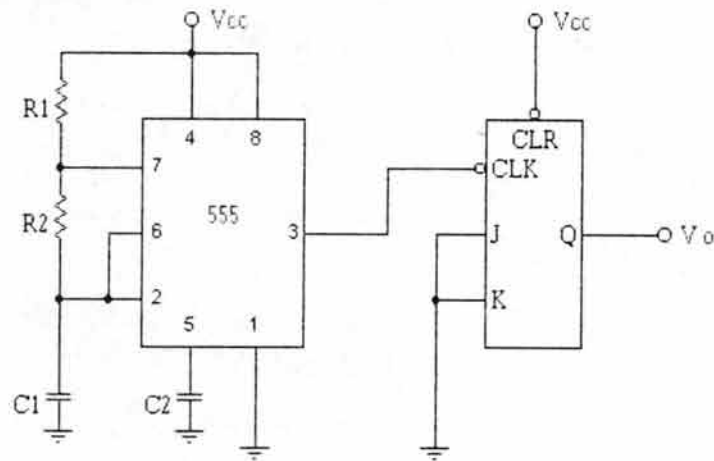


Figura 1.17 Configuración del 555 y el DM74LS73AN para proporcionar un pulso de salida cuadrado-exacto.

Donde el pulso entregado por el 555 (patilla 3) entrará al CLK en uno de los flip-flops del DM74LS73A, las patillas JK se mantendrán unidas, y sólo restará alimentar el dispositivo. El dispositivo opera con 5 voltios de corriente directa.

La señal cuadrada y exacta de salida se obtendrá en Q y sólo hay que hacer hincapié en que la frecuencia que proviene del 555 se verá recortada a la mitad al pasar por el DM74LS73A. Por ejemplo, si obtenemos del 555 una frecuencia de 800 KHz., al pasar por el flip-flop la frecuencia obtenida a la salida de Q será aproximadamente de 400 KHz.

Para más información consultar anexos 6 y 7.

1.14 El sensor de temperatura LM35.

Este dispositivo entrega una salida de corriente linealmente proporcional a la temperatura que este recibiendo (en escala Centígrada), de esta manera tiene la ventaja de que no se realizará ninguna conversión adicional, ya que otros sensores trabajan en otra escala (Kelvin) y entonces se tendría que buscar una forma para convertir esos grados obtenidos a la escala Centígrada que es la escala que conocemos y trabajamos con mayor frecuencia.

El LM35 no requiere ningún tipo de calibración externa y ofrece una exactitud de $\pm 1/4^{\circ}\text{C}$ a temperatura encerrada y $\pm 3/4^{\circ}\text{C}$ cerca del rango de -55 a $+150^{\circ}\text{C}$.

Existen otras presentaciones del sensor como son: el TO-46 en presentación metálica, el SO-8 en presentación de circuito integrado, el TO-92 en presentación plástica tipo transistor, y el TO-220 con disipador integrado. Algunas características del sensor son las siguientes:

- Calibrado directamente en escala centígrada.
- Factor de escala lineal 10.0 mV/°C
- 0.5°C de exactitud garantizada (a 25 °C)
- Puede operar en el rango de -55 a +150 °C
- Adecuado para aplicaciones remotas.
- Bajo costo.
- Opera desde 4 hasta 30 V.
- Bajo sobrecalentamiento.

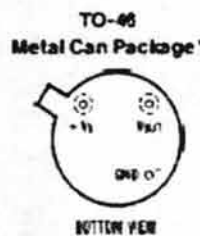


Figura 1.18 El LM35 en presentación metálica

1.15 Aplicaciones del LM35

Los LM35 pueden ser aplicados fácilmente de la misma manera como otros sensores de temperatura de circuito-integrado. Puede ser pegado o cementarse a una superficie y su temperatura será aproximadamente de 0.01 Grados Centígrados de la temperatura de la superficie.

Esto presume que la temperatura ambiente del aire es casi la misma que la temperatura de la superficie; si la temperatura del aire fuera mucho más alta o más baja que la temperatura de la superficie, la temperatura real del LM35 estaría dada en una temperatura intermedia entre la temperatura de la superficie y la temperatura del aire. Esto es particularmente cierto para el TO-92 de paquete plástico, donde los conductores de cobre son el principal camino térmico para llevar calor al dispositivo, para que su temperatura pudiera ser lo mas aproximada a la temperatura del aire con la temperatura de la superficie.

Para minimizar este problema, hay que estar seguro que el cableado del LM35, cuando deja el dispositivo, se tiene la misma temperatura como la superficie de interés. La manera más fácil de hacer esto es cubriendo estos alambres con una gota de epoxy el cual asegurará que los conductores y alambres están todos a la misma temperatura como la superficie, y que la temperatura del LM35 no será afectada por la temperatura del aire.

El TO-46 de presentación metálica también puede soldarse a una superficie de metal o conductor sin daño. Por supuesto, en ese caso el Voltaje terminal del circuito se conectará con tierra a ese metal.

Alternativamente, los LM35 pueden montarse dentro de un tubo de metal sellado al final, sumergirse entonces en un baño o puede atornillarse en un agujero enhebrado en un tanque. Como con cualquier C.I., los LM35 y el alambrado acompañante y demás circuitos deben aislarse y estar secos, para evitar goteo y corrosión. Esto es especialmente cierto si el circuito puede operar a temperaturas frías donde la condensación pueda ocurrir. Barnice el circuito impreso con barnices tales como el Epoxy esta sustancia es utilizada para prevenir que la humedad no corra a los LM35 o sus respectivas conexiones.

Estos dispositivos a veces se sueldan a un pequeño aleta de calor de luz-peso, para disminuir el tiempo de temperatura constante, y acelera la respuesta en movimientos lentos de aire. Por otro lado, una pequeña masa termal puede agregarse al sensor, para dar la lectura más firme a pesar de desviaciones pequeñas en la temperatura aérea.

1.16 El buffer SN74LS244.

El buffer SN74LS244 es un buffer octal y controlador de líneas, el cuál se puede emplear como manejador de dirección de memoria, como manejador de reloj ó bien para proporcionar la corriente suficiente que puede ser necesitada en algún circuito en específico.

Específicamente para nuestro sistema de adquisición de datos lo utilizaremos para emplear la corriente que es capaz de entregar cuando es configurado en la forma que se muestra en el capítulo 3 en el tema: "3.9: Análisis del sistema de adquisición de datos", con el fin de redireccionar ó elegir otro canal de los 7 disponibles del ADC0808, con la cantidad de corriente necesaria para que dicho redireccionamiento se lleve a cabo satisfactoriamente.

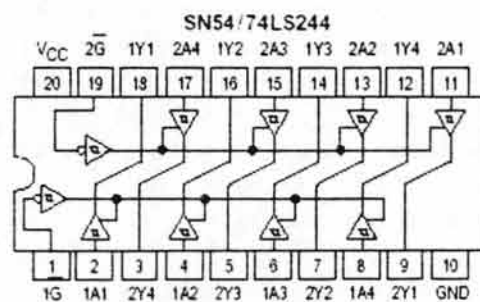


Figura 1.19 Diagrama de conexión interno del buffer SN74LS244.

Este dispositivo opera con una tensión de por lo menos 4.5v. y como máximo 5.5V. siendo la tensión típica de trabajo a 5v.

Para mayor información relacionada con este dispositivo consultar el anexo 4.

Ahora que conocemos los diferentes conceptos referentes al hardware del sistema de adquisición de datos veamos a continuación los mecanismos de software, de los cuales nos valdremos para vincular todos estos elementos para que se relacionen entre si y desempeñen su tarea encomendada.

CAPÍTULO 2.

CONCEPTOS FUNDAMENTALES REFERENTES AL SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS.

Una vez que sabemos el funcionamiento de cada uno de los componentes que se verán involucrados en el desarrollo del prototipo del sistema de adquisición de datos, será ahora necesario conocer como podemos tomar esos datos biomédicos con la PC, para ello haremos uso de potentes lenguajes de programación tales como Visual Basic y Visual C++.

2.1 El lenguaje de programación Visual Basic 6.0

¿Qué es Visual Basic? La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede agregar objetos prefabricados en su lugar dentro de la pantalla. Si ha utilizado alguna vez un programa de dibujo como Paint, ya se tiene la mayor parte de las habilidades necesarias para crear una interfaz de usuario efectiva.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows. Los principiantes pueden crear aplicaciones útiles con sólo aprender unas pocas palabras clave, pero, al mismo tiempo, la eficacia del lenguaje permite a los profesionales acometer cualquier objetivo que pueda alcanzarse mediante cualquier otro lenguaje de programación de Windows.

La capacidad de Visual Basic para realizar tareas y aplicaciones es ilimitada, es por ello que resulta muy atractivo la idea de utilizarlo como lenguaje de programación, con dicho lenguaje se pueden realizar aplicaciones sencillas hasta muy potentes aplicaciones, además utiliza ciertas técnicas que se pueden considerar hasta cierto punto lógicas, así como las palabras claves que utiliza son bastante obvias brindándonos de esta forma una manera muy fácil de interactuar con el mundo exterior, Visual Basic proporciona un juego completo de herramientas que facilitan el desarrollo rápido de aplicaciones.

2.2 Razones para utilizar el lenguaje Visual Basic 6.0.

Son muchas las razones para utilizar Visual Basic 6.0 entre otras, algunas muy importantes son:

- Las características de acceso a datos le permiten crear bases de datos, aplicaciones cliente, y componentes de servidor escalables para los formatos de las bases de datos más conocidas, incluidos Microsoft SQL Server y Access.
- Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet o Intranet desde su propia aplicación, o la creación de aplicaciones de servidor para Internet ó un servidor FTP anónimo ó protegido.
- La aplicación terminada es un auténtico archivo.exe que utiliza una máquina virtual de Visual Basic que se puede distribuir con absoluta libertad.

2.3 La programación orientada a objetos.

El diseño orientado a objetos y la programación orientada a objetos representan un cambio de perspectiva con respecto a la programación estándar por procedimientos. En lugar de pensar en el flujo del programa desde la primera hasta la última línea de código, se debe pensar en la creación de objetos: componentes auto contenidos de una aplicación que tienen funcionalidad privada además de la funcionalidad que se puede exponer al usuario.

Los objetos son el fundamento de la programación en Visual Basic. Los formularios y los controles son objetos. Las bases de datos son objetos. Hay objetos por todas partes.

Los formularios y controles de Visual Basic son objetos que exponen sus propios métodos, propiedades y eventos. Las propiedades se pueden considerar como atributos de un objeto, los métodos como sus acciones y los eventos como sus respuestas.

Un objeto de uso diario como el globo de un niño tiene también propiedades, métodos y eventos. Entre las propiedades de un globo se incluyen atributos visibles como el alto, el diámetro y el color. Otras propiedades describen su estado (inflado o desinflado) o atributos que no son visibles, como su edad. Por definición, todos los globos tienen estas propiedades; lo que varía de un globo a otro son los valores de estas propiedades.

Un globo tiene también métodos o acciones inherentes que puede efectuar. Tiene un método inflar (la acción de llenarlo de helio) o un método desinflar (expeler su contenido) y un método elevarse (si se deja escapar). De nuevo, todos los globos pueden efectuar estos métodos.

Los globos tienen, además, respuestas predefinidas a ciertos eventos externos. Por ejemplo, un globo respondería al evento de pincharlo desinflándose o al evento de soltarlo elevándose en el aire.

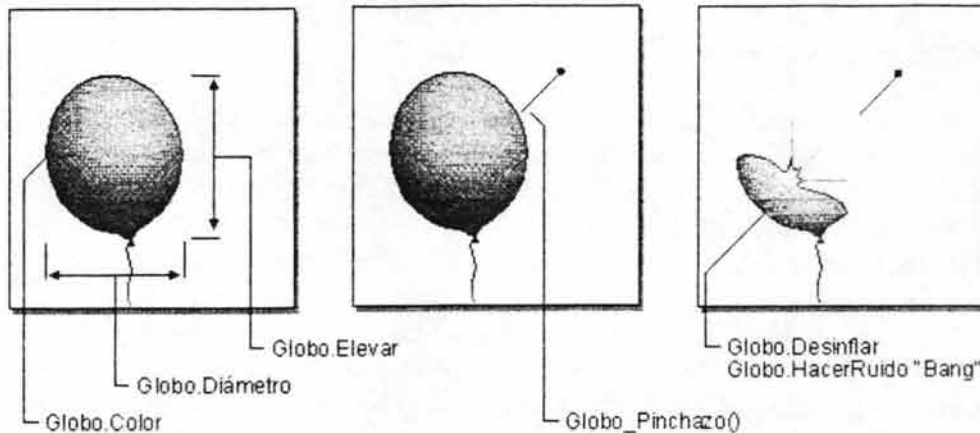


Figura 2.1 Los objetos tienen propiedades, responden a eventos y ejecutan métodos

2.4 Librerías de vínculos dinámicos DLL.

Las librerías de enlace dinámico son uno de los elementos principales del sistema operativo Windows 9x. Normalmente las Bibliotecas de Vínculos Dinámicos aparecen con la extensión ".dll"; En su concepto básico, se tratan de archivos ejecutables independientes que contienen funciones y recursos que pueden ser llamados por los programas y por otras DLL para realizar ciertos trabajos. Los desarrolladores utilizan las DLL para poder reciclar el código y aislar las diferentes tareas. Una DLL no puede ser ejecutada en forma independiente, entra en acción hasta que un programa u otra DLL llama a una de las funciones de la librería, es decir, Las DLL no pueden ejecutarse directamente, es necesario llamarlas desde un código externo. El término "enlace dinámico" se refiere al hecho de que el código que contiene la DLL se incorpora al programa ejecutable que la llama sólo hasta el momento en que es requerido, en tiempo de ejecución, al contrario del enlace estático que es el que se lleva a cabo durante el proceso de enlazado para crear un programa Windows 9x.

El lenguaje Visual Basic Ver. 6.0 ofrece en sus tipos de proyectos la posibilidad de crear una librería de vínculos dinámicos .DLL ActiveX, pero, desafortunadamente, este lenguaje no ofrece librerías o instrucciones para trabajar con los puertos de la PC (por lo menos con lo que respecta al puerto paralelo) es por ello que se recurre al lenguaje de programación Visual C++, el cuál es un potentísimo lenguaje de programación que incorpora funciones realmente útiles además de que, entre otros muchos tipos de proyectos que se pueden crear, nos ofrece la posibilidad de crear librerías de vínculos dinámicos, pero lo más trascendental radica en el hecho de tener la posibilidad de incluir funciones para poder manipular los puertos de la PC.

2.5 El archivo de definiciones en una DLL.

Un archivo de definición (.DEF) es un archivo de texto que contiene declaraciones para definir un archivo .EXE ó una DLL. En nuestro caso, al crear una DLL necesitaremos este tipo de archivo con el fin de que, al crear las funciones dentro de nuestra DLL dichas funciones se puedan utilizar desde algún otro programa externo, en este caso, Visual Basic.

Los archivos de definición: DEF contienen una sección denominada EXPORT, pues bien, en esta sección debemos de incluir las funciones que vamos a utilizar desde otro programa externo. Más adelante se analizará el código de la librería de vínculos dinámicos: REDIRECC.DLL y su archivo de definición .DEF para observar como se logra llevar a cabo la inclusión de las funciones de la .DLL en el lenguaje de programación Visual Basic

La palabra clave EXPORT indica que las funciones están disponibles para otros programas externos a la DLL, con esto se garantiza que las funciones sean exportables a otros programas que así las requieran, en otras palabras, se trata del punto de entrada a las funciones de la DLL.

Si no se está usando la palabra clave EXPORT (No será necesaria) o el archivo de definiciones: DEF) otro punto importante a considerar es la convención de llamadas a función. Lenguajes como C/C++ utilizan la convención `__cdecl` que se caracteriza por incluir un código que sirve para realizar labores de limpieza previa (inicialización con valor igual a cero) en el stack. En comparación, las funciones de la API de Windows utilizan la convención `__stdcall` la cual realiza un decorado de los nombres de función utilizando un símbolo de subrayado seguido de `@#nn` donde `#nn` representa el número total de bytes que se requieren en los parámetros de la función. Tanto en la convención `__cdecl` como en `__stdcall` los argumentos de las funciones se pasan al stack de derecha a izquierda.

En Windows se declara a las convenciones de llamadas a función WINAPI, PASCAL y CALLBACK como de tipo `__stdcall` y resulta que Visual Basic utiliza la convención de tipo PASCAL la cual no distingue entre letras mayúsculas y minúsculas, al contrario de la convención utilizada por C/C++ que es sensible al tamaño de caja de las letras, por lo tanto los argumentos utilizados en nuestras funciones entran en conflicto al tratar de utilizarlas desde Visual Basic. Para resolver este inconveniente se requiere establecer una equivalencia de exportación, es decir, especificamos que el nombre de función NOMBRE sea asociado con la función llamada Nombre () y que ésta requiere 1 parámetro. Para el caso de la función Nombre2 se sigue un proceso similar pero especificando que ésta función requiere dos parámetros, de ésta manera se definen los valores de exportación así:

EXPORTS

NOMBRE=Nombre @1

NOMBRE2=Nombre2 @2

2.6 Nombres decorativos para invocar funciones en Visual Basic.

Cuando creamos una .DLL en Visual C++, nos podemos valer de una utilería especial de Visual C++ denominada AppWizard para crear un archivo MFC DLL, así obtendremos un archivo especial que se puede incluir cuando compilamos nuestra librería de vínculos dinámicos .DLL. Dicho archivo es muy útil porque nos indica el número total de bytes que se requieren en los parámetros de la función ó funciones que se incluyen en la .DLL, además de crear el esqueleto del archivo .DEF, agrega el nombre de las funciones a ser exportadas y automáticamente se agrega al proyecto.

2.7 Conexión de Bases de datos Access sobre Visual Basic utilizando el modelo de datos ADO.

Microsoft® ActiveX® Data Objects (ADO) es el modelo para bases de datos que, además de ser el modelo más reciente con respecto al modelo de bases de datos DAO, incorpora muchas mejoras tales como: su facilidad de uso, su gran velocidad, el bajo consumo de memoria y la pequeña ocupación en disco.

Los siguientes elementos son la parte clave del modelo de programación ADO:

- Connection
- Command
- Parameter
- Recordset
- Field
- Error
- Property
- Collection
- Event

2.7.1 Comparativa de modelos de datos.

ADO no proporciona compatibilidad automática de código con las aplicaciones de acceso a datos existentes. Aunque ADO encapsula la *funcionalidad* de DAO y de RDO, sin embargo es necesario convertir muchos de los elementos del lenguaje a la sintaxis de ADO. En ciertos casos, esto implica solamente una sencilla operación de conversión de algunas funciones del código existente. En otros casos, es preferible volver a escribir la aplicación para emplear las nuevas características de ADO.

DAO (Objetos de acceso a datos) fue la primera interfaz orientada a objetos que exponía el motor de base de datos Microsoft Jet (empleado por Microsoft Access) y permitía a los programadores de Visual Basic usar ODBC para conectar directamente con tablas Access, así como con otras bases de datos. DAO es apropiado para el desarrollo de aplicaciones dentro

de un único sistema o para despliegues locales de pequeño tamaño, pero carece de funciones innovadoras las cuáles el modelo ADO ahora ofrece.

RDO (Objetos de datos remotos) es una interfaz de acceso a datos para ODBC orientada a objetos, que incorpora un estilo sencillo de DAO y cuya interfaz expone prácticamente toda la flexibilidad y eficacia de bajo nivel de ODBC. Sin embargo, RDO presenta limitaciones al no proporcionar un acceso apropiado a las bases de datos Jet o ISAM, y al sólo permitir el acceso a bases de datos relacionales a través de los controladores ODBC existentes. A pesar de ello, se ha constatado que RDO es la interfaz elegida por muchos programadores de SQL Server, de Oracle y de otras bases de datos relacionales de gran tamaño. RDO proporciona los objetos, las propiedades y los métodos necesarios para tener acceso a los aspectos más complejos relacionados con los procedimientos almacenados y los conjuntos de resultados.

ADO es el sucesor de DAO y RDO. La funcionalidad de ADO 2.0 es prácticamente idéntica a la de RDO y en general existe una correspondencia de funciones similares en ambos modelos. ADO permite "allanar" el modelo de objetos empleado por DAO y RDO, en el sentido de que contiene menos objetos y más propiedades, métodos (y argumentos) y eventos. Por ejemplo, en ADO no existe un equivalente de los objetos `rdoEngine` y `rdoEnvironment` expuestos por el administrador de controladores ODBC y las interfaces `hEnv`. Tampoco es posible crear orígenes de datos ODBC desde ADO, a pesar de que puede exponer su propia interfaz a través del proveedor de servicios OLE DB de ODBC.

Muchas de las características incluidas en los modelos DAO y RDO fueron refundidas en objetos únicos para simplificar en gran medida el modelo de objetos. No obstante, debido a esto es posible que inicialmente encontremos dificultades cuando intentemos localizar un objeto, colección, propiedad, método o evento de ADO apropiado a nuestras necesidades. A diferencia de DAO y RDO, aunque los objetos ADO son jerárquicos, se pueden crear también fuera del alcance de la jerarquía.

Sin embargo, conviene tener en cuenta que ADO no admite actualmente toda la funcionalidad de DAO. ADO incluye principalmente funcionalidad para interactuar con los orígenes de datos OLE DB en una forma similar a RDO, así como operaciones remotas y tecnología DHTML.

En general, dado el estado de la evolución de ADO, probablemente sea prematuro migrar directamente a ADO la mayoría de las aplicaciones DAO (excepto posiblemente las que utilizan `ODBCDirect`), ya que ADO no admite por el momento definición de datos (bibliotecas DDL), usuarios, grupos y otras características. Sin embargo, probablemente podrá migrar desde ahora a ADO si utiliza DAO únicamente para aplicaciones cliente-servidor y no se basa en el motor de base de datos Jet ni utiliza bibliotecas DDL.

Hablando específicamente del modelo de datos a utilizar (ADO), un elemento fundamental que se utilizará a lo largo de la aplicación `Servidor.exe` es el elemento `Recordset`, por ello ampliamos su definición:

2.7.2 Recordset.

Si el comando es una consulta que devuelve datos como filas de información de una tabla (esto es, es una consulta que devuelve filas), entonces esas filas se colocan en almacenamiento local.

El modelo de objetos que encarna este almacenamiento es un objeto Recordset. Sin embargo, no existe un objeto que represente una única fila de un Recordset.

El Recordset es el medio principal para examinar y modificar los datos de las filas. El objeto Recordset nos permite:

- Especificar qué filas están disponibles para examinar.
- Recorrer las filas.
- Especificar el orden en el que se pueden recorrer las filas.
- Agregar, cambiar o suprimir las filas.
- Actualizar el origen de datos con las filas cambiadas.
- Gestionar el estado general del Recordset.

Por otra parte, el elemento Field será también trascendental en la aplicación, sobretodo para la búsqueda de datos.

Una fila de un Recordset consta de uno o más *campos*. Si imagina el Recordset como una cuadrícula de dos dimensiones, los campos se alinean para formar *columnas*. Cada campo (columna) tiene entre sus atributos un nombre, un tipo de datos y un valor. Es este valor el que contiene los datos reales del origen de datos.

El modelo de objetos que encarna un campo es el objeto Field.

Para modificar los datos en el origen de datos, se modifica el valor de los objetos Field de las filas Recordset. Finalmente, los cambios a un Recordset se envían al origen de datos. Como opción, los métodos de administración de transacción del objeto Connection pueden garantizar que los cambios se realizan correctamente o fallan a la vez.

2.8 El protocolo FTP.

Una de las alternativas más importantes que nos permite Internet es la transferencia de archivos de un ordenador a otro desde cualquier parte del mundo. Para ello utilizaremos el protocolo de transferencia de archivos o "ftp" (file transfer protocol).

FTP, perteneciente a la familia TCP/IP, comenzó siendo una utilidad incluida en el sistema operativo Unix, usada para transferir archivos entre los distintos equipos conectados en una red. No en vano, FTP es un acrónimo de File Transfer Protocol o Protocolo de Transferencia de Archivos.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

Comienza a ser usado en 1971 con el fin de implementarlo en el MIT (instituto tecnológico de Massachussets), sin embargo, no fue hasta 1973 cuando el protocolo traería considerables mejoras para su utilización.

Concepto Cliente y Servidor.

Para poder comunicarse mediante este protocolo es preciso que un PC se conecte como cliente y exista otro que haga de servidor. Esto quiere decir que nosotros mismos podemos realizar una comunicación como clientes hacia otro PC servidor, que puede estar en nuestra red o Internet (por ejemplo). Esta operación se realiza con el fin de navegar e investigar qué es lo que tiene este PC servidor, para poder subir, bajar, borrar, escribir....dependiendo de los permisos que tengamos.

Nosotros como cliente, debemos usar un programa denominado "Cliente de FTP" ya sea en modo texto o gráfico. En el PC servidor, deberá haber instalado un programa "Servidor de FTP" que pueda atender las peticiones de los clientes y permitir la conexión de todos ellos.

En resumen podemos decir que el protocolo FTP y los programas que lo manejan y usan, nos servirán para compartir archivos y permitir la transmisión fiable de los mismos, con opciones avanzadas.

Mediante ftp podemos compartir (recibir y enviar) nuestros ficheros con otras computadoras, siempre que el administrador de estos últimos nos lo permita.

2.9 Configuración de un servidor FTP.

Al conectarnos por ftp, nos encontramos con el problema de disponer de un identificador (login) y una clave de acceso (password) para cada máquina a la que nos conectemos. Esto se debe a protección que da el servidor a la información que posee para mantenerla fuera del alcance de intrusos. Es posible también, aunque no recomendable que los usuarios de Internet pueden realizar ftp sobre los servidores de la red configurados con el título "*anonymous ftp*" (ftp anónimo).

La comunicación por ftp anónimo es una forma de conectarse a los servidores de ficheros de Internet sin necesidad de poseer una palabra clave. Existen muchos servidores públicos que admiten conexiones ftp anónimas llamados *ftp-sites*. Estos servidores se dedican a distribuir software de dominio público y shareware para cualquier tipo de ordenador o sistema operativo existente.

Para utilizar un ftp anónimo, el usuario debe introducir el identificador de usuario *anonymous* y como palabra clave de acceso se considera una muestra de cortesía introducir la dirección de correo electrónico aunque también se suele introducir la palabra *guest* (huésped).

Los objetivos principales de este protocolo son:

- Posibilitar el *compartimiento de archivos* entre computadoras (programas y/o datos)
- Posibilitar el *uso remoto* de las computadoras
- *Transferir datos* de una forma segura y optima entre computadoras.

FTP mas que para ser usado por un usuario directamente es usado por los programas para comunicarse, lo que facilita al usuario despreocuparse de las características del sistema con que conecta. Se creo en 1971 con un modelo de transferencia llamado RFC 141 en M.I.T. Fue hasta después de muchas revisiones que llegó a RFC 265 cuando ya se le considera un protocolo de transferencia de archivos completa entre HOSTs (servidores de archivos) de ARPHANET. Al final de la edición de RFC 765 se incluyeron algunos de los que son ahora los comandos de este protocolo:

CDUP Change to Parent Directory (Cambio hacia el directorio paterno ó raíz)
SMNT Structure Mount (Montar estructura)
STOU Store Unique (Establecimiento único)
RMD Remove Directory (Borrar Directorio)
MKD Make Directory (Crear Directorio)
PWD Print Directory (Imprimir Directorio)
SYST System (Sistema)

2.9.1 Terminología relacionada.

ASCII.

Solo se usan todos los caracteres dentro de los 8 bits en su valor bajo

Access controls.

Este sirve para hablar a cerca de los privilegios (derechos en la red) de cada usuario tanto en archivos como en dispositivos.

Data connection.

Habla de cuando hay una comunicación Full Duplex entre dos computadoras.

El concepto Full Duplex se lleva a cabo cuando los datos pueden viajar en ambas direcciones.

DTP Proceso de la transferencia (Error Recovery).

Este es un procedimiento que le permite al usuario en algunos casos recuperar información perdida en el proceso de transferencia.

2.9.2 Tipos de datos en la transferencia por FTP.

El tipo ASCII, es el más común en el protocolo FTP. Se usa cuando se transfieren archivos de texto, la computadora que envía (sender), debe convertir cualquiera que sea su estructura de archivos interna, debe convertir sus datos al formato genérico de 8 bits, y el que recibe (receiver) lo debe convertir de nuevo a su formato propio.

El tipo EBCDIC es el más eficiente cuando ambos el que recibe y el que envía lo usan como formato propio, este tipo se representa también en 8 bits pero de forma EBCDIC. Lo único en lo que cambian es en la forma de reconocer los códigos de los caracteres.

El formato de IMAGEN es cuando se compacta todo lo que se quiere enviar en cadenas seguidas de paquetes de 8 bits, esto es no importa el formato en que internamente se maneje la información, cuando se va a enviar se tiene que hacer una conversión de 8 bits en 8 bits y cuando el que recibe tiene todo el paquete, el mismo debe codificarlos de nuevo para que la transmisión sea completada.

En la estructura de datos en FTP se consideran tres tipos diferentes de archivos:

File - structure donde no hay estructuras internas y el archivo es considerado una secuencia continua de bytes

Record - structure donde los archivos contienen registros iguales en estructura.

Page - structure donde los archivos contienen páginas enteras indexadas separadas.

Al establecer una conexión por FTP se debe tomar en cuenta que el mecanismo de transferencia consiste en colocar bien la transferencia de datos en los puertos adecuados y al concluir la conexión estos puertos deben ser cerrados adecuadamente. El tamaño de transferencia es de 8 bits, en ambos. El que va a transferir, debe escuchar desde el puerto hasta que el comando enviado sea recibido y este será el que de la Dirección de la transferencia. Una vez recibido el comando y establecido una transferencia del servidor a que solicita se inicializa la Comunicación de la transferencia para verificar la conexión, esta es una cabecera con un formato específico, después de esto se comienza a enviar las tramas de 8 bits sin importar el tipo de datos que sea (antes mencionado), y al finalizar se envía otra trama cabecera ya establecida confirmando la transferencia completada.

2.9.3 Modos de transferencia en FTP.

STREAM MODE (Modo de Cadena)

BLOCK MODE (Modo de Bloque)

COMPRESSED MODE (Modo Comprimido)

2.9.4 Comandos utilizados frecuentemente.

De acceso:

USER NAME (USER) Nombre de usuario
PASSWORD (PASS) Contraseña
ACCOUNT (ACCT) Tipo de cuenta
CHANGE WORKING DIRECTORY (CWD) Cambiar el directorio de trabajo
CHANGE TO PARENT DIRECTORY (CDUP) Cambiar al directorio paterno ó raíz
REINITIALIZE (REIN) Reinicializar
LOGOUT (QUIT) Finalizar la sesión

De transferencia:

DATA PORT (PORT) Puerto de datos
PASSIVE (PASV) Pasiva
FILE STRUCTURE (STRU) Estructura de archivos
TRANSFER MODE (MODE) Modo de transferencia

De servicio:

RETRIEVE (RETR) Recuperación
STORE (STOR) Acopio
STORE UNIQUE (STOU) Establecimiento único
APPEND (with create) (APPE) Anexar
ALLOCATE (ALLO) Asignación
RENAME TO (RNTO) Renombrar a
ABORT (ABOR) Abortar
DELETE (DELE) Borrar
REMOVE DIRECTORY (RMD) Eliminar directorio
MAKE DIRECTORY (MKD) Crear directorio
PRINT WORKING DIRECTORY (PWD) Imprimir directorio de trabajo
LIST (LIST) Lista
HELP (HELP) Ayuda

Algunos de los códigos usados en la transferencia son los siguientes, estos códigos no son más que mensajes enviados por el protocolo:

Códigos normales

- 200 Command okay.
- 500 Syntax error, command unrecognized. This may include errors such as command line too long.
- 501 Syntax error in parameters or arguments.
- 202 Command not implemented, superfluous at this site.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.

110 Restart marker reply. In this case, the text is exact and not left to the particular implementation; it must read:

211 System status or systems help reply.

212 Directory status.

213 File status.

214 Help message. On how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.

215 NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.

120 Service ready in nnn minutes.

220 Service ready for new user.

221 Service closing control connection. Logged out if appropriate.

421 Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.

125 Data connection already open; transfer starting.

225 Data connection open; no transfer in progress.

425 Can't open data connection.

226 Closing data connection. Requested file action successful (for example, file transfer or file abort).

426 Connection closed; transfer aborted.

227 Entering Passive Mode (h1,h2,h3,h4,p1,p2).

230 User logged in, proceed.

530 not logged in.

331 User name okay, need password.

332 Need account for login.

532 Need account for storing files.

150 File status okay; about to open data connection.

250 Requested file action okay, completed.

257 "PATHNAME" created.

350 Requested file action pending further information.

450 Requested file action not taken. File unavailable (e.g., file busy).

550 Requested action not taken. File unavailable (e.g., file not found, any access).

451 Requested action aborted. Local error in processing.

551 Requested action aborted. Page type unknown.

452 Requested action not taken. Insufficient storage space in system.

552 Requested file action aborted Exceeded storage allocation (for current directory or dataset).

553 Requested action not taken. File name not allowed. Códigos de mensajes con operaciones numéricas

110 Restart marker reply.

120 Service ready in nnn minutes.

125 Data connection already opens; transfer starting.

150 File status okay; about to open data connection.

200 Command okay.

202 Command not implemented, superfluous at this site.

- 211 System status, or system help reply.
- 212 Directory status.
- 213 File status.
- 214 Help message. On how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.
- 215 NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.
- 220 Service ready for new user.
- 221 Service is closing control connection. Logged out if appropriate.
- 225 Data connection open; no transfer in progress.
- 226 Closing data connection. Requested file action successful (for example, files transfer or file abort).
- 227 Entering Passive Mode (h1, h2, h3, h4, p1, p2).
- 230 User logged in, proceed.
- 250 Requested file action okay, completed.
- 257 "PATHNAME" created.
- 331 User names okay need password.
- 332 Need account for login.
- 350 Requested file action pending further information.
- 421 Service not available, closing controls connection. This may be a reply to any command if the service knows it must shut down.
- 425 can't open data connection.
- 426 Connection closed; transfer aborted.
- 450 Requested file action not taken. File unavailable (e.g., file busy).
- 451 Requested action aborted: local error in processing.
- 452 Requested action not taken. Insufficient storage space in system.
- 500 Syntax error, command unrecognized. This may include errors such as command line too long.
- 501 Syntax error in parameters or arguments.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.
- 530 Not logged in.
- 532 Need account for storing files.
- 550 Requested action not taken. File unavailable (e.g., file not found, no access).
- 551 Requested action aborted: page type unknown.
- 552 Requested file action aborted. Exceeded storage allocation (for current directory or dataset).
- 553 Requested action not taken. File name not allowed.

Con el fin de tener un control absoluto de las personas que ingresarán al servidor FTP es ampliamente recomendable manejar el uso de contraseñas y un login de identificación para, así saber en todo momento quienes serán las personas autorizadas para ingresar a dicho servidor y enviar sus archivos.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

La forma de conexión con el servidor es muy similar a la utilizada en Telnet: al contactar con el servidor, la computadora pedirá la correspondiente identificación (*login y password*), tras lo cual, ya se podrá transferir cualquier fichero en una ó ambas direcciones.

2.10 El control Transfer de Visual Basic 6.0

El control Internet Transfer implementa dos protocolos de Internet de gran difusión: Protocolo de transferencia de hipertexto (HTTP) y Protocolo de transferencia de archivos (FTP). Con el control Internet Transfer se puede conectar con cualquier sitio que utilice alguno de estos protocolos y obtener archivos mediante los métodos OpenURL ó Execute.

La funcionalidad del control Internet Transfer depende del protocolo que desee usar. Como los dos protocolos contemplados funcionan de forma distinta, las operaciones que puede realizar dependen del protocolo elegido.

2.10.1 Aplicaciones posibles.

- Agregar un explorador FTP a cualquier aplicación.
- Crear una aplicación que transfiera automáticamente archivos de un sitio público FTP.
- Analizar un sitio del World Wide Web para buscar referencias gráficas y transferir únicamente los gráficos.
- Mostrar una vista personalizada de datos dinámicos obtenidos de una página Web.

2.10.2 Funcionamiento básico.

La funcionalidad del control Internet Transfer depende del protocolo que desee usar. Como los dos protocolos contemplados funcionan de forma distinta, las operaciones que puede realizar dependen del protocolo elegido. Por ejemplo, el método GetHeader sólo funciona con HTTP (documentos HTML).

Sin embargo, hay algunos procedimientos que son comunes a ambos protocolos. Básicamente, para poder usar cualquiera de los protocolos, debe hacer lo siguiente:

1. Establecer la propiedad AccessType a un servidor proxy válido.
2. Invocar el método OpenURL con una dirección URL válida.
3. Invocar el método Execute con una dirección URL válida y un comando adecuado para el protocolo.
4. Usar el método GetChunk para recuperar datos del búfer.

2.10.3 Transmisión síncrona y asíncrona.

El método `OpenURL` produce una transmisión *síncrona* de los datos. En este contexto, "síncrona" significa que la operación de transferencia se produce antes de que se ejecute cualquier otro procedimiento. Por tanto, la transferencia de datos debe haber terminado antes de que pueda ejecutarse otro código.

Que queremos decir con esto, que cuando se lleva a cabo una transmisión síncrona de datos se necesita llevar a cabo la totalidad de transferencia de los mismos, mientras tanto la aplicación se detendrá y no se ejecutará ninguna otra línea posterior de código hasta que este completa dicha transferencia de datos.

Por otra parte, el método `Execute` produce una transmisión *asíncrona*. Cuando se invoca el método `Execute`, la operación de transferencia se produce independientemente de otros procedimientos. Por tanto, después de invocar el método `Execute` es posible ejecutar otro código mientras se reciben datos en segundo plano.

Es decir, que con el método `Execute` podemos llevar a cabo otras tareas de programación y al mismo tiempo llevar a cabo la transferencia de la información que se desea enviar.

¿Qué significa esto para el usuario del control `Internet Transfer`? En pocas palabras, el uso del método `OpenURL` produce una corriente directa de datos que puede guardar en disco ó ver directamente en un control `TextBox` (si los datos son de texto). Por otra parte, si utiliza el método `Execute` para obtener datos, deberá observar el estado de conexión del control mediante el evento `StateChanged`. Cuando se llegue al estado adecuado.

El método `Execute` tiene cuatro argumentos: *url*, *operation*, *data* y *requestHeaders*. Las operaciones FTP sólo utilizan el argumento *operation* y el argumento *url*, que es opcional. Por ejemplo, para obtener un archivo de un equipo remoto, puede usar el código siguiente:

```
Inet1.Execute "FTP://ftp.microsoft.com", _  
"GET disclaimer.txt C:\Temp\Disclaimer.txt"
```

Por ejemplo, para cambiar de directorio con el protocolo FTP se utiliza el comando "CD" con la ruta de acceso del directorio al que desea cambiar.

Para las operaciones más comunes, como llevar un archivo a un servidor o traer un archivo desde él, el control **Internet Transfer** utiliza el mismo comando, o uno similar, con el método **Execute**. Por ejemplo, en el código siguiente se utiliza el comando "CD" como argumento del método **Execute** para cambiar de directorio:

‘El cuadro de texto `txtURL` contiene la ruta de acceso que se va a abrir. El cuadro de texto `txtRemotePath` contiene la ruta de acceso a la que hay que cambiar.

```
Inet1.Execute txtURL.Text, "CD " & txtRemotePath.Text
```


Nota Al usar el método **Execute** con comandos FTP no se emplean los argumentos *data* y *requestHeaders*. En su lugar, todas las operaciones y sus parámetros se pasan como una única cadena en el argumento *operation*, con los parámetros separados por un espacio en blanco. En las descripciones siguientes no se deben confundir los términos "archivo1" y "archivo2" con los argumentos *data* y *requestHeaders*.

Para poder enviar archivos por Internet es necesario configurar un servidor el cuál se encuentre dispuesto a recibir y mandar dichos archivos, existen muchas aplicaciones comerciales para entablar y configurar un servidor tipo FTP, pero la opción más viable sería la de encontrar una aplicación que no represente un desembolso para adaptarlo a nuestra aplicación, más adelante se hablará con mayor detalle de la aplicación Cesar.exe, la cuál es de distribución totalmente gratuita.

Por ahora hablemos de la manera de operar de las funciones mayor usadas por el control Internet Transfer:

La sintaxis de las operaciones FTP es la siguiente:

nombre_operación archivo1 archivo2

Por ejemplo, para enviar un archivo, el código siguiente incluye el nombre de la operación ("PUT") y los dos nombres de archivo que la operación requiere:

```
Inet1.Execute , "PUT " & RutaLocal1 & " " & RutaDestino1
```

En donde:

RutaLocal1 = "C:\Mis Documentos\Dato.txt"

RutaLocal2 = "C:\Folder1\Entrada\Dato.txt"

Aquí se utiliza el método Execute, el cuál como ya hemos observado lleva a cabo una transmisión asíncrona, el envío de los datos se lleva a cabo mediante la función "PUT" la cuál, como veremos en la tabla siguiente necesita de dos parámetros un Archivo1 y un archivo2, en la línea anterior estos archivos son almacenados en dos variables.

En la tabla 2.1 se enumeran los comandos FTP admitidos por el control:

Operación	Descripción	Ejemplo
CD <i>archivo1</i>	Cambia de directorio. Cambia al directorio especificado en <i>archivo1</i> .	Execute , "CD docs\misdocs"
CDUP	Cambia al directorio primario. Equivale a "CD".	Execute , "CDUP"
DELETE <i>archivo1</i>	Elimina el archivo especificado en <i>archivo1</i> .	Execute , "DELETE discard.txt"
DIR [<i>archivo1</i>]	Busca en el directorio especificado en <i>archivo1</i> . Si no se especifica <i>archivo1</i> , se busca en el directorio de trabajo actual. Utilice el método GetChunk para obtener los datos.	Execute , "DIR /misdocs"
GET <i>archivo1</i> <i>archivo2</i>	Recupera el archivo remoto especificado en <i>archivo1</i> y crea el nuevo archivo local especificado en <i>archivo2</i> .	Execute , _ "GET tomame.txt C:\metienes.txt"
MKDIR <i>archivo1</i>	Crea el directorio especificado en <i>archivo1</i> . El éxito de la operación depende de los privilegios del usuario en el host remoto.	Execute , "MKDIR /miDir"
PUT <i>archivo1</i> <i>archivo2</i>	Copia el archivo local especificado en <i>archivo1</i> al host remoto especificado en <i>archivo2</i> .	Execute , _ "PUT C:\colócame.txt /colócame.txt"
PWD	Indica el directorio de trabajo. Devuelve el nombre del directorio actual. Puede usar el método GetChunk para obtener los datos.	Execute , "PWD"
QUIT	Cierra la conexión actual	Execute , "QUIT"
RECV <i>archivo1</i> <i>archivo2</i>	Igual que GET.	Execute , _ "RECV tomame.txt C:\metienes.txt"
RENAME <i>archivo1</i> <i>archivo2</i>	Cambia el nombre de un archivo. El éxito de la operación depende de los privilegios del usuario en el host remoto.	Execute , "RENAME viejo.txt nuevo.txt"
RMDIR <i>archivo1</i>	Quita un directorio. El éxito de la operación depende de los privilegios del usuario en el host remoto.	Execute , "RMDIR DirAntiguo"
SEND <i>archivo1</i>	Copia un archivo al sitio FTP. (Igual que PUT.)	Execute , _ "SEND C:\colócame.txt /colócame.txt"

Tabla 2.1 Comandos FTP admitidos por el control

Hablando específicamente de la aplicación Cliente.exe, se utiliza el comando SEND ó PUT para enviar únicamente los archivos desde la máquina cliente hacia el servidor ftp.

2.11 Derechos y restricciones de acceso al servidor FTP.

Cuando configuramos un servidor FTP debemos establecer derechos y restricciones a las máquinas clientes que van a interactuar con dicho servidor, estos derechos y restricciones serán de acuerdo a las necesidades propias de cada cliente en particular y de las operaciones necesarias a realizar.

Podemos clasificar como más importantes derechos y restricciones de acceso a un servidor los siguientes:

- Descargar archivos hacia el servidor.
- Cantidad de Kb que se pueden descargar al servidor.
- Establecer el número de conexiones al servidor de cada cliente.
- Establecer un tiempo determinado de conexión para cada cliente.

2.11.1 Descargar archivos hacia el servidor.

Este derecho se refiere cuando necesitamos o no que la maquina cliente pueda descargar archivos hacia el servidor ó únicamente se otorga el acceso para consultar un archivo, por ejemplo.

2.11.2 Cantidad de Kb que se pueden descargar hacia el servidor.

Algunos programas que se utilizan para configurar un servidor ftp, permiten establecer este tipo de parámetros, es decir, únicamente podrán descargar archivos de una máquina cliente con un espacio no mayor a una cantidad especificada.

2.11.3 Establecer el número de conexiones al servidor de cada cliente.

Es posible establecer el número de accesos permitidos de la máquina cliente hacia el servidor con el fin de controlar los accesos y de alguna manera restringir las descargas posibles de la máquina que se este conectando a dicho servidor y evitar posibles saturaciones de manejo de información.

2.11.4 Establecer un tiempo determinado de conexión para cada cliente.

Esto se refiere al lapso de tiempo que puede estar conectada una máquina cliente con el servidor, y es útil en ciertas aplicaciones cuando se necesita tener el servidor libre de "transito" y que sólo ocupe el tiempo necesario entre una conexión y otra.

Algunos otros derechos y restricciones se relacionan directamente con los archivos que se están trabajando y podemos clasificarlos específicamente como atributos de archivos, he aquí los más importantes:

- Atributos de sólo lectura.
- Atributos de sólo escritura.
- Atributo de lectura/escritura.
- Atributo para borrar archivos.

2.11.5 Atributo de sólo lectura.

Se refiere a que la máquina cliente sólo puede tener acceso a determinado archivo únicamente para consultarlo y no podrá obtener una copia de este bajo ninguna circunstancia, mucho menos eliminarlo.

2.11.6 Atributo de sólo escritura.

Este atributo se refiere cuando es posible que la máquina cliente puede descargar un archivo hacia el servidor, pero no podrá acceder a los archivos que se encuentren en él inclusive del que esta descargando en ese momento, tanto para consultarlos mucho menos para eliminarlos.

2.11.7 Atributo de lectura/escritura.

Este atributo le da el derecho pleno a la máquina cliente de acceder a los archivos del servidor tanto para copiarlos como para poder eliminarlos, así como para descargar los archivos que desee, en función con otros derechos y restricciones que le sean impuestos.

2.11.8 Atributo para borrar archivos.

Con este atributo la máquina cliente podrá eliminar archivos que se encuentran en el servidor, cabe señalar que este atributo es difícilmente otorgado a una máquina cliente, por la razón de que se pueden perder archivos importantes del mismo, siendo el más usual el atributo de sólo escritura, pero marcado, limitado y con ciertas restricciones. Como en este caso, no es importante otorgar este atributo, pues sólo es necesario que el archivo dictamen sea enviado de: la máquina cliente hacia la máquina servidor, será absolutamente conveniente y recomendable prescindir u otorgar, este atributo a las máquinas cliente.

Los controles de imagen, forma y línea se consideran controles ligeros; es decir, sólo admiten un subconjunto de las propiedades, métodos y eventos que se encuentran en el cuadro de imagen. Por tanto, normalmente requieren menos recursos del sistema y se cargan más rápidamente que el control de cuadro de imagen.

2.12 Operaciones lógicas BITWARE (&, ^, |)

Los operadores de bits dirigidos ejecutan operaciones lógicas como son:
-AND DE BIT DIRIGIDO. (&)

- OR EXCLUSIVA DE BIT DIRIGIDO. (^)
- OR INCLUSIVA DE BIT DIRIGIDO. (|)

Hablando específicamente de: - OR INCLUSIVA DE BIT DIRIGIDO. (|)
exclusiva-OR

Expresión inclusiva-OR:

Expresión

Expresión inclusiva-OR | Expresión exclusiva-OR

Los operandos de operadores de bit's dirigidos deben tener tipo integral, pero sus tipos pueden ser diferentes. Estos operadores realizan las conversiones de la aritmética más usuales; el tipo del resultado es el tipo del operando después de la conversión.

Los operadores de bit's dirigidos en C son descritos a continuación:

2.12.1 Operación AND (&).

El operador de bit dirigido AND compara cada bit de su primer operando con el correspondiente bit de su segundo operando. Si ambos bit's son 1, el respectivo bit resultante es puesto a 1. De la otra forma, el respectivo bit resultante es puesto a 0.

&

$1 \& 1 = 1$

2.12.2 Operación Exclusivo-OR (^)

El operador exclusivo-OR de bit dirigido compara cada bit de su primer operando con el correspondiente bit de su segundo operando. Si un bit es 0 y el otro bit es 1, el bit de resultado correspondiente es puesto a 1. De la otra forma, el correspondiente bit de resultado es puesto a 0.

Veamos su tabla

:

^

$0 \wedge 1 = 1$

$1 \wedge 1 = 0$

$1 \wedge 0 = 1$

2.12.3 Operación Inclusivo-OR (|)

El operador inclusivo-OR de bit dirigido compara cada bit de su primer operando con el correspondiente bit de su segundo operando. Si cualquier bit es 1, el bit de resultado correspondiente es puesto a 1. De la otra forma, el bit de resultado correspondiente es puesto a 0.

Ejemplos:

Estas declaraciones son usadas para los siguientes tres ejemplos:

```
short i = OxAB00;  
short j = OxABCD;  
short n;  
n = i & j;
```

El resultado asignado a n en este primer ejemplo es el mismo que el de i (OxAB00 en hexadecimal).

```
n = i | j;  
n = i ^ j;
```

El bit dirigido inclusivo-OR en el segundo ejemplo resulta en el valor OxABCD (en hexadecimal), mientras que el bit dirigido exclusivo-OR en el tercer ejemplo produce OxCD (en hexadecimal).

2.13 La parte grafica de Visual Basic.

Para no ver afectada la funcionalidad de las aplicaciones creadas en Visual Basic utilizando gran cantidad de gráficos, se puede hacer uso de los controles ligeros de manejo de gráficos de dicho lenguaje.

Los controles de imagen, forma y línea se consideran controles ligeros; es decir, sólo admiten un subconjunto de las propiedades, métodos y eventos que se encuentran en el cuadro de imagen. Por tanto, normalmente requieren menos recursos del sistema y se cargan más rápidamente que el control de cuadro de imagen.

Los gráficos (las imágenes y los métodos gráficos) pueden consumir una gran cantidad de memoria. En cierto modo es inevitable: los gráficos contienen mucha información y suelen ser grandes. Pero en muchos casos puede reducir el impacto de los gráficos sobre el tamaño de la aplicación si emplea alguna de las técnicas siguientes:

- Usar el control Image para presentar mapas de bits.
- Cargar los mapas de bits desde archivos cuando sean necesarios y compartir imágenes.

- Usar el método PaintPicture.
- Liberar la memoria utilizada por los gráficos.
- Usar mapas de bits en formato .rle o metarchivos.

2.13.1 Usar el control Image para presentar mapas de bits

Los controles Picture de muchas aplicaciones de Visual Basic sólo existen para hacer clic en ellos o para arrastrarlos y colocarlos. Si esto es todo lo que va a hacer con un control Picture, está malgastando muchos recursos de Windows. Para estos mismos propósitos, los controles Image son mejores que los controles Picture. Cada control Picture es una ventana real y utiliza bastantes recursos del sistema. El control Image es un control "ligero", no una ventana, y no utiliza tantos recursos. De hecho, normalmente puede usar entre cinco y diez controles Image por cada control Picture. Además, los controles Image se dibujan más rápido que los controles Picture. Utilizaremos los controles Picture únicamente cuando necesitemos alguna de las características que sólo estos proporcionan, como el intercambio dinámico de datos (DDE), los métodos gráficos o la posibilidad de contener otros controles ó representar gráficas en plena ejecución.

Cargar los mapas de bits desde archivos cuando sean necesarios y compartir imágenes

Cuando establece una propiedad Picture en tiempo de diseño, se agrega la imagen al formulario e incrementa la memoria consumida por el formulario en tiempo de ejecución. Podemos reducir el consumo de memoria si almacenamos las imágenes en un archivo de recursos y utilizar la función LoadResPicture para cargarlos en tiempo de ejecución. Si nunca utilizamos todas las imágenes asociadas con un formulario al mismo tiempo, esta técnica ahorra memoria en comparación con la de almacenar todas las imágenes en los controles del formulario. Podemos acelerar la velocidad de carga del formulario porque no todas las imágenes tienen que cargarse antes de presentar el formulario.

Podemos compartir la misma imagen entre varios controles Picture, controles Image y formularios. Si utilizamos *código* de este tipo, sólo podemos mantener una copia de la imagen:

```
Picture = LoadPicture("C:\Windows\Chess.bmp")  
Image1.Picture = Picture ' Utiliza la misma imagen.  
Picture1.Picture = Picture ' Utiliza la misma imagen.
```

Contraste lo anterior con el siguiente *código*, que hace cargar tres mapas de bits y, por tanto, necesita más memoria y más tiempo:

```
Picture = LoadPicture("C:\Windows\Chess.bmp")  
Image1.Picture = LoadPicture("C:\Windows\Chess.bmp")  
Picture1.Picture = LoadPicture("C:\Windows\Chess.bmp")
```

De forma similar, si cargamos la misma imagen en varios formularios o controles en tiempo de diseño, se guarda una copia de dicha imagen con cada formulario o con cada control. En vez de esto, podemos colocar la imagen en un formulario y después compartirla con los demás formularios y controles de la forma descrita anteriormente. Esto hace que la aplicación sea más pequeña (porque no tiene copias redundantes de la misma imagen) y más rápida (porque no hay que cargar la imagen varias veces desde el disco). Solamente si son distintas las imágenes que se quieren cargar en cada uno de los formularios será necesario establecer la imagen determinada para cada uno de ellos.

2.13.2 Usando el método PaintPicture.

En lugar de colocar mapas de bits en los controles, puede usar el método PaintPicture para presentar mapas de bits en cualquier parte de los formularios. Esto es especialmente útil cuando queremos dibujar varias veces en mosaico un mismo mapa de bits en un formulario: sólo necesitamos cargar el mapa de bits una vez y usar PaintPicture para dibujarlo varias veces.

2.13.3 Liberar la memoria utilizada por los gráficos.

Cuando ya no vamos a usar una imagen de la propiedad Picture de un formulario, de un cuadro de imagen o de un control Image, podemos establecer la propiedad Picture a Nothing para dejarla vacía:

```
Set Picture1.Picture = Nothing
```

Si utilizamos la propiedad Image de un cuadro de imagen o de un formulario, Visual Basic crea un mapa de bits AutoRedraw (incluso si la propiedad AutoRedraw de dicho formulario o cuadro de imagen es False). Cuando hemos terminado de usar la propiedad Image, podemos liberar la memoria utilizada por ese mapa de bits si utilizamos el método Cls antes de establecer AutoRedraw a False. Por ejemplo, el código siguiente libera la memoria utilizada por la propiedad Image en un control llamado midib:

```
midib.AutoRedraw = True      ' Activa el mapa de bits AutoRedraw.  
midib.Cls                  ' Lo borra.  
midib.AutoRedraw = False    ' Desactiva el mapa de bits.
```

2.13.4 Usando mapas de bits en formato .rle o metarchivos

Aunque el formato de imagen predeterminado es el mapa de bits (.bmp), Visual Basic también puede usar otros formatos de archivos gráficos. Varios programas de dibujo y de gráficos nos permiten guardar los mapas de bits en un formato de mapa de bits comprimido estándar llamado Run Length Encoded (.rle). Los mapas de bits .rle pueden ser varias veces más pequeños que sus equivalentes no comprimidos, especialmente los mapas de bits que contienen grandes partes de colores sólidos. Además, no se aprecia que sean más lentos de

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

cargar o presentar. El uso de metarchivos (.wmf) nos puede ahorrar aún más espacio, hasta 10 veces o más en algunos casos. Hay que intentar usar los metarchivos en su tamaño normal: es más lento dibujarlos si es necesario ampliarlos o reducirlos.

También podemos usar los formatos .gif y .jpg. Estos formatos suelen ser mucho más pequeños; sin embargo, hay algunos inconvenientes en cuanto a la calidad de la imagen y la velocidad de carga.

2.13.5 El control PictureBox.

El control PictureBox se utiliza para presentar gráficos, para actuar como contenedor de otros controles y para presentar el resultado de los métodos gráficos o texto con el método Print.

El control PictureBox es similar al control Image en que los dos se pueden usar para presentar gráficos en una aplicación y los dos aceptan los mismos formatos gráficos. Sin embargo, el control PictureBox tiene funcionalidades que no tiene el control Image; por ejemplo, la posibilidad de actuar como contenedor de otros controles y de aceptar métodos gráficos.

2.13.6 Formatos gráficos aceptados.

El control PictureBox puede presentar archivos de imágenes en los formatos siguientes: mapa de bits, icono, metarchivo, metarchivo mejorado o archivos JPEG o GIF.

2.13.7 Cargar un gráfico en el control PictureBox.

Podemos cargar imágenes en el control PictureBox en tiempo de diseño si seleccionamos la propiedad Picture en la ventana Propiedades del control o en tiempo de ejecución, si utilizamos la propiedad Picture y la función LoadPicture.

```
Set Picture1.Picture = _  
LoadPicture("c:\Windows\Winlogo.cur", vbLPLarge, vbLPColor)
```

Es posible que queramos usar archivos de iconos (.ico) y cursores (.cur) que contengan imágenes separadas a diferentes tamaños y profundidades de color para soportar una gama de dispositivos de visualización. Los valores de la función LoadPicture nos permiten seleccionar imágenes con determinados tamaños y profundidades de color de un archivo .ico o .cur. En los casos en los que no hay disponible una pareja exacta de los valores requeridos, LoadPicture carga la imagen con los valores más próximos disponibles.

Para borrar un gráfico del control PictureBox, se utiliza la función LoadPicture sin especificar el nombre de archivo. Por ejemplo:
Set Picture1.Picture = LoadPicture

De esta forma se borra el contenido del control PictureBox incluso si el gráfico se cargó con la propiedad Picture en tiempo de diseño.

2.13.8 Usar el Portapapeles.

También podemos agregar un gráfico a un control PictureBox en tiempo de diseño si lo pegamos desde otra aplicación. Por ejemplo, podemos agregar una imagen de mapa de bits creada con Windows Paint. Simplemente copiando la imagen al Portapapeles, seleccionando el control PictureBox y utilizando la combinación de teclas Ctrl-V o el comando Pegar del menú Edición.

2.13.9 Ajustar el tamaño de una imagen.

De forma predeterminada, los gráficos se cargan en un cuadro de imagen con su tamaño original, lo que significa que si el gráfico es mayor que el control, la imagen aparecerá recortada, ya que el control PictureBox no proporciona barras de desplazamiento. Para hacer que el control PictureBox cambie su tamaño automáticamente para presentar un gráfico completo, hay que establecer la propiedad AutoSize a True. Para ajustarse al gráfico, el control cambiará de tamaño; es decir, aumentará o disminuirá.

Al contrario que el control Image, el control PictureBox no puede cambiar el tamaño de la imagen para ajustarla al tamaño del control.

Crear una vista de gráficos desplazable", anteriormente en este mismo capítulo.

Usar el control PictureBox como contenedor

Se puede usar el control PictureBox como contenedor de otros controles. Por ejemplo, como el cuadro de imagen puede colocarse dentro del área interna de un formulario MDI, se suele usar para crear manualmente barras de herramientas o barras de estado.

2.13.10 Métodos gráficos.

Los cuadros de imagen, como los formularios, se pueden usar para recibir el resultado de métodos gráficos como Circle, Line y Point. Por ejemplo, podemos usar el método Circle para dibujar un círculo en un cuadro de imagen si establecemos la propiedad AutoRedraw del control a True.

Picture1.AutoRedraw = True

Picture1.Circle (1200, 1000), 750

Al establecer `AutoRedraw` a `True` conseguimos que el resultado de estos métodos se dibuje en el control y que se vuelva a dibujar automáticamente cuando el control `PictureBox` cambia de tamaño o se vuelve a presentar después de estar oculto detrás de algún otro objeto.

2.13.11 Usar el método Print

Podemos usar el control `PictureBox` para presentar texto si utilizamos el método `Print` y establecemos la propiedad `AutoRedraw` a `True`. Por ejemplo:

```
Picture1.Print "Una cadena de texto"
```

Cuando utilicemos el método `Print`, también podemos modificar el estilo y el tamaño de la fuente o emplear las propiedades `CurrentX`, `CurrentY`, `Height` y `Width` para alinear el texto dentro del cuadro de imagen.

2.14 El Objeto FileSystemObject (FSO) de Visual Basic 6.0

Al programar en Windows, es muy importante poder agregar, mover, cambiar, crear o eliminar carpetas (directorios) y archivos, y obtener información y gestionar unidades.

Visual Basic nos permite trabajar con unidades, carpetas y archivos de dos formas diferentes: mediante métodos tradicionales como las instrucciones `Open`, `Write#` y así sucesivamente, y mediante un nuevo conjunto de herramientas, el modelo de objetos del sistema de archivos (FSO).

El modelo de objetos del sistema de archivos (FSO) es una nueva característica de Visual Basic, que proporciona una solución basada en objetos para trabajar con carpetas y archivos. Esto nos permite usar la sintaxis familiar `objeto.método` con un conjunto extenso de propiedades, métodos y eventos para procesar carpetas y archivos, además de usar las instrucciones y comandos tradicionales de Visual Basic.

El modelo de objetos FSO proporciona a las aplicaciones la posibilidad crear, modificar, mover y eliminar carpetas, o detectar si existen carpetas en particular, y si existen, dónde. También nos permite obtener información acerca de las carpetas, sus nombres, la fecha cuando se crearon o la fecha de la última modificación, etc.

El modelo de objetos FSO también facilita el procesado de archivos. Al procesar archivos, el principal objetivo es almacenar datos en un espacio con un formato eficiente con los recursos y fácil de tener acceso. Necesitamos ser capaces de crear archivos, insertar y cambiar datos y escribir los datos. Mientras tanto, almacenar datos en una base de datos, como Jet, SQL ó Access, agrega una sobrecarga importante a la aplicación. Por muchas razones, puede que no se desee esta sobrecarga, o su acceso a los datos puede que no necesite todas las características extras asociadas con una base de datos con características plenas. En

este caso, almacenar los datos en un archivo binario o de texto puede ser la solución más eficiente.

El modelo de objetos FSO, que está contenido en la biblioteca de tipos Scripting (Srrun.Dll), permite la creación y gestión de archivos de texto mediante el objeto TextStream. Sin embargo, todavía no permite la creación o gestión de archivos binarios. Para manipular archivos binarios, se usa el comando Open con el indicador Binary.

El modelo de objetos FSO tiene estos objetos:

Objeto	Descripción
Drive	Nos permite obtener información acerca de las unidades del sistema, el espacio disponible, el nombre de la unidad compartida, etc. Hay que tener en cuenta que "unidad" no es necesariamente un disco duro. Puede ser una unidad de CD-ROM, un disco RAM, etc.
Folder	Nos permite crear, eliminar o mover carpetas, además nos permite consultar al sistema para conocer sus rutas y nombres.
Files	Nos permite crear, eliminar o mover archivos, además nos permite consultar al sistema para conocer sus rutas y nombres.
FileSystemObject	El objeto principal del grupo, repleto de métodos, nos permite crear, eliminar y obtener información y generalmente controlar unidades, carpetas y archivos. Muchos de los métodos asociados con este objeto duplican los de los otros objetos.
TextStream	Nos permite leer y escribir archivos de texto.

Tabla 2.2 Objetos del modelo de objetos FSO

Programar con el modelo de objetos FSO implica tres tareas principales:

- Usar el método CreateObject, o dimensionar una variable como un objeto FileSystemObject para crear un objeto FileSystemObject.
- Usar el método adecuado en el objeto que acaba de crear.
- Tener acceso a las propiedades del objeto.

El modelo de objetos FSO está contenido en una biblioteca de tipos denominada Scripting, situada en el archivo Srrun.Dll. Si no se tiene establecida una referencia a él, hay que seleccionar "Microsoft Scripting Runtime" en el cuadro de diálogo Referencias del menú Propiedades. Podemos usar el Examinador de objetos para ver sus objetos, colecciones, propiedades, métodos y eventos, así como sus constantes.

Crear un objeto FileSystemObject

El primer paso es crear un objeto FileSystemObject para trabajar con él. Podemos hacerlo de dos formas:

- Dimensionando una variable como del tipo FileSystemObject:
- Usar el método CreateObject para crear un objeto FileSystemObject:

Podemos trabajar con archivos en Visual Basic usando los nuevos objetos FSO orientados a objetos como Copy, Delete, Move y OpenAsTextStream, además de otros, o usando las antiguas funciones existentes como Open, Close, FileCopy, GetAttr, etc. Hay que tener en cuenta que podemos mover, copiar o eliminar archivos a pesar del tipo de archivo que este sea.

Hay dos categorías principales para trabajar con archivos:

- Crear, agregar o eliminar datos, y leer archivos.
- Mover, copiar y eliminar archivos.

2.15 El objeto Internet Explorer de Microsoft Windows.

Desde los inicios de la Internet hasta nuestros días se han creado varios programas para acceder a la información que brinda ésta red de redes. A estos programas para acceder a Internet se les conoce como navegadores (en inglés Web browsers). Desde el primer navegador más popular, el Mosaic.

En la historia de los navegadores podríamos mencionar a muchos de éstos, pero sin lugar a duda la aparición del navegador Internet Explorer marcó la diferencia. Éste navegador, desarrollado por Microsoft, fue introducido al mercado junto al lanzamiento del sistema operativo Windows 95 en el PLUS pack en el año 1995. Microsoft encontró la mejor forma de difundir su navegador incluyéndolo como un programa más de su sistema operativo. No pasó ni un año y el Internet Explorer había dejado atrás a muchos navegadores y se había situado en el tercer lugar de los navegadores más populares.

En aquellos años de 1995 hasta 1998, lideraba el mercado el famoso navegador de ese entonces; el Netscape, llamado también Communicator. Éste era un navegador libre de uso y fue el archirival de Internet Explorer hasta 1999, cuando sucedió lo previsto, Internet Explorer se coronó como el más popular y así se mantiene hasta nuestros días siendo usado por casi el 85% del mercado.

El Internet Explorer es el navegador de Microsoft, que luego de varias versiones de prueba y versiones estables ha llegado a consolidarse como el navegador por excelencia para los sistemas operativos de Microsoft, se ejecuta muy bien en estos sistemas, pues no podía -

ser de otra forma; es el navegador de la misma empresa. Las versiones del Internet Explorer siempre han venido asociadas a las versiones de los sistemas operativos de Microsoft.

Internet Explorer 1.0	Ésta fue la primera versión que vino junto con el lanzamiento de Windows 95 en el Plus pack. Fecha: Agosto de 1995
Internet Explorer 2.0	Liberada en Noviembre de 1995 y fue asociada al sistema operativo Windows NT 4.0
Internet Explorer 3.0	Publicada en Agosto de 1996. Se agregaron nuevas especificaciones del lenguaje HTML de ese entonces.
Internet Explorer 4.0	Ésta fue publicada en Octubre de 1997 y vino junto al tan esperado sistema operativo Windows 98.
Internet Explorer 5.0	Ésta versión fue una de las más estables y gustó mucho por su rapidez y a la vez es muy ligera. Publicada en marzo de 1999. Viene junto al Windows 98SE (Segunda Edición) y al Windows 2000.
Internet Explorer 5.5	En Julio del 2000 sale esta versión, que venía junto al Windows Millennium Edition (ME)
Internet Explorer 6.0	En octubre del 2001 publican esta versión que viene integrada al sistema operativo Microsoft Windows XP.
Internet Explorer 7	Microsoft anunció en Mayo del 2003 que ya no publicaría más versiones del Internet Explorer como programa independiente, sino que a partir de ahora sólo saldrían versiones del Internet Explorer junto a los nuevos sistemas operativos.

Tabla 2.3 Versiones del Internet Explorer.

Adicionalmente de poder utilizar el Objeto Internet Explorer para visualizar páginas web, otro uso que le podemos dar, conociendo todas y cada una de las funciones y métodos que este objeto presenta, es el de realizar labores propias de la navegación (como ir a la página anterior, siguiente, actualizar información, etc.) con la ayuda de un lenguaje de programación como en este caso Microsoft Visual Basic.

Las propiedades, métodos y sucesos del objeto Internet Explorer quedan expuestos en Visual Basic, haciendo referencia a ellos desde el menú.

Para observar la sintaxis de cada uno de éstos métodos y sucesos basta con presionar la tecla F2 dentro de un proyecto desarrollado en Visual Basic y el aspecto mostrado en la pantalla será el siguiente:



Figura 2.2 El examinador de objetos de Visual Basic.

Algunos de los métodos y propiedades más comunes que se utilizan con el objeto Internet Explorer son:

2.15.1 El método Navigate.

El método Navigate abre la URL especificada, que puede ser una dirección de Internet ó un documento HTML, almacenado en algún lugar del sistema. El argumento Flags especifica donde se debe añadir este URL; a la lista historial de Internet Explorer ó a la memoria caché del disco.

Los argumentos TargetFrameName, postData y Headers describen cómo se deben de abrir los documentos HTML y cómo se deben identificar en la ventana del navegador (los argumentos TargetFrameName, postData y Headers todos son opcionales). Aunque puede parecer complicado, el método Navigate es muy fácil de utilizar. En muchos casos es todo lo que se necesita para visualizar un documento HTML desde determinada aplicación.

2.15.2 La propiedad LocationURL.

La propiedad LocationURL contiene el nombre de ruta del documento HTML que se encuentra abierto en el Internet Explorer. Si se desea llevar un registro de todos los servidores web que visita un usuario en una sesión de trabajo, se podrá utilizar el método Navigate para copiar la cadena contenida en la propiedad LocationURL a un cuadro de texto o combo cada vez que se realice con éxito una conexión a una nueva página web.

2.15.3 El suceso NavigateComplete2.

El objeto Internet Explorer utiliza el suceso NavigateComplete2 cuando el método Navigate haya finalizado con éxito y se haya cargado un nuevo documento HTML en la ventana del navegador.

2.15.4 Propiedad AddressBar.

Controla si la dirección de la URL es mostrada.

2.15.5 El evento BeforeNavigate2.

Este evento ocurre mientras no se este solicitando la carga de alguna URL.

2.15.6 Propiedad Busy.

Esta propiedad averigua si alguna tarea esta aún en progreso.

2.15.7 Propiedad GoBack.

Esta propiedad es útil para implementar el procedimiento cuando queremos regresar a la página previa de navegación.

2.15.8 Propiedad GoForward .

Esta propiedad es útil para implementar el procedimiento cuando queremos ir a la página siguiente de navegación.

2.15.9 Propiedad GoHome.

Esta propiedad es útil para implementar el procedimiento cuando queremos ir a la página principal de navegación.

2.15.10 Propiedad Refresh.

Refresca el contenido de la página actual, es decir actualiza la información presentada en ella.

2.15.11 Propiedad Stop.

Con esta propiedad implementamos el procedimiento para detener la carga de la página que queremos mostrar.

Cada vez que Microsoft mejora el modelo de objetos de Internet Explorer suele crear versiones actualizadas de las propiedades, métodos y sucesos existentes. Para ello, añade nuevos números (incrementados) a sus nombres. Observando la sintaxis y descripción de estos nuevos miembros se puede determinar con frecuencia los nuevos argumentos y funciones que soportan. Este es el motivo por el cuál se pueden observar métodos como `Navigate`, `Navigate2`, `NavigateComplete2`, etc.

2.16 Archivos de Ayuda en Visual Basic.

El archivo de ayuda es importante en una aplicación Visual Basic ya que si se tiene previsto que el usuario de la aplicación pueda sentir alguna inquietud en el manejo de cierto control ó tema en específico, con la ayuda proporcionada está problemática quedará resuelta.

Agregar compatibilidad para la Ayuda a la aplicación de Visual Basic es bastante sencillo. Basta con que se establezca la propiedad `HelpFile` (y, por supuesto, escribir y compilar un archivo de Ayuda, para más detalles ver: "creación del Archivo de Ayuda: el programa `Salomón.exe`" en el capítulo 3) para poder mostrar la Ayuda cuando el usuario presiona la tecla `F1` o solicita Ayuda en un menú ó en un control. Es posible establecer una propiedad adicional, `HelpContextID`, para proporcionar un tema de Ayuda contextual para cualquier elemento de la interfaz de usuario de la aplicación. El proceso de implementación de la Ayuda es básicamente el mismo para `WinHelp` y para el formato `HTML`.

2.16.1 La propiedad `HelpFile`.

La propiedad `HelpFile` del objeto `App` permite especificar el nombre del archivo de Ayuda de la aplicación. Requiere un archivo de tipo `WinHelp` (`.hlp`) o `HTML` (`.chm`) válido. Si el archivo no existe, se producirá un error.

Para establecer la propiedad `HelpFile`

1. Hay que seleccionar la opción `Propiedades` del proyecto del menú `Proyecto` para abrir el cuadro de diálogo `Propiedades` del proyecto.
2. En el campo `Nombre del archivo de Ayuda` de la ficha `General`, escribir la ruta de acceso y el nombre de archivo del archivo de Ayuda de la aplicación (`.hlp` o `.chm`).

También podemos establecer la propiedad `HelpFile` por programa. El código siguiente puede especificar un archivo de Ayuda en formato `HTML` ubicado en el mismo directorio que el archivo ejecutable de la aplicación:

```
Private Sub Form_Load()  
    App.HelpFile = App.Path & "\Ayuda.chm"  
End Sub
```

El objeto ErrObject ofrece también una propiedad HelpFile, que permite especificar un archivo de Ayuda diferente para los mensajes de error. Por ejemplo, si tenemos varias aplicaciones que comparten los mismos mensajes de error, se puede ubicar la Ayuda correspondiente a los mensajes de error en un único archivo de Ayuda al que podemos invocar mediante la propiedad Err.Helpfile de cada aplicación.

2.16.2 La propiedad HelpContextID.

La propiedad HelpContextID nos permite vincular un elemento de la interfaz de usuario (como un control, un formulario o un menú) con un tema relacionado de un archivo de Ayuda. La propiedad HelpContextID debe ser un valor de tipo Long que coincida con el Id. de contexto de un tema en un archivo WinHelp (.hlp) o HTML (.chm).

Por ejemplo, podría escribir 10000 en la propiedad HelpContextID de un control TextBox. Cuando el usuario selecciona el control TextBox y presiona F1, Visual Basic busca un tema cuyo Id. de contexto sea 10000 en el archivo de Ayuda especificado por la propiedad HelpFile de la aplicación. Si lo encuentra, se abrirá una ventana de Ayuda con el tema en pantalla; si no lo encuentra, se producirá un error y se mostrará el tema predeterminado del archivo de Ayuda.

Se debe hacer coincidir un único valor de HelpContextID con cada tema de Ayuda del archivo de Ayuda. En ciertos casos, puede que deseemos asignar la misma propiedad HelpContextID a varios objetos si comparten un tema de Ayuda común.

No tenemos que escribir necesariamente un valor en la propiedad HelpContextID de cada control en un formulario. Si el usuario presiona F1 en un control con un valor de la propiedad HelpContextID de 0 (el valor predeterminado), Visual Basic buscará un valor de HelpContextID válido para el contenedor del control.

Para asignar la propiedad HelpContextID de un control o de un formulario

1. Seleccionar el control o el formulario para el que deseamos escribir una propiedad HelpContextID.
2. Hacer doble clic en la propiedad HelpContextID en la ventana Propiedades y escribir un entero de tipo Long válido.

Apuntar el valor escrito para usar el mismo valor en el Id. de contexto del tema de Ayuda asociado.

Nota Para el control CommonDialog y, posiblemente, para otros controles, el nombre de esta propiedad es HelpContext en vez de HelpContextID.

Para asignar un Id. de contexto a un menú:

1. Seleccionar Editor de menús en el menú Herramientas.
2. Elegir el elemento del menú para el que deseamos escribir una propiedad HelpContextID.
3. Escribir un valor de tipo Long válido en el cuadro HelpContextID.

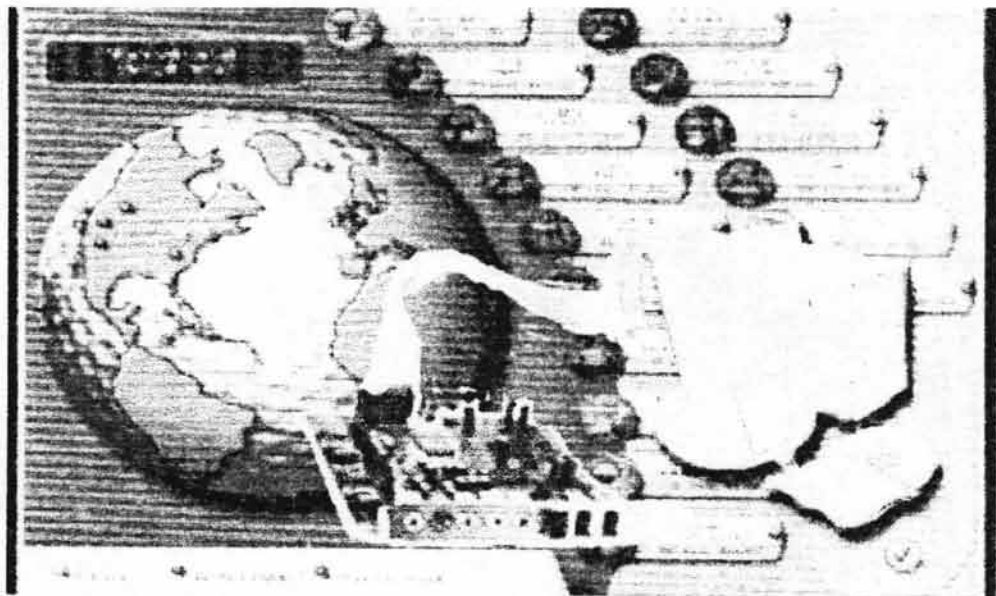
Apuntar el valor escrito para usar el mismo valor en el Id. de contexto del tema de Ayuda asociado.

También podemos establecer la propiedad HelpContextID mediante código de la manera siguiente:

```
Private Sub Form_Load()  
    Command1.HelpContextID = 12345  
    MenuHelp.HelpContextID = 23456  
    Err.HelpContext = 34567  
End Sub
```

CAPÍTULO TRES

Implementación y resultados del sistema de adquisición de datos



By JLJT

CAPITULO 3.

IMPLEMENTACIÓN Y RESULTADOS DEL SISTEMA DE ADQUISICIÓN DE DATOS.

3.1 Visión general del sistema de adquisición de datos.

Un sistema de adquisición de datos es un conjunto de elementos interrelacionados entre si con el fin de llevar a cabo una ó más tareas determinadas, estos elementos forman parte de una automatización.

El objetivo primordial de un sistema de adquisición de datos es precisamente por un lado, obtener valores analógicos, y por otro, automatizar las labores que se pudieran repetir constantemente, ó que representen alguna labor repetitiva y que no se pudieran llevar a cabo, simplemente por razones de distancia ó por el alto costo que representan para que se lleven a cabo.

Con el desarrollo de éste trabajo se pretende lograr implementar un sistema de adquisición de datos y control, que posea una interfase de uso basada en un programa propio y que pueda hacer uso de Internet, de modo que, permita realizar controles no sensibles en el tiempo, y que la distancia deje de ser un factor de peso. Esto se logrará construyendo una interfase por medio de un lenguaje de programación altamente utilizado en la actualidad como es: Microsoft Visual Basic Versión 6.0, así como del uso de una base de datos entre la parte servidor-monitor de control y el sistema de adquisición-cliente; el programa servidor es operado por medio de una interfase altamente manejable, que guarda las nuevas altas de registros en una base de datos hecha en Microsoft Access vinculada al lenguaje antes descrito mediante el uso de líneas de programación, al mismo tiempo que controla la adquisición y envío de datos haciendo uso de uno de los protocolos más utilizados y difundidos para el envío de archivos FTP por medio de Internet y el control de Microsoft Visual Basic: Microsoft Internet Transfer Control 6.0.

3.2 Convertidor analógico a digital ADC0808 configurado en modo de corrida libre.

Se alambra el convertidor de tal manera que el pulso de entrada inicio (Start) así como también la ALE sea alimentado por la salida EOC, en el momento en que una conversión está completa el ADC nos enviará un "alto" mismo que servirá para iniciar una nueva conversión (Start) y habilitar la lectura de los datos digitales obtenidos (ALE) lo único que debemos tomar en cuenta es que el reloj que tengamos operando en el convertidor deberá ser menor a los 500KHz. De esta manera estaremos creando un ciclo infinito de trabajo del ADC0808 para poder realizar las conversiones hasta que el sistema de adquisición de datos sea detenido.

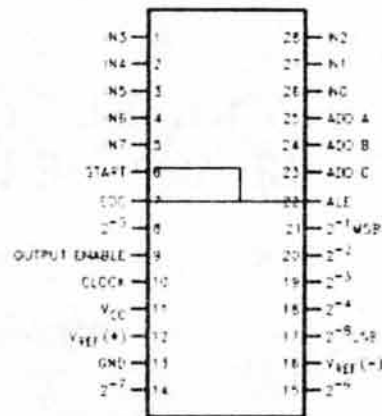


Figura 3.1 El ADC0808 funcionando en modo de corrida libre.

3.3 El termómetro electrónico, rango elegido e interpretación de la salida de datos digitales obtenidos en el ADC0808.

Se utiliza un sensor de temperatura de la Nacional Semiconductor, para crear un termómetro electrónico de precisión con un rango de 0 °C a 128 °C.

Para poder utilizar este sensor se soldó a una pieza metálica tubular para dar la apariencia de un verdadero termómetro digital; el rango elegido se determinó de la siguiente manera:

Según el fabricante (Anexo 8) la temperatura entregada por el sensor diferirá aproximadamente en 0.01 Grados Centígrados con respecto a la temperatura de la superficie. Esto quiere decir que la temperatura entregada por el sensor será aproximadamente igual a la temperatura de la superficie.

Además, nos especifica que 1 grado centígrado será igual a 10mv.

Considerando que proporcionamos un voltaje de referencia negativo Vref. (-) = 0v. al ADC y un voltaje de referencia positivo Vref. (+) = 1.28v. el convertidor estará en la disponibilidad de recibir, como valor máximo 1.28v. y como mínimo 0v.

Por otro lado en la hoja de datos del ADC0808 (Anexo 3) nos dice que si el Vref.(+)= X

Los cambios entre bit y bit de la cantidad digital obtenida serán: Vref. (+) / 256, ya que si calculamos el numero de pasos disponibles entre cambio y cambio en el convertidor, como este es de 8 bits el resultado es 256 (2^8). Haciendo operaciones, obtenemos lo siguiente:

$$\begin{aligned} \text{Vref. (+)} / 256 &= 1.28 / 256 \\ &= 0.005\text{v} = 5\text{mv} \end{aligned}$$

Por lo que, por cada paso que se lleve a cabo en las salidas digitales del ADC esto equivaldrá a la mitad de un grado (según las especificaciones del sensor de temperatura).

Entonces, para obtener la temperatura real debemos realizar la siguiente regla de tres:

$$\begin{aligned} 00000001. &= \frac{1}{2}^{\circ}\text{C} \\ \text{Salida Digital Obtenida} &= X \end{aligned}$$

3.4 El 555 configurado como multivibrador astable.

Se utiliza el integrado LM555 de la Nacional Semiconductor y se configura como oscilador astable (anexo 6 Pág.7) con el fin de aprovechar la característica de este circuito integrado de poder entregar una onda cuadrada que será el pulso de reloj necesario para el correcto funcionamiento del convertidor analógico a digital ADC0808.

Según las especificaciones del ADC0808 de la Nacional Semiconductor (Anexo 3) el convertidor analógico a digital necesita (a parte de otras señales de control) un pulso de reloj (Clock) para poder realizar las conversiones analógicas a digitales, como hemos configurado el conversor analógico a digital en modo de corrida libre, esto significa, que el convertidor se mantendrá en un ciclo infinito de conversiones hasta que nosotros manualmente detengamos el funcionamiento del sistema de adquisición de datos.

Para poder operar el convertidor analógico a digital ADC0808 en modo de corrida libre es necesario que la frecuencia de reloj no sobrepase de los 500Khz., ya que si esto sucede, no pasará el tiempo suficiente para que el multiplexor cargue la dirección correspondiente del canal elegido de manera correcta, mediante la patilla ALE, Cerrojo habilitador de dirección (Address Latch Enable).

El modo en que el LM55 fue configurado es el siguiente:

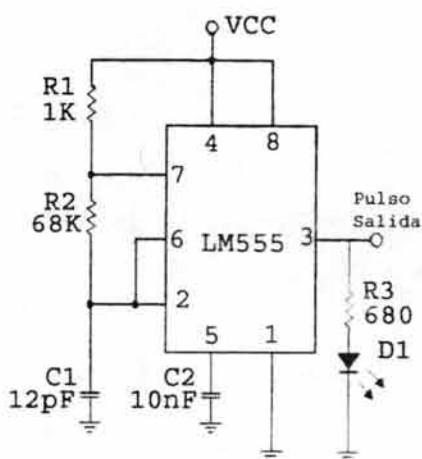


Figura 3.2 El LM555 configurado como oscilador astable.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

Los componentes que nos marcan las características de salida del LM555 (Figura 3.2) son:

$$R1 = 1 \text{ K}\Omega.$$

$$R2 = 68 \text{ K}\Omega.$$

$$C1 = 12 \text{ pF}.$$

Según la hoja de datos del LM555 (Anexo 6 Pág.8) para obtener la frecuencia de salida del LM555 configurado como oscilador astable tenemos las siguientes fórmulas:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B) C}$$

Fórmula para obtener la frecuencia de oscilación.

Realizando los cálculos correspondientes:

$$F = \frac{1}{T} = \frac{1.44}{(R1 + 2(R2)) C1} = \frac{1.44}{1 \times 10^3 + 2(68 \times 10^3) 12 \times 10^{-12}}$$

$$F = \frac{1}{T} = \frac{1.44}{(1 \times 10^3 + (136000)) 12 \times 10^{-12}}$$

$$F = \frac{1}{T} = \frac{1.44}{(137000) (12 \times 10^{-12})}$$

$$F = \frac{1}{T} = \frac{1.44}{0.000001644} = 875912.4088 \text{ Hz.}$$

Pasando a Khz.:

1Khz. = 1000 Hz.:

$$\frac{875912.4088}{1000} = 875.9124088 \text{ Khz.}$$

Como podemos observar, con los valores de los componentes dados (R1, R2 y C1) la frecuencia obtenida es mayor a la que puede trabajar el ADC0808 siendo éste configurado en modo de corrida libre, la razón de obtener esta frecuencia de salida es la siguiente:

Si calculamos algunos otros parámetros del pulso de salida obtenido tales como el ciclo de trabajo y sobre todo, el tiempo de descarga (salida en bajo) y el tiempo de carga (salida en alto) (Anexo 6 Pág.8) observamos que la señal obtenida no es del todo cuadrada, por lo cuál se puede ver afectado el funcionamiento del convertidor analógico a digital, ya que, puede ser que este trabaje bien después de un tiempo determinado, pero considerando que el sistema de adquisición de datos pudiera estar trabajando día y noche, es preferible proporcionarle una señal estrictamente cuadrada al ADC0808 y con esto obtener un óptimo funcionamiento y conversiones analógico a digitales y evitar un mal funcionamiento.

El tiempo de carga (salida en alto) esta dado por:

$$t_1 = 0.693 (R_A + R_B) C$$

Realizando operaciones:

$$t_1 = 0.693 (1 \times 10^3 + 68 \times 10^3) 12 \times 10^{-12}$$

$$t_1 = 0.0000005738 \text{ seg.} = 0.573804 \mu\text{seg}$$

Y el tiempo de descarga (salida en bajo) esta dado por:

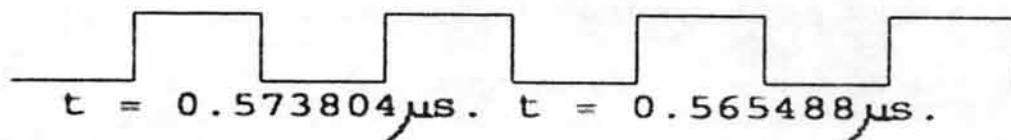
$$t_2 = 0.693 (R_B) C$$

Realizando operaciones:

$$t_2 = 0.693 (68 \times 10^3) 12 \times 10^{-12}$$

$$t_2 = 0.0000005654 \text{ seg.} = 0.565488 \mu\text{seg.}$$

El pulso de salida sería el siguiente:



Y el ciclo de trabajo será:

$$D = \frac{R_B}{(R_A + 2R_B)} = \frac{68 \times 10^3}{(1 \times 10^3 + 2(68 \times 10^3))}$$

$$D = 0.496350365 \times 100\% = 49.635$$

Observamos que el ciclo de trabajo es del 49.635 %, no del todo cuadrada.

3.5 El flip-flop JK DM74LS73A para generar pulso de reloj cuadrado-exacto.

La idea es retomar el pulso de salida del LM555 y pasarlo a través de un flip-flop para que dicho integrado convierta a la señal en un pulso de reloj totalmente cuadrado, como lo muestra la figura 3.3:

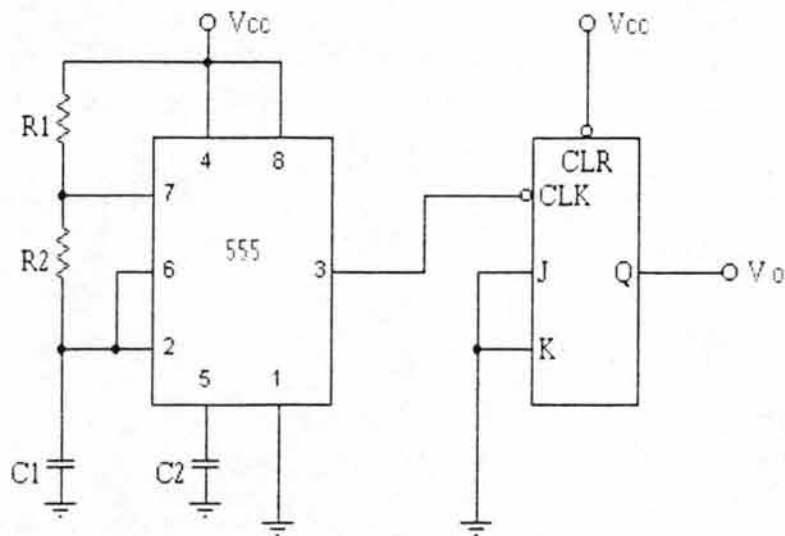
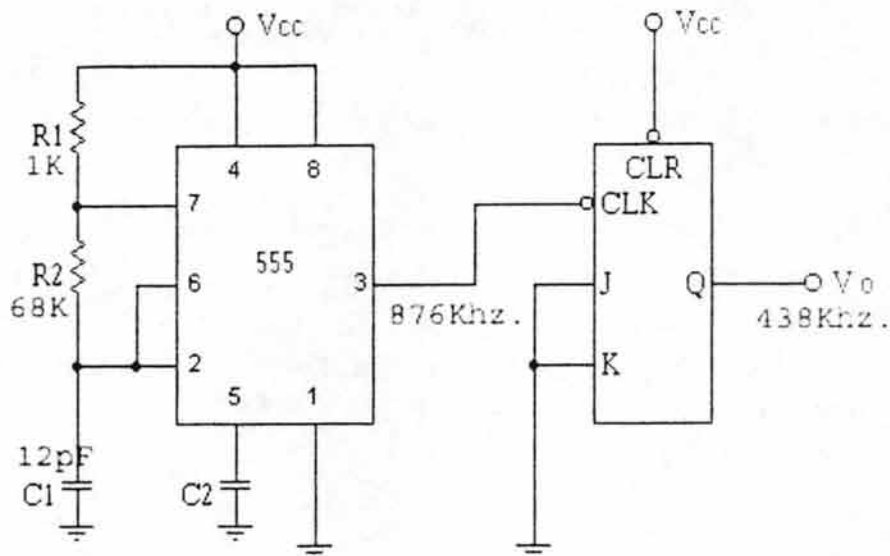


Figura 3.3 La señal V_o será un pulso de reloj totalmente cuadrado con ayuda del flip-flop tipo JK.

Para tal efecto se hace uso del flip-flop tipo JK DM74LS73A de la FairChild Semiconductor, el único inconveniente es, que la frecuencia obtenida a la salida del flip-flop, (en Q) se ve recortada a la mitad con respecto a la frecuencia de salida en el LM555 (patilla 3), de ahí la razón de obtener a la salida del LM555 una frecuencia de alrededor de 876 KHz., ya que al recortar esta frecuencia a la mitad en la salida del flip-flop obtendremos alrededor de los 438 KHz., un pulso de reloj exacto y sobre todo, podemos operar el convertidor analógico a digital ADC0808 en modo de corrida libre, como lo muestra la figura 3.4:



ESTA TESIS NO SALE
DE LA BIBLIOTECA

Figura 3.4 El pulso obtenido será permitido para operar el ADC0808 en modo de corrida libre.

3.6 El Voltaje de referencia del ADC0808.

El voltaje de referencia del ADC0808 es determinante para seleccionar el rango de trabajo que queremos emplear en el sistema de adquisición de datos. Se compone por 2 tipos de voltaje $V_{ref. (+)}$ y $V_{ref. (-)}$. Con lo que respecta al $V_{ref. (-)}$ simplemente conectamos la patilla que corresponde a ese voltaje (patilla 16) a tierra (GND) con el fin de indicarle al ADC0808 que el mínimo valor que puede convertir a digital será a partir de cero, es decir, si en las entradas analógicas del convertidor recibimos un voltaje de 0v. entonces a la salida del ADC en sus salidas digitales encontraremos "00000000"; por otro lado, el valor mayor a convertir de analógico a digital será lo que introduzcamos en el $V_{ref. (+)}$ (patilla 12), esto es, si introducimos en $V_{ref. (+)}$ una tensión = 5v., en el momento en que recibamos en cualquiera de los canales analógicos (previamente direccionado y cargado) los 5v. máximos permisibles, entonces lo que obtendremos a la salida del ADC0808 en sus salidas digitales será: "11111111".

3.7 Determinación de rangos de trabajo.

En el caso del termómetro digital, el rango de trabajo se determinó de la siguiente manera:

Se introdujo en el ADC0808 un $V_{ref. (-)}$ de 0v. (patilla 16)

Se introdujo en el ADC0808 un $V_{ref. (+)}$ de 1.28v. (patilla 12)

1 °C equivale a 10mV. (Según anexo 8 Pág.1)

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

Como el $V_{ref. (+)} = 1.28v.$ y el convertidor ADC0808 es de 8 bits, los máximos pasos permitidos son $2^8 = 256.$

Para calcular a cuanto equivale cada cambio en el ADC0808 realizamos lo siguiente:
 $1.28v. / 256 = 0.005v. = 5mV.$

Es decir, que por cada 2 cambios de señales digitales corresponderá a $1^{\circ}C.$

Ejemplo:

Si obtenemos a la salida digital del ADC0808: "00111100", que en el sistema decimal es igual a "60" la temperatura real será determinada por una regla de tres:

$$\begin{aligned} 11111111 &= 128^{\circ}C \\ 00111100 &= X \end{aligned}$$

Realizando operaciones:

$$(128) * (60) / 256 = 30^{\circ}C$$

Con este ejemplo nos damos cuenta que si dividimos el valor digital entregado por el ADC0808 entre 2 obtendremos la temperatura real registrada en ese momento, ya que, como habíamos visto anteriormente los 5 mV. que corresponden a cada paso en el ADC0808 significan $\frac{1}{2}$ de grado Centígrado.

Esta sencilla operación será llevada a cabo mediante el software del sistema de adquisición de datos, como lo veremos más adelante.

3.8 El buffer SN74LS244.

Este buffer cuenta hasta con 8 posibles canales de entrada, para el caso del sistema de adquisición de datos, sólo fueron empleados tres canales de entrada con el objetivo de ampliar la entrega de corriente por parte de la PC hacia el sistema de adquisición de datos, específicamente para el direccionamiento apropiado del canal analógico elegido del ADC0808, ya que eléctricamente, el puerto paralelo entrega señales TTL y como tal, está muy limitado en cuanto a su capacidad de manejo de corriente.

Cabe mencionar que para poder utilizar el nibble de orden superior hay que alambrear la patilla 1 a nivel "alto" y la patilla 19 a "bajo"; por otro lado si queremos trabajar con el nibble de orden inferior necesitamos alambrear la patilla 19 a un nivel "alto" y la patilla 1 a nivel "bajo", ó bien para habilitar permanentemente tanto el nibble de orden superior e inferior necesitamos conectar ambas patillas (1 y 19) a tierra (GND).

3.9 Análisis del sistema de adquisición de datos.

El diagrama del sistema de adquisición de datos es el siguiente:

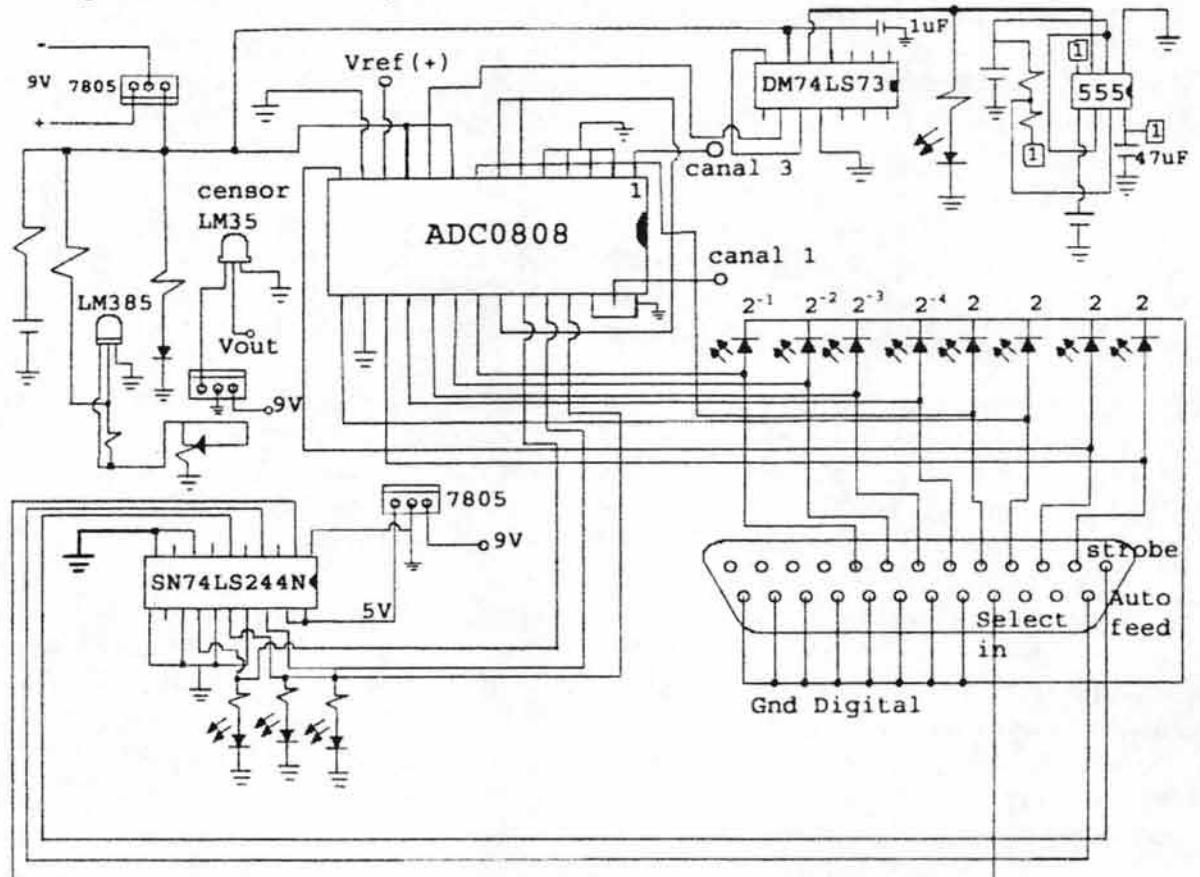


Figura 3.5 Diagrama general del sistema de adquisición de datos.

El sistema de adquisición de datos basa su funcionamiento en el convertidor analógico a digital ADC0808 y el sensor de temperatura LM35 ambos manufacturados de la Nacional Semiconductor, el proceso comienza, al configurar el ADC0808 en modo de corrida libre, para lograr esto conectamos juntos los pines 6, 7 y 22. (START, EOC y ALE respectivamente) En la patilla 9 (OE) introducimos 5V con el fin de que en todo momento este habilitada la salida de datos del convertidor. La patilla 10 (CLK) es muy importante, porque de este pulso de reloj se determinará la velocidad de respuesta del convertidor, dicho pulso de reloj se obtiene a partir de lo siguiente:

Se configura un LM55 en modo oscilador astable, es decir, que nos proporcione una señal cuadrada de reloj de la cuál, la frecuencia es obtenida mediante determinados valores de componentes externos, tales como resistencia y capacitores (para mayores detalles ver el tema: "3.4 El 555 configurado como multivibrador astable", en este mismo capítulo)

A partir de la frecuencia de pulso de reloj obtenida se procede a “limpiar” la señal haciéndola lo más cuadrada y exacta posible, esto es, con el objetivo de no perturbar el funcionamiento del convertidor cuando éste pueda ser operado en un lapso muy grande de tiempo, para lograr tal objetivo nos valemos del integrado DM74LS73, el cual se compone de dos flip-flops tipo JK este integrado nos entregará el pulso de reloj recortando la frecuencia a la mitad, con respecto a la que recibió de entrada, (876 Khz.) es decir, obtenemos un pulso de reloj aproximado de 438 Khz. listo para ser introducido a la patilla 10 (CLK) del ADC0808.

En la patilla 11 suministramos la alimentación del convertidor, mientras que en la patilla 13 (GND) la tierra necesaria, la patilla 12 (Vref (+)) es la encargada de establecer el rango de trabajo, es decir, dependiendo de la cantidad de voltaje suministrado a esa patilla se obtendrá el rango de trabajo, para obtener el voltaje de referencia necesario hacemos uso del LM385 variable el cuál puede ser modificado mediante el uso de un potenciómetro de 10 K Ω .

En las patillas 26, 27, 28, 1, 2, 3, 4 y 5 podemos introducir el valor analógico que deseamos convertir a digital, en este caso, el valor analógico se obtiene mediante el uso del sensor de temperatura LM35 en presentación metálica, igualmente manufacturado por la Nacional Semiconductor.

Las patillas 23, 24 y 25 (Add A, Add B y Add C, respectivamente) son los cerrojos habilitadores de dirección, los cuales son utilizados para elegir otro canal del convertidor, (para mayores detalles ver el tema: “1.8 El Convertidor Analógico a Digital ADC0808” en el capítulo 1. Dicho valor será proporcionado por el puerto paralelo que se este utilizando en ese momento específicamente los pines: Strobe (A), Autofeed (B) y Select in (C), en el bus de control de dicho puerto y con la ayuda del buffer integrado SN74LS244N, pues como ya vimos, proporcionará la corriente necesaria para el correcto redireccionamiento del convertidor.

El valor digital entregado por el ADC0808 será recibido por el puerto paralelo mediante su bus de datos, llevando a cabo, previamente, el redireccionamiento de dicho puerto, mediante la librería de vínculos dinámicos diseñada en el lenguaje de programación: Microsoft Visual C++.

Finalmente con el uso del software desarrollado en el lenguaje de programación: Visual Basic 6.0 se controlarán e interpretarán los valores biomédicos obtenidos y se podrá elaborar un archivo gráfico, el cuál se podrá enviar vía Internet a cualquier parte del mundo, mediante el protocolo de traslado de archivos FTP.

3.10 Puerto y tipo de bus a utilizar.

El puerto a utilizar es el LPT1 mejor conocido como 888 en sistema decimal y H378 en sistema hexadecimal.

Una PC soporta hasta tres puertos paralelo separados, por tanto puede haber hasta tres juegos de registros en un sistema en un momento dado. Existen tres direcciones base para el

puerto paralelo asociadas con tres posibles puertos paralelo: 0x3BCh, 0x378h y 0x278h, nos referimos a éstas como las direcciones base para el puerto LPT1, LPT2 y LPT3, respectivamente. El registro de datos se localiza siempre en la dirección base de un puerto paralelo, el registro de estado aparece en la dirección base + 1, y el registro de control aparece en la dirección base + 2. Por ejemplo, para un puerto LPT2 localizado en 0x378h, ésta es la dirección del registro de datos, al registro de estado le corresponde la dirección 0x379h y su respectivo registro de control está en la dirección 0x37Ah. Cuando la PC se enciende el BIOS ejecuta una rutina para determinar el número de puertos presentes en el sistema asignando la etiqueta LPT1 al primer puerto localizado, si existen más puertos entonces se asignarán consecutivamente las etiquetas LPT2 y LPT3, pero en ocasiones las direcciones pueden estar invertidas, por ejemplo habrá veces que el LPT1 corresponde con 0x378h en sistema hexadecimal pero en ocasiones corresponderá con el 0x632 esto va a depender de cómo se encuentre establecido el direccionamiento en la PC.

Para trabajar con el puerto paralelo necesitamos en primer lugar conocer la dirección base asignada por el BIOS (estamos hablando de una PC compatible con IBM), podemos utilizar un programa llamado `Debug.exe` que nos indique la(s) dirección(es) asignada(s): en la plataforma Windows vamos al menú *inicio*, seleccionamos *programas* y luego *MS-DOS* para abrir una ventana de *Símbolo de MS-DOS* y aquí podemos introducir los comandos indicados más abajo. Si se trabaja en ambiente DOS basta con teclear en la línea de comandos la palabra `debug`, el programa responde colocando un signo de menos - en donde tecleamos sin dejar espacios en blanco `d040:08L8` y presionamos la tecla *entrar*, entonces el programa `debug.exe` nos indica en una serie de números la(s) dirección(es) para el (los) puerto(s) paralelo(s) disponibles en nuestra computadora, la siguiente imagen muestra el resultado obtenido específicamente en la computadora que utilicé:

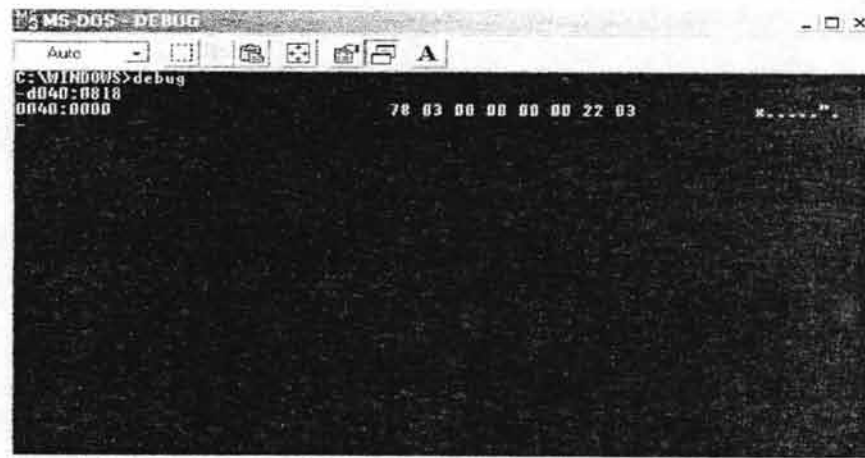


Figura 3.6 Resultado obtenido en mi computadora para observar los puertos disponibles dentro de la PC.

Se puede observar una serie de números de dos dígitos (ocho en total), se trata del volcado de memoria que empieza en la dirección 40:0008h. Los primeros seis pares de números representan las direcciones base para los puertos paralelo instalados, en la imagen de arriba se aprecia que el único puerto paralelo de mi máquina está en la dirección 0x378h (78 03). Los números están invertidos porque Intel almacena tal información en un formato de "byte de bajo orden - byte de alto orden". Una vez que obtenemos la información deseada cerramos el programa *Debug.exe* simplemente tecleando la letra q y presionando la tecla *entrar*. Para cerrar la ventana de *Símbolo de MS-DOS* tecleamos la palabra exit y presionamos la tecla *entrar*.

Para controlar al sistema de adquisición de datos haremos uso tanto del bus de datos como del bus de control, el bus de estado no va a ser necesario utilizarlo ya que una vez que el sistema de adquisición de datos este en modo de encendido éste se encontrará en un bucle infinito de conversión de datos analógico a digitales por parte del ADC0808, ya que como vimos anteriormente éste fue conectado en modo de corrida libre. Cabe mencionar que mediante software, específicamente haciendo uso del lenguaje visual Basic podemos darle opción a la aplicación para escoger otro tipo de puerto (por ejemplo el LPT2, H632 en sistema hexadecimal, 1586 en sistema decimal).

En este caso, como ya vimos que el LPT1 corresponde a la dirección 0xh888 en sistema hexadecimal (también conocido como puerto base), este corresponde al bus de datos del puerto, y el bus de control será el puerto base +2, es decir, $0xh888 + 2 = 0x37A$, 890 en sistema decimal. Ahora veremos como controlamos estos buses del puerto.

3.11 Metodología de programación de la aplicación Cliente.exe.

El modelo de programación de la aplicación Cliente es la siguiente:

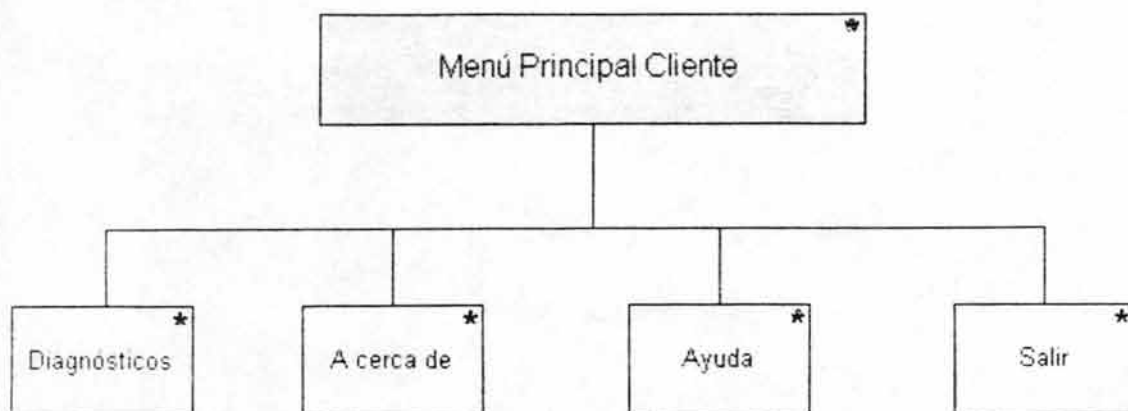


Figura 3.7 Modelo de programación para el menú principal de la aplicación Cliente.exe

Aquí podemos observar que al presentarnos el primer menú podremos elegir de entre cuatro opciones, dependiendo de la elección de cada una de ellas es el camino que tomará la aplicación.

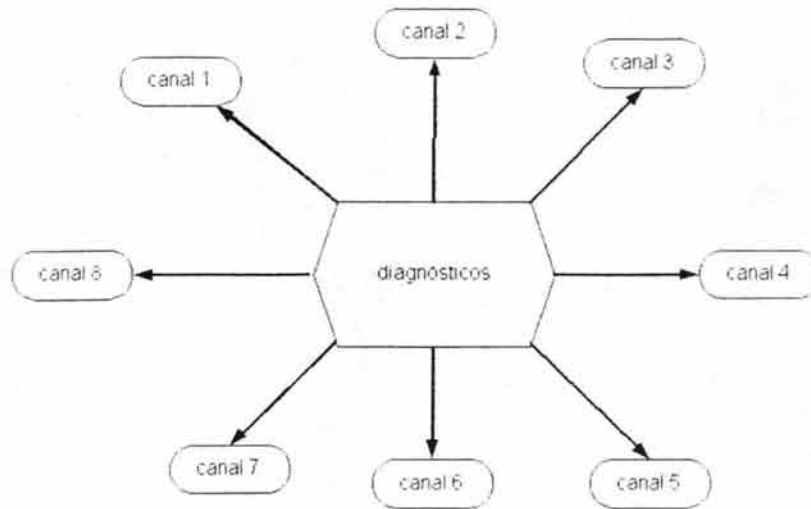


Figura 3.8 Representación de los canales disponibles del sistema de adquisición de datos.

Aquí podremos elegir entre los 8 canales disponibles del ADC0808, del cuál se utilizaron los canales 1 y 2 para la obtención de datos biomédicos

Las siguientes opciones dentro del sistema son:

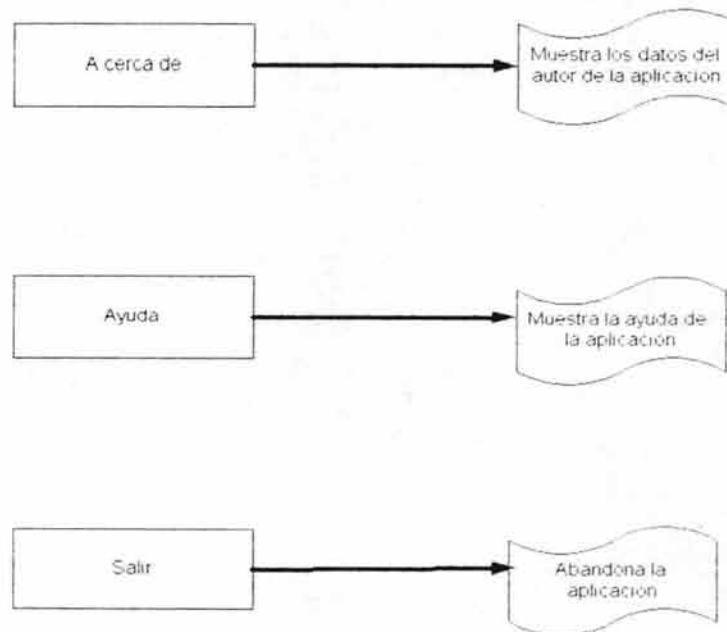


Figura 3.9 Otras opciones dentro del sistema de adquisición de datos.

Al elegir el canal 0 ó el canal 1 se llevará a cabo el redireccionamiento del puerto habilitando en el ADC0808 el canal correspondiente al canal elegido:

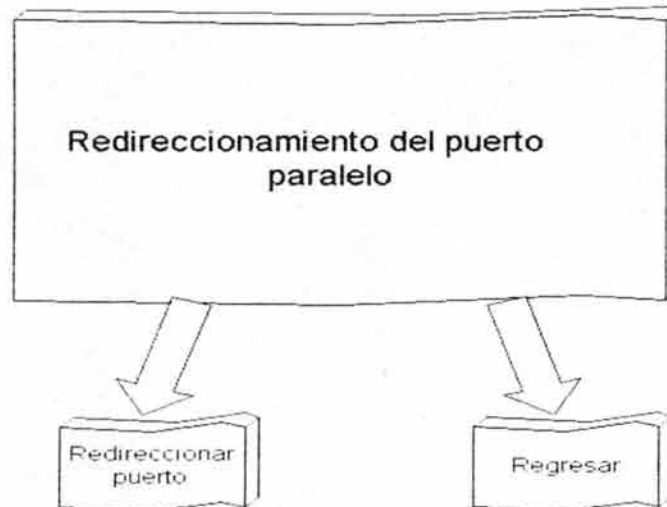


Figura 3.10 Redireccionamiento del puerto paralelo.

Al redireccionar el puerto paralelo, éste estará ahora en la disponibilidad de recibir valores en el bus de datos de dicho puerto elegido.

El canal a elegir está determinado por la cantidad de voltajes analógicos disponibles en el convertidor analógico a digital ADC0808, el siguiente diagrama del patillaje del ADC0808 nos muestra que los canales disponibles son 8 en total:

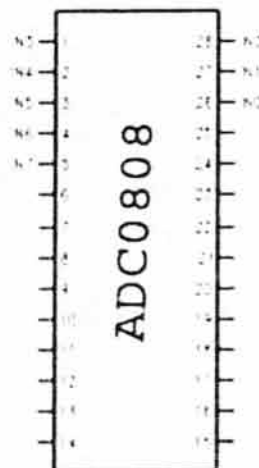


Figura 3.11 Canales analógicos de entrada disponibles en el ADC0808

Cuando obtenemos el dato biomédico tenemos diferentes opciones a elegir dentro de la aplicación Cliente.exe. Primeramente, lo que se deseará observar será la visualización del dato obtenido de modo gráfico para ello podemos elegir la opción mostrar gráfica, con la cual mostrará el dato biomédico obtenido en una representación gráfica en tiempo real. Además de esta opción podemos también almacenar el dato obtenido en nuestro disco duro, para que después pueda ser enviado, si es que así se desea.

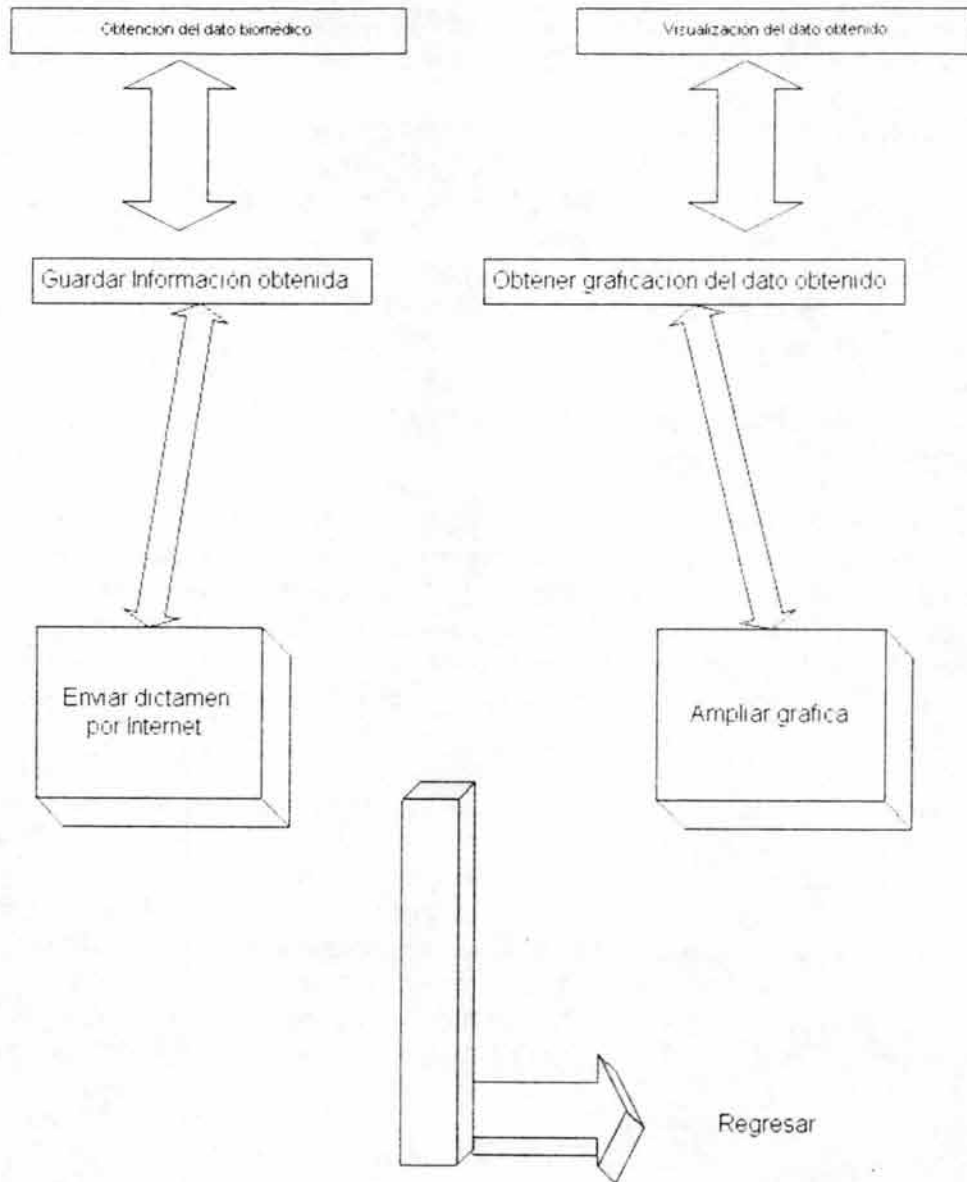


Figura 3.12 Opciones disponibles dentro de la pantalla de obtención del dato biomédico.

3.12 Redireccionamiento del puerto paralelo.

Como ya vimos, originalmente el puerto paralelo fue diseñado exclusivamente como sólo salida de datos (hablando de su bus de datos), es decir, que éstos solo pueden ir desde el interior de la PC hacia algún dispositivo externo. Existen algunos proyectos desarrollados, que, al no contar con un bus de datos de entrada en el puerto paralelo, se valen de los pines que pueden trabajar de modo entrada de datos, por ejemplo, utilizan del bus de estado: busy, acknowledge, paper out, error, select in; y de los pines del bus de control: select out, strobe, y printed reset, de los cuáles algunos de ellos pueden tanto recibir datos como transmitirlos.

La gran desventaja es que si los utilizamos para que la computadora pueda leer datos desde nuestro dispositivo externo, estaríamos restando capacidades a dichos pines, por ejemplo, si el dispositivo externo requiere algunas otras señales de control para realizar determinada tarea, además de que algunos de estos pines trabajan con la llamada lógica negativa, es decir, si nosotros recibiéramos un voltaje de 5V. éste podría representar ó un 0 ó un 1 lógico dependiendo de la manera en como trabaje dicho pin, lo que nos causaría cierta confusión, es por ello que la manera más práctica y efectiva es hacer que nuestro bus de datos sea redireccionado y con ello sea capaz de recibir datos y poder interpretarlos, sin tener que hacer uso de los otros pines que nos podrán ser útiles para realizar otra tarea y también evitar la lógica negativa.

Para poder llevar a cabo el redireccionamiento del puerto paralelo nos valdremos de instrucciones del lenguaje Visual C++, las cuáles nos permiten trabajar con los puertos de la computadora, específicamente el bus de control, que es en donde podemos escribir en su bit 5 pasándolo de 0 a 1 lógico y con ello, el bus de datos del puerto paralelo estará listo para leer datos desde el dispositivo externo hacia el interior. La codificación la podemos ver en el siguiente capítulo llamado: "3.13 Código fuente de la librería de vínculos dinámicos "REDIRECC.DLL".

Finalmente, es importante hacer notar que previamente se necesitará conocer la dirección de memoria de nuestro puerto paralelo, y si tenemos más de un puerto, tenemos que saber la dirección del cuál estemos trabajando. Por ejemplo: Si tenemos el puerto paralelo con dirección 888 (H378 en hexadecimal) entonces el bus de datos será el 889 (H379 en hexadecimal) y el bus de control el 890 (H37A) que es precisamente en el cuál debemos hacer el redireccionamiento. La dirección en donde empieza a trabajar el puerto paralelo (bus de datos) es llamada: dirección BASE + 0, la dirección del bus de estado es BASE + 1 y finalmente la dirección del bus de control será BASE + 2, de ahí que vayan aumentando en la unidad desde 888, 889 y 890.

Para habilitar uno de los canales del ADC0808 es necesario implementar una función, la cual permita mandar el dato necesario y se lleve a cabo, satisfactoriamente el redireccionamiento del canal.

3.13 Código fuente de la librería de vínculos dinámicos “REDIRECC.DLL”

Mediante el lenguaje de programación Microsoft Visual C++, el cuál trabaja con múltiples archivos de proyecto que le serán de utilidad para que se compile apropiadamente algún proyecto .dll, se puede realizar una librería capaz de interrelacionarse con el lenguaje que utilizaremos para la creación de la aplicación: Microsoft Visual Basic. En este caso el tipo de proyecto que se ha realizado sobre Visual C++ es de tipo Win 32 Dynamic-Link library el cual es una librería de vínculos dinámicos, la razón por la que se hizo dicha librería es porque el lenguaje de programación Visual Basic 6.0, no nos permitía el manejo del puerto paralelo para introducir los datos desde el dispositivo externo hacia la PC. Hay que hacer notar que Visual C++ genera el código necesario automáticamente, por ejemplo, para el de manejo de rutinas que utilizara el proyecto, y nosotros únicamente nos centramos en la creación de las rutinas o funciones necesarias para el redireccionamiento del puerto paralelo.

Por otra parte, el bus de control es bidireccional: es decir se pueden modificar sus bits tanto por software, como por hardware (señales de voltajes) conectadas al puerto paralelo. Este puerto controla 4 pines, mediante sus bits 3, 2, 1 y 0. La tabla de este puerto es la siguiente:

Nº DEL BIT	Nº DEL PIN	NOMBRE DEL PIN
7	-	
6	-	
5	-	DIRECTION
4	-	IRQ ENABLE
3	17 *	SELECT IN
2	16	INIT
1	14 *	AUTOFEED
0	1 *	STROBE

Tabla 3.1 Bit's disponibles dentro del bus de control del puerto paralelo.

Nota: Los pines que aparecen en la tabla marcados con un asterisco, trabajan con lógica negativa.

De acuerdo con esto entonces estamos leyendo un 0 en strobe, un 0 en Autofeed, un 1 en INIT y un 0 en Select in, un 0 en IRQ Enable, un 0 en Dirección, (que es el que nos interesa) y finalmente el bit 6 y 7 no se utilizan.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

Entonces podemos observar que el bus de datos es ahora de solo salida, (el bit 5 esta en 0) antes de llevar a cabo el redireccionamiento.

El código fuente principal de dllredirecc.dll (con extensión .cpp) es el siguiente:

// Este código es generado automáticamente por Visual C++ para el manejo de las funciones o subrutinas a utilizar.

// dllRedirecc.cpp : Defines the initialization routines for the DLL.

#include "stdafx.h"

#include "dllRedirecc.h"

#ifdef _DEBUG

#define new DEBUG_NEW

#undef THIS_FILE

static char THIS_FILE[] = __FILE__;

#endif

//

// Note!

//

*// If this DLL is dynamically linked against the MFC
// DLLs, any functions exported from this DLL which
// call into MFC must have the AFX_MANAGE_STATE macro
// added at the very beginning of the function.*

//

// For example:

//

// extern "C" BOOL PASCAL EXPORT ExportedFunction()

// {

// AFX_MANAGE_STATE(AfxGetStaticModuleState());

// // normal function body here

// }

//

*// It is very important that this macro appear in each
// function, prior to any calls into MFC. This means that
// it must appear as the first statement within the
// function, even before any object variable declarations
// as their constructors may generate calls into the MFC
// DLL.*

//

*// Please see MFC Technical Notes 33 and 58 for additional
// details.*

```

//
/////////////////////////////////////////////////////////////////
// CDllRedireccApp

BEGIN_MESSAGE_MAP(CDllRedireccApp, CWinApp)
//{{AFX_MSG_MAP(CDllRedireccApp)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////
// CDllRedireccApp construction

CDllRedireccApp::CDllRedireccApp()
{
// TODO: add construction code here,
// Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////
// The one and only CDllRedireccApp object

CDllRedireccApp theApp;

/Aquí empiezan las funciones para redireccionamiento del puerto
//*****
// dllredirecc.cpp
// ©2003, José Luis Juárez Trejo
//Funciones de la dll
//*****
#include <conio.h> //necesaria para _outp e _inp
extern "C"
int WINAPI Redirecc(int puerto)//función para redireccionar puerto paralelo
{
if (puerto = 888) //si el DATAPORT es el 888 (H378)
{
puerto = 0x37A; //es el valor correspondiente para el CONTROLPORT (890 BASE + 2)
    aquí el valor para redireccionar puerto
    aquí el valor para redireccionar puerto
}
if (puerto = 1586) //si el DATAPORT es el 1586 (H632 BASE + 0)
{

```

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

```
puerto = 0x634; //es el valor correspondiente para el CONTROLPORT (1588 BASE + 2)
    'aquí el valor para redireccionar puerto
    'aquí el valor para redireccionar puerto
}
return(0);
};
int WINAPI MandarDato(int puerto, int dato)//función para mandar datos
{

    _outp(puerto,dato);

    return (dato);
};
```

Para elegir otro canal del ADC0808 utilizamos el bus de control del puerto que se este utilizando, específicamente los pines: strobe, (MSB) Autofeed y Select in (LSB)

Es importante hacer notar en todo momento que algunos de estos pines trabajan con lógica negativa por lo que se tendrá que poner extremo cuidado con este tipo de pines a la hora de enviar datos.

Regresando a la función, observemos que esta requiere ó necesita dos parámetros para enviar el dato hacia el puerto.

El primer parámetro (int puerto) necesitamos especificar un valor que dependerá de la dirección del puerto que estemos usando, por ejemplo, si se trata del 888 (H378 en hexadecimal), entonces el bus de control será el 890 (H37A en hexadecimal), por otro lado si se trata de la dirección 1586 (H632 en hexadecimal) el bus de control será H634.

El otro parámetro (dato) será el valor necesario para elegir otro canal del ADC0808, aquí tendrá lugar el buffer SN74LS244N como abastecedor de la corriente necesaria para que el redireccionamiento del canal específico sea llevado de manera satisfactoria. Para observar como se conecta el buffer SN74LS244N abastecedor de corriente, referirse al tema 3.9: "Análisis del sistema de adquisición de datos.", en este mismo capítulo.

Los valores a utilizar para elegir otro canal en el ADC0808 fueron elegidos tomando en cuenta la lógica negativa con la que trabajan algunos pines y el valor resultante que obtendríamos al mandar determinado valor, dichos valores que serán enviados a través de los pines del bus de control, están los mostrados en la tabla 3.2:

Valor hexadecimal	Valor decimal	Salida	Canal habilitado
HB	11	0 0 0	1
H7	7	0 0 1	2
H9	9	0 1 0	3
H5	5	0 1 1	4
HA	10	1 0 0	5
H2	2	1 0 1	6
H8	8	1 1 0	7
H0	0	1 1 1	8

Tabla 3.2 Valores a enviar para habilitar canales del ADC0808

Analicemos un par de valores, por ejemplo cuando mandamos el valor H2 y el valor H9 hacia el bus de control del puerto paralelo utilizado, consideremos que el puerto que se está utilizando es el LPT888, por ser el más común (el bus de control será entonces: H37A)

Utilizando la función:

```
_outp(puerto,dato);
```

Sustituyendo valores:

```
puerto = H37A  dato = H2
_outp(H37A, H2);
```

Es decir, con estos valores, enviaría hacia el bus de control el siguiente valor en binario:
0 0 0 0 0 1 0

Pero, considerando que algunos bits del bus de control trabajan con lógica negativa como son: STROBE, AUTOFEED Y SELECT IN, bits 0, 1 y 3 respectivamente, la salida quedará de la siguiente manera:

```
0 0 0 0 1 0 0 1
```

Como se están utilizando en el sistema de adquisición de datos los pines:

STROBE.- Para el MSB del ADC0808, es decir, el ADD C. (Anexo 3 Pág.2)

AUTOFEED.- Para el bit medio del ADC0808 es decir el ADD B. (Anexo 3 Pág.2)

SELECT IN.- Para el bit LSB del ADC0808 es decir, ADD A. (Anexo 3 Pág.2)

Tomando los datos enviados el valor obtenido es:

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

1 0 1 (el valor correspondiente para habilitar el canal 6)

De la misma manera con el valor H9:

Nuevamente, utilizando la función:

```
_outp(puerto,dato);
```

Y sustituyendo valores:

```
puerto = H37A dato = H9
```

```
_outp(H37A, H9);
```

El valor en binario obtenido sería:

0 0 0 0 1 0 0 1

Igualmente, considerando que algunos bits del bus de control trabajan con lógica negativa (STROBE, AUTOFEED Y SELECT IN), bits 0, 1 y 3 respectivamente, la salida quedará entonces, de la siguiente manera:

0 0 0 0 0 0 1 0

Y nuevamente, utilizando los pines como se utilizaron en el sistema de adquisición de datos:

STROBE.- Para el MSB del ADC0808, es decir, el ADD C. (Anexo 3 Pág.2)

AUTOFEED.- Para el bit medio del ADC0808. es decir el ADD B. (Anexo 3 Pág.2)

SELECT IN.- Para el bit LSB del ADC0808. es decir, ADD A. (Anexo 3 Pág.2)

Tomando los datos enviados el valor obtenido es:

0 1 0 (el valor correspondiente para habilitar el canal 3)

Observando los valores de la tabla 1 con los valores obtenidos el resultado es correcto.

La siguiente función es:

```
int WINAPI LeerDato(int puerto)//función para leer datos
{
    int dato;
    if (puerto = 888)
    {
        puerto = 0x378;
        dato = _inp(puerto);
        return (dato);
    }
}
```



```

    }
    if (puerto = 1586)
    {
        puerto = 0x632;
        dato = _inp(puerto);
        return (dato);
    }
};

```

Como podemos apreciar el código toma en cuenta el puerto que se este trabajando, es decir, de entre el LPT888 o bien si se tiene un LPT1586. Para comenzar, es necesario el uso de la librería conio.h la cuál contiene las instrucciones `_outp` e `_inp` necesarias para llevar a cabo el redireccionamiento si el puerto es el 888 entonces el bus de control será el 890 de otra forma si se trata del puerto 1586 entonces el bus de control será el 1588.

La función: `_outp` (mandar un dato hacia el puerto paralelo) trabaja con 2 parámetros: la dirección de memoria del bus del puerto a mandarle el dato, y el dato que se quiere mandar:

```
_outp(numero_puerto, dato_a_mandar)
```

La función `_inp` (leer un dato desde el puerto paralelo) trabaja únicamente con un solo parámetro: la dirección de memoria del bus del puerto para leer el dato.

```
_inp(numero_puerto)
```

En ambos casos tanto si se tiene un puerto LPT888 ó un LPT1586 tendremos que trabajar sobre el bus de control del puerto (890 y 1588 respectivamente) pues es precisamente ahí en donde se escribirá en el bit 5 el redireccionamiento del puerto, por lo que la parte más importante del código es la siguiente:

```
'aquí el valor para redireccionar puerto
'aquí el valor para redireccionar puerto
```

Aquí lo que estamos programando es lo siguiente, (considerando que se tratara de un LPT888):

Enviar un dato hacia el bus de control 890 (determinado por la variable: `puerto`) ese dato será lo que resulte de la función bitware inclusiva-OR: lo que lee la función `_inp` (el dato que hay en el bus de control del puerto) y (el valor) por lo tanto lo que se haría es:

```
_outp(valor,valor2|valor3);
```

ya que, siempre que prendemos la computadora el dato que podemos leer en el bus de control del puerto paralelo es 4 o sea:

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

00 000100 (los bit's 6 y 7 son reservados)

el resultado de la operación de bit dirigido Inclusivo-OR es:

```
  0 0 0 0 0 0
|
  0 0 0 0 0 0
-----
  0 0 0 0 0 0 * estos valores se encuentran protegidos
```

Si observamos el bit No.5, tenemos que ha cambiado ya a 1 lógico, por lo que ahora el bus de datos del puerto ahora puede recibir valores, pero observamos que se ve afectado el bit IRQ Enable el cuál ha pasado de 0 a 1 lógico por lo que necesitamos pasar nuevamente el bit IRQ Enable (bit 4) a su estado original (0 lógico), ya que si se deja así, entonces nuestro sistema tendrá conflictos en su funcionamiento de IRQ's. Para ello es la siguiente instrucción, pero ahora con una operación de bit dirigido AND:

```
_outp(890,inp(puerto)&valor4);

  0 0 0 0 0 0 0 0
&
  1 1 1 0 1 1 1 1
-----
  1 1 1 1 1 1 1 1 * estos valores se encuentran protegidos
```

Con esto observamos que el bit IRQ Enabled vuelve a su estado original (0 lógico) y el bit de redireccionamiento (bit 5) queda listo (1 lógico) para recibir valores mediante el bus de datos del puerto paralelo.

Existen otras funciones dentro de la librería de vínculos dinámicos creada, que se utilizarán en nuestro sistema de adquisición de datos: una es para mandar los correspondientes valores para seleccionar un canal del convertidor analógico a digital ADC0808 y la otra es para leer los valores entregados por parte de dicho convertidor:

```
int WINAPI MandarDato(int puerto, int dato)//función para mandar datos
{
    _outp(puerto,dato);

    return (dato);
};
int WINAPI LeerDato(int puerto)//función para leer datos
{
    int dato;
```

```

    if (puerto = 888)
    {
        puerto = 0x378;
        dato = _inp(puerto);
        return (dato);
    }
    if (puerto = 1586)
    {
        puerto = 0x632;
        dato = _inp(puerto);
        return (dato);
    }
};

```

Podemos observar que cuando se trata de leer datos desde el ADC0808 entonces el parámetro ha utilizar de la instrucción `_inp` será el que corresponde al bus de datos del puerto paralelo y no al bus de control del mismo.

El lenguaje Visual C++ también necesita otro archivo importante cuando se está trabajando con una librería de vínculos dinámicos, es el llamado archivo de definiciones (.DEF), el cuál es necesario para declarar las funciones ó subrutinas que queremos incluir en nuestra librería de vínculos dinámicos, el archivo de definiciones que corresponde a la librería de vínculos dinámicos es el siguiente:

Este código lo genera automáticamente Visual C++:

; dllRedirecc.def : Declares the module parameters for the DLL.

LIBRARY dllRedirecc.dll

DESCRIPTION 'dllRedirecc Windows Dynamic Link Library'

EXPORTS

; Explicit exports can go here

Aquí empieza la declaración de funciones:

LUISREDIRECC = _Redirecc@4 ; el alias sería: LUISREDIRECC

_Redirecc@4

LUISMANDARDATO = _MandarDato@8 ; el alias sería: LUISMANDARDATO

_MandarDato@8

LUISLEERDATO = _LeerDato@4 ; el alias sería: LUISLEERDATO

_LeerDato@4

Observamos que además de declarar la función. También es necesario declarar un número el cual representa el número total de bytes que se requieren en los parámetros de la función, y finalmente un "alias" el cuál es un sobrenombre que nosotros queramos ponerle a la función

para que la pueda identificar, en este caso, el lenguaje de programación Visual Basic, todo esto tendrá que ser declarado en ésta sección.

3.14 Proceso y creación del archivo a transferir por Internet.

Lo primero que se debió hacer fue planificar el documento que se deseaba transmitir, es decir, diseñar como queríamos que se encontraran ubicados cada uno de los elementos que se formarían parte de dicho documento y a partir de eso buscar las alternativas posibles para el envío del mismo tomando en cuenta varios aspectos muy importantes como son: los gráficos que tendrán que ser enviados, el tamaño de los mismos, el tiempo que se pudiera llevar en la traslación del documento, etc.

Para llevar a cabo la trasmisión del documento se planificó que debería ser en un tipo de archivo el cuál no tardase mucho tiempo en llegar a su destino, se llegó a la conclusión de de las páginas Web son los archivos por excelencia que se utilizan para el traslado de información además de que resultan ser muy llamativos por su aspecto gráfico y sobretodo porque con ellas podemos enviar la información que nosotros queramos, no solo en modo texto sino también con la posibilidad de incluir imágenes.

La metodología que se siguió fue la de crear la página Web con el programa Microsoft Front Page© versión 2002, el cual facilita la labor de crear páginas, ya que se puede trabajar en modo grafico y solo fue cuestión de arrastrar y soltar los elementos que queríamos que aparecieran en nuestra página. Teniendo la codificación de la página HTML se buscó la manera de trasladarla a Visual Basic con la consideración de poder manejar variables que se incorporaran en la página en modo real, ya que serían los datos más importantes que contendría dicha página, además de la utilización de instrucciones en Visual Basic para poder crear la pagina Web con su correspondiente extensión para posteriormente tener una vista preeliminar de cómo sería la página Web que se tendrá que mandar todo ello se programó utilizando Visual Basic (Anexo I, Pág.4)

El procedimiento fue el siguiente:

Teniendo una codificación como por ejemplo un encabezado en la parte central de una página, la codificación en HTML sería la siguiente:

```
<html>

<head>
<meta http-equiv="Content-Language" content="es-mx">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Pagina nueva 1</title>
</head>
```

```

<body>

<p align="center">UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO.</p>

</body>

</html>

```

el primer paso fue utilizar una instrucción en Visual Basic para poder crear un documento en formato texto (.TXT) la función OPEN la cuál necesita como parámetros básicamente: la ruta para escribir el archivo y el numero de archivo que se le asignará.

Open Nombre_Archivo For Output As #numero

Posteriormente empezamos a escribir lo que contiene la página HTML: utilizando la instrucción: PRINT de Visual Basic para escribir en el archivo de texto, y aquí nos encontramos con un detalle. La sintaxis de la función print es la siguiente:

Print #*númeroarchivo*, [*listasalida*]

Parte	Descripción
<i>númeroarchivo</i>	Requerido. Cualquier número de archivo válido.
<i>listasalida</i>	Opcional. Expresión o lista de expresiones para imprimir.

Tabla 3.3 Parámetros para la apertura de un archivo en Visual Basic 6.0

Utilizamos el número correspondiente de archivo, pero el detalle es que Visual Basic trabaja las cadenas de texto utilizando comillas, es decir, lo que queramos que aparezca en el archivo de texto lo tenemos que encerrar entre comillas, además de que tenemos que trabajar cada una de las líneas de la página HTML ya que HTML también utiliza muchas comillas cuando trabaja con sus comandos, aquí el problema es que Visual Basic necesita ser “informado” correctamente si se trata de comillas que corresponder a la programación propia de la función ó se trata de comillas existentes en el texto que queremos escribir al archivo .TXT, para solucionar este problema utilizamos las funciones:

Chr(34) la cuál nos es de gran utilidad para decirle a Visual Basic que se trata de comillas que queremos que aparezcan en el archivo .TXT, observamos que el carácter 34 corresponde al código ASCII asociado a las comillas, adicionalmente también le tenemos que indicar cuando queramos introducir una línea en blanco en el archivo .TXT:

Print #numero, 'imprime una línea en blanco

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

Así que utilizando estas convenciones nuestra codificación en Visual Basic utilizando la codificación HTML descrita para un encabezado en la parte centra de la página sería la siguiente (suponiendo que el archivo creado fuera el No.1):

```
Print #1, "<html> "
```

```
Print #1, "<head> "
```

```
Print #1, "<meta http-equiv="; Chr(34); "Content-Language"; Chr(34); " content=";
```

```
Chr(34); "es-mx"; Chr(34); ">"
```

```
Print #1, "<meta name="; Chr(34);"GENERATOR";Chr(34);"content Microsoft
```

```
FrontPage 5.0";Chr(34);">"
```

```
Print #1, " <meta name=" ;Chr(34) ; "ProgId";Crh(34); "content=";Chr(34);
```

```
"FrontPage Editor Document" ;Chr(34) ; ">"
```

```
Print#1, "<meta http-equiv=";Chr(34);"Content-Type"Ch(34);
```

```
"content="Chr(34);"text/html; charset=windows-1252";Chr(34); ">"
```

```
Print #1, "<title>Pagina nueva 1</title>"
```

```
Print #1, "</head>"
```

```
Print #1, "<body>"
```

```
Print #1, "<p align=";Chr(34);"center";Chr(34);">UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO.</p>"
```

```
Print #1, "</body>"
```

```
Print #1, "</html>"
```

Será también necesario el manejo de variables que se incorporen a el archivo de texto en tiempo real, es decir que en cuanto se tenga una adquisición de algún dato biomédico este valor pase a ser parte de los elementos que compondrán nuestra pagina HTML a enviar (además de otros parámetros como son: la hora y la fecha en la que se está enviando nuestro dictamen, y su gráfico correspondiente si es que se desea enviar en modo gráfico).

Para incorporar el valor biomédico correspondiente únicamente incorporamos el valor que se esta registrando en determinado control de Visual Basic (como por ejemplo en una etiqueta) al archivo de texto y los demás mensajes que queramos por ejemplo:

```
Print #1, "<p align="; Chr(34); "left"; Chr(34); "><font face="; Chr(34);  
"TEMPERATURA REGISTRADA: &nbsp;"; formularioX.etiquetaX.Caption, "
```


Con esta codificación en Visual Basic la línea de de texto que se creará en el archivo de texto será el siguiente:

TEMPERATURA REGISTRADA: *Valor Correspondiente*

Aquí el *valor correspondiente* será el que se este registrando en tiempo real por nuestro sistema de adquisición de datos. El valor se toma de la propiedad "caption" del control utilizado, en este caso una etiqueta.

Para incorporar la fecha y la hora en que se crea la página HTML nos valemos de las siguientes funciones Date() y Time():

Incorporamos la fecha y hora en una etiqueta y posteriormente las incluimos en el archivo de texto que se va a crear, utilizando las mismas convenciones que fueron descritas anteriormente:

```
Print #1, "<p align="; Chr(34); "left"; Chr(34); "><font face="; Chr(34); "Comic Sans MS"; Chr(34); " color="; Chr(34); "#0000FF"; Chr(34); ">; FECHA:",  
formularioX.etiqueta_fechaX.Caption, "</font></p>"  
Print #1, "<p align="; Chr(34); "center"; Chr(34); ">&nbsp;</p>"  
Print #1, "<p align="; Chr(34); "left"; Chr(34); "><font face="; Chr(34); "Comic Sans MS"; Chr(34); " color="; Chr(34); "#0000FF"; Chr(34); ">; HORA:",  
formularioX.etiqueta_relojX.Caption, "</font></p>"
```

Ahora bien, si el la página a enviar es elegida en modo gráfico, es decir, que incluya el gráfico correspondiente a la temperatura registrada entonces el procedimiento para incluir dicho gráfico será el siguiente:

Primero que nada necesitamos saber la ubicación del gráfico que queremos que se incluya, primeramente en el archivo de texto .TXT como codificación y posteriormente en la página HTML ya como gráfico, todo esto llevado a cabo en tiempo real (para mayores detalles ver el siguiente tema: "Creación del dictamen en modo gráfico").

Una vez que se conoce la ubicación del gráfico procedemos a codificarlo en Visual Basic, en el archivo .TXT, nuevamente utilizando las convenciones antes descritas tendremos:

```
Print #1, "&nbsp;<font face="; Chr(34); "Comic Sans MS"; Chr(34); " color="; Chr(34); "#0000FF"; Chr(34); "
```

Aquí vemos que la ruta es RutaX (cualquier ruta que elijamos) seguida del nombre del archivo (la gráfica.bmp) con algunos otros parámetros como son: el ancho del gráfico, el alto, color, etc.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

El archivo de texto queda exactamente como se encuentra desarrollado en la codificación HTML, ahora sólo basta cambiar la extensión de dicho archivo para que quede terminada nuestra página:

Para ello nos valemos de la función *Name* de Visual Basic la cual tiene la siguiente sintaxis:

Name nombre_antiguo As nombre_nuevo

En *nombre_antiguo* va el nombre que le asignamos al archivo de texto y en *nombre_nuevo* va el nombre y la extensión que queramos para nuestra página HTML, cabe mencionar que en el parámetro *nombre_antiguo* y *nombre_nuevo* debe recibir la ubicación exacta de en donde se encuentra el archivo y donde será ubicado el nuevo archivo, respectivamente.

La instrucción *Name* cambia el nombre de un archivo y lo mueve a un directorio o carpeta diferente, si es necesario. *Name* puede mover un archivo a través de unidades pero sólo puede cambiar el nombre a un directorio o carpeta existente cuando tanto *nombre_ruta_antigua* como el *nombre_ruta_nueva* están ubicados en la misma unidad.

Name no puede crear un nuevo archivo, directorio o carpeta.

Si se utiliza la instrucción *Name* con un archivo abierto se producirá un error. Tenemos que cerrar un archivo antes de cambiarle el nombre, de la siguiente manera:

Close #1 'cierra el archivo

Cuando se ejecuta la instrucción *Close*, la asociación de un archivo con su número de archivo termina.

En el momento en que se toman cada una de los datos en tiempo real por parte de la codificación e instrucciones dentro de la aplicación cliente.exe, el resultado del dictamen obtenido puede darse en dos formas: uno con sólo los datos escritos, es decir en modo texto excluyendo la gráfica de la temperatura registrada, y de la otra forma, la cuál, además de los datos biomédicos en modo texto se incluye la gráfica correspondiente, así, el resultado obtenido es el que se muestra en la figura 3.13:



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

TELEMEDICA: "ATENCIÓN MÉDICA A CADA SEGUNDO".

PACIENTE: JOSÉ LUIS JUÁREZ TREJO.

DIRECCIÓN: CALLE 19 #55 COL. OLIVAR DEL CONDE

TELÉFONO: 044-55-10-59-28-63

FECHA: 04/06/2004

HORA: 06:54:14 p.m.

TEMPERATURA REGISTRADA: 33 Grados Centígrados

GRÁFICA TEMPERATURA

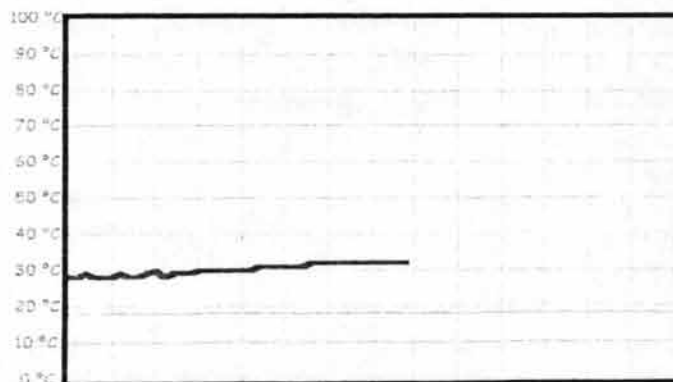


Figura 3.13 Grafica obtenida en el sistema de adquisición de datos.

3.15 Creación del dictamen en modo gráfico.

Para crear un archivo HTML con su gráfico correspondiente, necesitamos un control en Visual Basic que nos permita la obtención de un archivo gráfico con lo que se encuentre en ese momento en dicho control, para tal efecto nos valimos del control intrínseco de Visual Basic llamado: PictureBox, del cuál presentamos una breve descripción:

3.15.1 Control PictureBox.

El control PictureBox se utiliza para presentar gráficos, para actuar como contenedor de otros controles y para presentar el resultado de los métodos gráficos o texto con el método Print.

En este caso, el control PictureBox nos servirá para poder visualizar el dato biomédico obtenido dentro de una gráfica. Antes de poder trabajar con este control, es conveniente establecer el modo de trabajo en el cuál vamos a graficar el dato obtenido.

En Visual Basic existen varios modos de trabajo dentro de éste control, dependiendo de cuáles vayan a ser las necesidades de trabajo para representar la información. La función a utilizar es la llamada: ScaleMode, en la cuál debemos establecer el modo de trabajo a utilizar, dicha función se vera con mayor detalle en las líneas precedentes, así como los valores disponibles que se pueden establecer para obtener mejores resultados al trabajar con este potente control de Visual Basic.



Figura 3.14 Control PictureBox de Visual Basic.

El control PictureBox es similar al control Image en que los dos se pueden usar para presentar gráficos en una aplicación y los dos aceptan los mismos formatos gráficos. Sin embargo, el control PictureBox tiene funcionalidades que no tiene el control Image; por ejemplo, la posibilidad de actuar como contenedor de otros controles y de aceptar métodos gráficos.

El control PictureBox puede presentar archivos de imágenes en los formatos siguientes: mapa de bits, (que es el que utilizaremos en nuestro sistema) icono, metarchivo, metarchivo mejorado o archivos JPEG o GIF.

Una característica también muy importante que diferencia a estos dos tipos de controles es que el control Image tiene una propiedad conocida como Stretch, la cuál puede adaptar automáticamente el tamaño de la imagen que se quiera vincular a este control, por lo que este control se utiliza más a menudo como contenedor de imágenes. Con el control PictureBox podemos vincular ó contener, otros tipos de controles.

Formato de imagen	Descripción
Mapa de bits	<p>Un <i>mapa de bits</i> define una imagen como un patrón de puntos (píxeles). Un mapa de bits tiene la extensión de nombre de archivo .bmp o .dib. Los mapas de bits también se denominan gráficos de "tipo pintura".</p> <p>Puede usar mapas de bits de varios tonos de colores, incluidos los de 2, 4, 8, 16, 24 y 32 bits, pero un mapa de bits sólo se presenta correctamente si el dispositivo de presentación acepta los tonos de colores del mapa de bits. Por ejemplo, un mapa de bits de 8 bits por píxeles (256 colores) sólo presenta 16 colores cuando se muestra en un dispositivo de 4 bits por píxeles (16 colores).</p>
Icono	<p>Un <i>icono</i> es un tipo especial de mapa de bits. Los iconos tienen un tamaño máximo de 32 por 32 píxeles, pero en Microsoft Windows 95 los iconos también pueden tener un tamaño de 16 por 16 píxeles. Los iconos tienen la extensión de nombre de archivo .ico.</p>
Cursor	<p>Los cursores, como iconos, son básicamente mapas de bits. Los cursores, no obstante, también contienen una zona caliente, un píxel que hace un seguimiento de la ubicación del cursor por sus coordenadas x e y. Los cursores tienen la extensión de nombre de archivo .cur.</p>
Metafile	<p>Un metarchivo define una imagen como líneas y formas codificadas. Los metarchivos convencionales tienen la extensión de nombre de archivo .wmf. Los metarchivos mejorados tienen la extensión .emf. Sólo se pueden cargar los archivos compatibles con Microsoft Windows. Los metarchivos también se denominan gráficos de "tipo dibujo".</p>
JPEG	<p>PEG (Joint Photographic Experts Group) es un formato de mapa de bits comprimido que acepta colores de 8 y 24 bits. Es un formato de archivo popular en Internet.</p>
GIF	<p>GIF (Graphic Interchange Format) es un formato de mapa de bits comprimido desarrollado originalmente por CompuServe. Acepta hasta 256 colores y es un formato de archivo popular en Internet.</p>

Tabla 3.3 Formatos de imagen disponibles en PictureBox.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

El procedimiento a seguir para crear el entorno de trabajo en el PictureBox es el siguiente:

Establecer el modo de trabajo: *ScaleMode = EscalaX*

Sintaxis.

objeto.ScaleMode [= valor]

La sintaxis de la propiedad ScaleMode consta de las siguientes partes:

Parte	Descripción
<i>Objeto</i>	El objeto que se este utilizando en este caso el control PictureBox.
<i>Valor</i>	Un número entero que especifica la unidad de medida, como se describe en Valores.

Tabla 3.4 Sintaxis de la propiedad ScaleMode.

3.15.3 Valores.

Los valores admitidos para *valor* son:

Constante	Valor	Descripción
vbUser	0	Para indicar valores personalizados.
vbTwips	1	(Predeterminado) Twip (1440 twips por pulgada lógica; 567 twips por centímetro lógico).
vbPoints	2	Punto (72 puntos por pulgada lógica).
vbPixels	3	Pixel (la unidad mínima de la resolución del monitor o la impresora).
vbCharacters	4	Carácter (horizontal = 120 twips por unidad; vertical = 240 twips por unidad).
vbInches	5	Pulgada.
vbMillimeters	6	Milímetro.
vbCentimeters	7	Centímetro.
VbHimetric	8	HiMetric
VbContainerPosition	9	Unidades usadas por el contenedor del control para determinar la posición del control.
VbContainerSize	10	Unidades usadas por el contenedor del control para determinar el tamaño del control.

Tabla 3.5 Valores de trabajo permitidos para la propiedad ScaleMode.

En este caso, en un primer formulario de nuestro sistema de adquisición, en donde podemos observar un tipo de gráfico compacto de la temperatura registrada, la escala de trabajo esta estipulada en píxeles (valor = 3), y en la grafica en modo de visualización grande, la escala de trabajo esta establecida en milímetros (valor = 6).

Posteriormente con la ayuda del método Line y la propiedad DrawWidth establecemos el cuadriculado del control PictureBox. El método Line es utilizada para dibujar líneas rectas y necesita algunos de los parámetros siguientes:

Line (X1,Y1)-(X2,Y2)

(X1,Y1).- Indican las coordenadas del punto inicial.

(X2,Y2).- Indican las coordenadas del punto final.

Es importante hacer notar que si establecemos un control Picture box las coordenadas disponibles del área de trabajo quedarán establecidas con el ancho (ScaleWidth) y el alto (ScaleHeight) del control, por ejemplo: Si establecemos un Picture box de aproximadamente 30mm. x 30mm. (trabajando en esa escala) entonces los valores máximos de las coordenadas serán (30,30).

La cuestión más importante radica en el hecho de que cuando establecemos alguna coordenada dentro del control este tomará el primer punto (X1) de la coordenada empezando por la esquina superior izquierda recorriéndose hacia la derecha según el valor introducido, y en el segundo punto (Y1) empezando en la esquina superior izquierda y recorriéndose hacia abajo según el valor introducido.

Con esto nos podemos dar cuenta que difiere el modo de encontrar los puntos utilizando este lenguaje de programación que el que normalmente utilizamos en los planos cartesianos.

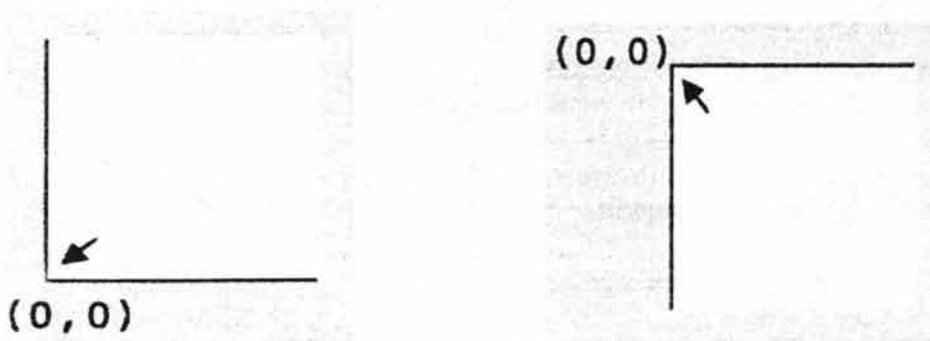


Figura3.14 El modo de localizar un punto difiere entre control PictureBox de Visual Basic y el sistema de coordenadas cartesiano normal.

La manera de obtener el dato biomédico es llevado a cabo mediante la librería de vínculos dinámicos redirecc.dll, el cuál es mostrado en tiempo real con una etiqueta de Visual Basic; para poder mostrar la graficación se lleva a cabo mediante la misma función descrita anteriormente (Line) y el siguiente procedimiento:

- Establecer un temporizador.

Un temporizador es un control intrínseco de Visual Basic, Los controles Timer responden al paso del tiempo. Son independientes del usuario y se pueden programar para que ejecuten acciones a intervalos periódicos de tiempo. No obstante, cabe hacer mención que se utilizaron muchos de estos controles para otros objetivos (visualizar la hora actual, realizar tareas de verificación de archivos, realizar algún proceso periódicamente, etc.)

En el caso del presente tema en particular, se empleó para realizar la graficación del dato biomédico obtenido, mediante un ciclo de obtención del valor que se tenía que presentar en pantalla. Después del siguiente punto se mostrará la forma en que trabaja dicho control.

- Inicializar los puntos de coordenadas que corresponden a (X1,Y1) en:

(0, valor_máximoX)

valor_máximoX.- Es el valor de la altura máxima del control Picture box.

Este punto es necesario para que la graficación comience desde la parte inferior izquierda del control.

El primer punto que corresponderá a la coordenada (X2,Y2) y que será la primera línea a trazar nos la dará, por un lado (X2), el valor que haya en el punto X1 más un valor extra que puede ser determinado con un control de Visual Basic que es una barra deslizante llamada: ScrollBar la cuál puede ser variada en tiempo real, con el objetivo de aparentar un mejor tiempo de respuesta por parte de la graficación que se está llevando a cabo en ese momento, y por otro lado (Y2), el valor máximo de altura del control Picture box menos el valor que esté registrando en ese momento la librería de vínculos dinámicos.

Con la ayuda del temporizador podemos realizar éste procedimiento una y otra vez, todo el tiempo que se desee, Lo importante es actualizar los valores de las coordenadas.

- Actualizar los valores de las coordenadas

Esto será muy importante establecer ya que ahora el punto inicial de la línea a graficar (X1,Y1), será el valor que haya quedado en el punto (X2,Y2) y el punto destino seguirá siendo el valor que se encuentre en el ScrollBar (X2),que podrá ser modificado en todo momento) y el valor máximo de altura del control Picture box menos el valor que esté registrando en ese momento la librería de vínculos dinámicos (Y2).

Cuando se llegue al final del Picture box, es decir, cuando ya no haya más espacio en el control para graficar, entonces se “limpiará” el control mediante la propiedad: Cls. Posteriormente, el valor que tendrá ahora la coordenada (X1,Y1) será:

En X1 será igual a 0 y en X2 tomará el último valor que tenía Y2 antes de que se “limpiara” el control. Nuevamente se llevara a cabo todo el proceso anteriormente descrito hasta que la graficación sea detenida por el usuario.

Para mayores referencias e información ver el anexo I Pág.11.

3.16 Visualización de documentos HTML con Internet Explorer

A medida que se conoció que existen objetos dentro de nuestra computadora de los cuales podemos utilizar sus métodos y propiedades pudimos añadir el objeto Internet Explorer al sistema de adquisición de datos hecho en Visual Basic mediante código, el objeto Internet Explorer, un componente programable que incluye propiedades, métodos y sucesos que están disponibles en cada computadora que tenga instalada una copia del programa Internet Explorer es el encargado de hacer que podamos ver nuestra página preeliminar a enviar, utilizando las propiedades, métodos y sucesos de dicho objeto.

Microsoft Internet Explorer es un navegador de propósito general que muestra documentos HTML ubicados en Internet o en el disco duro. Microsoft diseñó el programa Internet Explorer para que pudiéramos utilizarlo como una aplicación individual (poniéndolo en marcha desde el menú Inicio de Windows) o para que extrajéramos de él sus objetos componentes y los empleáramos en cualquier programa que desarrolláramos. Según esto, Internet Explorer expone sus funciones como una colección de propiedades, métodos y sucesos siguiendo la estructura de un modelo de objetos comprensible.

Para poder observar este modelo de objetos podemos utilizar el Examinador de objetos de Visual Basic e introduciendo sus funciones en los programas que desarrollemos.

El *objeto Internet Explorer* no es un control ActiveX. Por el contrario, se trata de una biblioteca de objetos que reside en todas las computadoras que tengan instalado el programa Internet Explorer (en otras palabras, Internet Explorer tiene que estar presente en el registro del sistema).

El primer paso para utilizar el objeto Internet Explorer es añadir una referencia a la biblioteca de objetos en la aplicación. Llevamos esto a cabo utilizando el mandato: Referencias del menú Proyecto de Visual Basic, tal y como se muestra en la figura 3.15:

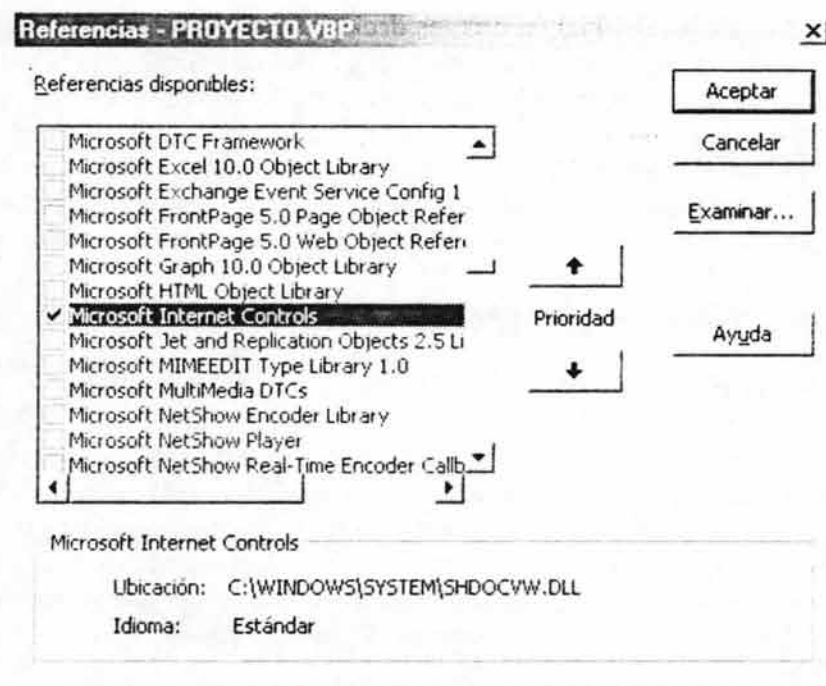


Figura 3.15 Referencias Microsoft Internet Controls en Visual Basic 6.0

Algunos de los métodos que utilizamos mayormente es el método Navigate, el cual es el encargado de abrir ya sea una dirección en Internet ó un documento HTML almacenado en algún lugar en nuestro disco duro.

Para mostrar algún documento HTML con el objeto Internet Explorer sólo se requirieron unas cuantas líneas de programación (Anexo 1, Pág.27):

Observamos que se deberá crear una variable objeto para representar la clase Internet Explorer de la biblioteca de objetos: shdocvw.dll. Si observamos un mensaje de error al cargar dicha librería y no se encuentra en nuestro sistema será conveniente actualizar la versión de el Internet Explorer y con ello actualizar también sus métodos y propiedades, pero sólo en casos muy raros se muestra este tipo de mensaje.

3.17 La aplicación CesarFTP.exe

Esta aplicación es capaz de configurar un servidor ftp anónimo ó protegido con contraseña, se hace uso de la misma para establecer los derechos de cada máquina cliente y determinar su login y contraseña respectiva que le será necesaria para establecer la conexión con el servidor. Los requerimientos necesarios para la utilización óptima de la aplicación son los siguientes: Para una máquina servidor: la dirección IP ó dominio de ella, establecer una carpeta de

descargas para cada una de las máquinas cliente así como su contraseña respectiva para controlar el acceso del exterior.

La aplicación CesarFTP.exe puede ser usada y distribuida gratuitamente, según los autores.

3.18 Envío del dictamen médico (Derechos, login, password, control de transferencia de archivos, propiedades, métodos, etc.)

Para iniciar el traslado de archivos desde la maquina cliente hasta el servidor configurado como Servidor FTP se debió establecer una serie de derechos y restricciones, además de seguir procedimiento a continuación descrito:

El primer paso es crear un acceso directo hacia la aplicación cesarFtp.exe con Visual Basic, ya que este programa será fundamental para establecer derechos, restricciones, carpetas de trabajo, password, login, etc. La función a utilizar es Shell de Visual Basic, de la siguiente manera:

Shell (Función)

Ejecuta un programa ejecutable y devuelve un tipo Variant (Double) que representa la identificación de la tarea del programa si se ha ejecutado con éxito, en caso contrario devuelve cero.

Por ejemplo, para abrir la aplicación CesarFTP.exe tenemos la siguiente línea:

```
Shell "d:\Archivos de programa\CesarFTP\CesarFTP.exe"
```

En la aplicación Servidor.exe, al momento de dar de alta a un nuevo paciente en la base de datos pacientes.mdb se tendrá que habilitar un botón en Visual Basic capaz de ejecutar la aplicación cesarFtp.exe, en donde, podremos establecer, entre otras cosas, lo siguiente:

- Establecer un nombre de trabajo.

Es decir, el nombre con el cuál se podrá entrar al servidor FTP (login)

- Establecer un grupo de trabajo.

Este puede ser el nombre de la asociación ó grupo al que pertenece cada uno de los usuarios dados de alta en el servidor.

- Establecer contraseña de usuario.

Un usuario podría acceder al servidor FTP de manera anónima, es decir, sin el uso de contraseña alguna, pero es altamente recomendable establecer ésta, para que de esta manera el

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

sistema de información del servidor FTP vea incrementada su seguridad y restrinja el acceso desde el exterior hacia éste. Además con ello tendremos absolutamente controlado que maquina cliente ha entrado a la máquina servidor y a que hora lo hizo.

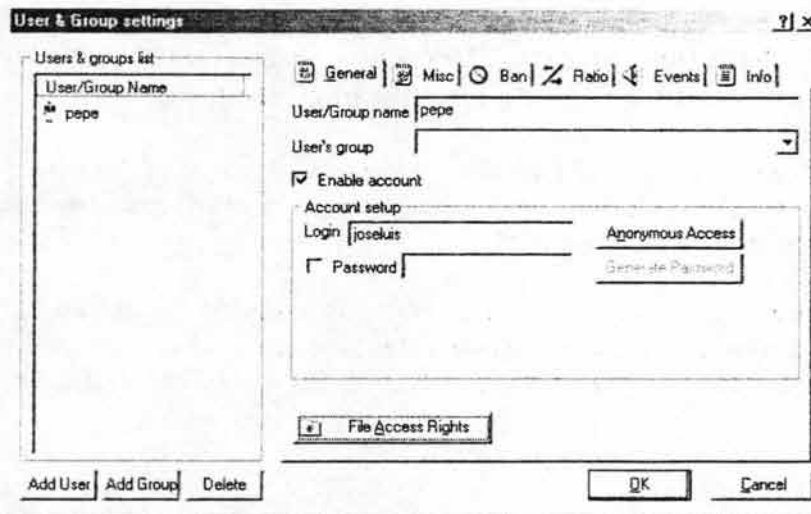


Figura 3.16 Establecer una contraseña para acceder al servidor agrega seguridad a ambos sistemas.

- Establecer el área de trabajo y tipo de información que podrá ser enviada y recibida.
- Establecer derechos de archivo tales como: lectura, escritura, borrar, agregar, listar, etc.

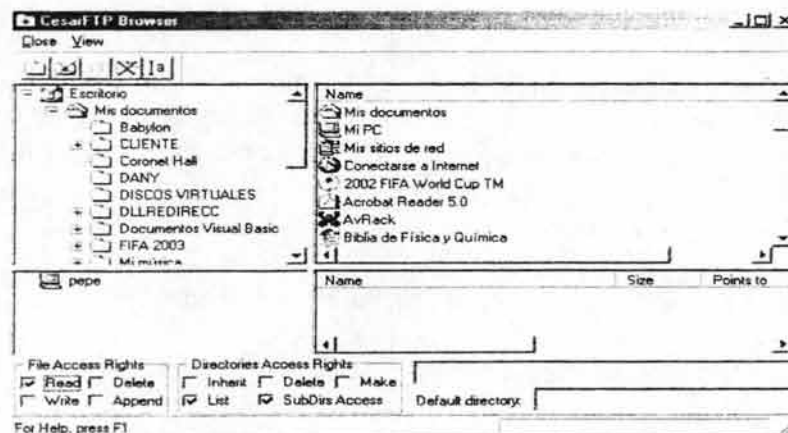


Figura 3.17 Privilegios y restricciones dentro del Servidor

3.19 Metodología de programación de la aplicación Servidor.exe.

El modelo de programación de la aplicación Servidor.exe es el siguiente:

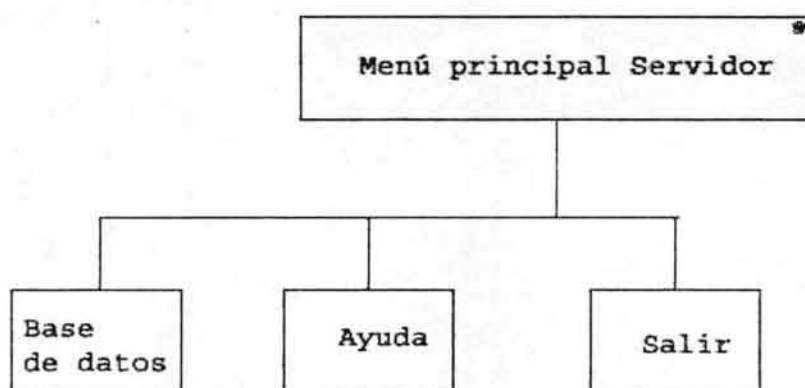


Figura 3.18 Modelo de programación de la aplicación Servidor.exe

El registro de datos estará protegido por contraseña:

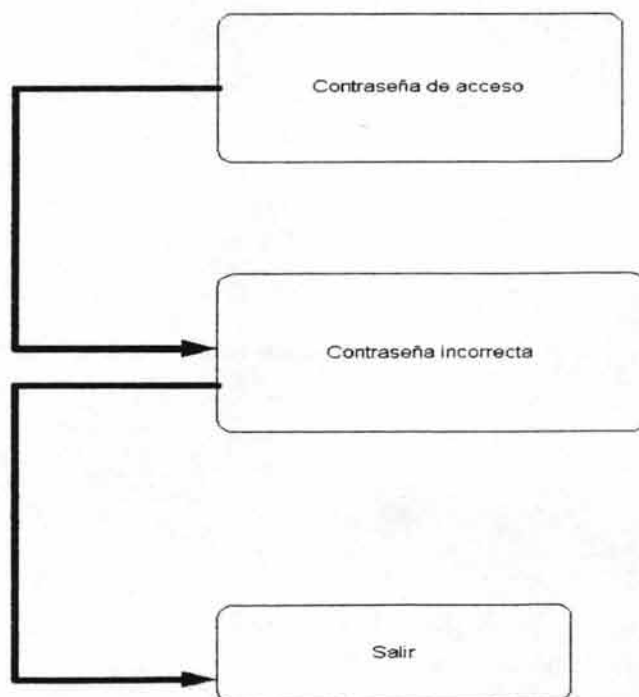


Figura 3.19 Inmediatamente se abandona la aplicación al detectar una contraseña incorrecta.

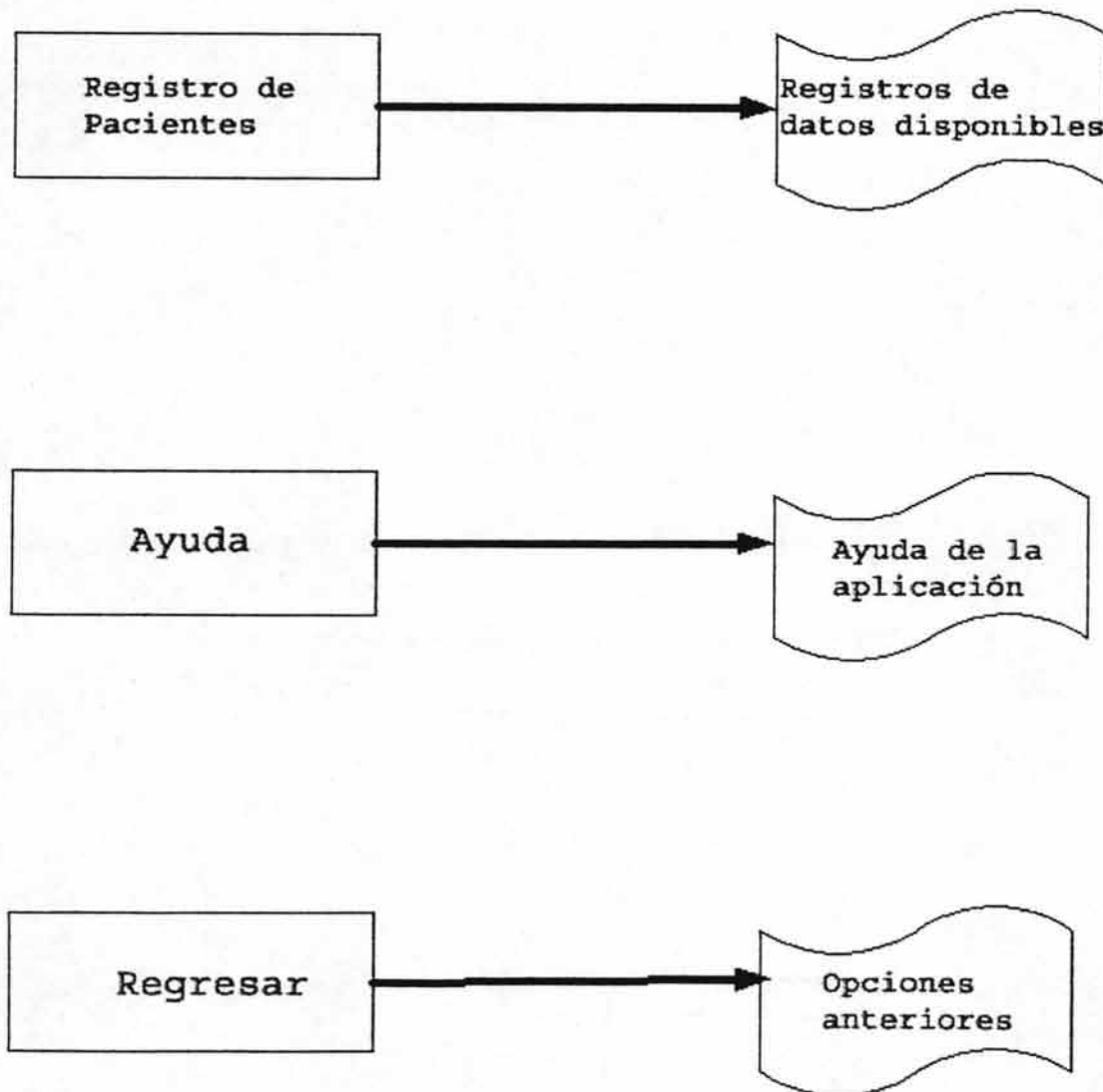


Figura 3.20 Opciones disponibles dentro de la aplicación Servidor.exe

Al ingresar a la opción del registro de pacientes, tenemos una serie de opciones a elegir, desde la configuración de un nuevo paciente, el establecimiento de su nuevo login y password de trabajo, carpeta de entrada de archivos al servidor FTP, establecer su identificación dentro de la base de datos, modificar parte ó toda la información de su perfil dentro del sistema, etcétera.

Algunas otras opciones de la aplicación Servidor.exe son:

Registro General de Pacientes:

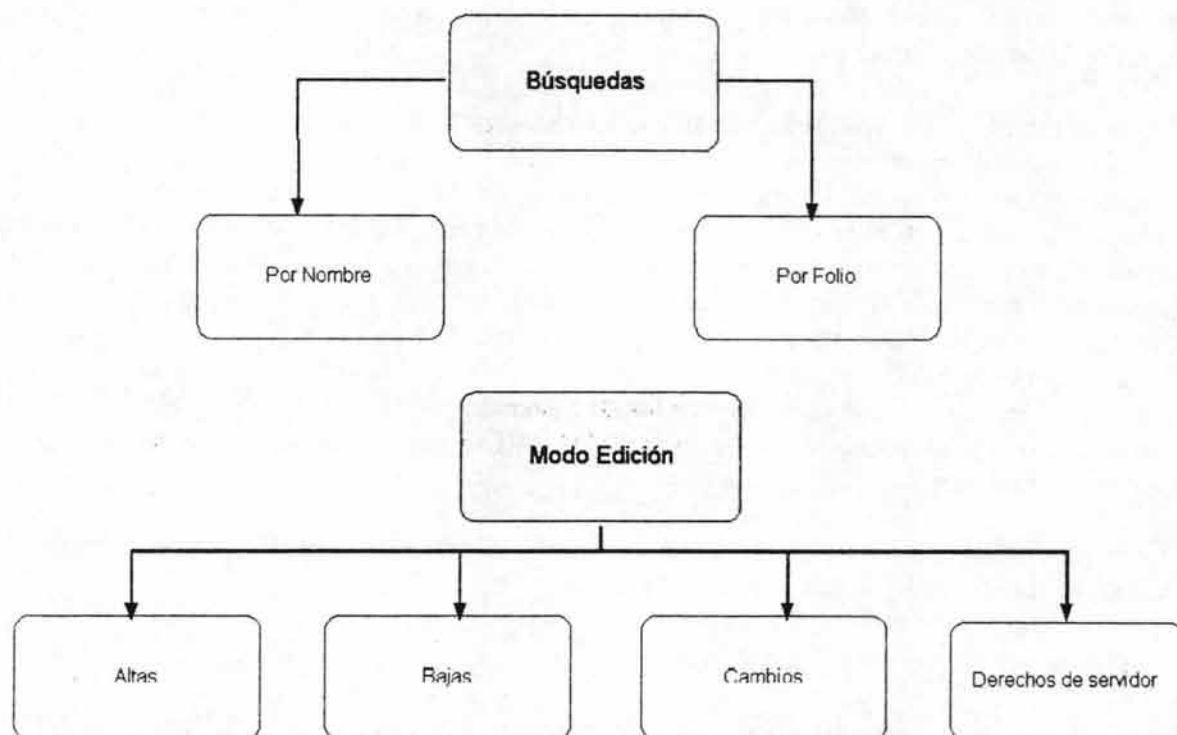


Figura 3.21 Diagramas de árbol de la opción búsqueda y Modo edición de la aplicación Servidor.exe

3.20 La base de datos termometro.mdb

Esta base de datos está vinculada a Visual Basic y consiste únicamente en una sola tabla llamada "pacientes" por lo que el diagrama entidad relación en este caso no existe ya que se ha utilizado solamente una tabla, el registro general de pacientes, por lo tanto, no hay relaciones establecidas, aunque no se descarta la oportunidad de establecer otras nuevas tablas dentro de la aplicación Servidor.exe, con el fin de ampliar la funcionalidad del sistema de adquisición de datos, para tal efecto, se deberán ir analizando éstas conforme a las necesidades que se vayan presentando a lo largo del desarrollo del proyecto de adquisición de datos, por lo pronto con la tabla establecida para almacenar los datos de los registros de pacientes es más que suficiente. Sólo hay que poner en claro que Visual Basic pone a nuestra disposición tres interfaces de acceso a datos: objetos de datos ActiveX (ADO), objetos de datos remotos (RDO) y objetos de acceso a datos (DAO). Una interfaz de acceso a datos es un modelo de objetos que presenta diversas formas de tener acceso a datos. Visual Basic permite controlar por programa la conexión, los generadores de instrucciones y los datos devueltos que se usarán en cualquier aplicación.

¿Por qué hay tres interfaces de acceso a datos en Visual Basic? La tecnología de acceso a datos evoluciona constantemente y cada una de estas tres interfaces corresponde a un estado de la evolución. La tecnología más reciente es ADO, con un modelo de objetos más sencillo (y aún más flexible) que RDO o DAO. Para este proyecto en particular se utilizó ADO como la interfaz de acceso a datos.

La interfaz ADO se ha diseñado como una interfaz de nivel de aplicación fácil de usar para el más nuevo y eficaz paradigma de acceso a datos de Microsoft, OLE DB. OLE DB proporciona un acceso de alto rendimiento a cualquier origen de datos, incluidos bases de datos relacionales y no relacionales (como lo es en este caso), correo electrónico y sistemas de archivos, texto y gráficos, objetos de negocios personalizados y mucho más. La implementación de ADO genera una cantidad mínima de transferencias a través de la red en escenarios clave de Internet y utiliza un número mínimo de capas entre el servidor y el origen de datos para proporcionar una interfaz compacta de alto rendimiento. Una metáfora habitual es hacer referencia a ADO como la interfaz de automatización de OLE. Además ADO utiliza convenciones y funciones similares a las de DAO y RDO, con una semántica simplificada que facilita su aprendizaje.

El procedimiento para llevar a cabo la vinculación de una base de datos de Microsoft Access hacia Microsoft Visual Basic es el siguiente:

Dentro del entorno de Microsoft Access:

- Crear la tabla en Microsoft Access que será vinculada a Microsoft Visual Basic.

Esta tabla contendrá tanto el número como el tipo de datos que podrán ser almacenados en los registros de ella. Utilizamos el tipo de datos Memo por si necesitamos almacenar más de 255 caracteres. Un campo Memo puede contener hasta 65.536 caracteres. Si deseamos almacenar texto con formato o documentos largos, es más conveniente crear un campo Objeto OLE en lugar de un campo Memo.

Los tipos de datos Texto y Memo almacenan sólo los caracteres introducidos en un campo: los caracteres de espacio correspondientes a las posiciones no utilizadas no se almacenan.

Se puede ordenar o agrupar en un campo Texto o Memo, pero Access sólo utiliza los 255 primeros caracteres cuando se ordena o agrupa en un campo Memo.

Utilizamos también un campo de tipo OLE en el cuál podremos almacenar un objeto de tipo imagen, es decir, la fotografía de identificación de determinada persona dada de alta en el sistema.

Objeto OLE.-Se utiliza para objetos OLE (como documentos de Microsoft Word, hojas de cálculo de Microsoft Excel, imágenes, sonidos u otros datos binarios) que se crearon en otros programas mediante el protocolo OLE (OLE: tecnología de integración de programas que se puede utilizar para compartir información entre programas. Todos los programas de Office

admiten OLE, de manera que puede compartir información a través de objetos vinculados e incrustados.).

Dentro del entorno Microsoft Visual Basic:

- Crear variables de tipo objeto para poder acceder a las funciones y procedimientos de la base de datos.

Una de las variables utilizará el objeto Recordset.

El objeto Recordset.

Utilizamos el objeto Recordset para manipular datos en una base de datos a nivel de registro. Cuando utilizamos objetos de acceso de datos, interactuamos con los datos prácticamente utilizando objetos Recordset. Todos los objetos Recordset se construyen utilizando registros (filas) y campos (columnas).

Con otra variable estableceremos la conexión entre la base de datos en Microsoft Access y Microsoft Visual Basic, mediante la propiedad: Connection, de la siguiente forma:

```
Set Nombre_Variable = New ADODB.Connection
```

Utilizando esta misma variable establecemos el nombre del proveedor de una base de datos, como en este caso estamos vinculando una base de datos hecha en Microsoft Access, entonces el nombre del proveedor de la base de datos es:

```
Nombre_Variable.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & sBase  
rst.Open "SELECT * FROM Pacientes", Nombre_Variable, adOpenDynamic,  
adLockOptimistic
```

Observamos que se utiliza la palabra: FROM esta indica el origen de datos mediante el cuál se realizará el vínculo, es decir, desde la tabla: "Pacientes" se leerán los campos de la base de datos hacia Microsoft Visual Basic.

- Establecer los controles en Microsoft Visual Basic que albergaran la información ó campos vinculados de la base de datos en Microsoft Access.

En este caso, fueron utilizados seis TextBox, un ComboBox y un control PictureBox para la foto de identificación, de los cuáles se establecieron algunas propiedades con efecto de llevar a cabo la conexión de la base de datos y además el mecanismo para llevar a cabo la carga de identificación mediante la utilización de un Modulo dentro de Visual Basic, algunas propiedades establecidas para estos controles fueron:

3.20.1 DataSource.

Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.

Sintaxis.

objeto.DataSource [=datasource]

3.20.3 DataField.

Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos, es decir los campos del registro

Sintaxis.

objeto.DataField [= cadena]

Para llevar a cabo el alta de un objeto OLE (imagen) en la base de datos utilizamos la propiedad DataFormat del control Picture box para indicarle a visual Basic que se tratará de un objeto de tipo imagen, de la siguiente forma:

PictureBox.DataFormat.Type = fmtPicture ' ó 2, la cuál corresponde a una imagen.

Para las operaciones básicas de navegación entre registros tales como: primero, siguiente, anterior y ultimo, no se utilizó el control que ofrece Microsoft Visual Basic ADODC por la sencilla razón de que no acepta alguna propiedad de imagen con fines de hacer al control más atractivo en su aspecto, por lo que se prefirió hacer uso de botones ordinarios y establecer cada una de las acciones descritas anteriormente mediante las propiedades:

MoveFirst, MovePrevious, MoveNext y MoveLast todas ellas disponibles en el objeto Recordset.

Como decíamos anteriormente, para cargar la imagen de identificación de la base de datos de un registro en específico, ó bien, para dar de alta una nueva imagen de identificación se hizo uso de una subrutina dentro de un módulo en Visual Basic llamado: DBbinary.bas, con la cuál se establece en tiempo real la imagen a mostrar dentro de la base de datos.

Es importante mencionar que como se utilizó un PictureBox, este control no tiene la propiedad llamada: Stretch, la cuál si se encuentra habilitada permite modificar automáticamente el tamaño de la imagen por lo que será necesario establecer un estándar de tamaño para las fotos de identificación. Para mayores detalles de codificación referirse al anexo 2 página 23.

3.21 Manejo de archivos y carpetas utilizando el objeto FileSystemObject de Microsoft Visual Basic 6.0

Para poder utilizar el objeto FileSystemObject es necesario establecer una variable de tipo objeto dentro de Microsoft Visual Basic, con el fin de trabajar con archivos y carpetas y poder realizar importantes funciones como: movimiento de archivos, copiar, crear, eliminar, etc., una manera típica de declaración de variable de tipo FileSystemObject es la siguiente:

```
Set Nombre_Variable = CreateObject("Scripting.FileSystemObject")
```

A partir de este momento, con la declaración anterior podemos hacer uso de las propiedades y métodos relacionados con este objeto, las cuáles resultan de gran trascendencia e importancia en una aplicación en donde se ven involucrados el manejo de archivos.

Algunos métodos y propiedades que se utilizaron son las siguientes:

3.21.1 CopyFile.

Copia uno o más archivos desde una ubicación a otra.

Sintaxis:

```
objeto.CopyFile origen, destino [, sobrescribir]
```

Con el parámetro: sobrescribir establecemos un valor tipo Boolean (booleano) que indica si los archivos existentes se van a sobrescribir. Si es True, los archivos se sobrescriben; si es False, no se sobrescriben.

3.21.2 CopyFolder.

Copia recursivamente una carpeta desde una ubicación a otra.

Sintaxis.

```
objeto.CopyFolder origen, destino[, sobrescribir]
```

3.21.3 CreateFolder.

Crea una carpeta.

Sintaxis.

```
objeto.CreateFolder(nombreDeLaCarpeta)
```

3.21.4 DeleteFile.

Elimina un archivo especificado.

Sintaxis.

objeto.DeleteFile especArchivo[, forzar]

3.21.5 DeleteFolder.

Elimina una carpeta especificada y su contenido.

Sintaxis.

objeto.DeleteFolder especCarpeta[, forzar]

3.21.6 FileExists.

Devuelve True si existe un archivo especificado y False si no existe.

Sintaxis.

objeto.FileExists(especArchivo)

3.21.7 FolderExists.

Devuelve True si existe una carpeta especificada y False si no existe.

Sintaxis.

objeto.FolderExists(especCarpeta)

3.21.8 MoveFile.

Mueve uno o más archivos desde una ubicación a otra.

Sintaxis.

objeto.MoveFile origen, destino

3.21.9 MoveFolder.

Mueve una o más carpetas desde una ubicación a otra.

Sintaxis.

objeto.MoveFolder origen, destino

El objeto FileSystemObject es el objeto principal de Visual Basic, repleto de métodos, que nos permite crear, eliminar y obtener información y generalmente controlar unidades, carpetas y archivos. Muchos de los métodos asociados con este objeto duplican los de los otros objetos.

3.22 Creación del Archivo de Ayuda: el programa Salomón.exe

Este programa permite crear archivos de ayuda que pueden ser vinculados a Visual Basic, su entorno de trabajo es muy accesible y fácil de utilizar, funciona conjuntamente con el programa de Microsoft: Help Workshop y sólo será necesario establecer la ruta del programa, para que de esta manera se enlace el archivo de ayuda con el compilador de Microsoft Visual Basic 6.0 y así obtener el archivo.

Una vez obtenido el archivo de ayuda se podrá vincular a las aplicaciones hechas en Visual Basic mediante el uso de sus propiedades de archivos de ayuda. Para mayor información referirse al capítulo 2 tema 2.16: "Archivos de Ayuda en Visual Basic"

CONCLUSIONES.

Inicialmente el puerto paralelo se utilizó para la comunicación con impresoras. Actualmente se utiliza también para manejar otros periféricos como CD ROM, cintas de copia de respaldo, discos duros, tarjetas de red, protectores de copia, scanners, etc.

Tipos de puerto paralelo:

En la actualidad se conoce cuatro tipos de puerto paralelo:

- Puerto paralelo estándar (Standart Parallel Port SPP)
- Puerto Paralelo PS/2 (bidireccional)
- Enhanced Parallel Port (EPP)
- Extended Capability Port (ECP)

Con la presente transformación y desarrollo de nuevas funciones del Puerto paralelo es posible hoy en día el manejo de este, a fin de recibir datos mediante su bus de datos del mismo, y que antes sólo se utilizaba para el envío de ellos desde el interior de una computadora.

El Protocolo de Transferencia de Ficheros (*File Transfer Protocol* o, para abreviar, FTP) es uno de los pilares de Internet junto con el acceso a páginas Web (*http*). Durante mucho tiempo ha sido el servicio *TCP/IP* de Internet más utilizado, debido a que cualquier distribución Linux (así como en otros Sistemas Operativos) incluye el software necesario, muy fácil de usar y configurar, y de reducido tamaño.

Es un protocolo de la familia *TCP/IP* usado para transferir ficheros entre máquinas que ejecutan *TCP/IP* (aunque hay programas estilo *FTP* para otros protocolos). Este protocolo permite transferir ficheros y manipular directorios en máquinas remotas. No está diseñado para permitir el acceso a otra máquina con el fin de ejecutar programas (para eso tenemos telnet), pero resulta la mejor utilidad para manipulación de ficheros.

Esto quiere decir que una máquina con un servidor de ftp dispone de una estructura de directorios con ficheros y que además es capaz de permitir a cientos o miles de usuarios la recogida de ficheros o envío de nuevos ficheros al servidor.

Por lo que podemos concluir que con el presente trabajo, efectivamente se logró establecer un mecanismo capaz de llevar a cabo el redireccionamiento del puerto paralelo, adicionalmente a ello, establecer un software de manufactura propia para prevenir casos en que se requieran ciertas modificaciones ó adicciones que posteriormente pudieran surgir con el uso del sistema, estas se pudieran ver solventadas mediante el uso del potente lenguaje de programación Visual Basic, también algo sumamente trascendental fue el hecho de vincular el lenguaje de programación Visual C++ con todo y las potentes y útiles funciones que se pueden hacer uso dentro de él, para agregar funcionabilidad y combinar el uso de herramientas entre

uno y otro lenguaje para el desarrollo del sistema, mediante la creación de librerías de vínculos dinámicos (.DLL).

Finalmente también se logró el traslado del archivo obtenido mediante el protocolo FTP haciendo uso de las propiedades y eventos que se ven involucrados en el uso del mismo, y adicionalmente a esto, también fue necesario la creación de una base de datos que nos permitiera llevar un control detallado de los registros que se iban mandando y recibiendo de la máquina cliente hacia la máquina servidor. La base de datos igualmente es posible modificarla en un futuro para que, ésta aumente en funcionalidad.

Consideramos que la mayor aportación de este trabajo de tesis, radica en el hecho de que es un sistema realmente útil, óptimo, confiable, que no requiere de mucha infraestructura y es de bajo costo, que puede ser utilizado por comunidades aisladas, por ejemplo, en un dispensario en el cual se cuente con una computadora y una línea telefónica con Internet, para enviar los signos vitales de un paciente, a hospitales y así obtener un diagnóstico de especialistas, más rápido, completo y sin tener que desplazarse a grandes distancias.

GLOSARIO.

ADC.- (Analogic to Digital Converter) Convertidor Analógico a Digital. Este dispositivo es el encargado de llevar a cabo las conversiones analógicas a digitales dentro del sistema de adquisición de datos.

ADO.- Microsoft® ActiveX® Data Objects permite a las aplicaciones cliente tener acceso y manipular datos de un servidor de bases de datos a través de cualquier proveedor de bases de datos.

BASIC.- (Beginners All Purpose Symbolic Instruction Code) Uno de los primeros lenguajes de programación cuyo objetivo fue el de programar utilizando el más básico y accesible código de programación.

B.D..- Base de Datos. Conjunto de información agrupada en campos y registros.

BIOS.- (Basic Input/Output System) Es en donde se cargan las opciones mínimas necesarias para el correcto funcionamiento del Hardware y arranque del Software así como de otros periféricos, es en donde se lleva el direccionamiento y mapeo de los diferentes dispositivos que intervienen en el sistema.

BUS DE CONTROL (CONTROL-BUS).- Es la parte del puerto paralelo en donde se pueden realizar determinadas tareas tales como el redireccionamiento del puerto entre otras. El bus de control esta conformado por los pines llamados: STROBE, AUTOFEED, INIT, SELECT IN.

BUS DE DATOS (DATA-BUS).- Es la parte del puerto paralelo en donde se reciben/envían los datos que se utilizan.

BUS DE ESTADO (STATUS-BUS).- Es la parte del puerto paralelo en donde se verifican algunos estados actuales de algún dispositivo que se este manipulando externamente (originalmente la impresora).

CIRCUITO INTEGRADO 555.- Se puede encontrar en el mercado con muy diversos nombres: NE555, μ A555, LM555, SN72555, XR-555, CA555, HA1755, NC1455, TA7555P, ECG955, etc. Dependiendo del fabricante. Algunas de las aplicaciones más comunes de él son:

- Controlador de sistemas secuenciales.
- Generador de tiempos de retraso.
- Divisor de frecuencias.
- Modulador.
- Repetidor.
- Pulsador
- Oscilador.

CLIENTE.- Se refiere a la máquina en donde se lleva a cabo la adquisición de los datos y que tiene la posibilidad de enviarlos a través de Internet mediante algún protocolo, en este caso el protocolo FTP.

CONTROL MICROSOFT INTERNET TRANSFER.- El control Internet Transfer implementa dos protocolos de Internet de gran difusión: Protocolo de transferencia de hipertexto (HTTP) y Protocolo de transferencia de archivos (FTP). Con el control Internet Transfer nos podemos conectar a cualquier sitio que utilice alguno de estos protocolos y obtener ó enviar archivos mediante el uso de métodos incluidos en dicho control.

DAC.- (Digital to Analogic Converter) Convertidor Digital a Analógico. Es el encargado de convertir señales digitales a analógicas esta implementado dentro del ADC0808 para llevar a cabo su correcto funcionamiento.

DAO.- (Data Access Objects) Los objetos de acceso a datos nos permiten habilitar para utilizar un lenguaje de programación con el fin de tener acceso y manipular datos en bases de datos locales o remotas y administrar bases de datos, los objetos y la estructura.

DEF.- Usualmente, es un archivo de texto que contiene declaraciones para definir un archivo .EXE ó una .DLL

DLL.- (Data Link Library) Las Librerías de Vínculos Dinámicos son archivos ejecutables independientes que contienen funciones y recursos que pueden ser llamados por los programas y por otras DLL para realizar ciertos trabajos en determinado tiempo.

ESTANDAR IEEE 1284.- Básicamente, define 5 modos de transferencia de datos:

- Nibble
- Modo Byte
- Bidireccional
- EPP
- ECP

EXE.- Archivo ejecutable.

FLIP-FLOP.- Es una celda binaria capaz de almacenar un bit indefinidamente hasta que se cambie por otra señal de entrada, los tipos más comunes de flip-flops son los siguientes: SR, JK y tipo D.

FRECUENCIA.- Es el número de vibraciones, ondas o ciclos por segundo de cualquier fenómeno periódico.

FTP.- (File Transfer Protocol) Protocolo de Transferencia de Archivos.

GUI.- (Graphic User Interface) La Interfaz Gráfica de Usuario, es cualquier entorno de trabajo ofrecido al usuario.

HARDWARE.- Conjunto de elementos materiales físicos de una computadora.

INTERNET.- Es la red de redes para llevar a cabo operaciones tales como: búsquedas de información, traslado/recepción de archivos, actividades didácticas, recreativas, etc.

LPT1.- Es la etiqueta que designa el BIOS para el primer puerto paralelo encontrado cuando se lleva a cabo el establecimiento ó direccionamiento de los diversos elementos de hardware del sistema.

LPT2.- Es la etiqueta que designa el BIOS para el segundo puerto paralelo encontrado cuando se lleva a cabo el establecimiento ó direccionamiento de los diversos elementos de hardware del sistema.

LP3.- Es la etiqueta que designa el BIOS para el tercer puerto paralelo encontrado cuando se lleva a cabo el establecimiento ó direccionamiento de los diversos elementos de hardware del sistema.

LSB.- (Low Significance Bit) Es el bit menos significativo, ó de menor peso dentro de un sistema digital.

MOD0 BIDIRECCIONAL.- En este modo de trabajo se pueden tanto recibir como enviar datos dentro de un sistema.

MSB.- (Mayor Significance Bit) Es el bit más significativo, ó de mayor peso dentro de un sistema digital.

MULTIPLEXOR.- Es un circuito lógico que tiene varias entradas de datos y que permite que sólo una de ellas pase en un tiempo a la salida.

NIBBLE.- Medio byte, o sea 4 bits.

OBJETO FSO.- (FILE SYSTEM OBJECT) Con el modelo de objetos FSO podemos trabajar con unidades y carpetas mediante programación tal como lo hacemos con el Explorador de Windows de forma interactiva. Entre otras cosas se puede copiar y mover carpetas, obtener información acerca de las unidades y carpetas, etc.

OBJETO INTERNET TRANSFER.- Es un control incluido en la Versión Visual Basic 6.0 Profesional capaz de manejar todos los detalles relacionados con la transferencia de archivos en una amplia variedad de contextos.

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

OBJETO RECORDSET.- Un objeto Recordset representa todo el conjunto de registros de una tabla o del resultado de un comando ejecutado. En cualquier momento, el objeto Recordset sólo hace referencia a un único registro dentro del conjunto, llamado registro actual, en esencia, es el medio principal para examinar y modificar registros de una base de datos.

OPERACIONES BITWARE.- Realizan operaciones de conversión aritméticas más usuales como son: AND, OR EXCLUSIVA y OR INCLUSIVA.

PIN.- Es la patilla de un Circuito Integrado.

PUERTO RS232.- Puerto Serie.

REDIRECCIONAMIENTO.- Es la acción de darle un nuevo sentido de trabajo (transmisión de datos) a un sistema de adquisición.

RDO.- (Remote Data Object) Objeto de Datos Remotos. Es una forma antigua de acceder a los registros de una base de datos. Para proporcionar compatibilidad con versiones anteriores, Visual Basic sigue admitiendo DAO y RDO en los posibles proyectos a crear.

SAR.- (Successive Approaches Register) El Registro de Aproximaciones Sucesivas es una parte funcional de las conversiones analógicas a digitales que se llevan a cabo dentro del ADC0808.

SERVIDOR.- Es la máquina encargada de monitorear las entradas y salidas de archivos que se envían desde una máquina cliente teniendo ésta, además, la capacidad de guardar los archivos recibidos dentro de una base de datos.

SOFTWARE.- Es el conjunto de programas de computadora y técnicas de informática

TRANSMISION SINCRONA.- Esto significa que la operación de transferencia se produce antes de que se ejecute cualquier otro procedimiento. Por tanto, la transferencia de datos debe haber terminado antes de que pueda ejecutarse otro código dentro de la aplicación.

TRANSMISION ASINCRONA.- Esto significa que, cuando se transfieren los datos, la operación de transferencia se produce independientemente de otros procedimientos. Por lo tanto, después de invocar dicha transferencia es posible ejecutar otro código dentro de la aplicación mientras se reciben ó envían datos en un segundo plano.

BIBLIOGRAFÍA.

Páginas electrónicas:

- 1) <http://ar.geocities.com/valdezda/informes/IEEE1284.htm>
- 2) <http://www.elrinconcito.com/DiccAmpliado/LPT.htm>
- 3) http://www.ii.uam.es/~gdrivera/varios/notas_lpt.htm
- 4) <http://www.pchardware.org/puertos/modoecp.php>
- 5) <http://www.redcientifica.com/doc/doc199903310006.html>
- 6) <http://www.glosarium.com/term/239,14,xhtml>
- 7) <http://www.eupmt.es/cra/inform/info1.html>
- 8) <http://www.monografias.com/trabajos/paralelos/paralelos.shtml>
- 9) <http://members.tripod.com/~MoisesRBB/adc.html>
- 10) <http://electronred.iespana.es/electronred/555.htm>
- 11) http://electro.webcindario.com/informacion/timer_555.htm
- 12) http://www.svetlian.com/dll/articulos_descripcion_dll.htm
- 13) <http://www.elguruprogramador.com.ar/zonas/ver.asp?cod=10>
- 14) http://alonso_m.tripod.com/visualc/creardll.htm
- 15) <http://www.modelo.edu.mx/univ/virtech/circuito/dllvb.htm>
- 16) <http://www.modelo.edu.mx/univ/virtech/prograc/win04.htm>
- 17) <http://www.mundopc.net/cursos/ftp/1.php>
- 18) <http://www.tel.uva.es/red/servidorftp.html>
- 19) <http://members.tripod.com/vteforte/ftp.htm>
- 20) <http://www.vc.ehu.es/wuagacaj/manual/ftp/ftp.html#PUNTO 1>
- 21) <http://burete.forodigital.es/angel/programacion/Redes/Protocolos/Ftp.html>
- 22) <http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/ftp.html>
- 23) <http://www.schematica.com>
- 24) <http://www.solont.com/z-net/ftp/ftp.htm>
- 25) <http://www.lacompu.com/soporte/linux/ftp/ftp.php3?template=imprimir>
- 26) <http://www.ignside.net/man/ftp/index.php>
- 27) Página de: Angel Abellán Torregrosa, 2000. Versión 210802000
- 28) CesarFTP 0.99g ACLogic, Alexandre Cesari

Libros de texto:

- 1) Bernard Zoller
Circuitos electrónicos con el PC
Editorial Marcombo.
- 2) Microsoft Corp. 1995-2000.
Colección de Ayuda de MSDN Library Studio 6.0
- 3) Michael Halvorson
Aprenda Visual Basic Ya 6.0
Editorial Mc Graw Hill
- 4) Malvino, Paul
Principios de Electrónica.
Editorial: Mc Graw-Hill
- 5) Serie Mundo Electrónico.
Interconexión de periféricos y microcomputadores.
Editorial: Marcombo.
- 6) Hostetter, G.H.
Sistemas de control.
Editorial: Mc Graw-Hill
- 7) Benjamín C. Kuo.
Sistemas Automáticos de Control.
Editorial C.E.C.S.A.
- 8) Boylestad
Electrónica teoría y circuitos.
Editorial: Prentice may.
- 9) Abraham Marcus.
Electrónica para técnicos.
Editorial Diana.
- 10) Henry Jacobowitz.
Electrónica simplificada.
Editorial Com.Gral.Ediciones.
- 11) Gilberto Melgarejo.
Electrónica básica por objetivos.
Editorial Herrero.
- 12) Manuales y Hojas de datos varios Nacional Semiconductors.
- 13) L. A. Geddes
Principles of Applied Biomedical Instrumentation 3ª.Edition

Anexo 1

ANEXO 1. CODIFICACION DE LA APLICACIÓN CLIENTE.EXE

frmAcerca (Código)

Option Explicit

Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hdc As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long

Private Declare Function GetTickCount Lib "kernel32" () As Long

Const DT_BOTTOM As Long = &H8

Const DT_CALCRECT As Long = &H400

Const DT_CENTER As Long = &H1

Const DT_EXPANDTABS As Long = &H40

Const DT_EXTERNALLEADING As Long = &H200

Const DT_LEFT As Long = &H0

Const DT_NOCLIP As Long = &H100

Const DT_NOPREFIX As Long = &H800

Const DT_RIGHT As Long = &H2

Const DT_SINGLELINE As Long = &H20

Const DT_TABSTOP As Long = &H80

Const DT_TOP As Long = &H0

Const DT_VCENTER As Long = &H4

Const DT_WORDBREAK As Long = &H10

Private Icono As Integer

Private Type RECT

Left As Long

Top As Long

Right As Long

Bottom As Long

End Type

'El texto actual a correr. Tambien se puede leer desde un archivo de texto

```
Const ScrollText As String = "Tema de Tesis" & vbCrLf & _
    vbCrLf & _
    "Asesor:" & vbCrLf & _
    "Ing. Ocampo Alvarez Arturo" & vbCrLf & _
    vbCrLf & "Programación:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Temas de referencia y Bibliografía:" & vbCrLf & _
    "Ing. Ocampo Alvarez Arturo" & vbCrLf & _
    vbCrLf & "Base de Datos:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Reportes tablas de Access:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Configuración Servidor FTP para Windows:" & vbCrLf & _
    "Ing. Ocampo Alvarez Arturo" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Diseño de página Web:" & vbCrLf & _
```

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

```
"José Luis Juárez Trejo" & vbCrLf & _  
vbCrLf & "Tema y Circutería propuesta por:" & vbCrLf & _  
"Ing. Ocampo Alvarez Arturo" & vbCrLf & _  
"Correo: luisca_74@hotmail.com" & vbCrLf & _  
"    aoa@servidor.unam.mx" & vbCrLf & _  
"Todos los Derechos Reservados" & vbCrLf & _  
"México MMIII" & vbCrLf & _  
vbCrLf & "Programa Exclusivamente" & vbCrLf & _  
"para uso Académico..." & vbCrLf
```

```
Dim EndingFlag As Boolean
```

```
Private Sub Form_Activate()
```

```
RunMain
```

```
End Sub
```

```
Private Sub Form_DblClick()
```

```
EndingFlag = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Icono = 1
```

```
picScroll.Font = "Arial" 'tipo de fuente
```

```
picScroll.ForeColor = &HFF00& ' color de fuente
```

```
picScroll.FontSize = 16.5 'tamaño de fuente
```

```
End Sub
```

```
Private Sub RunMain()
```

```
Dim LastFrameTime As Long
```

```
Const IntervalTime As Long = 40
```

```
Dim rt As Long
```

```
Dim DrawingRect As RECT
```

```
Dim UpperX As Long, UpperY As Long 'Punto izquierdo superior del PICSCROLL
```

```
Dim RectHeight As Long
```

```
'Muestra la forma
```

```
frmAcerca.Refresh
```

```
'Obtiene el tamaño del PICSCROLL y lo reemplaza por la constante DT_CALRECT
```

```
rt = DrawText(picScroll.hdc, ScrollText, -1, DrawingRect, DT_CALCRECT)
```

```
If rt = 0 Then 'Si marca error
```

```
    MsgBox "Error en el desplazamiento del texto", vbCritical
```

```
    EndingFlag = True
```

```
Else
```

```
    DrawingRect.Top = picScroll.ScaleHeight
```

```
    DrawingRect.Left = 0
```

```
    DrawingRect.Right = picScroll.ScaleWidth
```

```
'Arregla la altura del PICSCROLL
```

```
    RectHeight = DrawingRect.Bottom
```

```
    DrawingRect.Bottom = DrawingRect.Bottom + picScroll.ScaleHeight
```

```
End If
```

```
Do While Not EndingFlag
```

```

If GetTickCount() - LastFrameTime > IntervalTime Then
    picScroll.Cls

    DrawText picScroll.hdc, ScrollText, -1, DrawingRect, DT_CENTER Or
DT_WORDBREAK
    'Actualiza las coordenadas del rectangulo
    DrawingRect.Top = DrawingRect.Top - 1
    DrawingRect.Bottom = DrawingRect.Bottom - 1
    'Controla el PICSCROLL y lo reinicia si se sale de su limite(si termina)
    If DrawingRect.Top < -(RectHeight) Then 'Tiempo de reinicio
        DrawingRect.Top = picScroll.ScaleHeight
        DrawingRect.Bottom = RectHeight + picScroll.ScaleHeight
    End If
    picScroll.Refresh
    LastFrameTime = GetTickCount()
End If
DoEvents
Loop
Unload Me
Set frmAcerca = Nothing
End Sub
Private Sub Form_Unload(Cancel As Integer)
    EndingFlag = True
End Sub
Private Sub picScroll_Click()
    EndingFlag = True
End Sub
Private Sub Timer1_Timer()
    Me.Icon = Me.ImgIcono.ListImages(Icono).Picture
    If Icono <> 24 Then
        Icono = Icono + 1
    Else
        Icono = 1
    End If
End Sub
frmAcerca2 (Código)
Option Explicit
Private Sub Form_Click()
    Unload Me
End Sub
Private Sub Label1_Click(Index As Integer)
    Unload Me
End Sub

```



```

If Archivohtml2.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\Grafica.bmp") Then
  'BORRARLO
  Archivohtml2.DeleteFile ("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\Grafica.bmp")
End If
Load frmWebCreada
frmWebCreada.Show 1
' MsgBox "Pagina web creada...", , "Crear página Web"
Unload frmAviso
frmtermo.cmdVerPag.Visible = True
End If
End If
frmtermo.cmdVerPag.Enabled = True 'habilitar el boton ver pagina
frmtermo.cmdEnvio.Enabled = True 'habilitar el boton Enviar Dictamen
End Sub
Private Sub cmdRegresar_Click()
  frmAviso.Hide
  'frmtermo.Show
  frmtermo.cmdVerPag.Enabled = False
  frmtermo.cmdEnvio.Enabled = False
End Sub
frmCanalHabil (Código)
Option Explicit
Private Sub tmrAviso_Timer()
  Unload frmCanalHabil
End Sub
Private Sub cmdAceptar_Click()
  Unload frmCanalHabil
End Sub
Private Sub Form_Load()
  If Dato = &H5 Then 'si el dato mandado es el b = 11 (en hexadecimal)
    lblAviso.Caption = "El canal 3 ha sido habilitado satisfactoriamente..."
  ElseIf Dato = &H3 Then 'si el dato mandado es el 1 (1 en hexadecimal)
    lblAviso.Caption = "El canal 1 ha sido habilitado satisfactoriamente..."
  End If
End Sub
frmDiagnosticos (Código)
Option Explicit
Private Sub cmdRegresar_Click()
  Load frmMenu
  frmMenu.Show
  Unload frmDiagnosticos
End Sub
Private Sub cmdTermo_Click()

```

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

```
Load frmReDirec 'carga frmRedirec
frmReDirec.Show 'muestra frmRedirec
Unload frmDiagnosticos 'descarga frmDiagnosticos
Canal = 3 'boton del Termometro
End Sub
Private Sub cmdTermo2_Click()
Load frmReDirec 'carga frmRedirec
frmReDirec.Show 'muestra frmRedirec
Unload frmDiagnosticos 'descarga frmDiagnosticos
Canal = 1 'boton del Termometro
End Sub

Private Sub Form_Load()
End Sub
Private Sub submnuAcerca_Click()
Load frmAcerca2
frmAcerca2.Show 1
End Sub
Private Sub submnuAyuda_Click()
Load frmHelp
frmHelp.Show 1
End Sub
Private Sub submnuRegresar_Click()
cmdRegresar_Click
End Sub
Private Sub submnuTemperaturaAmbiental_Click()
cmdTermo2_Click
End Sub
Private Sub submnuTermometro_Click()
cmdTermo_Click
End Sub
frmEnviar (Código)
Dim Archivo As New FileSystemObject
Dim RutaLocal1 As String
Dim RutaLocal2 As String
Dim RutaLocal3 As String
Dim RutaLocal4 As String
Dim RutaLocal5 As String
Dim RutaLocal6 As String
Dim RutaDestino1 As String
Dim RutaDestino2 As String
Dim RutaDestino3 As String
Dim RutaDestino4 As String
Dim RutaDestino5 As String
Dim RutaDestino6 As String
```

```

Private Sub cmdEnviar_Click()
Inet1.UserName = txtNombre.Text
Inet1.Password = txtPassword.Text
Inet2.UserName = txtNombre.Text
Inet2.Password = txtPassword.Text
Inet3.UserName = txtNombre.Text
Inet3.Password = txtPassword.Text
Inet4.UserName = txtNombre.Text
Inet4.Password = txtPassword.Text
Inet5.UserName = txtNombre.Text
Inet5.Password = txtPassword.Text
Inet6.UserName = txtNombre.Text
Inet6.Password = txtPassword.Text
'DireccionServidor = "ftp://132.248.173.89"

'RUTA LOCAL FONDO DE LA PAGINA WEB
RutaLocal1 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\FONDO.JPG"
'RUTA LOCAL IMAGEN UNAM
RutaLocal2 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\UNAM.GIF"
'RUTA LOCAL IMAGEN TELEMEDICA
RutaLocal3 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\TELEMEDICA.JPG"
'RUTA LOCAL IMAGEN RANGO
RutaLocal4 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\RANGO.JPG"
'RUTA LOCAL IMAGEN GRAFICA
RutaLocal5 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\Grafica.bmp"
'RUTA PAGINA WEB
'AVISO: NO HACEPTA NOMBRES DE ARCHIVO CON ESPACIOS (COMO EN LA
FECHA CON FORMATO "LONG DATE")
RutaLocal6 = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & ".html"
'RUTA DESTINO FONDO DE LA PAGINA WEB:
RutaDestino1 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & "\FONDO.JPG"
'RUTA DESTINO IMAGEN UNAM:
RutaDestino2 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & "\UNAM.GIF"
'RUTA DESTINO IMAGEN TELEMEDICA:
RutaDestino3 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & "\TELEMEDICA.JPG"
'RUTA DESTINO IMAGEN RANGO:
RutaDestino4 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & "\RANGO.JPG"
'RUTA DESTINO IMAGEN GRAFICA:
RutaDestino5 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & "\Grafica.bmp"
'RUTA DESTINO PAGINA WEB:
RutaDestino6 = "\telemedica\" & Format(Date, "dd-mm-yyyy") & ".html"
If txtNombre.Text <> "pepe" Or txtPassword.Text <> "joseluis" Then
MsgBox "El password es incorrecto...! Compruebe que esté ó no habilitado la tecla de
Mayúsculas...", , "Enviar Dictamen"

```

```
txtPassword.Text = ""
txtPassword.SetFocus
Exit Sub
End If
'si existe el archivo "grafica.bmp" quiere decir que se mandará
'el dictamen en MODO GRAFICO
If Archivo.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\Grafica.bmp") Then
'MANDAR TODOS LOS ARCHIVOS
'si el control esta ocupado mandando el archivo entonces...
Inet1.Execute , "PUT " & RutaLocal1 & " " & RutaDestino1
Inet2.Execute , "PUT " & RutaLocal2 & " " & RutaDestino2
Inet3.Execute , "PUT " & RutaLocal3 & " " & RutaDestino3
Inet4.Execute , "PUT " & RutaLocal4 & " " & RutaDestino4
Inet5.Execute , "PUT " & RutaLocal5 & " " & RutaDestino5
Inet6.Execute , "PUT " & RutaLocal6 & " " & RutaDestino6
MsgBox "Su dictámen ha sido enviado con éxito..." , , "Dictamen Enviado
Exitosamente..."
'Res = MsgBox("Su dictámen ha sido enviado con éxito..." , , "Dictamen Enviado
Exitosamente...")
' If Res = vbOK Then
' Unload frmEnviar
' End If
'de lo contrario si NO existe el archivo "grafica.bmp" se mandará
'en MODO TEXTO
ElseIf Archivo.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\Grafica.bmp") = False Then
'MANDAR UNICAMENTE LOS QUE SE OCUPEN
Inet1.Execute , "PUT " & RutaLocal1 & " " & RutaDestino1
Inet2.Execute , "PUT " & RutaLocal2 & " " & RutaDestino2
Inet3.Execute , "PUT " & RutaLocal3 & " " & RutaDestino3
Inet4.Execute , "PUT " & RutaLocal6 & " " & RutaDestino6
MsgBox "Su dictámen ha sido enviado con éxito..." , , "Dictamen Enviado
Exitosamente..."
'LA ruta 4 y 5 no porque es de SOLO TEXTO (CORRESPONDEN A
'(LA GRAFICA Y EL RANGO((0°C - 100°C)
'Res = MsgBox("Su dictámen ha sido enviado con éxito..." , , "Dictamen Enviado
Exitosamente...")
'If Res = vbOK Then
' Unload frmEnviar
'End If
End If
End Sub
Private Sub cmdRegresar_Click()
Unload frmEnviar
```

```

End Sub
'''Private Sub Inet1_StateChanged(ByVal State As Integer)
'Este suceso se activa cuando el control finaliza varias tareas,
'como conectarse y registrar los errores
''' Select Case State
''' Case icError 'si es que existe un error, describirlo
'ResponseCode.- Devuelve el código de error de la conexión
'cuando aparece el estado icError (11) en el evento StateChanged
''' If Inet1.ResponseCode = 80 Then 'si es que existe el archivo
''' MsgBox "¡El archivo existe! ¿Desea sobrescribirlo?"
''' Else 'si el error no es el 80, describir el tipo de error
''' MsgBox Inet1.ResponseInfo, "Ha fallado la transferencia del archivo..."
''' End If
''' Case icResponseCompleted 'si la transferencia ha tenido éxito
'Load frmConfirma
' frmConfirma.lblAviso.Caption = "Archivo Enviado..."
' frmConfirma.Show

''' MsgBox "Su dictámen ha sido enviado con éxito...", RutaDestino
''' Unload frmEnviar
'case
''' End Select
'''End Sub
Private Sub Form_Load()
'Dim DireccionServidor As String 'la direccion del servidor ftp
Inet1.OpenURL ("ftp://132.248.173.89")
Inet2.OpenURL ("ftp://132.248.173.89")
Inet3.OpenURL ("ftp://132.248.173.89")
Inet4.OpenURL ("ftp://132.248.173.89")
Inet5.OpenURL ("ftp://132.248.173.89")
Inet6.OpenURL ("ftp://132.248.173.89")
End Sub
Private Sub submnuEnviar_Click()
cmdEnviar_Click
End Sub
Private Sub submnuRegresar_Click()
cmdRegresar_Click
End Sub
Private Sub txtPassword_Change()
If txtPassword.Text <> "" Then
cmdEnviar.Enabled = True
Else
cmdEnviar.Enabled = False
End If
End Sub

```

frmGrafica (Código)

```
Dim i As Integer
Dim X1 As Integer
Dim X2 As Integer
Dim Y1 As Integer
Dim Y2 As Integer
Dim PuntoX1 As Long
Dim PuntoY1 As Long
Dim PuntoX2 As Long
Dim PuntoY2 As Long
Dim Distancia As Integer
Dim NomGraficaTemp As String
Private Const ValorMaximoPictureBox As Integer = 100
Private Sub cmdCapturarGrafica_Click()
'quiere decir que la grafica ha sido guardada grafica = True
Grafica = True
'se establece la ruta, el nombre del archivo y la extensión
'se asigna todo a la variable: NomGraficaTemp
  NomGraficaTemp = "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\Grafica.bmp"
  CommonDialog1.FileName = NomGraficaTemp
  SavePicture PictureGde.Image, CommonDialog1.FileName ' Guarda la imagen en la ruta
correspondiente.
  Load frmGraficaGuardada
  frmGrafica.Hide
  frmGraficaGuardada.Show 1
  'MsgBox "Archivo guardado..."
End Sub
Private Sub cmdIniciar_Click()
'Call EdoGrafica
'If Reproduccion = True Then 'si se está reproduciendo la grafica chica
  cmdRegresar.Enabled = False 'inhabilitar el boton regresar
  cmdCapturarGrafica.Enabled = True 'habilitar el boton capturar grafica
  cmdPausa.Enabled = True 'habilitar el boton pausa
  submnuPausa.Enabled = True
  submnuDetener.Enabled = True
'Call EdoGrafica
  cmdIniciar.Enabled = False 'inhabilitar el boton de Reproducir grafica
  submnuRegresar.Enabled = False 'deshabilita el boton Regresar del menu
  cmdParar.Enabled = True 'habilitar el boton Parar grafica
  submnuCapturarGrafica.Enabled = True 'habilita el boton Capturar Grafica del menu
'se limpia el PictureBox
  PictureGde.Cls
'se cuadrícula
  Cuadros
```



```

frmGrafica.Show
'Se habilita el timer1
Timer1.Enabled = True 'habilitar el control que grafica en el PictureBox grande
End Sub
Private Sub cmdParar_Click()
PuntoX1 = 0 'inicializar el PuntoX1 = 0
'instrucciones para el grafico chico
frmtermo.Timer2.Enabled = False 'detener el timer que grafica
'en el PictureBox Chico del frmTermo
frmtermo.picGrafica.Cls 'limpiar el PictureBox Grande
frmtermo.cmdReproducir.Enabled = True 'habilitar el boton Reproducir grafica
submnuRegresar.Enabled = True 'habilita el boton Regresar del menu
frmtermo.cmdStop.Enabled = False 'inhabilitar el boton Detener grafica
submnuPausa.Enabled = False
submnuDetener.Enabled = False
submnuReproducir.Enabled = False
submnuCapturarGrafica.Enabled = False 'deshabilita el boton Capturar Grafica del menu
frmtermo.cmdPausa.Enabled = False 'inhabilitar el boton Pausa
Call Cuadros_Chicos 'dejar el PictureBox Chico cuadriculado
'Instrucciones para el gráfico grande
Timer1.Enabled = False 'detener el timer que grafica en el PictureBox Grande
cmdCapturarGrafica.Enabled = False 'deshabilitar el boton Capturar grafica
cmdRegresar.Enabled = True 'habilitar el botón Regresar
cmdIniciar.Enabled = True 'habilitar el boton Reproducir grafica
cmdPausa.Enabled = False 'deshabilitar el boton Pausa
cmdParar.Enabled = False 'deshabilitar el boton Detener grafica
PictureGde.Cls 'Limpiar el PictureBox Grande
Cuadros 'dejar cuadriculado el PictureBox Grande
End Sub
Private Sub cmdPausa_Click()
If submnuPausa.Checked = True Then
submnuPausa.Checked = False
ElseIf submnuPausa.Checked = False Then
submnuPausa.Checked = True
End If
If Timer1.Enabled = True Then
Timer1.Enabled = False 'pausa en la gráfica
Else
Timer1.Enabled = False
Timer1.Enabled = True 'reanuda la gráfica
End If
End Sub
Private Sub cmdRegresar_Click()
frmtermo.TimerTemp.Enabled = True 'habilitar el timer que muestra la
'temperatura en el lblTemp del frmTermo

```

```
frmtermo.Timer1.Enabled = True 'habilitar el timer que muestra la
'hora en el frmtermo
'frmtermo.Timer2.Enabled = True 'habilitar el timer que grafica en
'el PictureBox Chico
Load frmtermo
frmtermo.Show
Unload frmGrafica
End Sub
Private Sub Form_Load()
PictureGde.ScaleMode = 6 'mm modo de trabajo para la Grafica grande
lblFecha.Caption = Date
Call EdoGrafica 'Llamar a la funcion Estado de la Grafica (Chica)
If Reproduccion = True Then 'si se está reproduciendo la grafica chica
cmdIniciar_Click 'nos vamos directamente a lo que hay en el boton
'reproducir de la grafica grande
submnuReproducir.Enabled = False 'deshabilita el boton reproducir del menu
Elseif Reproduccion = False Then 'si no se está reproduciendo nada
submnuCapturarGrafica.Enabled = False 'deshabilita el boton Capturar Grafica del menu
cmdRegresar.Enabled = True 'el boton regresar estará habilitado
cmdCapturarGrafica.Enabled = False 'el boton capturar grafica estará deshabilitado
cmdPausa.Enabled = False 'el boton pausa estará deshabilitado
submnuPausa.Enabled = False
submnuDetener.Enabled = False
'Call EdoGrafica
cmdIniciar.Enabled = True 'el boton reproducir estará habilitado
cmdParar.Enabled = False 'el boton Parar estará deshabilitado
'se limpia el PictureBox
PictureGde.Cls
'se cuadrícula
Cuadros
frmGrafica.Show
'Se habilita el timer1
Timer1.Enabled = False 'detener el que grafica en el PictureBox Grande
End If
Distancia = 4
ScrollDistancia.Value = ScrollDistancia.Max - 1
PictureGde.Cls
Cuadros
End Sub
Public Sub Cuadros() 'para el PictureBox Grande
PictureGde.DrawWidth = 1 'ancho de lineas
X1 = 0
X2 = 145
Y1 = 0
Y2 = 105
```

```

PictureGde.ForeColor = &HE0E0E0 'gris claro
'lineas horizontales delgadas
For i = 1 To 100
    PictureGde.Line (X1, i)-(X2, i)
Next i
'lineas verticales delgadas
For i = 1 To 140
    PictureGde.Line (i, Y1)-(i, Y2)
Next i
PictureGde.ForeColor = &H808080 'gris fuerte
'lineas horizontales gruesas
For i = 10 To 100 Step 10
    PictureGde.Line (X1, i)-(X2, i)
Next i
'lineas verticales gruesas
For i = 10 To 140 Step 10
    PictureGde.Line (i, Y1)-(i, Y2)
Next i
PictureGde.ForeColor = &H8080FF 'color rojo suave linea divisoria (temperatura alta)
'entre temperatura normal y temperatura excedida
PictureGde.DrawWidth = 2
PictureGde.Line (0, 60)-(145, 60)
'la otra linea divisoria (temperatura baja)
PictureGde.Line (0, 82)-(145, 82)

'frmGrafica.Show
End Sub
Private Sub ScrollDistancia_Change()
    Distancia = ScrollDistancia.Max + 1 - ScrollDistancia.Value
    lblDistancia.Caption = Distancia
End Sub
Private Sub submnuAcercade_Click()
    Load frmAcerca2
    frmAcerca2.Show 1
End Sub
Private Sub submnuDetener_Click()
    cmdParar_Click
    submnuReproducir.Enabled = True
    submnuPausa.Checked = False
End Sub
Private Sub submnuPausa_Click()
    cmdPausa_Click
End Sub

```

```
Private Sub submnuRegresar_Click()
    cmdRegresar_Click
End Sub
Private Sub submnuReproducir_Click()
    cmdIniciar_Click 'nos vamos directamente a lo que hay en el boton
    submnuReproducir.Enabled = False
End Sub
Private Sub submnuTemaAyuda_Click()
    Load frmHelp
    frmHelp.Show 1
End Sub
Private Sub Timer1_Timer()
'Graficar en el PictureBox Grande
'(PuntoX2, PuntoY2) punto destino para trazar la linea
PuntoX2 = PuntoX1 + Distancia 'distancia horizontal punto destino
'distancia vertical punto destino
PuntoY2 = ValorMaximoPictureBox - (LeerDato(DataPort) / 2)
lblTemp.Caption = Int(LeerDato(DataPort) / 2) & " °C"
lblCoord.Caption = "(" & PuntoX1 & "," & PuntoY1 & ")-(" & PuntoX2 & "," & PuntoY2 &
")"
'establece el ancho de la linea a dibujar
PictureGde.DrawWidth = 5
If PuntoY1 <= 82 And PuntoY1 >= 60 Then 'si la temperatura esta en el rango (18-40)
    PictureGde.ForeColor = &HC00000 'azul marino para graficar
'de lo contrario si la temperatura es mayor o menor del rango
Else: PictureGde.ForeColor = &HFF& 'rojo para graficar
End If
PictureGde.Line (PuntoX1, PuntoY1)-(PuntoX2, PuntoY2)
PuntoX1 = PuntoX2
PuntoY1 = PuntoY2
'Si puntoX1 es mayor que el espacio total del PictureBox
'en forma horizontal entonces:
If PuntoX1 > PictureGde.ScaleWidth Then
    Me.Cls
    PuntoX1 = 0
    PictureGde.Cls
    Cuadros
End If
End Sub
Private Sub TimerHora_Timer()
lblHora.Caption = Time
End Sub
Public Sub EdoGrafica()
'si el boton Reproducir de frmTermo esta habilitado
'(quiere decir que el gráfico no ha empezado) entonces:
```

```

If frmtermo.cmdReproducir.Enabled = True Then
    'Inicializa los valores
    '(PuntoX1, PuntoY1) punto inicial para trazar linea
    'PuntoX1 avanza en forma horizontal de izq. a der.
    'comenzando en la parte superior
    PuntoX1 = 0
    Reproduccion = False
    'Punto Y1 avanza en forma vertical de arriba hacia abajo
    'comenzando en la parte superior
    PuntoY1 = 100 'parte mayor de la ventana (vertical)
'de lo contrario si el boton Reproducir de frmTermino no esta habilitado
'(quiere decir que ya se empezó a graficar) entonces:
ElseIf frmtermo.cmdReproducir.Enabled = False Then
    'PuntoX1 avanza en forma horizontal de izq. a der.
    'comenzando en la parte superior
    PuntoX1 = 0
    Reproduccion = True
    'Punto Y1 avanza en forma vertical de arriba hacia abajo
    'comenzando en la parte superior
    PuntoY1 = ValorMaximoPictureBox - (LeerDato(DataPort) / 2)

```

End If

End Sub

frmGraficaGuardada (Código)

Option Explicit

Private Sub cmdAceptar_Click()

 Unload frmGraficaGuardada

 frmGrafica.Show

End Sub

frmHelp (Código)

Option Explicit

Private Sub cmdAceptar_Click()

 Unload frmHelp

End Sub

frmMenu (Código)

Private Sub cmdAcerca_Click()

 Load frmAcerca2

 frmAcerca2.Show 1

End Sub

Private Sub cmdAyuda_Click()

 Load frmHelp

 frmHelp.Show 1

End Sub

Private Sub cmdDiagnosticos_Click()

 Load frmDiagnosticos

```
frmDiagnosticos.Show
Unload frmMenu
End Sub
Private Sub cmdSalir_Click()
    End
End Sub
Private Sub Form_Load()
End Sub
Private Sub submnuAcercade_Click()
    cmdAcerca_Click
End Sub
Private Sub submnuAuyda_Click()
    cmdAyuda_Click
End Sub
Private Sub submnuDiagnosticos_Click()
    cmdDiagnosticos_Click
End Sub
Private Sub submnuSalir_Click()
    cmdSalir_Click
End Sub
frmReDirec (Código)
'para que se cierre automaticamente el programa Dos le añadimos antes
'de la Ruta en donde se encuentra el programa: command.com /C
'Private Const Ruta As String = "C:\Mis
documentos\FOLDER_3\JOSE_LUIS\CLIENTE\PROG_MS_DOS\REDIRECC.EXE"
'Private Const Ruta As String = "command.com /C C:\WINDOWS\Escritorio\Lo que me
lleve\cliente\prog MS-dos\REDIRECC.EXE"
Private Sub cmdRedirecc_Click()
    Select Case Canal
        Case 3
            Dato = &H5 'para habilitar el canal 3
            Load frmtermo
            frmtermo.Show
            Unload frmReDirec
            ControlPort = &H37A '(890 en decimal) 'ésta dirección le corresponde
            'Manda un valor al ControlPort
            MandarDato ControlPort, Dato
            Load frmCanalHabil
            frmCanalHabil.Show
            'de acuerdo a la dirección del DataPort (Base+2 = 888+2)
            'llama a la funcion Para Redireccionar el puerto
            Redirecc (&H378) 'redireccionar el puerto 888
            'Shell Ruta, vbNormalNoFocus 'ejecutar prog ms-dos
        Case 1
            Dato = &H3 'para habilitar el canal 1 (del 0 al 7)
```



```

Load frmTermino2
frmTermino2.Show
Unload frmReDirec
ControlPort = &H37A '(890 en decimal) 'ésta dirección le corresponde
'Manda un valor al ControlPort
MandarDato ControlPort, Dato
Load frmCanalHabil
frmCanalHabil.Show
'de acuerdo a la dirección del DataPort (Base+2 = 888+2)
'llama a la función Para Redireccionar el puerto
Redirecc (&H378) 'redireccionar el puerto 888
'Shell Ruta, vbNormalNoFocus 'ejecutar prog ms-dos
End Select
End Sub
Private Sub cmrRegresar_Click()
frmReDirec.Hide
frmDiagnosticos.Show
End Sub
Private Sub Form_Load()
End Sub
Private Sub submenuRegresar_Click()
cmrRegresar_Click
End Sub
Private Sub submnuAcerca_Click()
Load frmAcerca2
frmAcerca2.Show 1
End Sub
Private Sub submnuRedireccPuerto_Click()
cmdRedirecc_Click
End Sub
Private Sub submnuTemaAyuda_Click(Index As Integer)
Load frmHelp
frmHelp.Show 1
End Sub
frmtermo (Código)
'para visualizar la página Web creada
'Declarar una variable para el URL actual
Public Explorer As SHDocVw.InternetExplorer
'para gráfica
Dim i As Integer
Dim X1 As Integer
Dim Y1 As Integer
Dim X2 As Integer
Dim Y2 As Integer
Dim PuntoX1 As Long

```

```
Dim PuntoY1 As Long
Dim PuntoX2 As Long
Dim PuntoY2 As Long
Dim Distancia As Integer
Private Const ValorMaximoPictureBox As Integer = 100
Dim RutaPagGrafica As String
Dim FechaActual As String
Private Sub cmdAmpliarGraf_Click()
    Timer2.Enabled = False 'detener temporalmente el timer que grafica en el
    'PictureBox Chico
    Timer1.Enabled = False 'detener temporalmente el timer que muestra la hora en
    'el frmTermo
    Timertemp.Enabled = False 'detener temporalmente el timer que muestra la
    'temperatura en la etiqueta lblTemp
    Load frmGrafica 'cargar form
    frmGrafica.Show 'mostrar form
    frmtermo.Hide
End Sub
Private Sub cmdEnvio_Click()
    Load frmEnviar 'cargar frmEnviar
    frmEnviar.Show 'Mostrar frmEnviar
    frmEnviar.txtPassword.SetFocus
End Sub
Private Sub cmdOcultarGrafica_Click()
    cmdAmpliarGraf.Enabled = False
    cmdVerGrafica.Enabled = True
    cmdOcultarGrafica.Enabled = False
    Frame1.Visible = True
    lblCoord.Visible = False
    shpControles.Visible = False
    VSDistancia.Visible = False
    picGrafica.Visible = False
    lblDistancia.Visible = False
    cmdStop.Visible = False
    cmdPausa.Visible = False
    cmdReproducir.Visible = False
    lblCord.Visible = False
    lbl100.Visible = False
    lbl40.Visible = False
    lbl0.Visible = False
End Sub
Private Sub cmdPausa_Click()
    If Timer2.Enabled = True Then
        Timer2.Enabled = False
    Else
```

```

    Timer2.Enabled = True
End If
End Sub
Private Sub cmdAcercade_Click()
'Presenta formulario a cerca de...
Load frmAcerca
frmAcerca.Show 1
End Sub
Private Sub cmdMedirtemp_Click()
If cmdMedirtemp.Caption = "&Medir Temperatura" Then
    Timertemp.Enabled = True 'habilitar el timer que lee la temperatura actual
    cmdMedirtemp.Caption = "Detener &medición"
    cmdMedirtemp.ToolTipText = "Detiene la medición"
    cmdGuardar.Enabled = True
    Frame1.Enabled = False
    Option1.Enabled = False 'inhabilitar opciones de escoger otro puerto
    Option2.Enabled = False 'inhabilitar opciones de escoger otro puerto
    cmdRegresar.Enabled = False
    cmdVerGrafica.Enabled = True
Else ' si el boton dice: "Detener Medición"
    cmdGuardar.Enabled = False
    cmdMedirtemp.Caption = "&Medir Temperatura"
    cmdMedirtemp.ToolTipText = "Medir temperatura"
    cmdVerGrafica.Enabled = False
    Timertemp.Enabled = False
    lblTemp.Caption = "0 Grados Centígrados"
    Frame1.Enabled = True
    Option1.Enabled = True
    Option2.Enabled = True
    cmdAmpliarGraf.Visible = True
    cmdRegresar.Enabled = True
End If
If cmdOcultarGrafica.Enabled = True Then 'si el boton ocultar grafica
'esta habilitado
    Frame1.Visible = True
    cmdOcultarGrafica.Enabled = False
    cmdVerGrafica.Enabled = False
    lblCoord.Visible = False
    shpControles.Visible = False
    VSDistancia.Visible = False
    picGrafica.Visible = False
    lblDistancia.Visible = False
    cmdStop.Visible = False
    cmdPausa.Visible = False
    cmdReproducir.Visible = False

```

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

```
cmdAmpliarGraf.Visible = True
cmdAmpliarGraf.Enabled = False
lblCord.Visible = False
lbl100.Visible = False
lbl40.Visible = False
lbl0.Visible = False
End If
If Option1.Value = True Then
    DataPort = &H378 'Base (888 en decimal)
Elseif Option2.Value = True Then
    DataPort = &H632 'Base (1586 en decimal)
End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
    MMControl1.Command = "Close" 'descargar el control para liberar recursos
End Sub
Private Sub Timertemp_Timer()
    'lee el valor del DataPort y lo divide entre 2
    'esto es por el rango elegido
    Temperatura = LeerDato(DataPort) / 2
    picGrafica.ToolTipText = "Temperatura Actual: " & lblTemp.Caption
    If Temperatura > 39 Then 'si la temperatura es mayor a 39°C
        lblTemp.ForeColor = &H8080FF 'cambia el color a rojo
        lblTemp.Caption = Temperatura & " Grados Centígrados"
        MMControl1.Command = "Open" 'Preparar control para Reproducción
        'Abre un dispositivo mediante el comando MCI_OPEN.
        MMControl1.Command = "play" 'Reproducir sonido de alarma
    Elseif Temperatura > 18 And Temperatura < 40 Then 'si la temperatura
        'está entre 19°C y 39°C entonces
        lblTemp.ForeColor = &HC0FFFF 'mantiene el color a amarillo
        MMControl1.Command = "Close" 'cerrar el control para Reproducción
        lblTemp.Caption = Temperatura & " Grados Centígrados"
    Elseif Temperatura < 19 Then 'si temperatura es menor que 19°C
        lblTemp.ForeColor = &H8080FF 'cambia el color a rojo
        lblTemp.Caption = Temperatura & " Grados Centígrados"
        MMControl1.Command = "Open" 'Preparar control para Reproducción
        'Abre un dispositivo mediante el comando MCI_OPEN.
        MMControl1.Command = "play" 'Reproducir sonido de alarma
    End If
    Comenzar = Timer
    Do While Timer < Comenzar + 1
        DoEvents
    Loop
End Sub
Private Sub cmdGuardar_Click()
```



```
Name "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & ".txt" As
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & ".html"
'Ahora hay que copiar los archivos que se van a mandar al servidor:
'de la carpeta DICTAMEN_ARCHIVOS
'a la carpeta creada con la fecha actual
'fondo.jpg, unam.gif, telemedica.jpg
'(la grafica.bmp ya fue creada en la ubicación: "C:\JOSE LUIS\CLIENTE\" &
Format(Date, "dd-mm-yyyy"))
If Archivohtml.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\FONDO.JPG") _
And Archivohtml.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\UNAM.GIF") _
And Archivohtml.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\TELEMEDICA.JPG") _
And Archivohtml.FileExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\RANGO.JPG") = True Then
    Archivohtml.DeleteFile "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\FONDO.JPG", True
    Archivohtml.DeleteFile "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\UNAM.GIF", True
    Archivohtml.DeleteFile "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\TELEMEDICA.JPG", True
    Archivohtml.DeleteFile "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
"\RANGO.JPG", True
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\FONDO.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\FONDO.JPG", True
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\UNAM.GIF",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\UNAM.GIF", True
    Archivohtml.CopyFile
"D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\TELEMEDICA.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\TELEMEDICA.JPG", True
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\RANGO.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\RANGO.JPG", True
Else
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\FONDO.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\FONDO.JPG", True
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\UNAM.GIF",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\UNAM.GIF", True
    Archivohtml.CopyFile
"D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\TELEMEDICA.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\TELEMEDICA.JPG", True
    Archivohtml.CopyFile "D:\JLUIS\CLIENTE\DICTAMEN_ARCHIVOS\RANGO.JPG",
"D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy") & "\RANGO.JPG", True
End If
'MsgBox "Pagina web creada...", , "Crear página Web"
```

```

Load frmWebCreada
frmWebCreada.Show 1
cmdVerPag.Enabled = True 'habilitar el boton para ver Dictamen creado
'habilitar el boton Enviar Dictamen
cmdEnvio.Enabled = True
End If
End If
End If
End Sub
Private Sub cmdRegresar_Click()
    frmtermo.Hide
    Load frmDiagnosticos
    frmDiagnosticos.Show
End Sub
Private Sub cmdReproducir_Click()
cmdStop.Enabled = True
cmdReproducir.Enabled = False
cmdPausa.Enabled = True
'inicializar valores
PuntoX1 = 0
'PictureBox Pantalla
'PointY es igual al total de largo en forma vertical
'menos uno
PuntoY1 = picGrafica.ScaleHeight - 1
'se establece el ancho
BorderWidth = 1
'se limpia el picturebox
picGrafica.Cls
Timer2.Enabled = True
End Sub
Private Sub cmdStop_Click()
PuntoX1 = 0 'inicializar el PuntoX1 = 0
'para el grafico
Timer2.Enabled = False
picGrafica.Cls
cmdReproducir.Enabled = True
cmdStop.Enabled = False
cmdPausa.Enabled = False
Call Cuadros_Chicos 'llama funcion para dejar los cuadros mostrados
PuntoX1 = 0
PuntoY1 = ValorMaximoPictureBox - (LeerDato(DataPort) / 2)
End Sub
Private Sub cmdVerGrafica_Click()
Timer2.Enabled = True 'habilitar el timer que grafica en el PictureBox Chico
If Option1.Value = True Then

```

Sistema de adquisición y envío de datos biomédicos a través de Internet utilizando el protocolo FTP

```
DataPort = &H378 'si el DataPort elegido es el 888
Elseif Option2.Value = True Then
    DataPort = &H632 'si el DataPort elegido es el 1586
End If
cmdVerGrafica.Enabled = False 'deshabilita el boton ver grafica
cmdOcultarGrafica.Enabled = True 'habilita el boton ocultar grafica
Frame1.Visible = False 'oculta el frame1
lblCoord.Visible = True 'hace visible las coordenadas
VSDistancia.Visible = True 'hace visible el scroll de distancia
picGrafica.Visible = True 'hace visible el picturebox
shpControles.Visible = True 'hace visible el recuadro de controles
lbl100.Visible = True 'hace visible la etiqueta de 100°C
lbl40.Visible = True 'hace visible la etiqueta de 50°C
lbl0.Visible = True 'hace visible la etiqueta de 10°C
lblDistancia.Visible = True 'hace visible lblDistancia
cmdReproducir.Visible = True 'hace visible el boton reproducir
cmdReproducir.Enabled = False 'inhabilita el boton reproducir
cmdStop.Visible = True 'hace visible el boton stop
cmdStop.Enabled = True 'habilita el boton stop
cmdPausa.Visible = True 'hace visible el boton pausa
cmdAmpliarGraf.Enabled = True 'habilitar el boton ampliar grafica
'inicializar valores
'(PuntoX1, PuntoY1) punto inicial para trazar linea
'PuntoX1 avanza en forma horizontal de izq. a der.
'comenzando en la parte superior
    PuntoX1 = 0
'Punto Y1 avanza en forma vertical de arriba hacia abajo
'comenzando en la parte superior
    PuntoY1 = 100
'se limpia el picturebox
picGrafica.Cls
'se cuadricula
Cuadros_Chicos
End Sub
Private Sub cmdVerPag_Click()
ComDialogAbrirPag.FileName = "D:\JLuis\CLIENTE\" & Format(Date, "dd-mm-yyyy") &
".html"
'se abre el cuadro de dialogo abrir Ver Página Web
ComDialogAbrirPag.ShowOpen
If ComDialogAbrirPag.FileName <> "" Then
    On Error GoTo manejadorerror 'si hay error ir al manejador de error
    Set Explorer = New SHDocVw.InternetExplorer 'crea el objeto
'para hacer uso del explorador:
    Explorer.Visible = True
    Explorer.Navigate ComDialogAbrirPag.FileName ' se declara la dirección en donde esta
```

```

    'la pagina web
    Exit Sub
End If
manejadorerror:
    MsgBox "Error visualizando el archivo", , Err.Description
End Sub
Private Sub Form_Load()
'crear el folder que se enviará con el nombre de la fecha actual
'primero crear el objeto para el manejo de archivos, carpetas, etc.
Set Folder = CreateObject("Scripting.FileSystemObject")
'si existe el folder, es decir, si ya se había creado antes
If Folder.FolderExists("D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy")) Then
    'Borrarlo y ...
    Folder.DeleteFolder "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy"), True
    'Volverlo a crear
    Folder.CreateFolder "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy")
Else 'Es decir, si no existe el folder (no se ha creado)
    'crearlo
    Folder.CreateFolder "D:\JLUIS\CLIENTE\" & Format(Date, "dd-mm-yyyy")
End If
'para reproducir la alarma cuando la temperatura exceda de 39°C
'o baje a menos de 18°C
MMControl1.DeviceType = "WaveAudio"
'Especifica el tipo de dispositivo MCI que se va a abrir.
MMControl1.Notify = False
'No queremos que nos avise lo que va ejecutando
MMControl1.Wait = True
'Determina si el control Multimedia MCI espera a completar el
'siguiente comando MCI antes de devolver el control a la aplicación.
MMControl1.Shareable = False
'Ningún otro control o aplicación puede tener acceso a este dispositivo.
MMControl1.FileName = "D:\JLuis\CLIENTE\ALARMA.WAV"
'Ruta del archivo a reproducir
'grafica es igual a falso
'quiere decir que no se ha entrado a capturar la grafica
Grafica = False
'para grafica
Distancia = 4
VSDistancia.Value = VSDistancia.Max - 1
'valor scrollDistancia = Valor Maximo - 1
'para grafica
picGrafica.ScaleMode = 3 'trabajar en pixeles
'Cuadros_Chicos
lbldate.Caption = Date
Activar = False

```

```
Temperatura = 0
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
  Cuadros_Chicos
```

```
  'graficar en el PicGrafica
```

```
  '(PuntoX2, PuntoY2) punto destino para trazar la linea
```

```
    PuntoX2 = PuntoX1 + Distancia 'distancia horizontal punto destino
```

```
    'distancia vertical punto destino
```

```
    PuntoY2 = ValorMaximoPictureBox - LeerDato(DataPort) / 2
```

```
    'lblTemp.Caption = Inp(DataPort)
```

```
    lblCord.Caption = "(" & PuntoX1 & "," & PuntoY1 & ")-( " & PuntoX2 & "," & PuntoY2  
& ")"
```

```
    'establece el ancho de la linea a dibujar
```

```
    picGrafica.DrawWidth = 2
```

```
    If PuntoY1 <= 82 And PuntoY1 >= 60 Then 'si la temperatura esta en el rango (18-40)
```

```
      picGrafica.ForeColor = &HC00000 'azul marino para graficar
```

```
    'de lo contrario si la temperatura es mayor o menor del rango
```

```
    Else: picGrafica.ForeColor = &HFF& 'rojo para graficar
```

```
    End If
```

```
    picGrafica.Line (PuntoX1, PuntoY1)-(PuntoX2, PuntoY2)
```

```
    PuntoX1 = PuntoX2
```

```
    PuntoY1 = PuntoY2
```

```
    'Si puntoX1 es mayor que el espacio total del PictureBox
```

```
    'en forma horizontal (ScaleWidth) entonces:
```

```
    If PuntoX1 > picGrafica.ScaleWidth Then
```

```
      Me.Cls 'Limpiar el picGrafica
```

```
      PuntoX1 = 0 'Empezar en 0 otra vez
```

```
      picGrafica.Cls
```

```
      Cuadros_Chicos
```

```
    End If
```

```
End Sub
```

```
Private Sub VSDistancia_Change()
```

```
  'Al cambiar el valor del scroll:
```

```
    'con el fin de incrementar o decrementar el valor de
```

```
    'distancia segun el valor actual del scroll
```

```
    Distancia = VSDistancia.Max + 1 - VSDistancia.Value
```

```
    lblDistancia.Caption = Distancia
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
  lblreloj.Caption = Time
```

```
End Sub
```


frmTermino2 (Código)

```
Option Explicit
Private Sub cmdRegresar_Click()
    Load frmDiagnosticos
    frmDiagnosticos.Show
    Unload frmTermino2
End Sub
Private Sub Form_Load()
    DataPort = &H378
End Sub
Private Sub submnuAcerca_Click()
    Load frmAcerca2
    frmAcerca2.Show 1
End Sub
Private Sub submnuAyuda_Click()
    Load frmHelp
    frmHelp.Show 1
End Sub
Private Sub submnuRegresar_Click()
    cmdRegresar_Click
End Sub
Private Sub Timer1_Timer()
    lblHora.Caption = Time
    lblFecha.Caption = Date
End Sub
Private Sub Timer2_Timer()
    TempAmbiental = LeerDato(DataPort) / 2
    lblTempAmb.Caption = TempAmbiental & " °C"
End Sub
```

frmWebCreada (Código)

```
Option Explicit
Private Sub cmdAceptar_Click()
    Unload frmWebCreada
End Sub
```

frmInicial (Código)

```
Private Sub Timer1_Timer()
    Unload Inicial
    Load frmMenu
    frmMenu.Show
End Sub
```

Modul1 (Código)

```
'Declara funcion para Redireccionar el puerto paralelo*
'Variable para cargar canal apropiado
'DEL CANAL 0 AL 7
Global Dato As Integer
Global Activar As Boolean
Global Temperatura As Integer
Global TempAmbiental As Integer
Global DataPort As Integer
Global ControlPort As Integer
Global Grafica As Boolean
Global Reproduccion As Boolean
Global Canal As Integer
'Declaremos una variable para el URL actual
'Public Explorer As SHDocVw.InternetExplorer
Public Sub Cuadros_Chicos()
frmtermo.picGrafica.DrawWidth = 1 'ancho de lineas
  X1 = 0
  X2 = 745
  Y1 = 0
  Y2 = 105
  frmtermo.picGrafica.ForeColor = &H808080 'gris fuerte
  'lineas horizontales gruesas
  For i = 10 To 100 Step 10
    frmtermo.picGrafica.Line (X1, i)-(X2, i)
  Next i
  'lineas verticales gruesas
  For i = 10 To 580 Step 10
    frmtermo.picGrafica.Line (i, Y1)-(i, Y2)
  Next i
  frmtermo.picGrafica.ForeColor = &H8080FF 'color rojo suave linea divisoria
  'entre temperatura normal y temperatura excedida ó baja
  frmtermo.picGrafica.DrawWidth = 2 'Devuelve o establece el ancho de línea para el
  resultado de los métodos gráficos.
  frmtermo.picGrafica.Line (0, 60)-(580, 60) 'excedida
  frmtermo.picGrafica.Line (0, 82)-(580, 82) 'baja
End Sub
```

Anexo 2

ANEXO 2. CODIFICACION DE LA APLICACION SERVIDOR.EXE

frmAcerca (Código)

Option Explicit

Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hdc As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long

Private Declare Function GetTickCount Lib "kernel32" () As Long

Const DT_BOTTOM As Long = &H8

Const DT_CALCRECT As Long = &H400

Const DT_CENTER As Long = &H1

Const DT_EXPANDTABS As Long = &H40

Const DT_EXTERNALLEADING As Long = &H200

Const DT_LEFT As Long = &H0

Const DT_NOCLIP As Long = &H100

Const DT_NOPREFIX As Long = &H800

Const DT_RIGHT As Long = &H2

Const DT_SINGLELINE As Long = &H20

Const DT_TABSTOP As Long = &H80

Const DT_TOP As Long = &H0

Const DT_VCENTER As Long = &H4

Const DT_WORDBREAK As Long = &H10

Private Icono As Integer

Private Type RECT

Left As Long

Top As Long

Right As Long

Bottom As Long

End Type

'El texto actual a correr. Tambien se puede leer desde un archivo de texto

```
Const ScrollText As String = "Tema de Tesis" & vbCrLf & _
    vbCrLf & _
    "Asesor:" & vbCrLf & _
    "Ing. Ocampo Alvarez Arturo" & vbCrLf & _
    vbCrLf & "Programacion:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Base de Datos:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Reportes tablas de Access:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    vbCrLf & "Diseño de página Web:" & vbCrLf & _
    "José Luis Juárez Trejo" & vbCrLf & _
    "Correo: luiscar_74@hotmail.com" & vbCrLf & _
    "Todos los Derechos Reservados" & vbCrLf & _
    "México MMIII" & vbCrLf & _
    vbCrLf & "Programa Exclusivamente para uso Académico..."
```

```
Dim EndingFlag As Boolean
Private Sub Form_Activate()
RunMain
End Sub
Private Sub Form_DblClick()
EndingFlag = True
End Sub
Private Sub Form_Load()
Icono = 1
picScroll.Font = "Arial" 'tipo de fuente
picScroll.ForeColor = &HFF00& ' color de fuente
picScroll.FontSize = 16.5 'tamaño de fuente
End Sub
Private Sub RunMain()
Dim LastFrameTime As Long
Const IntervalTime As Long = 40
Dim rt As Long
Dim DrawingRect As RECT
Dim UpperX As Long, UpperY As Long 'Punto izquierdo superior del PICSCROLL
Dim RectHeight As Long
'Muestra la forma
frmAcerca.Refresh
'Obtiene el tamaño del PICSCROLL y lo reemplaza por la constante DT_CALRECT
rt = DrawText(picScroll.hdc, ScrollText, -1, DrawingRect, DT_CALCRECT)
If rt = 0 Then 'Si marca error
MsgBox "Error scrolling text", vbCritical
EndingFlag = True
Else
DrawingRect.Top = picScroll.ScaleHeight
DrawingRect.Left = 0
DrawingRect.Right = picScroll.ScaleWidth
'Arregla la altura del PICSCROLL
RectHeight = DrawingRect.Bottom
DrawingRect.Bottom = DrawingRect.Bottom + picScroll.ScaleHeight
End If
Do While Not EndingFlag
If GetTickCount() - LastFrameTime > IntervalTime Then
picScroll.Cls
DrawText picScroll.hdc, ScrollText, -1, DrawingRect, DT_CENTER Or
DT_WORDBREAK
'Actualiza las coordenadas del rectangulo
DrawingRect.Top = DrawingRect.Top - 1
DrawingRect.Bottom = DrawingRect.Bottom - 1
'Controla el PICSCROLL y lo reinicia si se sale de su limite(si termina)
```

```

    If DrawingRect.Top < -(RectHeight) Then 'Tiempo de reinicio
        DrawingRect.Top = picScroll.ScaleHeight
        DrawingRect.Bottom = RectHeight + picScroll.ScaleHeight
    End If
    picScroll.Refresh
    LastFrameTime = GetTickCount()
End If
DoEvents
Loop
Unload Me
Set frmAcerca = Nothing
End Sub
Private Sub Form_Unload(Cancel As Integer)
    EndingFlag = True
End Sub
Private Sub picScroll_Click()
    EndingFlag = True
End Sub
Private Sub Timer1_Timer()
    Me.Icon = Me.ImgIcono.ListImages(Icono).Picture
    If Icono <> 24 Then
        Icono = Icono + 1
    Else
        Icono = 1
    End If
End Sub
frmAcerca2 (Código)
Option Explicit
Private Sub Form_Click()
    Unload Me
End Sub
Private Sub Label1_Click(Index As Integer)
    Unload Me
End Sub
frmAviso (Código)
Option Explicit
Dim h As Integer
Private Sub cmdAceptar_Click()
    'inhabilitar los botones para mover registros
    'cuando se vaya a Editar un Nuevo Registro
    For h = 0 To 3
        frmRegistros.cmdMover(h).Enabled = False
    Next h
    frmRegistros.cmdCancelar.Enabled = False
    frmRegistros.cmdNuevo.Enabled = False

```



```
frmRegistros.cmdRefrescar.Visible = True
frmRegistros.cmdRefrescar.Enabled = False
frmRegistros.cmdBorrar.Enabled = False
frmRegistros.cmdBuscar.Enabled = False 'Inhabilita el boton Busqueda
frmRegistros.cmdLoadPicture.Enabled = False
frmRegistros.cmdActualizar.Enabled = False
' Añadir un nuevo registro al recordset
rst.AddNew
frmRegistros.Timer1.Enabled = True
' Cargar la fotografía desde el inicio
frmRegistros.commonDialog1.ShowOpen
If frmRegistros.commonDialog1.FileName <> "" Then
    frmRegistros.Picture1.Picture = LoadPicture(frmRegistros.commonDialog1.FileName)
End If
' cmdActualizar.Enabled = True 'habilita el boton Actualizar
' Picture1.Picture = LoadPicture()
' Si no están ligados los controles al recordset
'#If TextLigados = 0 Then
' text1(0) = ""
' text1(1) = ""
' text1(2) = ""
' text1(3) = ""
' text1(4) = ""
' text1(5) = ""
' text1(6) = ""
' cmdActualizar_Click
'#End If
' For k = 0 To 5
' text1(k).Locked = False 'permite la escritura en campos
' Next k
frmRegistros.cmbTemp.Locked = False
frmRegistros.cmdActualizar.Enabled = True
frmRegistros.Show
frmRegistros.text1(0).SetFocus
Unload frmAviso
frmRegistros.cmbTemp.Text = "0 °C"
End Sub
Private Sub cmdAyuda_Click()
    Load frmHelp
    frmHelp.Show 1
End Sub
Private Sub cmdCancelar_Click()
    frmRegistros.Show
    Unload frmAviso
End Sub
```

```

Private Sub submnuAceptar_Click()
    cmdAceptar_Click
End Sub
Private Sub submnuCancelar_Click()
    cmdCancelar_Click
End Sub
frmBorrar (Código)
Private Sub cmdAceptar_Click()
    Call Refrescar
End Sub
Private Sub cmdCancelar_Click()
    'Exit Sub
    Unload frmBorrar
    frmRegistros.text1(0).SetFocus
End Sub
Public Sub Refrescar()
If optCampos.Value = True Then
    For i = 0 To 5
        frmRegistros.text1(i).Text = ""
    Next i
    frmRegistros.cmbTemp.Text = ""
    Unload frmBorrar
    frmRegistros.text1(0).SetFocus
Elseif optFoto.Value = True Then
    frmRegistros.Picture1.Picture = LoadPicture("")
    frmRegistros.commonDialog1.ShowOpen
    If frmRegistros.commonDialog1.FileName <> "" Then
        frmRegistros.Picture1.Picture =
LoadPicture(frmRegistros.commonDialog1.FileName)
    End If
    Unload frmBorrar
    frmRegistros.text1(0).SetFocus
Elseif optAmbos.Value = True Then
    For i = 0 To 5
        frmRegistros.text1(i).Text = ""
    Next i
    frmRegistros.cmbTemp.Text = ""
    frmRegistros.Picture1.Picture = LoadPicture("")
    frmRegistros.commonDialog1.ShowOpen
    If frmRegistros.commonDialog1.FileName <> "" Then
        frmRegistros.Picture1.Picture =
LoadPicture(frmRegistros.commonDialog1.FileName)
    End If
    Unload frmBorrar
    frmRegistros.text1(0).SetFocus

```

```
End If
End Sub
Private Sub Form_Load()
'si todos los campos estan vacios y hay foto entonces:
If frmRegistros.text1(0).Text = "" And frmRegistros.text1(1).Text = "" And
frmRegistros.text1(2).Text = ""
    And frmRegistros.text1(3).Text = "" And frmRegistros.text1(4).Text = "" And
frmRegistros.text1(5).Text = ""
    And frmRegistros.cmbTemp.Text = "" And frmRegistros.Picture1.Picture <>
LoadPicture("") Then
    'habilitar solo la opción foto
    optCampos.Enabled = False
    optFoto.Enabled = True
    optFoto.Value = True
    optAmbos.Enabled = False
    lblEdo.Caption = "Refrescar unicamente Foto"
'de lo contrario si algun campo esta lleno (incluyendo foto)
ElseIf frmRegistros.text1(0).Text <> "" Or frmRegistros.text1(1).Text <> "" Or
frmRegistros.text1(2).Text = ""
    Or frmRegistros.text1(3).Text <> "" Or frmRegistros.text1(4).Text <> "" Or
frmRegistros.text1(5).Text = ""
    Or frmRegistros.cmbTemp.Text <> "" Or frmRegistros.Picture1.Picture <>
LoadPicture("") Then
    'Habilitar las tres opciones y el value estará en Campos
    optCampos.Enabled = True
    optAmbos.Enabled = True
    optFoto.Enabled = True
    optCampos.Value = True
    lblEdo.Caption = "Refrescar unicamente Campos"
End If
End Sub
Private Sub optAmbos_Click()
    lblEdo.Caption = "Refrescar Todo"
End Sub
Private Sub optCampos_Click()
    lblEdo.Caption = "Refrescar unicamente Campos"
End Sub
Private Sub optFoto_Click()
    lblEdo.Caption = "Refrescar unicamente Foto"
End Sub
frmBuscar (Código)
Private Sub cmdAceptar_Click()
If OptFolio.Value = True Then
    submnuBusqueda.Caption = "Búsqueda por &Folio"
ElseIf OptNombre.Value = True Then
```

```

    submnuBusqueda.Caption = "Búsqueda por &Nombre"
End If
Dim Camp As String
If OptFolio.Value = True Then
    Camp = "Folio"
    frmBuscar.Hide
ElseIf OptNombre.Value = True Then
    Camp = "Nombre"
    frmBuscar.Hide
End If
If frmBuscar.OptFolio.Value = True Then
    'cadBusqueda = txtBuscar.Text
    campo = "Folio"
    frmRegistros.txtBuscar.Text = "F-"
ElseIf OptNombre.Value = True Then
    'cadBusqueda = txtBuscar.Text
    campo = "Nombre"
    frmRegistros.txtBuscar.Text = ""
End If
'frmRegistros.Show
frmRegistros.cmdBuscar.Caption = "Bu&scar Registro"
frmRegistros.lblInstruc.Caption = "Introduzca el " & Camp & " a buscar:"
frmRegistros.txtBuscar.Visible = True
frmRegistros.lblInstruc.Visible = True
' obtiene la longitud de la cadena y coloca el
' punto de inserción al final
frmRegistros.txtBuscar.SelStart = Len(frmRegistros.txtBuscar.Text)
' establece el enfoque sobre el cuadro de texto para que podamos ver
' el resultado
frmRegistros.txtBuscar.SetFocus
'frmBuscar.Hide
End Sub
Private Sub cmdCancelar_Click()
    Unload frmBuscar
End Sub
Private Sub Form_Load()
    Load frmRegistros
    OptFolio.Value = True
End Sub
Private Sub OptFolio_Click()
If OptNombre.Value = False Then 'si el optNombre no está seleccionado
    submnuBusqueda.Caption = "Búsqueda por &Folio"
ElseIf OptNombre.Value = True Then 'si el optNombre si está seleccionado
    submnuBusqueda.Caption = "Búsqueda por &Nombre"
End If

```

```
End Sub
'Private Sub txtBuscar_Change()
'If txtBuscar <> "" Then
' cmdBuscar.Enabled = True
' Else
' cmdBuscar.Enabled = False
'End If
'End Sub
Private Sub OptNombre_Click()
If OptFolio.Value = False Then 'si el optFolio no está seleccionado
  submnuBusqueda.Caption = "Búsqueda por &Nombre"
ElseIf OptFolio.Value = True Then 'si el optFolio si está seleccionado
  submnuBusqueda.Caption = "Búsqueda por &Folio"
End If
End Sub
Private Sub submnuAyuda_Click()
  Load frmHelp
  frmHelp.Show
End Sub
Private Sub submnuBusqueda_Click()
  cmdAceptar_Click
End Sub
Private Sub submnuCancelar_Click()
  cmdCancelar_Click
End Sub
frmClave (Código)
Private Sub cmdCancelar_Click()
  Unload Me
  Load frmMenu
  frmMenu.Show
End Sub
Private Sub cmdEntrar_Click()
'si la clave es "correct" entomces
If frmClave.text1.Text = "I" Then
  Load frmOpciones 'carga frmBase
  frmOpciones.Show 'muestra frmBase
  Unload frmClave 'descarga frmClave
Else
  MsgBox "Clave Incorrecta...!"
  Unload frmClave
End If
End Sub
Private Sub Text1_Change()
  If text1.Text <> "" Then
    cmdEntrar.Enabled = True
```

```

Else
    cmdEntrar.Enabled = False
End If
End Sub
frmHelp (Código)
Option Explicit
Private Sub cmdAceptar_Click()
    Unload frmHelp
End Sub
frmMenu (Código)
Private Sub cmdAcercade_Click()
'Presenta formulario a cerca de...
    Load frmAcerca
    frmAcerca.Show 1
End Sub
Private Sub cmdAyuda_Click()
    Load frmHelp
    frmHelp.Show 1
End Sub
Private Sub cmdBaseDatos_Click()
    Load frmClave 'carga frmClave
    frmClave.Show 'muestra frmClave
    Unload frmMenu 'descarga frmMenu
End Sub
Private Sub cmdSalir_Click()
    End
End Sub
Private Sub submnuAcerca_Click()
    cmdAcercade_Click
End Sub
Private Sub submnuAyudaPrincipal_Click()
    cmdAyuda_Click
End Sub
Private Sub submnuRegistro_Click()
    cmdBaseDatos_Click
End Sub
Private Sub submnuSalir_Click()
    cmdSalir_Click
End Sub
frmNuevoDictamen (Código)
Option Explicit
Private Sub cmdAceptar_Click()
'Si existe un Dictamen con el mismo nombre en la carpeta donde
'se mueven los dictamenes recién recibidos
'es decir, si ya antes había llegado otro dictamen (el cuál tendrá

```



```
'(el mismo nombre) y ya se ha movido a la nueva ubicación segun el nombre
'del paciente
If Archivo.FileExists("d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-
yyyy") & ".html") Then
    'Primeramente, borrar pagina web junto con la carpeta con Nombre de la Fecha Actual
    'de tal manera que solo existirá un dictamen por día
    'Borra pagina web
    Archivo.DeleteFile ("d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-
yyyy") & ".html")
    'Borra Carpeta
    Archivo.DeleteFolder ("d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-
yyyy")
    'Ahora si, mover el Dictamen Recibido
    'lo mueve a la ubicación ya que si se queda ahí
    '(a la carpeta dentro de telemedica: "Dictamenes Recibidos")
    'como el timer sigue habilitado
    'seguirá tocando el sonido wav y mandando el mensaje...
    Archivo.MoveFile "d:\telemedica\" & Format(Date, "dd-mm-yyyy") & ".html",
"d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-yyyy") & ".html"
    'mover folder que acompaña a la pagina web
    Archivo.MoveFolder "d:\telemedica\" & Format(Date, "dd-mm-yyyy"),
"d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-yyyy")
    'crear otro folder vacío para recibir otro dictamen
    Archivo.CreateFolder "d:\telemedica\" & Format(Date, "dd-mm-yyyy")
    'cerrar el control multimedia
    frmRegistros.MMControl1.Command = "close"
    .....
```

```
'Rehabilitar el timer DictaRecib:
frmRegistros.tmrDictaRecib.Enabled = True
.....
```

```
'si no ha llegado ningun archivo y éste es el primero en llegar...
ElseIf Archivo.FileExists("d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-
mm-yyyy") & ".html") = False Then
    'simplemente lo mueve a la ubicación (a la carpeta dentro de telemedica: "Dictamenes
Recibidos")
    'ya que si se queda ahí como el timer sigue habilitado
    'seguirá tocando el sonido wav y mandando el mensaje
    Archivo.MoveFile "d:\telemedica\" & Format(Date, "dd-mm-yyyy") & ".html",
"d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-yyyy") & ".html"
    'mover folder que acompaña a la pagina web
    Archivo.MoveFolder "d:\telemedica\" & Format(Date, "dd-mm-yyyy"),
"d:\telemedica\Dictamenes Recibidos\" & Format(Date, "dd-mm-yyyy")
    'cerrar el control multimedia
    frmRegistros.MMControl1.Command = "close"
    .....
```

```
'Rehabilitar el timer DictaRecib:
frmRegistros.tmrDictaRecib.Enabled = True
*****

End If
'Descargar formulario
Unload frmNuevoDictamen
'Habilita el boton Ver Nuevo Dictamen
frmRegistros.cmdVerArchivos.Enabled = True
End Sub
frmOpciones (Código)
Private Sub cmdAyuda_Click()
  Load frmHelp
  frmHelp.Show 1
End Sub
Private Sub cmdRegistro_Click()
  Load frmRegistros 'carga frmRegistros
  frmRegistros.Show '
  frmOpciones.Hide
End Sub
Private Sub cmdRegresar_Click()
  Load frmMenu
  frmMenu.Show
  frmOpciones.Hide
End Sub
Private Sub Form_Load()
  lblFecha.Caption = Date
End Sub
Private Sub submnuAcercade_Click()
  Load frmAcerca2
  frmAcerca2.Show 1
End Sub
Private Sub submnuAyuda_Click()
  cmdAyuda_Click
End Sub
Private Sub submnuRegistro_Click()
  cmdRegistro_Click
End Sub
Private Sub submnuRegresar_Click()
  cmdRegresar_Click
End Sub
Private Sub Timer1_Timer()
  lblHora.Caption = Time
End Sub
```

frmRegistros (Código)

```
Option Explicit
Dim Folder As New FileSystemObject
Dim Siguiete As Integer 'para renombrar dictamen recibido en caso de que
'lleguen mas de 1 dictamen
'para visualizar la pagina Web
'Declarar una variable para el URL actual
Public Explorer As SHDocVw.InternetExplorer
'-----

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim L As Integer
Dim NuevoFolio As Integer
Dim ElemCmbBox As Variant
Dim Responder As Integer
' Para probar usando controles ligados al recordset
' Si se ligan los Text1 al recordset, asignar -1 a esta constante
' Si no se ligan los Text1, asignar 0 (o simplemente comentar la línea)
#Const TextLigados = -1

Private sBase As String      ' Nombre de la base de datos
' Constantes para el movimiento dentro del recordset
Const cPrimero = 0
Const cAnterior = 1
Const cSiguiete = 2
Const cUltimo = 3
Private Sub cmdCancelar_Click()
'rst.MoveFirst 'si se aprieta cancelar mover al primero
cmdCancelar.Visible = False
cmdCesar.Visible = False
Frame1.BackColor = &H800000 'fondo azul (como estaba)
lblNombre.BackColor = &H800000 'fondo azul (como estaba)
lblFolio.BackColor = &H800000 'fondo azul (como estaba)
lblFecha.BackColor = &H800000 'fondo azul (como estaba)
lblEdad.BackColor = &H800000 'fondo azul (como estaba)
lblDireccion.BackColor = &H800000 'fondo azul (como estaba)
lblTel.BackColor = &H800000 'fondo azul (como estaba)
lblTemp.BackColor = &H800000 'fondo azul (como estaba)
For j = 0 To 5
    text1(j).Locked = True 'No permitir la escritura en campos
Next j
    cmbTemp.Locked = True
    cmdLoadPicture.Enabled = False
```

```

cmdRefrescar.Enabled = False
cmdRefrescar.Visible = False
cmdBorrar.Enabled = False
cmdActualizar.Enabled = False
cmdNuevo.Enabled = False
cmdEdit.Enabled = True
lblEstado.Caption = "Modo de sólo Lectura"
End Sub
Private Sub cmdCesar_Click()
Dim Ruta As String
Ruta = "d:\Archivos de programa\CesarFTP\CesarFTP.exe"
Shell Ruta, vbMaximizedFocus
End Sub
Private Sub cmdNuevo_Click()
Load frmAviso
frmRegistros.Hide
frmAviso.Show
End Sub
Private Sub cmdBuscar_Click()
If cmdBuscar.Caption = "Bu&sqveda" Then
Load frmBuscar
frmBuscar.Show 1
'txtBuscar.Visible = False
'lblInstruc.Visible = False
ElseIf cmdBuscar.Caption = "Bu&scar Registro" Then
Call Buscar(campo)
cmdBuscar.Caption = "Bu&sqveda"
End If
End Sub
Private Sub cmdBorrar_Click()
Responder = MsgBox("Con ésta acción se perderán todos los datos, ¿Desea continuar?",
vbOKCancel, "Confirmacion para borrar Registro")
If Responder = vbCancel Then
Exit Sub
ElseIf Responder = vbOK Then
' Eliminar el registro actual
rst.Delete
rst.MovePrevious
MsgBox "El Registro ha sido eliminado...", , "Borrar Registro"
On Local Error Resume Next
' Mover al siguiente registro (primero)
rst.Move 1
' Si no puede moverse al siguiente, moverlo al anterior
If Err Then
rst.MovePrevious

```

```
End If
End If
Err = 0
End Sub
Private Sub cmdLoadPicture_Click()
commonDialog1.ShowOpen
If commonDialog1.FileName <> "" Then
Picture1.Picture = LoadPicture(commonDialog1.FileName)
End If
End Sub
Private Sub cmdMover_Click(Index As Integer)
'Cuando las propiedades BOF y EOF son True,
'es que no hay ningun registro, por lo tanto
'salir del procedimiento
If rst.BOF = True And rst.EOF = True Then
Exit Sub
End If
'Moverse por el recordset
On Local Error Resume Next
Err = 0
Select Case Index
Case cPrimero ' Primero
rst.MoveFirst
Case cAnterior ' Anterior
rst.MovePrevious
Case cSiguiente ' Siguiente
rst.MoveNext
Case cUltimo ' Último
rst.MoveLast
End Select
If rst.BOF Then 'si estamos antes del primero,
rst.MoveFirst 'Mover al primero
ElseIf rst.EOF Then 'si estamos despues del último,
rst.MoveLast 'Mover al último
End If
If Err Then
Debug.Print Err.Description
End If
If (Not rst.EOF) And (Not rst.BOF) Then
'¿Como transfiero la imagen desde el
' campo FOTO del recordset rst
' al control Picture1?
'Picture1.Picture = rst!FOTO
'Picture1.Cls
'Picture1.Print "<Falta cargar la imagen>"
```

```

'Leer desde el campo FOTO (la foto)para ponerla en Picture1
LeerBinary rst!FOTO, Picture1
End If
Err = 0
End Sub
Private Sub cmdRefrescar_Click()
If text1(0).Text <> "" Or text1(1).Text <> "" Or text1(2).Text <> ""
Or text1(3).Text <> "" Or text1(4).Text <> "" Or text1(5).Text <> ""
Or cmbTemp.Text <> "" Or Picture1.Picture <> LoadPicture("") Then
Load frmBorrar
frmBorrar.Show 1
End If
'Esto es para el caso de que se pulse en Update,
'sin que se haya dado a AddNew
'If rst.EditMode = dbEditNone Then
'rst.EditMode
'End If
'rst!ID = text1(0)
'rst!NOMBRE = text1(1)
'¿Como transfiero la imagen desde el
'control Picture1 al campo FOTO del
'recordset rContactos?
'rst!FOTO = Picture1.Picture
'GuardarBinary rst!FOTO, Picture1
'rst.Update
End Sub
Private Sub cmdActualizar_Click()
For j = 0 To 3
cmdMover(j).Enabled = True
Next j
cmdCancelar.Enabled = True
cmdCancelar.Visible = False
Frame1.BackColor = &H800000 'fondo azul (como estaba)
lblNombre.BackColor = &H800000 'fondo azul (como estaba)
lblFolio.BackColor = &H800000 'fondo azul (como estaba)
lblFecha.BackColor = &H800000 'fondo azul (como estaba)
lblEdad.BackColor = &H800000 'fondo azul (como estaba)
lblDireccion.BackColor = &H800000 'fondo azul (como estaba)
lblTel.BackColor = &H800000 'fondo azul (como estaba)
lblTemp.BackColor = &H800000 'fondo azul (como estaba)
'cmdEdit.Caption = "&Editar"
If text1(0).Text = "" Or text1(1).Text = "" Or text1(2).Text = ""
Or text1(3).Text = "" Or text1(4).Text = "" Or text1(5).Text = ""
Or cmbTemp.Text = "" Or Picture1.Picture = LoadPicture("") Then
MsgBox "Es necesario llenar todos los campos..."

```

```
Frame1.BackColor = &H80&      'fondo rojo
lblNombre.BackColor = &H80&    'fondo rojo
lblFolio.BackColor = &H80&     'fondo rojo
lblFecha.BackColor = &H80&     'fondo rojo
lblEdad.BackColor = &H80&      'fondo rojo
lblDireccion.BackColor = &H80&  'fondo rojo
lblTel.BackColor = &H80&       'fondo rojo
lblTemp.BackColor = &H80&      'fondo rojo
text1(0).SetFocus
' For i = 1 To 6
'   text1(i).Text = ""
' Next i
' Exit Sub
'End If
' Esto no es necesario si los Text1 están ligados al recordset,
' cuando se cambie de registro, se actualizarán los datos
' De todas formas, se puede dejar para actualizar los datos cuando
' se de de alta un nuevo registro.
' Cuando no se ligen los controles, después de introducir los datos,
' es necesario pulsar en este botón
Else
'para guardar la fotografia
GuardarBinary rst.Fields(Picture1.DataField), Picture1
rst.Update
rst!Nombre = text1(0)
rst!Folio = text1(1)
rst!Fecha = text1(2)
rst!Edad = text1(3)
rst!Direccion = text1(4)
rst!Telefono = text1(5)
rst!Temperatura = cmbTemp
'rst.Update
For j = 0 To 5
    text1(j).Locked = True 'No permitir la escritura en campos
Next j
    cmbTemp.Locked = True
    cmdLoadPicture.Enabled = False
    cmdRefrescar.Enabled = False
    cmdRefrescar.Visible = False
    cmdCesar.Visible = False
    cmdBorrar.Enabled = False
    cmdActualizar.Enabled = False
    cmdNuevo.Enabled = False
    cmdEdit.Enabled = True
MsgBox "Registro Actualizado...", , "Actualizar Registro"
```



```

    lblEstado.Caption = "Modo de sólo Lectura"
    End If
End Sub
Private Sub Buscar(Field)
'Procedimiento para buscar el dato indicado
Dim sADOBuscar As String
'Dim Cadena As String
'Dim Field As String
'iniciamos la deteccion de errores
On Error Resume Next
'Busca la primera coincidencia en el Recordset
'los asteriscos son para encontrar cualquier similitud de la
'cadena introducida
sADOBuscar = Field & " Like *" & txtBuscar.Text & "*"
rst.MoveFirst
rst.Find sADOBuscar
If rst.BOF Or rst.EOF Then
    Err.Clear
    MsgBox "No existe el Folio buscado o ya no hay mas datos que mostrar..." , , "Buscar
Registro"
    lblInstruc.Visible = False
    txtBuscar.Visible = False
    'Posicionar el Recordset en el primer registro
    rst.MoveFirst
Else
    MsgBox "Registro encontrado..."
    lblInstruc.Visible = False
    txtBuscar.Visible = False
End If
End Sub
Private Sub cmdEdit_Click()
cmdCesar.Visible = True 'mostrar el boton Establecer Acceso

cmdCancelar.Visible = True 'hacer visible el boton cancelar
lblEstado.Caption = "Modo de Edición"
Frame1.BackColor = &H80&           'fondo rojo
lblNombre.BackColor = &H80&        'fondo rojo
lblFolio.BackColor = &H80&         'fondo rojo
lblFecha.BackColor = &H80&         'fondo rojo
lblEdad.BackColor = &H80&          'fondo rojo
lblDireccion.BackColor = &H80&     'fondo rojo
lblTel.BackColor = &H80&           'fondo rojo
lblTemp.BackColor = &H80&          'fondo rojo
cmdEdit.Enabled = False 'Inhabilitar el boton Editar

```

```
For j = 0 To 5
    text1(j).Locked = False 'Permitir escritura en campos
Next j
    cmdActualizar.Enabled = True
    cmbTemp.Locked = False
    cmdLoadPicture.Enabled = True
    'cmdRefrescar.Enabled = True
    cmdBorrar.Enabled = True
    cmdNuevo.Enabled = True
    cmdRefrescar.Visible = True
    cmdRefrescar.Enabled = True
End Sub
Private Sub cmdRegresar_Click()
    Load frmOpciones
    frmOpciones.Show
    Unload frmRegistros
End Sub
Private Sub cmdVerArchivos_Click()
    If cmdVerArchivos.Caption = "&Ver Nuevo Dictamen" Then
        cmdVerArchivos.Caption = "Ban&deja de Entrada"
        cmdVerArchivos.Picture = LoadPicture("D:\JOSE
LUIS\SERVIDOR\BandejaEntrada.ico")
        'se abre el cuadro de diálogo DialogVerArchivos
        DialogVerArchivos.FileName = ("d:\telemedica\Dictamenes Recibidos\" &
Format(Date, "dd-mm-yyyy") & ".html")
        'Devuelve o establece el nombre para poder abrir el archivo.
        DialogVerArchivos.Filter = "Páginas Web (*.html)|*.html"
        DialogVerArchivos.DialogTitle = "Ver Archivo Recibido"
        DialogVerArchivos.ShowOpen
        If DialogVerArchivos.FileName <> "" Then
            On Error GoTo manejadorerror 'si hay error ir al manejador de error
            Set Explorer = New SHDocVw.InternetExplorer 'crea el objeto
            'para hacer uso del explorador:
            Explorer.Visible = True
            Explorer.Navigate DialogVerArchivos.FileName ' se declara la dirección en
donde esta
            'la pagina web
            'cmdVerArchivos.Enabled = False
            Exit Sub
        End If
    manejadorerror:
        MsgBox "Puede ver el archivo con el boton Ver Paginas", , "Ver Dictamen Recibido"
    'Err.Description
    ElseIf cmdVerArchivos.Caption = "Ban&deja de Entrada" Then
        'se abre el cuadro de dialogo DialogVerArchivos
```

```

DialogVerArchivos.InitDir = ("d:\telemedica\Dictamenes Recibidos\")
'Devuelve o establece el directorio inicial de los archivos.
DialogVerArchivos.Filter = "Páginas Web (*.html)*.html"
DialogVerArchivos.DialogTitle = "Archivos de la Bandeja de Entrada"
DialogVerArchivos.ShowOpen
If DialogVerArchivos.FileName <> "" Then
    On Error GoTo manejadorerror2 'si hay error ir al manejador de error
    Set Explorer = New SHDocVw.InternetExplorer 'crea el objeto
    'para hacer uso del explorador:
    Explorer.Visible = True
    Explorer.Navigate DialogVerArchivos.FileName ' se declara la dirección en
donde esta
    'la pagina web
    'cmdVerArchivos.Enabled = False
    Exit Sub
End If
manejadorerror2:
    MsgBox "Puede ver el archivo con el boton Bandeja de Entrada", , "Ver Dictamen"
'Err.Description
End If
End Sub
Private Sub Form_Load()
    'crear el folder que se utilizará con el nombre de la fecha actual
    'para recibir los dictamenes
    'primero crear el objeto para el manejo de archivos, carpetas, etc.
    Set Folder = CreateObject("Scripting.FileSystemObject")
    'si existe el folder, es decir, si ya se había creado antes
    If Folder.FolderExists("d:\telemedica\" & Format(Date, "dd-mm-yyyy")) Then
        'Borrarlo y ...
        Folder.DeleteFolder ("d:\telemedica\" & Format(Date, "dd-mm-yyyy"))
        'Volverlo a crear
        Folder.CreateFolder "d:\telemedica\" & Format(Date, "dd-mm-yyyy")
    Else 'Es decir, si no existe el folder (no se ha creado)
        'crearlo
        Folder.CreateFolder "d:\telemedica\" & Format(Date, "dd-mm-yyyy")
    End If

'para reproducir el sonido de alerta cuando se reciba un Nuevo Dictamen
MMControl1.DeviceType = "WaveAudio"
'Especifica el tipo de dispositivo MCI que se va a abrir
MMControl1.Notify = False
'No queremos que nos avise lo que va ejecutando el control
MMControl1.Wait = True
'Determina si el control Multimedia MCI espera a completar el

```

```
'siguiente comando MCI antes de devolver el control a la aplicación.
MMControl1.Shareable = False
'Ningún otro control o aplicación puede tener acceso a este dispositivo.
MMControl1.FileName = "D:\JOSE LUIS\SERVIDOR\Nuevo.WAV"
'Ruta del archivo a reproducir
tmrDictaRecib.Enabled = True
'crear el objeto para poder trabajar con la variable
'la cual sirve para copiar archivos, mover, eliminar, etc.
Set Archivo = CreateObject("Scripting.FileSystemObject")
'*****
For j = 15 To 50
    ElemCmbBox = j & " °C"
    cmbTemp.AddItem ElemCmbBox
Next j
' Asignar el nombre de la base de datos
' (si la aplicación se ejecuta en el directorio raíz, quitar el \)
sBase = App.Path & "\termometro.mdb"
'crear objetos
Set cnn = New ADODB.Connection
Set rst = New ADODB.Recordset
'Proveedor de la Base de datos para Access 2000/2002
cnn.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & sBase
rst.Open "SELECT * FROM Pacientes", cnn, adOpenDynamic, adLockOptimistic
' Crear los objetos
Set Picture1.DataSource = rst
Picture1.DataField = "FOTO"
'Se tendrá que agregar la referencia:
'Microsoft Data Formatting Object Library
'El objeto StdDataFormat permite dar formato a los datos cuando se leen
'y se escriben en la base de datos.
'DataFormat devuelve este objeto
'fmtPicture.- El valor se trata como una imagen.
'Un objeto binario que se lee desde la base de datos se convierte en un
'objeto Picture y se vuelve a escribir en la base de datos como un
'objeto binario. Equivale a el Numero 2
Picture1.DataFormat.Type = fmtPicture ' 2 (fmtPicture)
'
'Indica la ubicación del servicio de cursor
'Elemento de la base de datos que controla el desplazamiento
'a través de registros, si los datos se pueden actualizar
'y la visibilidad de los cambios efectuados en la base de
'datos por otros usuarios
'rst.CursorLocation = adUseClient
'Si se quiere usar los Text1 ligados al Recordset
#If TextLigados Then
```

```

' Conectar manualmente los Text1 al recordset
Dim i As Long
' Asignar el recordset al que estarán los datos conectados
' Esto funciona igual que si se conectaran a un DataControl,
' Por tanto no hay que preocuparse de actualizar el contenido, etc.
For i = 0 To 5
    Set text1(i).DataSource = rst
Next
'For i = 0 To 5
'    text1(i).DataFormat.Type = fmtGeneral 'Formato para campos
'Next
'asigna el recordset al combobox (que pertenece a la temperatura)
Set cmbTemp.DataSource = rst
' Asignar los nombres de los campos
text1(0).DataField = "Nombre"
text1(1).DataField = "Folio"
text1(2).DataField = "Fecha"
text1(3).DataField = "Edad"
text1(4).DataField = "Direccion"
text1(5).DataField = "Telefono"
cmbTemp.DataField = "Temperatura"
Picture1.DataField = "FOTO"
' Deshabilitar el botón de Actualizar
cmdActualizar.Enabled = False
rst.MoveFirst
#End If
'If (Not rst.EOF) And (Not rst.BOF) Then
'    ¿Como transfiero la imagen desde el
'    campo FOTO del recordset rst
'    al control Picture1?
'Picture1.Picture = rst!FOTO
'Picture1.Cls
'Picture1.Print "<Falta cargar la imagen>"
'Leer desde el campo FOTO (la foto)para ponerla en Picture1
'    LeerBinary rst!FOTO, Picture1
' End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
MMControl1.Command = "close" 'cerrar control para liberar recursos
On Local Error Resume Next
rst.Close
cnn.Close
Set rst = Nothing
Set cnn = Nothing

```

```
Err = 0
Set frmRegistros = Nothing
End Sub
Private Sub rst_MoveComplete(ByVal adReason As ADODB.EventReasonEnum, _
    ByVal pError As ADODB.Error, _
    adStatus As ADODB.EventStatusEnum, _
    ByVal pRecordset As ADODB.Recordset)
' Mostrar el ID del registro actual
' si se pasa del primero o del último, dará error
On Local Error Resume Next
' Mostrar el ID del registro actual usando el recordset pasado como parámetro
'lblEdo.Caption = "Registro Actual: " & pRecordset!Registro
'Registro es el nombre del campo (el cual debe ser autonumerico)
' También se puede usar:
'lblEdo.Caption = "Registro Actual: " & pRecordset.RecordCount
'rst.CursorLocation = adUseNone
' Si no están ligados los Text1
#If TextLigados = 0 Then
' Si no produce error actualizar los Text1
' (no es necesario si los Text1 están ligados al Recordset)
If Err = 0 Then
    text1(0) = rst!Nombre
    text1(1) = rst!Folio
    text1(2) = rst!Fecha
    text1(3) = rst!Edad
    text1(4) = rst!Direccion
    text1(5) = rst!Telefono
    text1(6) = rst!Temperatura
End If
#End If
Err = 0
End Sub
Private Sub Timer1_Timer()
If text1(0).Text <> "" Or text1(1).Text <> "" Or text1(2).Text <> "" Or _
text1(3).Text <> "" Or text1(4).Text <> "" Or text1(5).Text <> "" Or _
cmbTemp.Text <> "" Or Picture1.Picture <> LoadPicture("") Then
cmdRefrescar.Enabled = True
Else
cmdRefrescar.Enabled = False
End If
End Sub
Private Sub tmrDictaRecib_Timer()
Dim Num As Integer
Dim Resp As Integer
'chechar si ha llegado un nuevo Dictamen
```

```

If Archivo.FileExists("d:\telemedica\" & Format(Date, "dd-mm-yyyy") & ".html") Then
    .....
    'detener momentaneamente el timer: DictaRecib
    tmrDictaRecib.Enabled = False
    .....

    'Reproducir el sonido de Nuevo Dictamen Recibido
    MMControl1.Command = "Open" 'Preparar control para Reproducción
    'Abre un dispositivo mediante el comando MCI_OPEN.
    MMControl1.Command = "play" 'Reproducir sonido de alarma
    Load frmNuevoDictamen
    frmNuevoDictamen.Show 1 '(Formulario Modal)
    cmdVerArchivos.Caption = "&Ver Nuevo Dictamen"
    cmdVerArchivos.Picture = LoadPicture("D:\JOSE
LUIS\SERVIDOR\VER_DICTAMEN.ico")
End If
End Sub

Inicial (Código)
Private Sub Timer1_Timer()
Unload Inicial
Load frmMenu
frmMenu.Show
End Sub

DBBinary (Código)
' Código para grabar y leer imagenes en campos de bases
' Adaptado para usarlo con ADO
' ©Jose Luis 2003
' <luisca_74@hotmail.com>
'-----
Option Explicit
Private nFile As Long
Private Chunk() As Byte
Private Const mBuffer As Long = 16384&
Public Sub LeerBinary(ADOFIELD As ADODB.FIELD, unPicture As PictureBox)
' Leer la imagen del campo de la base y asignarlo al Picture
'-----
' Este procedimiento no es necesario usarlo
' si el Picture está ligado a un data control
'-----

Dim nChunks As Long
Dim nSize As Long
Dim Fragment As Long
Dim i As Long
' Se usa un fichero temporal para guardar la imagen
nFile = FreeFile
Open "pictemp" For Binary Access Write As nFile

```



```
' Calcular los trozos completos y el resto
nSize = ADOField.ActualSize
nChunks = Int(nSize / mBuffer)
Fragment = nSize Mod mBuffer
Chunk() = ADOField.GetChunk(Fragment)
Put nFile, , Chunk()
For i = 1 To nChunks
    Chunk() = ADOField.GetChunk(mBuffer)
    Put nFile, , Chunk()
Next
Close nFile
Erase Chunk
' Ahora se carga esa imagen en el control
unPicture.Picture = LoadPicture("pictemp")
' Ya no necesitamos el fichero, así que borrarlo
On Error Resume Next
If Len(Dir$("pictemp")) Then
    Kill "pictemp"
End If
Err = 0
End Sub
Public Sub GuardarBinary(ADOField As ADODB.Field, unPicture As PictureBox)
' Guardar el contenido del Picture en el campo de la base
Dim i As Long
Dim Fragment As Long
Dim nSize As Long
Dim nChunks As Long
' Guardar el contenido del picture en un fichero temporal
SavePicture unPicture.Picture, "pictemp"
' Leer el fichero y guardarlo en el campo
nFile = FreeFile
Open "pictemp" For Binary Access Read As nFile
nSize = LOF(nFile) ' Longitud de los datos en el archivo
If nSize = 0 Then
    Close nFile
    Exit Sub
End If
' Calcular el número de trozos y el resto
nChunks = nSize \ mBuffer
Fragment = nSize Mod mBuffer
ReDim Chunk(Fragment)
Get nFile, , Chunk()
ADOField.AppendChunk Chunk()
ReDim Chunk(mBuffer)
For i = 1 To nChunks
```

```
    Get nFile, , Chunk()
    ADOField.AppendChunk Chunk()
Next i
Close nFile
' Ya no necesitamos el fichero, así que borrarlo
On Local Error Resume Next
If Len(Dir$("pictemp")) Then
    Kill "pictemp"
End If
Err = 0
End Sub
Module1 (Código)
Public campo As String
' Objetos para acceder directamente a la base usando código
Public cnn As ADODB.Connection
' Si se declara con WithEvents, tenemos acceso a los mismos eventos que
' con el ADO DataControl
Public rst As ADODB.Recordset
'Esta variable se usa para trabajar con archivos
'para moverlos, copiar, borrar, etc.
Global Archivo As FileSystemObject 'chechar referencias (Microsoft Scripting Runtime)
.....
```

Anexo 3



October 2002

ADC0808/ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

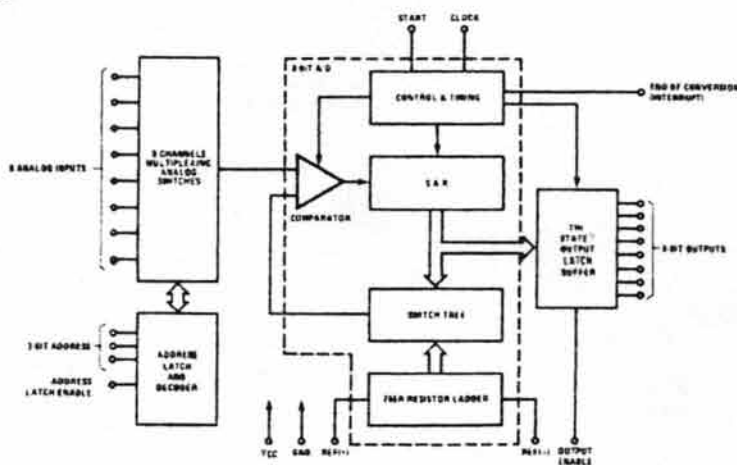
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- | | |
|--------------------------|-------------------------------|
| ■ Resolution | 8 Bits |
| ■ Total Unadjusted Error | $\pm 1/2$ LSB and ± 1 LSB |
| ■ Single Supply | 5 V _{DC} |
| ■ Low Power | 15 mW |
| ■ Conversion Time | 100 μ s |

Block Diagram



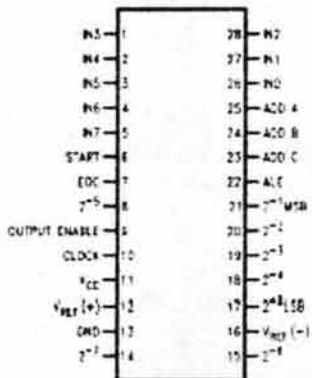
See Ordering Information

ADC0808/ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

ADC0808/A DC 0809

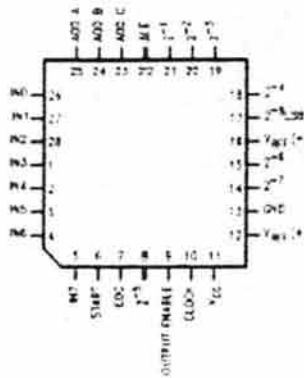
Connection Diagrams

Dual-In-Line Package



Order Number ADC0808CCN or ADC0809CCN
See NS Package J28A or N28A

Molded Chip Carrier Package



Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Ordering Information

TEMPERATURE RANGE		-40 C to +85 C	
Error	±1/2 LSB Unadjusted	ADC0808CCN	ADC0808CCV
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier

Absolute Maximum Ratings (Notes 2,

1)
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Pin	-0.3V to ($V_{CC}+0.3V$)

Except Control Inputs	
Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	-0.3V to +15V
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A=25^\circ\text{C}$	875 mW
Lead Temp. (Soldering, 10 seconds)	

Dual-In-Line Package (plastic)	260°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC0808CCN, ADC0809CCN	-40°C $\leq T_A \leq$ +85°C
ADC0808CCV, ADC0809CCV	-40°C $\leq T_A \leq$ +85°C
Range of V_{CC} (Note 1)	4.5 V_{CC} to 6.0 V_{CC}

Electrical Characteristics

Converter Specifications: $V_{CC}=5V_{CC}-V_{REF(+)}$, $V_{REF(-)}=GND$, $T_{MIN} \leq T_A \leq T_{MAX}$ and $f_{CLK}=640\text{ kHz}$ unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error (Note 5)	25°C T_{MIN} to T_{MAX}			$\pm 1\%$ $\pm 4\%$	LSB LSB
	ADC0809					
	Total Unadjusted Error (Note 5)	0°C to 70°C T_{MIN} to T_{MAX}			$\pm 1\%$ $\pm 1\%$	LSB LSB
	Input Resistance	From Rel(+) to Rel(-)	1.0	2.5		k Ω
	Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND-0.10		$V_{CC}+0.10$	V_{CC}
$V_{REF(+)}$	Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	$V_{CC}+0.1$	V
$\frac{V_{REF(+)} + V_{REF(-)}}{2}$	Voltage, Center of Ladder		$V_{CC}/2-0.1$	$V_{CC}/2$	$V_{CC}/2+0.1$	V
$V_{REF(-)}$	Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
I_N	Comparator Input Current	$f_{CLK}=640\text{ kHz}$, (Note 6)	-2	± 0.5	2	μA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \leq V_{CC} \leq 5.25V$, -40°C $\leq T_A \leq$ +85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
ANALOG MULTIPLEXER						
$I_{OFF(+)}$	OFF Channel Leakage Current	$V_{CC}=5V$, $V_{IN}=5V$, $T_A=25^\circ\text{C}$ T_{MIN} to T_{MAX}		10	200 1.0	nA μA
$I_{OFF(-)}$	OFF Channel Leakage Current	$V_{CC}=5V$, $V_{IN}=0$, $T_A=25^\circ\text{C}$ T_{MIN} to T_{MAX}	-200 -1.0	-10		nA μA
CONTROL INPUTS						
$V_{IN(1)}$	Logical '1' Input Voltage		$V_{CC}-1.5$			V
$V_{IN(0)}$	Logical '0' Input Voltage				1.5	V
$I_{IN(1)}$	Logical '1' Input Current (The Control Inputs)	$V_{IN}=15V$			1.0	μA
$I_{IN(0)}$	Logical '0' Input Current (The Control Inputs)	$V_{IN}=0$	-1.0			μA
I_{CC}	Supply Current	$f_{CLK}=640\text{ kHz}$		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \leq V_{CC} \leq 5.25V$, $-40 \leq T_A \leq 85$ C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical '1' Output Voltage	$V_{CC} = 4.75V$ $I_{OUT} = -360\mu A$ $I_{OUT} = -10\mu A$		2.4 4.5		V(min) V(min)
$V_{OUT(0)}$	Logical '0' Output Voltage	$I_O = 1.6$ mA			0.45	V
$V_{OUT(EOC)}$	Logical '0' Output Voltage EOC	$I_O = 1.2$ mA			0.45	V
I_{OUT}	TRI-STATE Output Current	$V_O = 5V$ $V_O = 0$	-3		3	μA μA

Electrical Characteristics

Timing Specifications $V_{CC} = V_{REF(+)}, = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20$ ns and $T_A = 25$ C unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{SPS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t_s	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t_h	Minimum Address Hold Time	(Figure 5)		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_S = 0\Omega$ (Figure 5)		1	2.5	μs
t_{OH}, t_{OL}	OE Control to Q Logic State	$C_L = 50$ pF, $R_L = 10k$ (Figure 6)		125	250	ns
t_{OH}, t_{OH}	OE Control to Hi-Z	$C_L = 10$ pF, $R_L = 10k$ (Figure 6)		125	250	ns
t_c	Conversion Time	$f_c = 640$ kHz, (Figure 5) (Note 7)	90	100	116	μs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(Figure 5)	0		8+2	μs Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists internally from V_{CC} to GND and has a typical breakdown voltage of $7 V_{CC}$.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The specification is 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute V_{IN} to 5V (0 to 5V) input voltage range will therefore require a minimum supply voltage of $4.900 V_{CC}$ over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full scale, linearity and multiplexer errors. See Figure 3. None of these A/Ds requires a zero or full scale adjust. However, if an all zero code is desired for an analog input other than 0V, or if a narrow full scale span exists (for example, 0.5V to 4.5V full scale) the reference voltages can be adjusted to achieve this. See Figure 1.3.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has 1% temperature dependence (Figure 6). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table 1 shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (Figure 1) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in Figure 1 are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached $\pm 1/2$ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n -iterations are required for an n -bit converter. Figure 2 shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion start pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

ADC0808/A DC 0809

Functional Description (Continued)

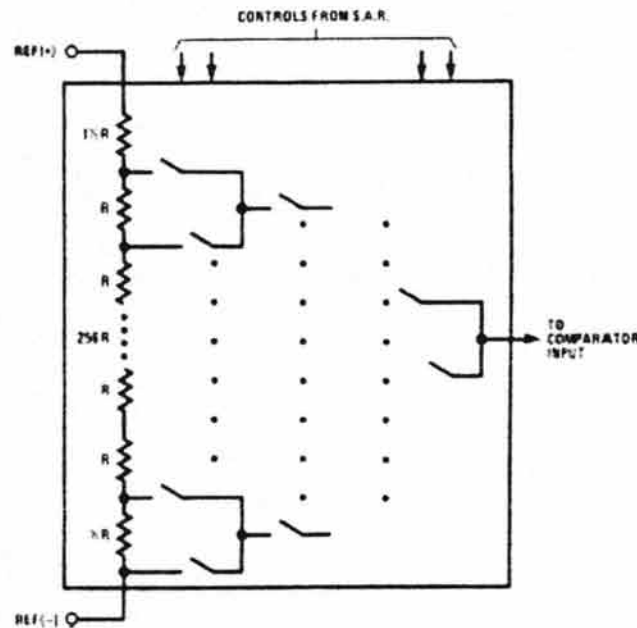


FIGURE 1. Resistor Ladder and Switch Tree

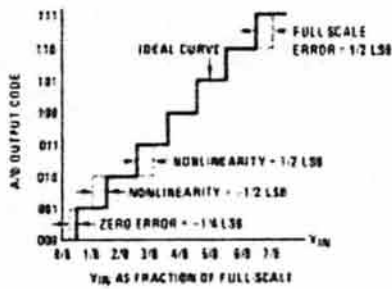


FIGURE 2. 3-Bit A/D Transfer Curve

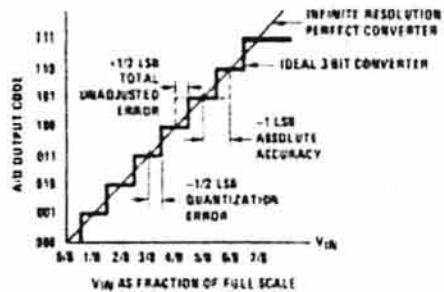


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

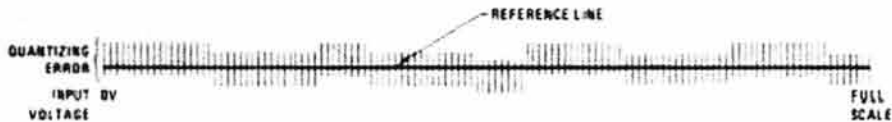


FIGURE 4. Typical Error Curve

Timing Diagram

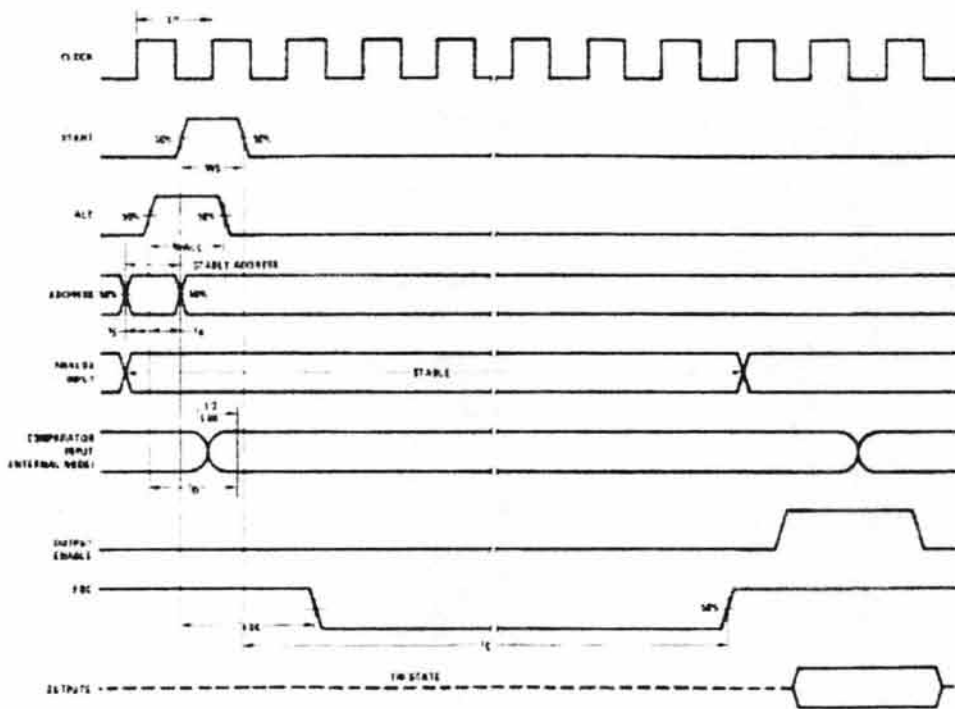
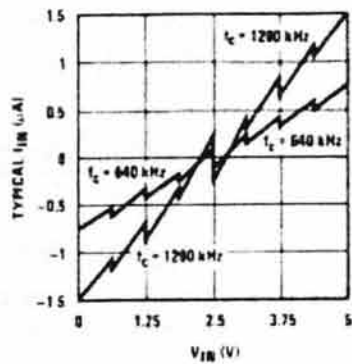


FIGURE 5.

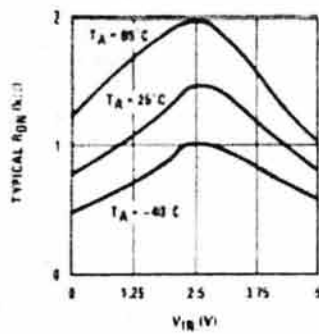
ADC0808/A DC0809

Typical Performance Characteristics



19A2716

FIGURE 6. Comparator I_{IN} vs V_{IN}
($V_{CC} = V_{REF} = 5V$)



19A2717

FIGURE 7. Multiplexer R_{ON} vs V_{IN}
($V_{CC} = V_{REF} = 5V$)

TRI-STATE Test Circuits and Timing Diagrams

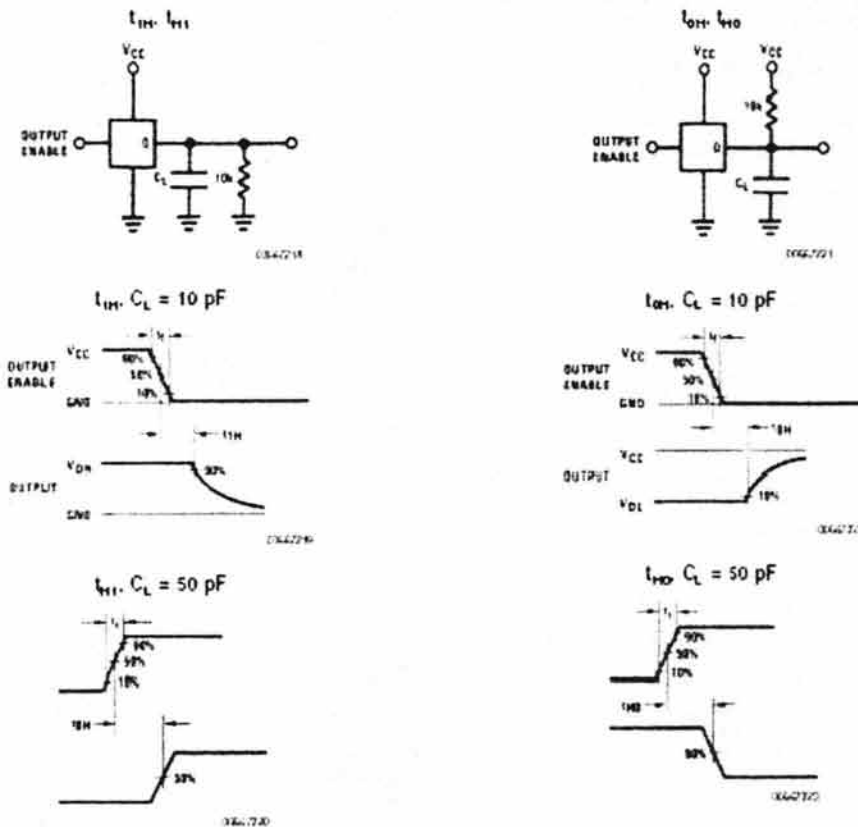


FIGURE 8.

Applications Information

OPERATION

1.0 RATIO-METRIC CONVERSION

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratio-metric conversion systems. In ratio-metric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN} - V_Z}{V_{IN} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}} \quad (1)$$

- V_{IN} = Input voltage into the ADC0808
- V_{FS} = Full-scale voltage
- V_Z = Zero voltage
- D_X = Data point being measured
- D_{MAX} = Maximum data limit

D_{MIN} = Minimum data limit

A good example of a ratio-metric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, (Figure 9).

Ratio-metric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if $V_{CC} - V_{REF} = 5.12V$, then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

ADC0808/A DC.08.09

Applications Information (Continued)

2.0 RESISTOR LADDER LIMITATIONS

The voltages from the resistor ladder are compared to the selected into 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(-), should

not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.

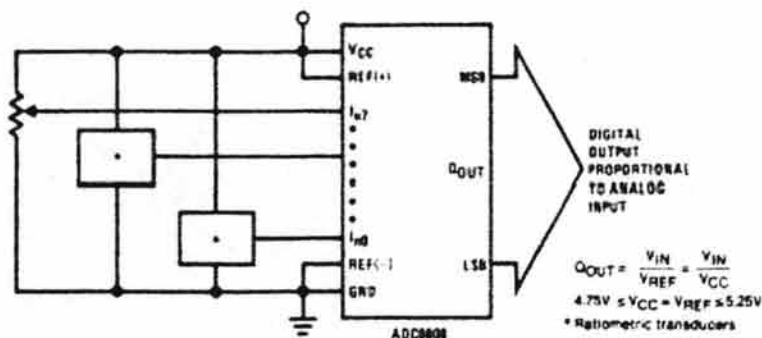
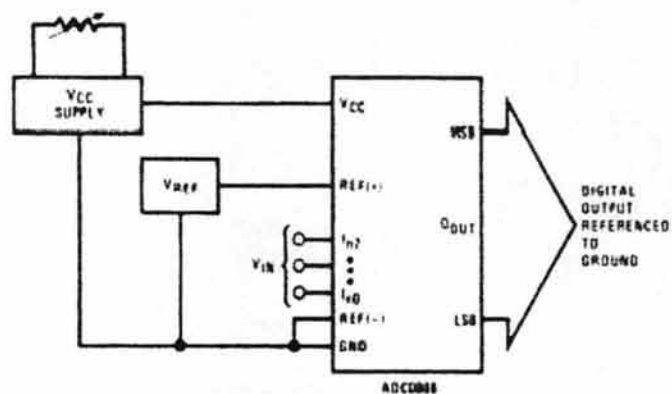


FIGURE 9. Ratiometric Conversion System

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In Figure 11 a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in Figure 12. The LM301 is overcompensated to insure stability when loaded by the 10 μ F output capacitor.

The top and bottom ladder voltages cannot exceed V_{CC} and ground, respectively, but they can be symmetrically less than V_{CC} and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In Figure 13, a 2.5V reference is symmetrically centered about $V_{CC}/2$ since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.

Applications Information (Continued)

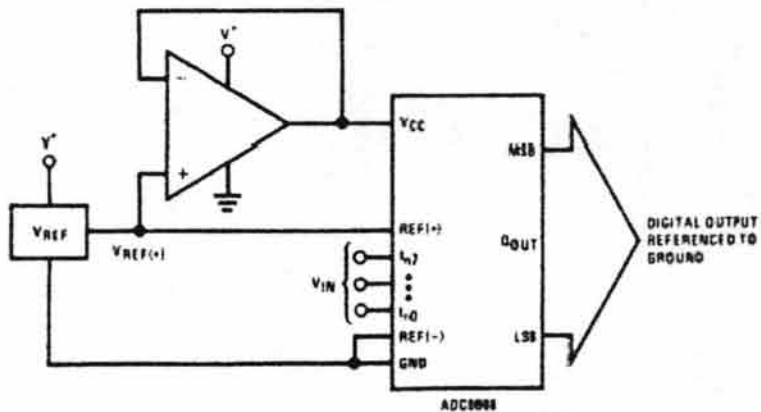


004724

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply



004725

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 11. Ground Referenced Conversion System with Reference Generating V_{CC} Supply

Applications Information (Continued)

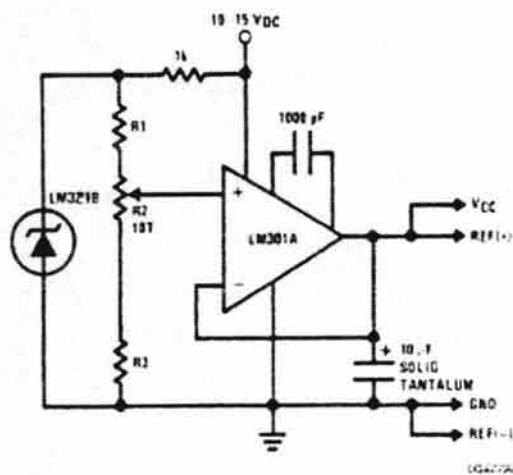
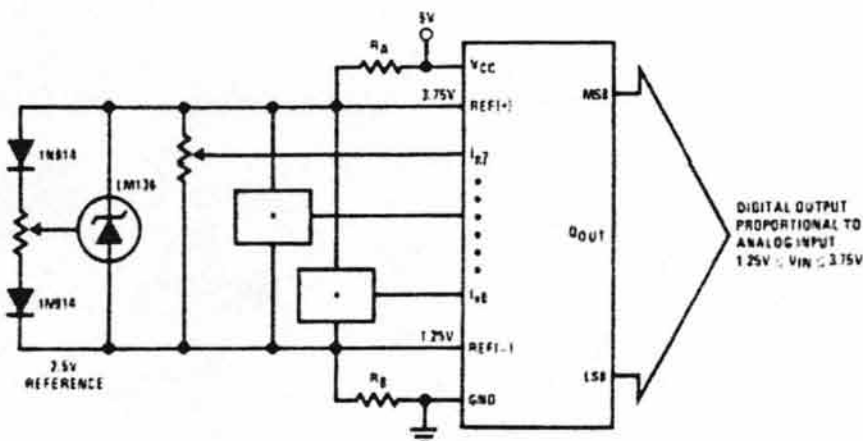


FIGURE 12. Typical Reference and Supply Circuit



$R_A = R_D$
 *Ratiometric transducers

FIGURE 13. Symmetrically Centered Reference

3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left[(V_{REF(+)} - V_{REF(-)}) \left(\frac{N}{256} + \frac{1}{512} \right) + V_{TUE} \right] + V_{REF(+)} \quad (2)$$

The center of an output code N is given by:

$$V_{IN} = \left[(V_{REF(+)} - V_{REF(-)}) \left(\frac{N}{256} \right) + V_{TUE} \right] + V_{REF(+)} \quad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \cdot 256 + \text{Absolute Accuracy} \quad (4)$$

Where: V_{IN} = Voltage at comparator input
 $V_{REF(+)}$ = Voltage at Ref(+)
 $V_{REF(-)}$ = Voltage at Ref(-)
 V_{TUE} = Total unadjusted error voltage (typically $V_{REF(+)} + 512$)

Applications Information (Continued)

4.0 ANALOG COMPARATOR INPUTS

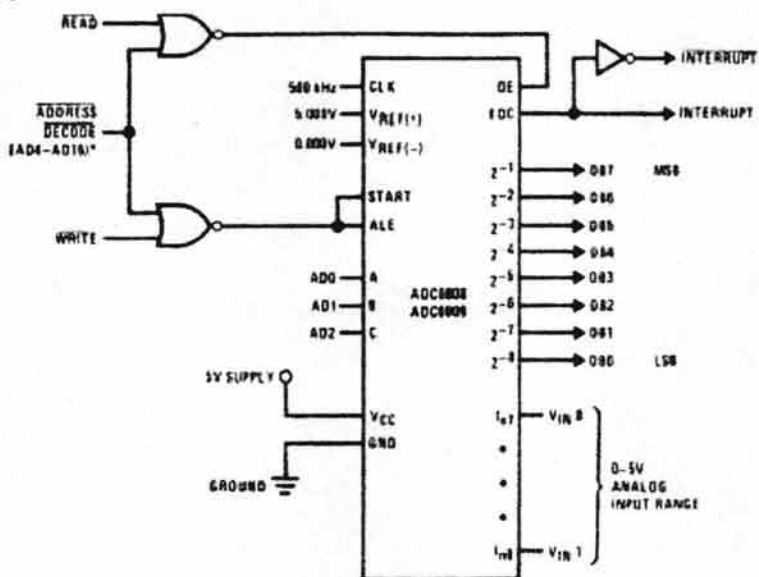
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in Figure 6.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

Typical Application



*Address latches needed for 8085 and SCMP interfacing the ADC0808 to a microprocessor

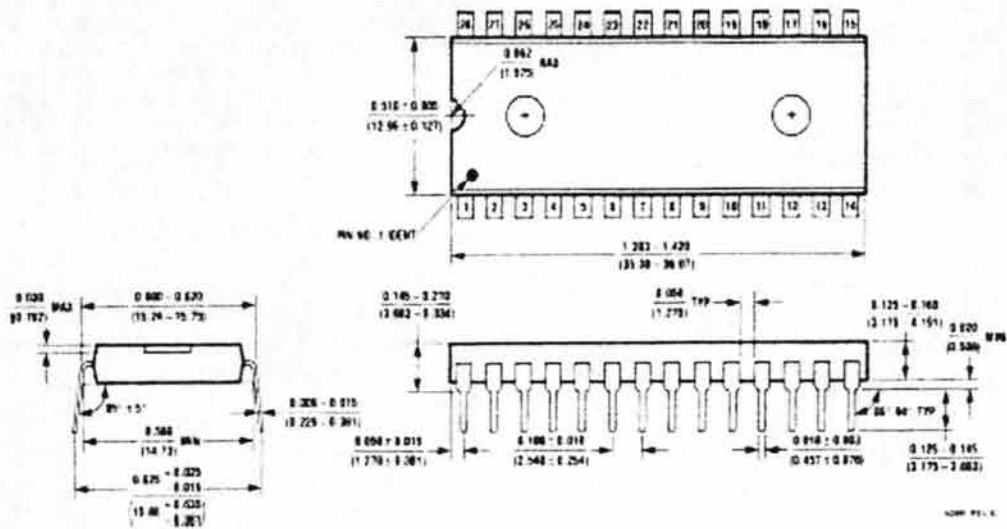
13A2719

TABLE 2. Microprocessor Interface Table

PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	MEMR	MEMW	INTR (Thru RST Circuit)
8085	RD	WR	INTR (Thru RST Circuit)
Z-80	RD	WR	INT (Thru RST Circuit, Mode 0)
SCMP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA* \overline{O} *R/W	VMA* \overline{O} *R/W	IRQA or IRQB (Thru PIA)

ADC0808/A DC 0809

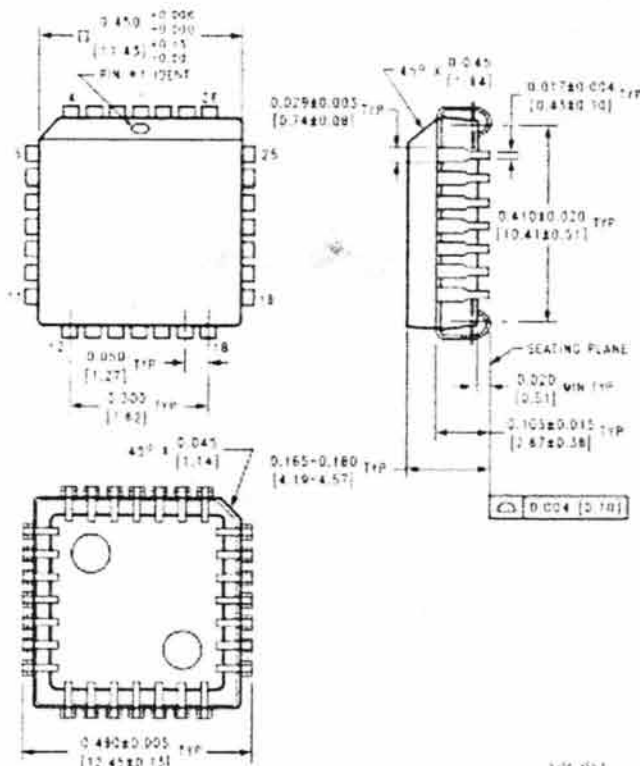
Physical Dimensions inches (millimeters)
unless otherwise noted



Molded Dual-In-Line Package (II)
 Order Number ADC0808CCN or ADC0809CCN
 NS Package Number N28B

NSM P11.6

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Chip Carrier (V)
Order Number ADC0808CCV or ADC0809CCV
NS Package Number V28A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180 630 85 86
Email: europe.support@nsc.com
Deutsch: Tel: +49 (0) 89 9508 6206
English: Tel: +44 (0) 870 24 0 2171
Francais: Tel: +33 (0) 1 41 91 8730

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544663
Fax: 65-2504460
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7500
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuitry described, in circuit board designs, or in other products, unless specifically stated in writing. National has the right at any time without notice to change said circuitry and specifications.

Anexo 4



March 1998

DM74LS244

Octal 3-STATE Buffers/Line Drivers/Line Receivers

General Description

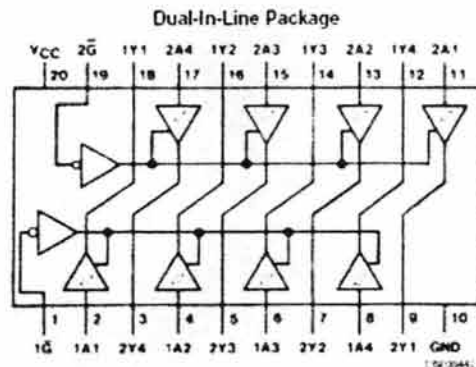
These buffers/line drivers are designed to improve both the performance and PC board density of 3-STATE buffers/drivers employed as memory address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133Ω.

Features

- 3-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at data inputs improves noise margins

- Typical I_{OL} (sink current)
 - 54LS 12 mA
 - 74LS 24 mA
- Typical I_{OH} (source current)
 - 54LS -12 mA
 - 74LS -15 mA
- Typical propagation delay times
 - Inverting 10.5 ns
 - Noninverting 12 ns
- Typical enable/disable time 18 ns
- Typical power dissipation (enabled)
 - Inverting 130 mW
 - Noninverting 135 mW

Connection Diagram



Order Number 54LS244DMQB, 54LS244FMOB, 54LS244LMOB,
DM74LS244WM or DM74LS244N
See Package Number E20A, J20A, M20B, N20A or W20A

Function Table

Inputs		Output
\bar{G}	A	Y
L	L	L
L	H	H
H	X	Z

L = Low Logic Level
H = High Logic Level
X = Either Low or High Logic Level
Z = High Impedance

DM74LS244 Octal 3-STATE Buffers/Line Drivers/Line Receivers

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V	54LS	-55 C to +125 C
Input Voltage	7V	DM74LS	0 C to +70 C
Operating Free Air Temperature Range		Storage Temperature Range	-65 C to +150 C

Recommended Operating Conditions

Symbol	Parameter	54LS244			DM74LS244			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			-12			-15	mA
I_{OL}	Low Level Output Current			12			24	mA
T_A	Free Air Operating Temperature	-55		125	0		70	°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted):

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units	
V_i	Input Clamp Voltage	$V_{CC} - \text{Min}$, $I_i = -18 \text{ mA}$			-1.5	V	
HYS	Hysteresis ($V_{i+} - V_{i-}$) Data Inputs Only	$V_{CC} - \text{Min}$	0.2	0.4		V	
V_{OH}	High Level Output Voltage	$V_{CC} - \text{Min}$, $V_{IH} - \text{Min}$ $V_{OL} - \text{Max}$, $I_{OH} = -1 \text{ mA}$	DM74	2.7		V	
		$V_{CC} - \text{Min}$, $V_{IH} - \text{Min}$ $V_{OL} - \text{Max}$, $I_{OH} = -3 \text{ mA}$	54LS/DM74	2.4	3.4		
		$V_{CC} - \text{Min}$, $V_{IH} - \text{Min}$ $V_{OL} = 0.5 \text{ V}$, $I_{OH} = \text{Max}$	54LS/DM74	2			
V_{OL}	Low Level Output Voltage	$V_{CC} - \text{Min}$ $V_{IL} - \text{Max}$ $V_{IH} - \text{Min}$	$I_{OL} = 12 \text{ mA}$ $I_{OL} = \text{Max}$	54LS/DM74 DM74	0.4 0.5	V	
I_{OZH}	Off-State Output Current, High Level Voltage Applied	$V_{CC} - \text{Max}$ $V_{OL} - \text{Max}$	$V_O = 2.7 \text{ V}$		20	μA	
I_{OZL}	Off-State Output Current, Low Level Voltage Applied	$V_{IH} - \text{Min}$	$V_O = 0.4 \text{ V}$		-20	μA	
I_i	Input Current at Maximum Input Voltage	$V_{CC} - \text{Max}$	$V_i = 7 \text{ V}$ (DM74) $V_i = 10 \text{ V}$ (54LS)		0.1	mA	
I_{iH}	High Level Input Current	$V_{CC} - \text{Max}$	$V_i = 2.7 \text{ V}$		20	μA	
I_{iL}	Low Level Input Current	$V_{CC} - \text{Max}$	$V_i = 0.4 \text{ V}$	-0.5	-200	μA	
I_{OS}	Short Circuit Output Current	$V_{CC} - \text{Max}$ (Note 3)	54LS	-50		mA	
			DM74	-40			
I_{CC}	Supply Current	$V_{CC} - \text{Max}$, Outputs Open	Outputs High		13	23	mA
			Outputs Low		27	46	
			Outputs Disabled		32	54	

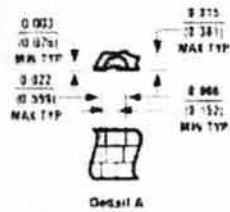
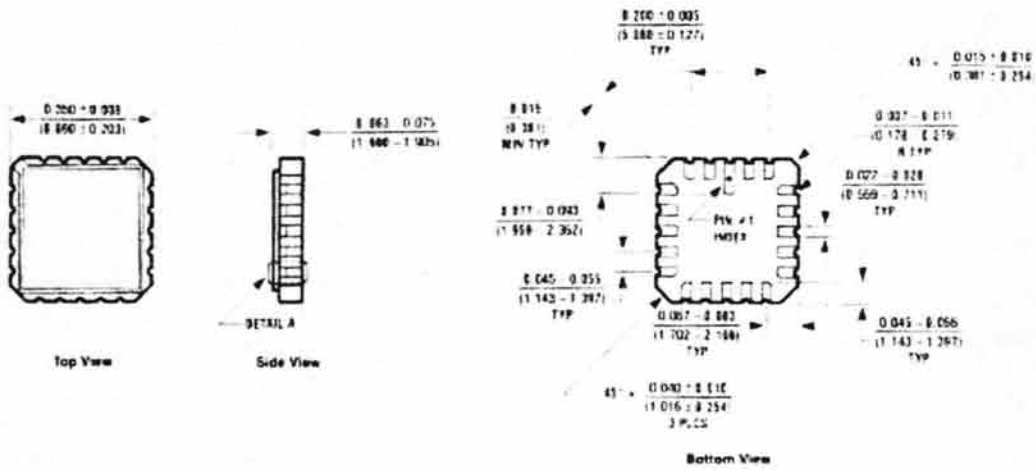
Note 2: All typicals are at $V_{CC} = 5 \text{ V}$, $T_A = 25 \text{ }^\circ\text{C}$.

Note 3: Not more than one output should be stored at a time, and the duration should not exceed one second.

Switching Characteristics					
at $V_{CC} = 5V$, $T_A = 25^\circ C$					
Symbol	Parameter	Conditions	54LS Max	DM74LS Max	Units
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 45 \text{ pF}$ $R_L = 667\Omega$	10	10	ns
t_{PLL}	Propagation Delay Time High to Low Level Output	$C_L = 45 \text{ pF}$ $R_L = 667\Omega$	10	10	ns
t_{OLN}	Output Enable Time to Low Level	$C_L = 45 \text{ pF}$ $R_L = 667\Omega$	30	30	ns
t_{OLH}	Output Enable Time to High Level	$C_L = 45 \text{ pF}$ $R_L = 667\Omega$	23	23	ns
t_{OLZ}	Output Disable Time from Low Level	$C_L = 5 \text{ pF}$ $R_L = 667\Omega$	25	25	ns
t_{OLV}	Output Disable Time from High Level	$C_L = 5 \text{ pF}$ $R_L = 667\Omega$	18	18	ns
t_{PZH}	Propagation Delay Time Low to High Level Output	$C_L = 150 \text{ pF}$ $R_L = 667\Omega$		21	ns
t_{PZL}	Propagation Delay Time High to Low Level Output	$C_L = 150 \text{ pF}$ $R_L = 667\Omega$		22	ns
t_{OLN}	Output Enable Time to Low Level	$C_L = 150 \text{ pF}$ $R_L = 667\Omega$		33	ns
t_{OLH}	Output Enable Time to High Level	$C_L = 150 \text{ pF}$ $R_L = 667\Omega$		26	ns

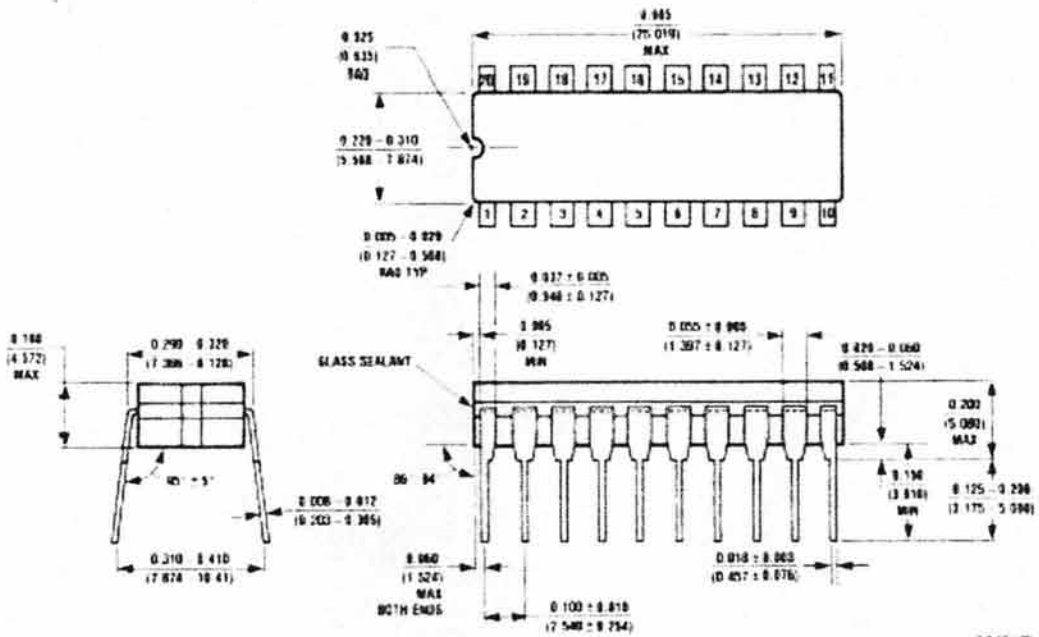
Note 4: 54LS Output Load is $C_L = 50 \text{ pF}$ for t_{PLH} , t_{PLL} , t_{OLN} and t_{OLH}

Physical Dimensions inches (millimeters) unless otherwise noted



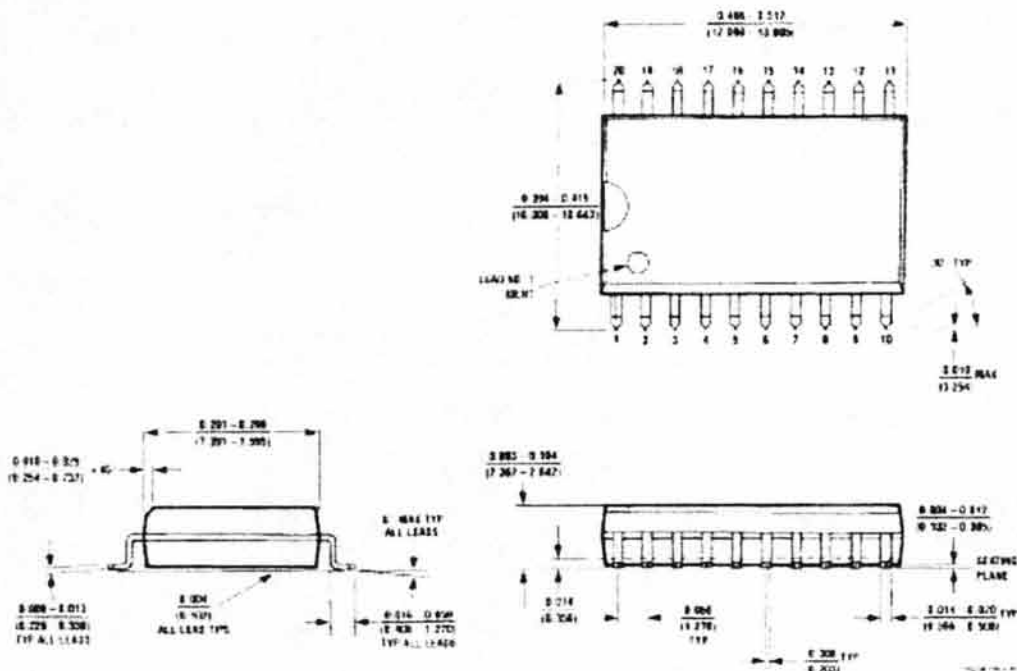
Ceramic Leadless Chip Carrier Package (E)
 Order Number 54LS244LMQB
 Package Number E20A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

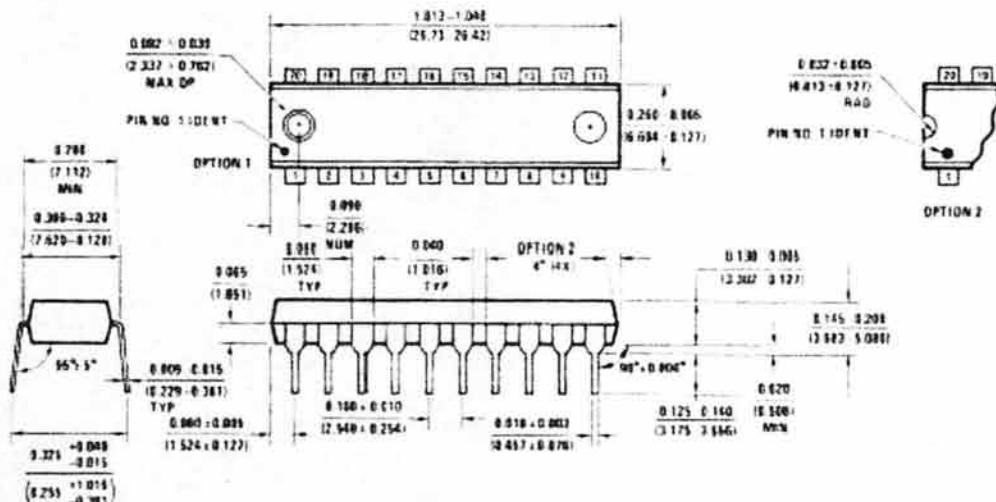


20-Lead Ceramic Dual-In-Line Package (J)
 Order Number 54LS244DMQB
 Package Number J20A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

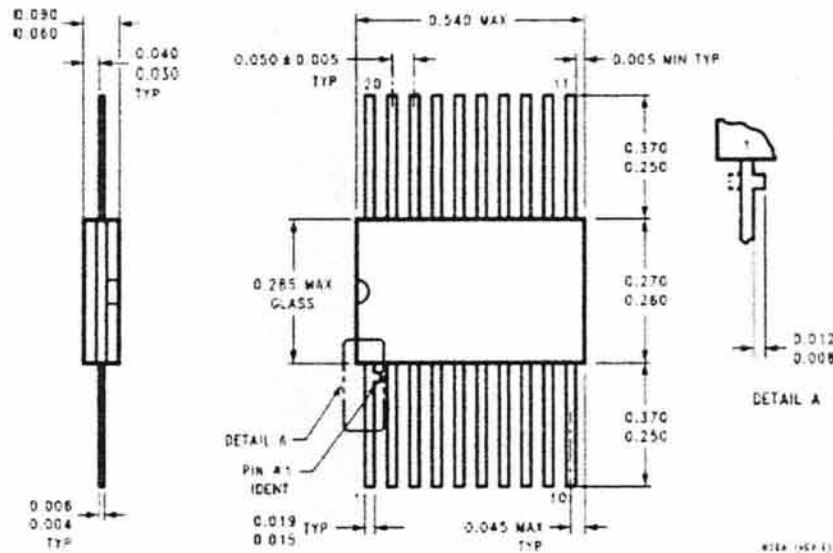


20-Lead Wide Small Outline Molded Package (M)
Order Number DM74LS244WM
Package Number M20B



20-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS244N
Package Number N20A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



20-Lead Ceramic Flat Package (W)
Order Number 54LS244FMOB
Package Number W20A

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Fairchild Semiconductor Corporation
Americas
Customer Response Center
Tel: 1-888-522-5372

Fairchild Semiconductor Europe
Fax: +49 (0) 1 80 530 95 96
Email: europe.support@fsc.com
Deutsch: Tel: +49 (0) 8 141 35-0
English: Tel: +44 (0) 1 793 85 68 56
Italy: Tel: +39 (0) 2 57 56 51

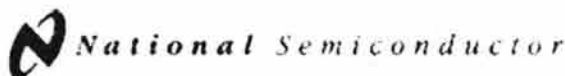
Fairchild Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd,
Tsimshatsui, Kowloon,
Hong Kong
Tel: +852 2737 7200
Fax: +852 2734 0061

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179

www.fairchildsemi.com

Fairchild does not assume any responsibility for use of any circuitry described in this data sheet except as expressly approved by Fairchild in writing. Fairchild reserves the right at any time without notice to change any circuitry and specifications.

Anexo 5



December 1994

LM185/LM285/LM385 Adjustable Micropower Voltage References

General Description

The LM185/LM285/LM385 are micropower 3-terminal adjustable band-gap voltage reference diodes. Operating from 1.24 to 5.3V and over a 10 μ A to 20 mA current range, they feature exceptionally low dynamic impedance and good temperature stability. On-chip trimming is used to provide tight voltage tolerance. Since the LM185 band-gap reference uses only transistors and resistors, low noise and good long-term stability result.

Careful design of the LM185 has made the device tolerant of capacitive loading, making it easy to use in almost any reference application. The wide dynamic operating range allows its use with widely varying supplies with excellent regulation.

The extremely low power drain of the LM185 makes it useful for micropower circuitry. This voltage reference can be used to make portable meters, regulators or general purpose an-

alog circuitry with battery life approaching shelf life. Further, the wide operating current allows it to replace older references with a tighter tolerance part.

The LM185 is rated for operation over a -55°C to 125°C temperature range, while the LM285 is rated -40°C to 85°C and the LM385 0°C to 70°C . The LM185 is available in a hermetic TO-46 package and a leadless chip carrier package, while the LM285/LM385 are available in a low-cost TO-92 molded package, as well as S.O.

Features

- Adjustable from 1.24V to 5.30V
- Operating current of 10 μ A to 20 mA
- 1% and 2% initial tolerance
- 1 Ω dynamic impedance
- Low temperature coefficient

Connection Diagrams

TO-92
Plastic Package



Bottom View

TL/16520-4

Order Number LM285BXZ,
LM285BYZ, LM285Z, LM385BXZ,
LM385BYZ, LM385BZ or LM385Z
See NS Package Number Z03A

TO-46
Metal Can Package

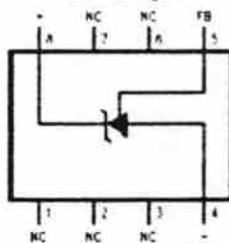


Bottom View

TL/16520-1

Order Number
LM185BH, LM185BH/883,
LM185BYH or LM185BYH/883
See NS Package Number H03H

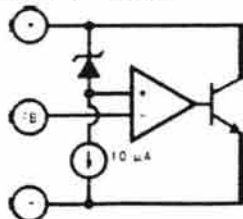
S.O. Package



TL/16520-10

Order Number LM285M, LM285BYM,
LM385BM or LM385M
See NS Package Number M08A

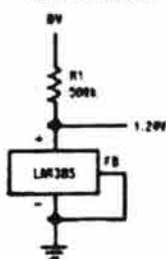
Block Diagram



TL/16520-13

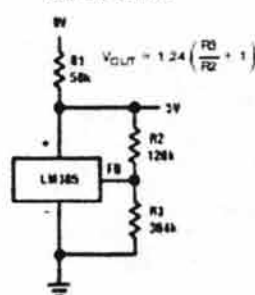
Typical Applications

1.2V Reference



TL/16520-14

5.0V Reference



TL/16520-2

LM185/285/385 Adjustable Micropower Voltage Reference

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications. (Note 2)

Reverse Current	30 mA
Forward Current	10 mA
Operating Temperature Range (Note 3)	
LM185 Series	-55°C to 125°C
LM285 Series	-40°C to 85°C
LM385 Series	0°C to 70°C
Storage Temperature	-55°C to 150°C

Soldering Information
 TO-92 Package (10 sec) 260°C
 TO-46 Package (10 sec) 300°C
 SO Package
 Vapor Phase (60 sec) 215°C
 Infrared (15 sec) 220°C
 See An-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics (Note 4)

Parameter	Conditions	LM185, LM285				LM385				Units (Limit)		
		LM185BX, LM185BY LM185B, LM285BX, LM285BY		LM285		LM385BX, LM385BY		LM385				
		Typ	Tested Limit (Note 5)	Design Limit (Note 6)	Tested Limit (Note 5)	Design Limit (Note 6)	Typ	Tested Limit (Note 5)	Design Limit (Note 6)		Tested Limit (Note 5)	Design Limit (Note 6)
Reference Voltage	$I_I = 100 \mu\text{A}$	1.240	1.252 1.288 1.228 1.218		1.265	1.270 1.215 1.208	1.240	1.252 1.226	1.288 1.218	1.265	1.270 1.208	V (max) V (min)
Reference Voltage Change with Current	$I_{IN} < I_I < 1 \text{ mA}$ $1 \text{ mA} < I_I < 20 \text{ mA}$	0.2 4	1 10	1.5 20	1 10	1.5 20	0.2 5	1 15	1.5 25	1 15	1.5 25	mV (max)
Dynamic Output Impedance	$I_I = 100 \mu\text{A}$, $f = 100 \text{ Hz}$ $A_C = 0.1$, $V_{OUT} = V_{REF}$ $V_{OUT} = 5.2 \text{ V}$	0.3 0.7					0.4 1					Ω
Reference Voltage Change with Output Voltage	$I_I = 100 \mu\text{A}$	1	3	6	3	6	2	5	10	5	10	mV (max)
Feedback Current		13	20	25	20	25	16	30	35	30	35	nA (max)
Minimum Operating Current (see curve)	$V_{OUT} = V_{REF}$ $V_{OUT} = 5.2 \text{ V}$	6 30	9 45	10 50	9 45	10 50	7 35	11 55	13 60	11 55	13 60	μA (max)
Output Wideband Noise	$I_I = 100 \mu\text{A}$, $10 \text{ Hz} < f < 10 \text{ kHz}$ $V_{OUT} = V_{REF}$ $V_{OUT} = 5.2 \text{ V}$	50 170					50 170					μV_{rms}
Average Temperature Coefficient (Note 7)	$I_I = 100 \mu\text{A}$, X Sulf Y Sulf All Others		30 50	150	30 50	150		30 50	150	30 50	150	ppm/°C (max)
Long Term Stability	$I_I = 100 \mu\text{A}$, $T = 1000 \text{ Hr}$, $T_A = 25^\circ\text{C} \pm 0.1^\circ\text{C}$	20					20					ppm

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed.

Note 2: Refer to RETS149H for military specifications.

Note 3: For elevated temperature operation, $T_{j \text{ max}}$
 LM185 150°C
 LM285 125°C
 LM385 100°C

Thermal Resistance	TO-92	TO-46	SO-8
θ_{JA} (Junction to Ambient)	180°C/W (0.41 in ² leads)	440°C/W	186°C/W
	170°C/W (0.125 in ² leads)		
θ_{JC} (Junction to Case)	N/A	60°C/W	N/A

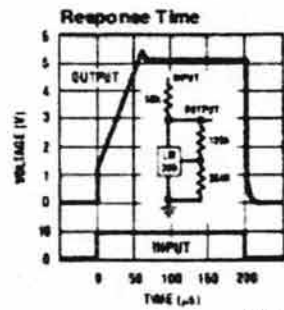
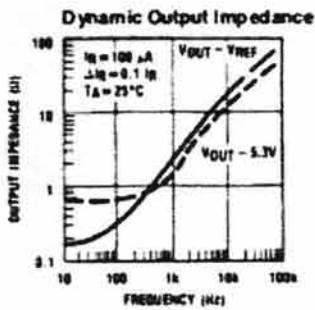
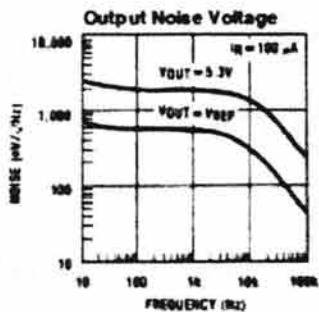
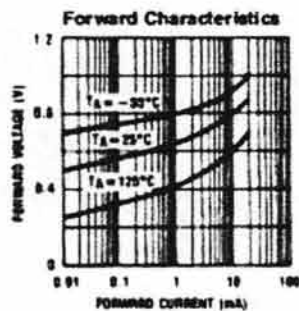
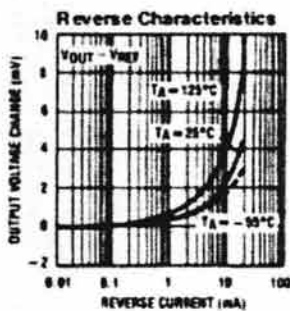
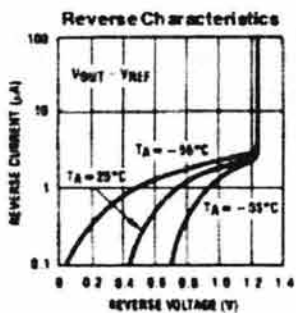
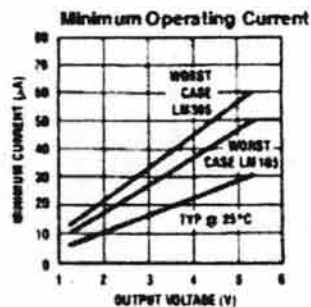
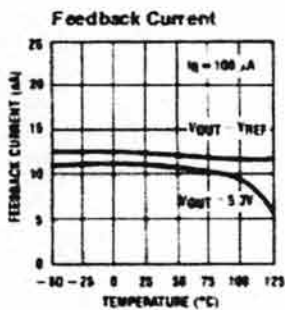
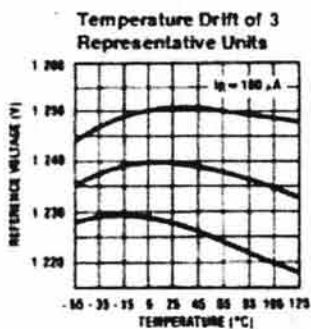
Note 4: Parameters identified with boldface type apply at temperature extremes. All other numbers apply at $T_A = T_J = 25^\circ\text{C}$. Unless otherwise specified, all parameters apply for $V_{IN} = 5 \text{ V}$, $V_{OUT} = 5.2 \text{ V}$.

Note 5: Guaranteed and 100% production tested.

Note 6: Guaranteed, but not 100% production tested. These limits are not to be used to calculate average output quality levels.

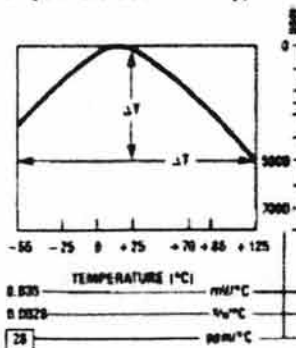
Note 7: The average temperature coefficient is defined as the maximum deviation of reference voltage at all measured temperatures from T_{ref} to T_{max} , divided by $T_{max} - T_{ref}$. The measured temperatures are -55, -40, 0, 25, 70, 85, 125°C.

Typical Performance Characteristics

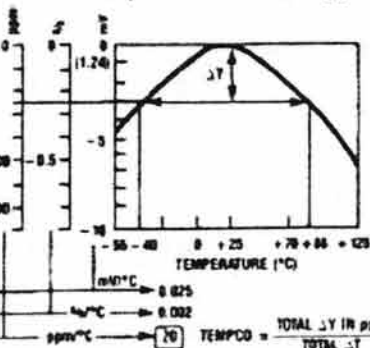


IL7145.292-3

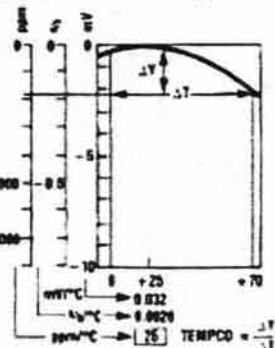
LM185
Temperature Coefficient Typical



LM285
Temperature Coefficient Typical



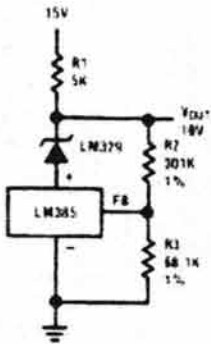
LM385
Temperature Coefficient Typical



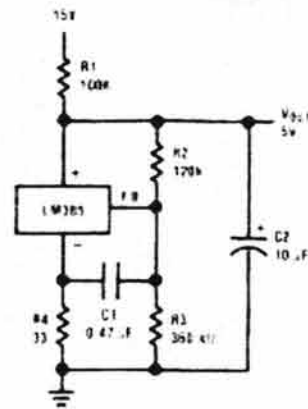
IL7145.292-4

Typical Applications (Continued)

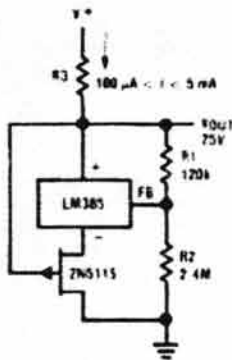
Precision 10V Reference



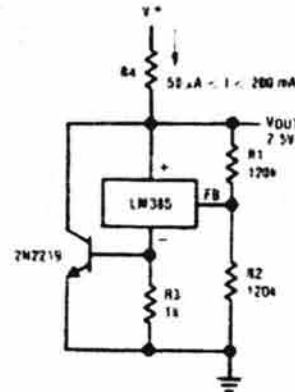
Low AC Noise Reference



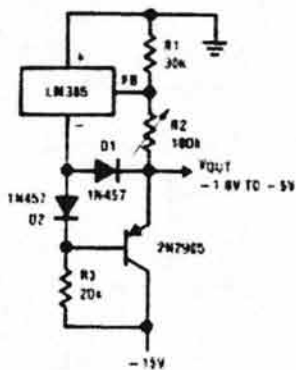
25V Low Current Shunt Regulator



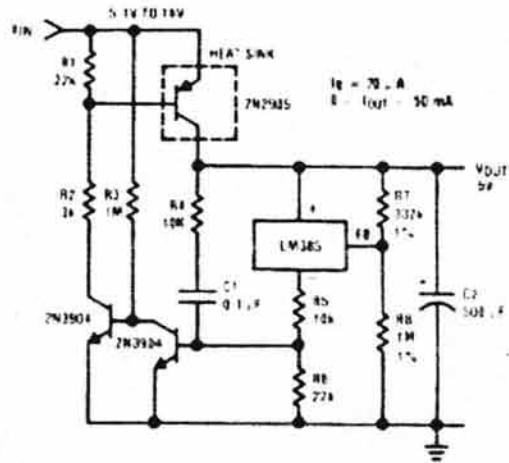
200 mA Shunt Regulator



Series-Shunt 20 mA Regulator



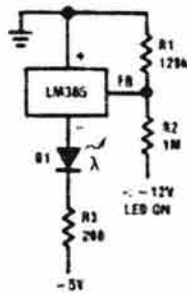
High Efficiency Low Power Regulator



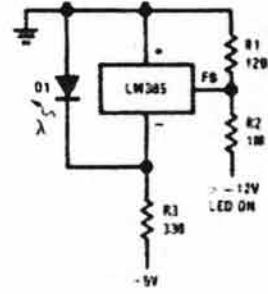
TL910520-5

Typical Applications (Continued)

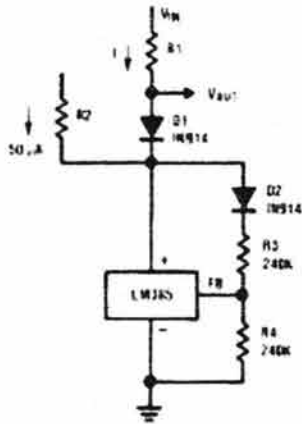
Voltage Level Detector



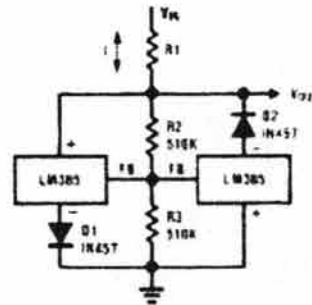
Voltage Level Detector



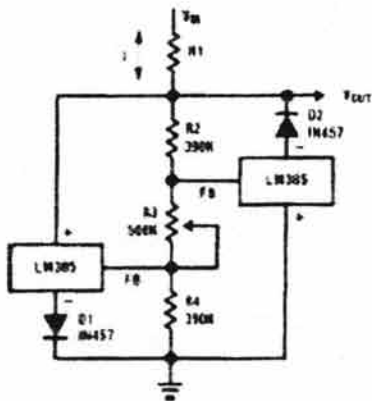
Fast Positive Clamp
 $2.4V + \Delta V_{D1}$



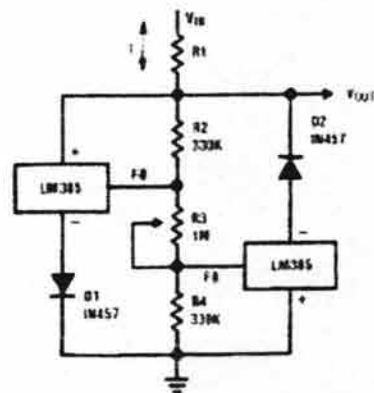
Bidirectional Clamp
 $+ 2.4V$



Bidirectional Adjustable Clamp
 $+ 1.8V$ to $+ 2.4V$



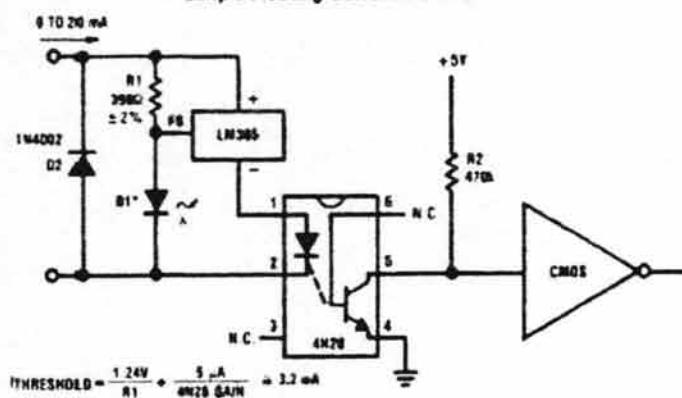
Bidirectional Adjustable Clamp
 $+ 2.4V$ to $+ 6V$



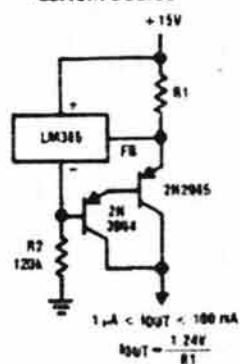
TL7919290-d

Typical Applications (Continued)

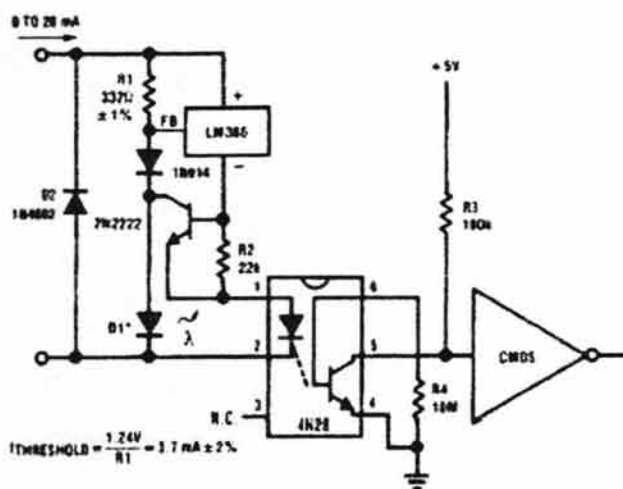
Simple Floating Current Detector



Current Source



Precision Floating Current Detector

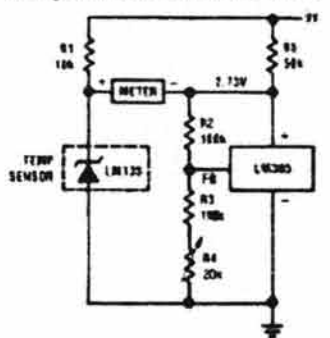


* D1 can be any LED. $V_f = 1.5\text{V to } 2.2\text{V at } 3\text{ mA}$. D1 may act as an indicator. D1 will be on if $I_{\text{THRESHOLD}}$ falls below the threshold current, except with I=0.

TL765210-7

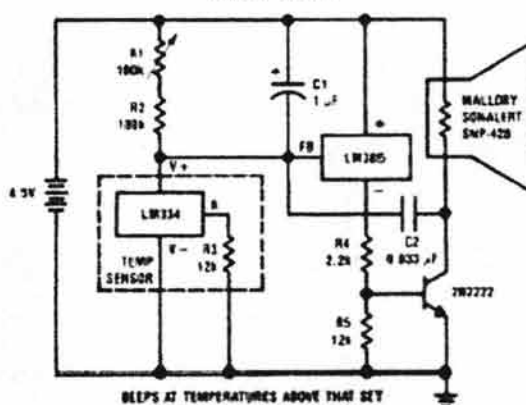
Typical Applications (Continued)

Centigrade Thermometer, 10 mV/°C



TLH-1520-11

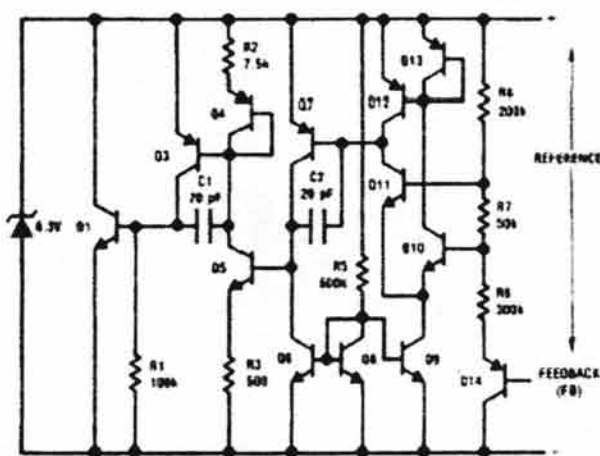
Freezer Alarm



BEEPS AT TEMPERATURES ABOVE THAT SET BY R1 (RANGE IS -30°F to +120°F)

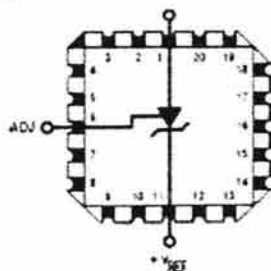
TLH-1520-12

Schematic Diagram



TLH-1520-8

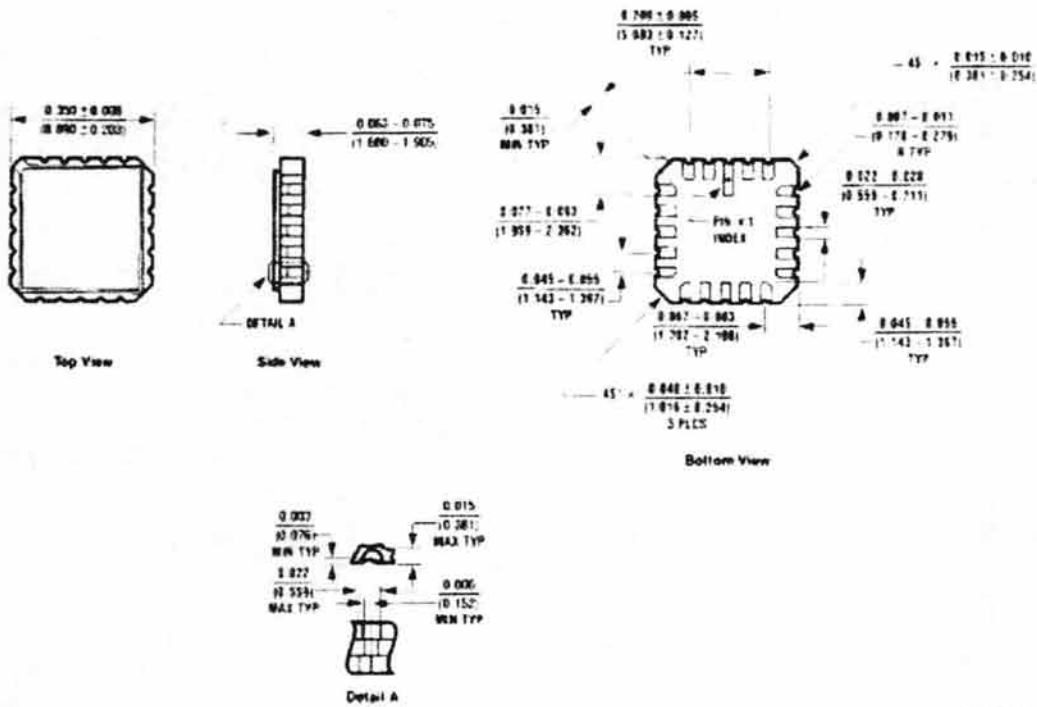
Connection Diagrams (Continued)



TLH-1520-15

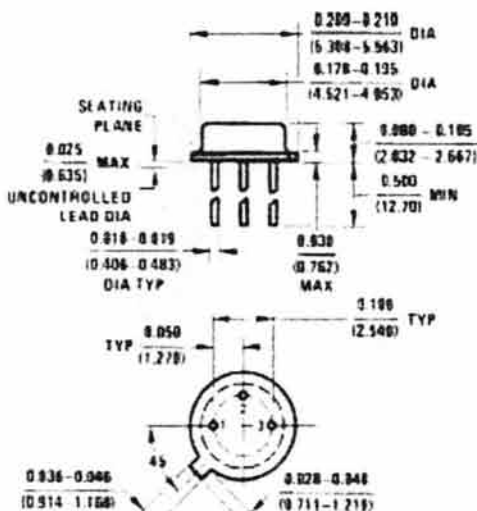
Order Number LM185BE/883
See NS Package Number E20A

Physical Dimensions inches (millimeters)

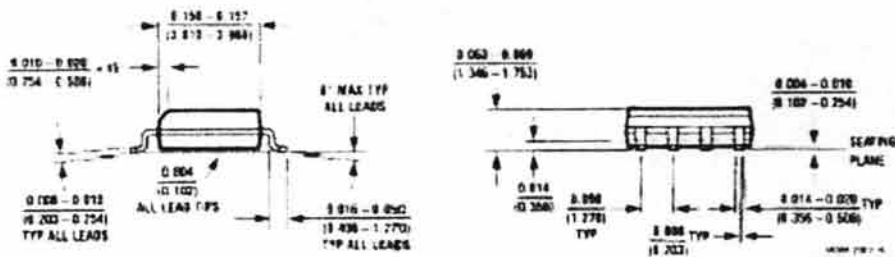
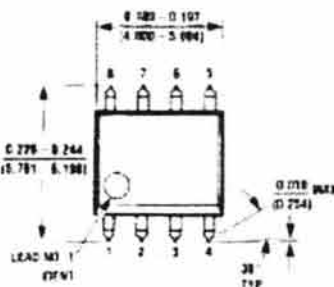


20-Leadless Chip Carrier (E)
 Order Number LM185BE/883
 NS Package Number E20A

Physical Dimensions inches (millimeters) (Continued)

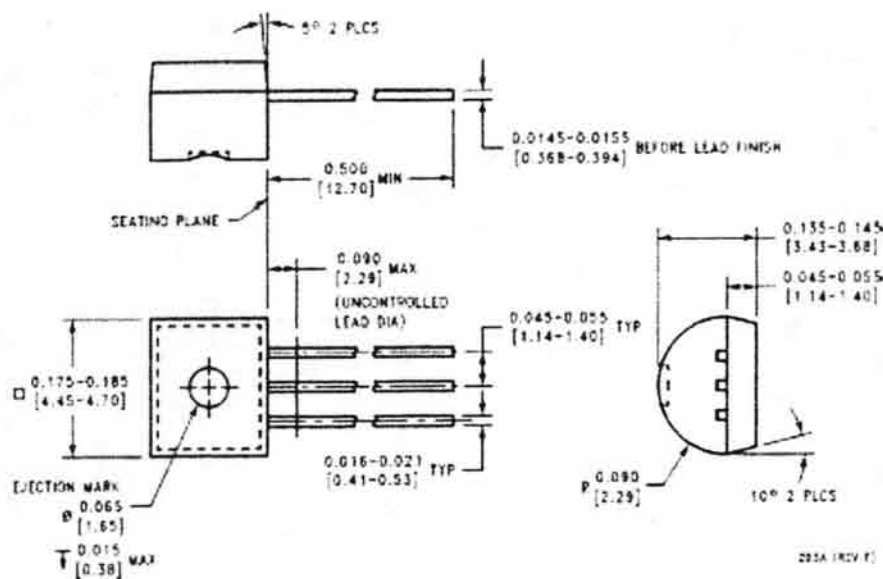


TO-48 Metal Can Package (M)
 Order Number LM185BH, LM185BH/883,
 LM185BYH or LM185BYH/883
 NS Package Number H03H



SO Package (M)
 Order Number LM285M, LM285YM, LM385M or LM385M
 NS Package Number M08A

Physical Dimensions inches (millimeters) (Continued)



TO-92 Plastic Package (Z)
Order Number LM385BZ, LM385Z, LM385BXZ, LM385BYZ, LM285Z, LM285BXZ or LM285BYZ
NS Package Number Z03A

LIFE SUPPORT POLICY

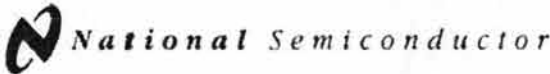
NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

<p>National Semiconductor Corporation 1111 West Bend Road Arlington, TX 76017 Tel: (817) 272-2000 Fax: (817) 272-2000</p>	<p>National Semiconductor Europe Fax: (+43) 0-780 521 89 86 Email: onyc@europe.nsc.com Dusseldorf Tel: (+49) 0-780 521 89 85 England Tel: (+49) 0-780 521 78 32 Frankfurt Tel: (+49) 0-780 521 93 58 Munich Tel: (+49) 0-780 521 18 90</p>	<p>National Semiconductor Hong Kong Ltd. 13th Floor, Straight Block, Ocean Centre, 5 Canton Rd, Tsimshatsui, Kowloon Hong Kong Tel: (852) 2717 7800 Fax: (852) 2716 9200</p>	<p>National Semiconductor Japan Ltd. Tel: 81-0-3-299-2322 Fax: 81-0-3-299-2409</p>
--	---	---	---

© National Semiconductor Corporation 1992. All rights reserved. This document is the property of National Semiconductor Corporation. No part of this document may be reproduced without the express written permission of National Semiconductor Corporation.

Anexo 6



February 2000

LM555 Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200mA or drive TTL circuits.

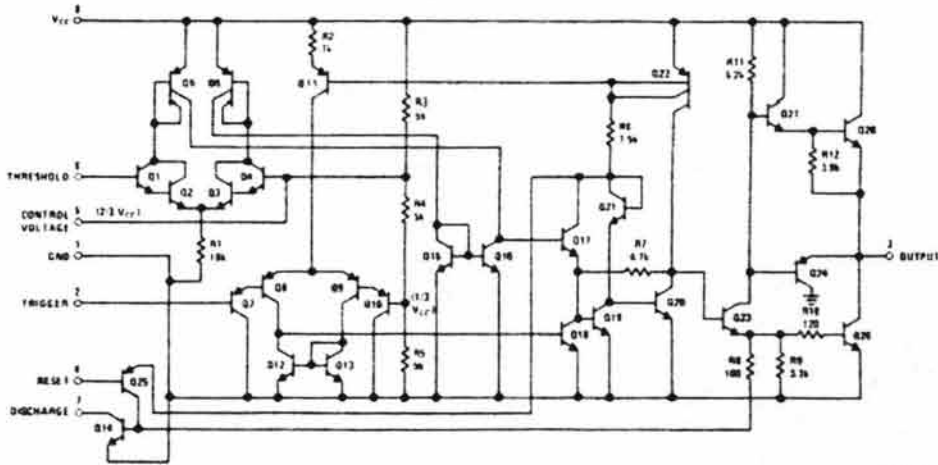
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per C
- Normally on and normally off output
- Available in 8-pin MSOP package

Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

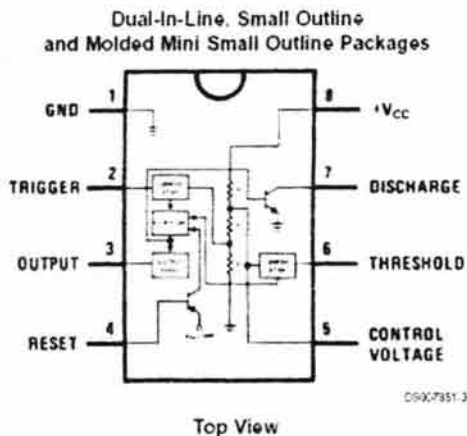
Schematic Diagram



2501985-1

LM555

Connection Diagram



Ordering Information

Package	Part Number	Package Marking	Media Transport	NSC Drawing
8-Pin SOIC	LM555CM	LM555CM	Rails	M08A
	LM555CMX	LM555CM	2.5k Units Tape and Reel	
8-Pin MSOP	LM555CMM	Z55	1k Units Tape and Reel	MUA08A
	LM555CMMX	Z55	3.5k Units Tape and Reel	
8-Pin MDIP	LM555CN	LM555CN	Rails	N08E

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+18V
Power Dissipation (Note 3)	
LM555CM, LM555CN	1180 mW
LM555CMM	613 mW
Operating Temperature Ranges	
LM555C	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Soldering Information

Dual-In-Line Package	
Soldering (10 Seconds)	260°C
Small Outline Packages (SOIC and MSOP)	
Vapor Phase (60 Seconds)	215°C
Infrared (15 Seconds)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics (Notes 1, 2)

($T_A = 25^\circ\text{C}$, $V_{CC} = +5\text{V}$ to $+15\text{V}$, unless otherwise specified)

Parameter	Conditions	Limits			Units
		LM555C			
		Min	Typ	Max	
Supply Voltage		4.5		16	V
Supply Current	$V_{CC} = 5\text{V}$, $R_L = \infty$ $V_{CC} = 15\text{V}$, $R_L = \infty$ (Low State) (Note 4)		3 10	6 15	mA
Timing Error, Monostable					
Initial Accuracy			1		%
Drift with Temperature	$R_A = 1\text{k}$ to $100\text{k}\Omega$ $C = 0.1\mu\text{F}$, (Note 5)		50		ppm/°C
Accuracy over Temperature			1.5		%
Drift with Supply			0.1		%/V
Timing Error, Astable					
Initial Accuracy			2.25		%
Drift with Temperature	$R_A, R_B = 1\text{k}$ to $100\text{k}\Omega$ $C = 0.1\mu\text{F}$, (Note 5)		150		ppm/°C
Accuracy over Temperature			3.0		%
Drift with Supply			0.30		%/V
Threshold Voltage			0.667		$\times V_{CC}$
Trigger Voltage	$V_{CC} = 15\text{V}$ $V_{CC} = 5\text{V}$		5 1.67		V V
Trigger Current			0.5	0.9	μA
Reset Voltage		0.4	0.5	1	V
Reset Current			0.1	0.4	mA
Threshold Current	(Note 6)		0.1	0.25	μA
Control Voltage Level	$V_{CC} = 15\text{V}$ $V_{CC} = 5\text{V}$	9 2.6	10 3.33	11 4	V
Pin 7 Leakage Output High			1	100	nA
Pin 7 Sat (Note 7)					
Output Low	$V_{CC} = 15\text{V}$, $I_T = 15\text{mA}$		180		mV
Output Low	$V_{CC} = 4.5\text{V}$, $I_T = 4.5\text{mA}$		80	200	mV

LM555

Electrical Characteristics (Notes 1, 2) (Continued)

($T_A = 25^\circ\text{C}$, $V_{CC} = +5\text{V}$ to $+15\text{V}$, unless otherwise specified)

Parameter	Conditions	Limits			Units
		LM555C			
		Min	Typ	Max	
Output Voltage Drop (Low)	$V_{CC} = 15\text{V}$				
	$I_{SINK} = 10\text{mA}$		0.1	0.25	V
	$I_{SINK} = 50\text{mA}$		0.4	0.75	V
	$I_{SINK} = 100\text{mA}$		2	2.5	V
	$I_{SINK} = 200\text{mA}$		2.5		V
	$V_{CC} = 5\text{V}$				
	$I_{SINK} = 8\text{mA}$		0.25	0.35	V
Output Voltage Drop (High)	$I_{SOURCE} = 200\text{mA}$, $V_{CC} = 15\text{V}$		12.5		V
	$I_{SOURCE} = 100\text{mA}$, $V_{CC} = 15\text{V}$	12.75	13.3		V
	$V_{CC} = 5\text{V}$	2.75	3.3		V
Rise Time of Output			100		ns
Fall Time of Output			100		ns

Note 1: All voltages are measured with respect to the ground pin, unless otherwise specified.

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For operating at elevated temperatures, the device must be derated above 25°C based on a $+150^\circ\text{C}$ maximum junction temperature and a thermal resistance of 106 $^\circ\text{C/W}$ (DIP), 170 $^\circ\text{C/W}$ (SO-8), and 204 $^\circ\text{C/W}$ (MSOP) junction to ambient.

Note 4: Supply current when output high typically 1 mA less at $V_{CC} = 5\text{V}$.

Note 5: Tested at $V_{CC} = 5\text{V}$ and $V_{CC} = 15\text{V}$.

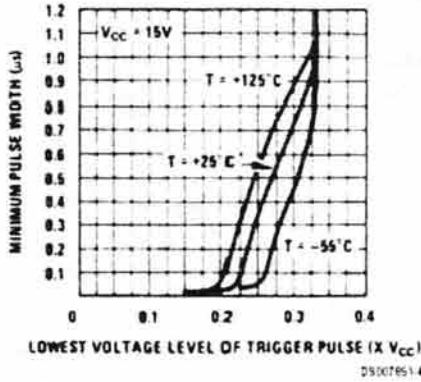
Note 6: This will determine the maximum value of $R_A + R_B$ for 15V operation. The maximum total ($R_A + R_B$) is 20M Ω .

Note 7: No protection against excessive pin 7 current is necessary providing the package dissipation rating will not be exceeded.

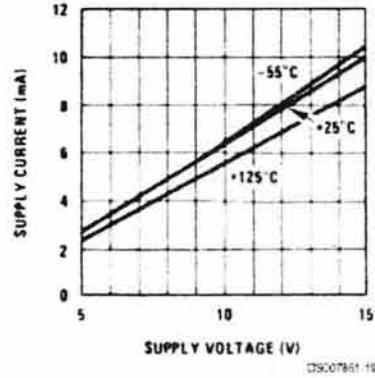
Note 8: Refer to RET555X drawing of military LM555H and LM555J versions for specifications.

Typical Performance Characteristics

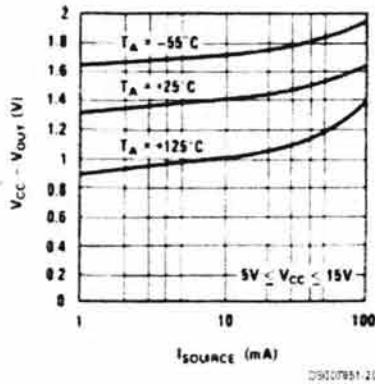
Minimum Pulse Width Required for Triggering



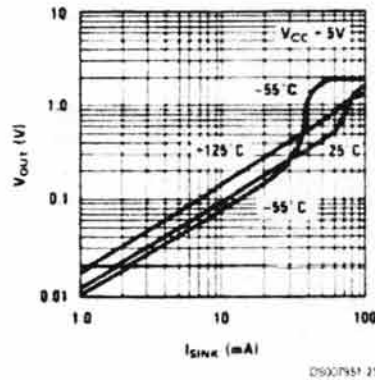
Supply Current vs. Supply Voltage



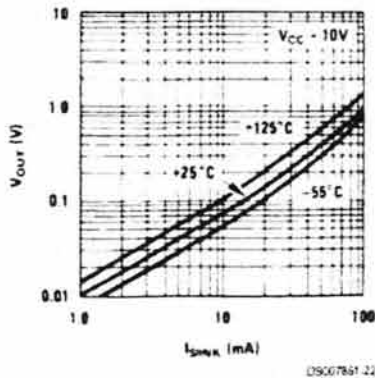
High Output Voltage vs. Output Source Current



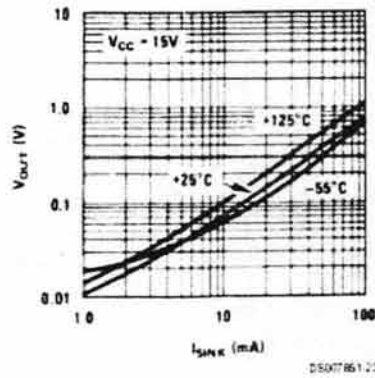
Low Output Voltage vs. Output Sink Current



Low Output Voltage vs. Output Sink Current



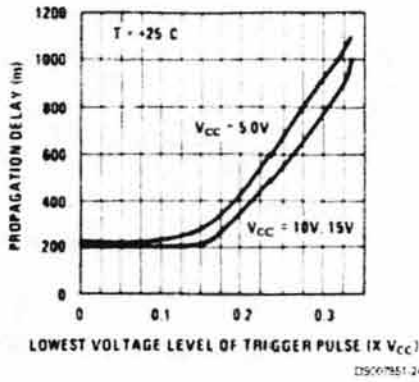
Low Output Voltage vs. Output Sink Current



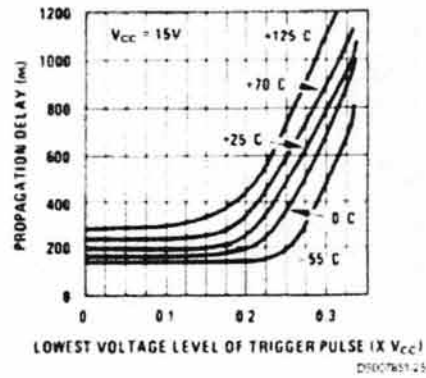
LM555

Typical Performance Characteristics (Continued)

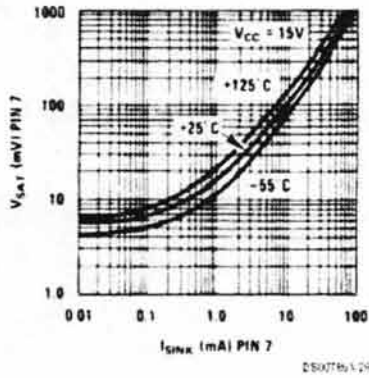
Output Propagation Delay vs. Voltage Level of Trigger Pulse



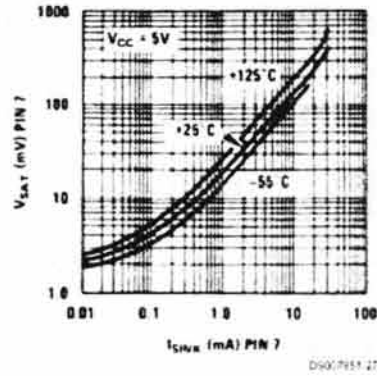
Output Propagation Delay vs. Voltage Level of Trigger Pulse



Discharge Transistor (Pin 7) Voltage vs. Sink Current



Discharge Transistor (Pin 7) Voltage vs. Sink Current



Applications Information

MONOSTABLE OPERATION

In this mode of operation, the timer functions as a one-shot (Figure 1). The external capacitor is initially held discharged by a transistor inside the timer. Upon application of a negative trigger pulse of less than $1/3 V_{CC}$ to pin 2, the flip-flop is set which both releases the short circuit across the capacitor and drives the output high.

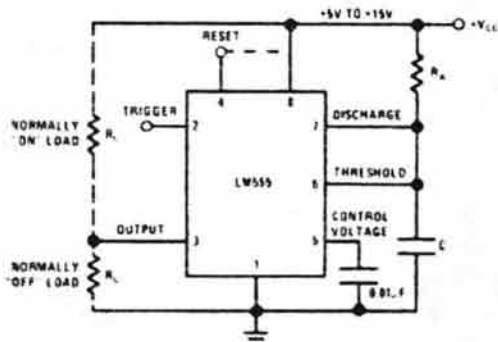
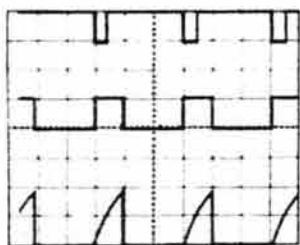


FIGURE 1. Monostable

The voltage across the capacitor then increases exponentially for a period of $t = 1.1 R_A C$, at the end of which time the voltage equals $2/3 V_{CC}$. The comparator then resets the flip-flop which in turn discharges the capacitor and drives the output to its low state. Figure 2 shows the waveforms generated in this mode of operation. Since the charge and the threshold level of the comparator are both directly proportional to supply voltage, the timing interval is independent of supply.



$V_{CC} = 5V$
 TIME = 0.1 ms/DIV
 $R_A = 9.1k\Omega$
 $C = 0.01\mu F$
 Top Trace: Input 5V/Div
 Middle Trace: Output 5V/Div
 Bottom Trace: Capacitor Voltage 2V/Div

FIGURE 2. Monostable Waveforms

During the timing cycle when the output is high, the further application of a trigger pulse will not effect the circuit so long as the trigger input is returned high at least $10\mu s$ before the end of the timing interval. However the circuit can be reset during this time by the application of a negative pulse to the reset terminal (pin 4). The output will then remain in the low state until a trigger pulse is again applied.

When the reset function is not in use, it is recommended that it be connected to V_{CC} to avoid any possibility of false triggering.

Figure 3 is a nomograph for easy determination of R, C values for various time delays.

NOTE: In monostable operation, the trigger should be driven high before the end of timing cycle.

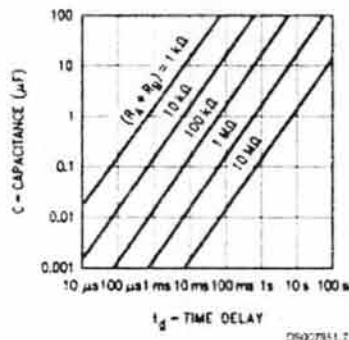


FIGURE 3. Time Delay

ASTABLE OPERATION

If the circuit is connected as shown in Figure 4 (pins 2 and 6 connected) it will trigger itself and free run as a multivibrator. The external capacitor charges through $R_A + R_B$ and discharges through R_B . Thus the duty cycle may be precisely set by the ratio of these two resistors.

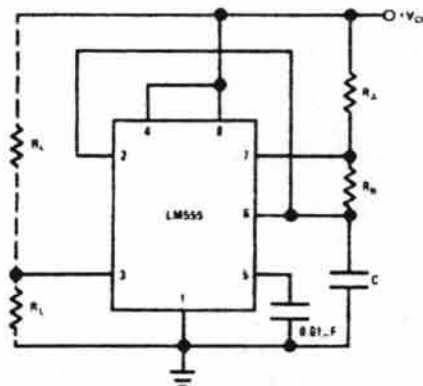
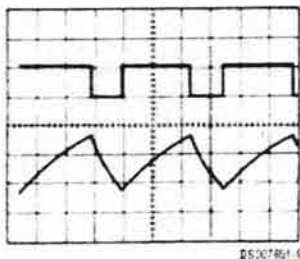


FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between $1/3 V_{CC}$ and $2/3 V_{CC}$. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

Applications Information (Continued)

Figure 5 shows the waveforms generated in this mode of operation.



$V_{CC} = 5V$
 TIME = 20 μ s/DIV
 $R_A = 3.9k\Omega$
 $R_B = 3k\Omega$
 $C = 0.01\mu F$

Top Trace: Output 5V/Div
 Bottom Trace: Capacitor Voltage 1V/Div

FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693 (R_A + R_B) C$$

And the discharge time (output low) by:

$$t_2 = 0.693 (R_B) C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693 (R_A + 2R_B) C$$

The frequency of oscillation is:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B) C}$$

Figure 6 may be used for quick determination of these RC values.

The duty cycle is:

$$D = \frac{R_B}{R_A + 2R_B}$$

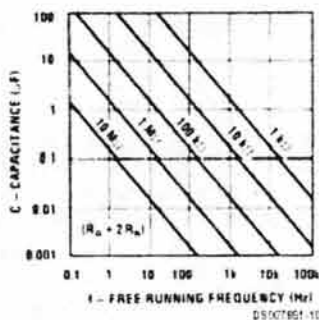
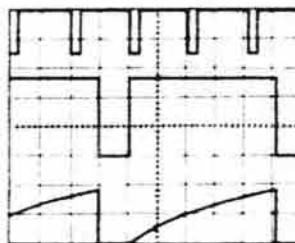


FIGURE 6. Free Running Frequency

FREQUENCY DIVIDER

The monostable circuit of Figure 1 can be used as a frequency divider by adjusting the length of the timing cycle. Figure 7 shows the waveforms generated in a divide by three circuit.



$V_{CC} = 5V$
 TIME = 20 μ s/DIV
 $R_A = 9.1k\Omega$
 $C = 0.01\mu F$

Top Trace: Input 4V/Div
 Middle Trace: Output 2V/Div
 Bottom Trace: Capacitor 2V/Div

FIGURE 7. Frequency Divider

PULSE WIDTH MODULATOR

When the timer is connected in the monostable mode and triggered with a continuous pulse train, the output pulse width can be modulated by a signal applied to pin 5. Figure 8 shows the circuit, and in Figure 9 are some waveform examples.

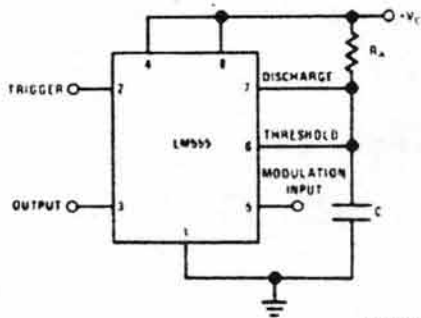
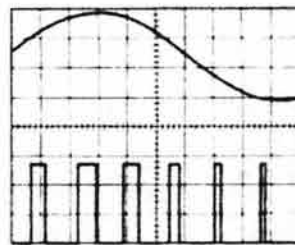


FIGURE 8. Pulse Width Modulator



$V_{CC} = 5V$
 TIME = 0.2 ms/DIV
 $R_A = 9.1k\Omega$
 $C = 0.01\mu F$

Top Trace: Modulation 1V/Div
 Bottom Trace: Output Voltage 2V/Div

FIGURE 9. Pulse Width Modulator

Applications Information (Continued)

PULSE POSITION MODULATOR

This application uses the timer connected for astable operation, as in Figure 10, with a modulating signal again applied to the control voltage terminal. The pulse position varies with the modulating signal, since the threshold voltage and hence the time delay is varied. Figure 11 shows the waveforms generated for a triangle wave modulation signal.

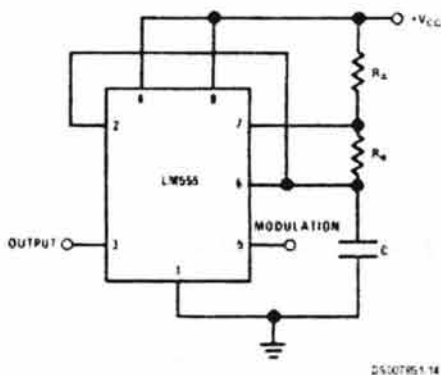
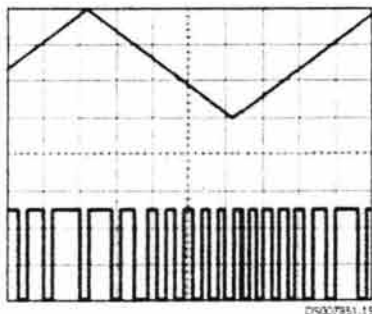


FIGURE 10. Pulse Position Modulator



$V_{CC} = 5V$
 TIME - 0.1 ms/DIV
 $R_A = 3.9k\Omega$
 $R_E = 3k\Omega$
 $C = 0.01\mu F$
 Top Trace: Modulation Input 1V/DIV
 Bottom Trace: Output 2V/DIV

FIGURE 11. Pulse Position Modulator

LINEAR RAMP

When the pullup resistor, R_A , in the monostable circuit is replaced by a constant current source, a linear ramp is generated. Figure 12 shows a circuit configuration that will perform this function.

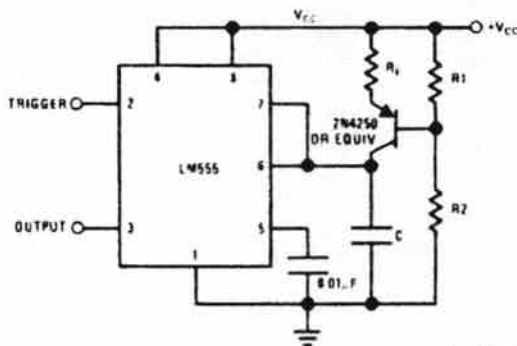
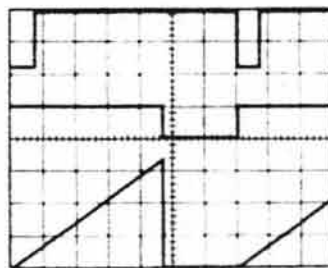


FIGURE 12.

Figure 13 shows waveforms generated by the linear ramp. The time interval is given by:

$$T = \frac{2/3 V_{CC} R_E (R_1 + R_2) C}{R_1 V_{CC} - V_{BE} (R_1 + R_2)}$$

$V_{BE} > 0.6V$
 $V_{BE} > 0.6V$



$V_{CC} = 5V$
 TIME - 20μs/DIV
 $R_1 = 47k\Omega$
 $R_2 = 100k\Omega$
 $R_E = 2.7k\Omega$
 $C = 0.01\mu F$
 Top Trace: Input 3V/DIV
 Middle Trace: Output 5V/DIV
 Bottom Trace: Capacitor Voltage 1V/DIV

FIGURE 13. Linear Ramp

Applications Information (Continued)

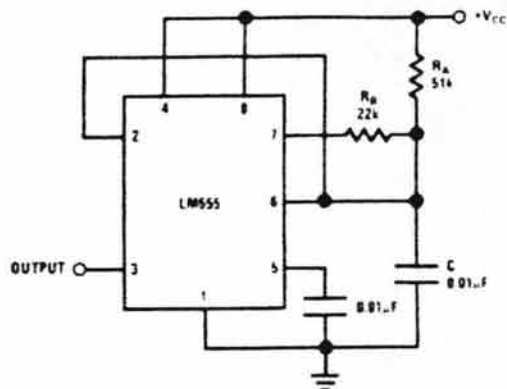
50% DUTY CYCLE OSCILLATOR

For a 50% duty cycle, the resistors R_A and R_B may be connected as in Figure 14. The time period for the output high is the same as previous. $t_1 = 0.693 R_A C$. For the output low it is $t_2 =$

$$\left[(R_A R_B) / (R_A + R_B) \right] C \ln \left[\frac{R_B - 2R_A}{2R_B - R_A} \right]$$

Thus the frequency of oscillation is

$$f = \frac{1}{t_1 + t_2}$$



2507661-18

FIGURE 14. 50% Duty Cycle Oscillator

Note that this circuit will not oscillate if R_B is greater than $1/2 R_A$ because the junction of R_A and R_B cannot bring pin 2 down to $1/3 V_{CC}$ and trigger the lower comparator.

ADDITIONAL INFORMATION

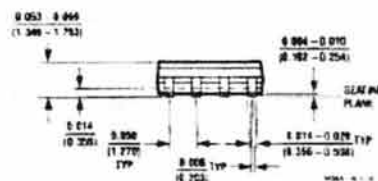
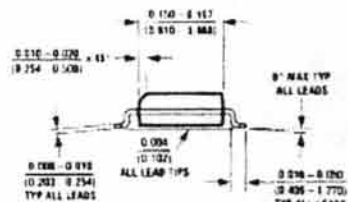
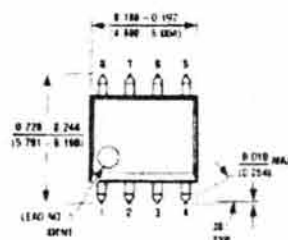
Adequate power supply bypassing is necessary to protect associated circuitry. Minimum recommended is $0.1\mu F$ in parallel with $1\mu F$ electrolytic.

Lower comparator storage time can be as long as $10\mu s$ when pin 2 is driven fully to ground for triggering. This limits the monostable pulse width to $10\mu s$ minimum.

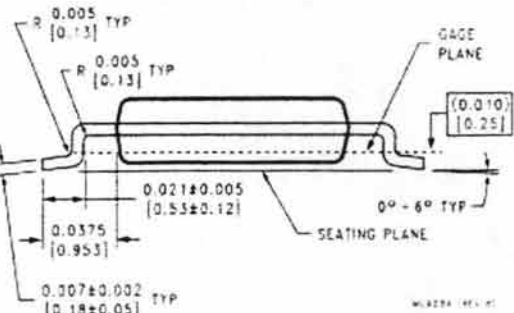
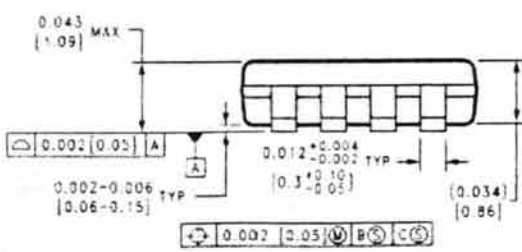
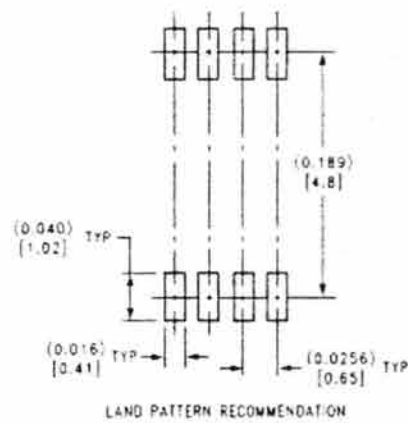
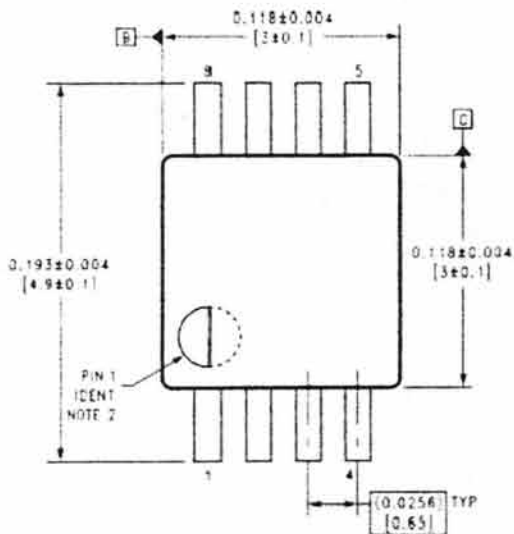
Delay time reset to output is $0.47\mu s$ typical. Minimum reset pulse width must be $0.3\mu s$ typical.

Pin 7 current switches within 30ns of the output (pin 3) voltage.

Physical Dimensions inches (millimeters) unless otherwise noted

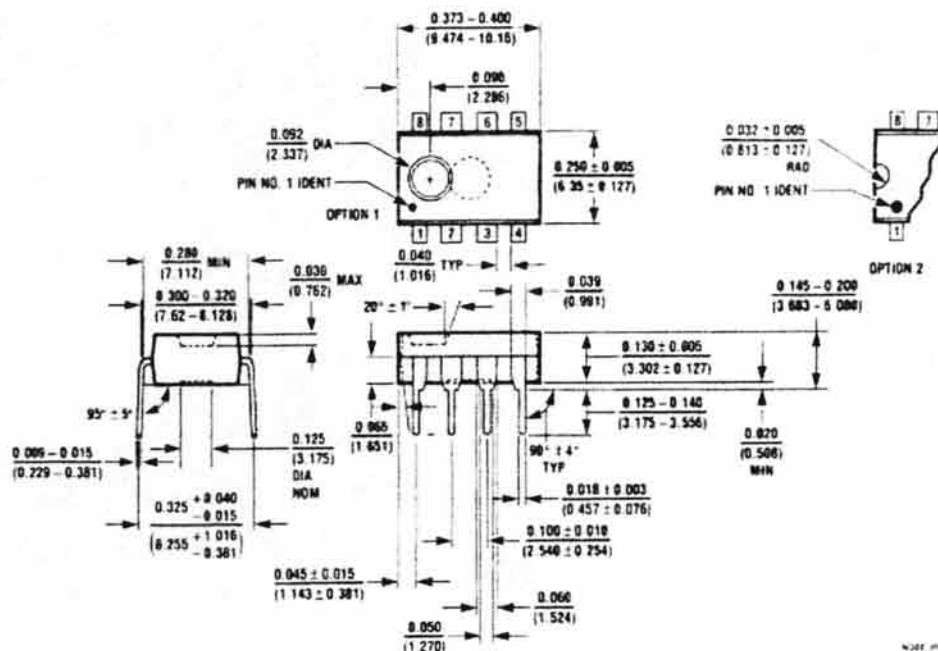


Small Outline Package (M)
NS Package Number M08A



8-Lead (0.118" Wide) Molded Mini Small Outline Package
NS Package Number MUA08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)




Moulded Dual-In-Line Package (N)
NS Package Number N08E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

 National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor
Europe
Fax: +49 (0) 180 530 65 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9608 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8799

National Semiconductor
Asia Pacific Customer Response Group
Tel: 65-2544666
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor
Japan Ltd.
Tel: 81-3-5639-7599
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuit described here; circuit patent licenses are implied and National reserves the right at any time without notice to change said circuit and specifications.

Anexo 7



August 1986
Revised March 2000

DM74LS73A

Dual Negative-Edge-Triggered Master-Slave J-K Flip-Flops with Clear and Complementary Outputs

General Description

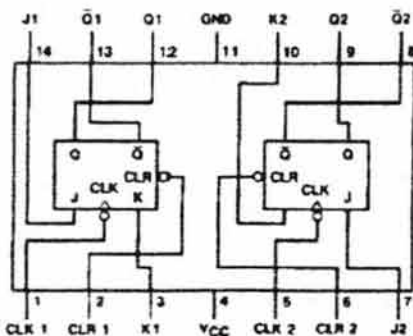
This device contains two independent negative-edge-triggered J-K flip-flops with complementary outputs. The J and K data is processed by the flip-flops on the falling edge of the clock pulse. The clock triggering occurs at a voltage level and is not directly related to the transition time of the negative going edge of the clock pulse. The data on the J and K inputs is allowed to change while the clock is HIGH or LOW without affecting the outputs as long as setup and hold times are not violated. A low logic level on the clear input will reset the outputs regardless of the levels of the other inputs.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS73AM	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS73AN	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

Inputs				Outputs	
CLR	CLK	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	↓	L	L	Q_0	\bar{Q}_0
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	Toggle	Toggle
H	H	X	X	Q_0	\bar{Q}_0

H - HIGH Logic Level
L - LOW Logic Level
X - Either LOW or HIGH Logic Level
↓ - Negative going edge of pulse
 Q_0 - The output logic level before the indicated input conditions were established
Toggle - Each output changes to the complement of its previous level on each falling edge of the clock pulse.

DM74LS73A Dual Negative-Edge-Triggered Master-Slave J-K Flip-Flops with Clear and Complementary Outputs

DM74LS73A

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0 °C to +70 °C
Storage Temperature Range	-65 °C to +150 °C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parameter values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _{IH}	HIGH Level Input Voltage	2			V
V _{IL}	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			8	mA
f _{CLK}	Clock Frequency (Note 2)	0		30	MHz
f _{CLK}	Clock Frequency (Note 3)	0		25	MHz
t _w	Pulse Width (Note 2)	Clock HIGH	20		ns
		Preset LOW	25		
		Clear LOW	25		
t _w	Pulse Width (Note 3)	Clock HIGH	25		ns
		Preset LOW	30		
		Clear LOW	30		
t _{SD}	Setup Time (Note 2)(Note 4)	20			ns
t _{SD}	Setup Time (Note 3)(Note 4)	25			ns
t _H	Hold Time (Note 2)(Note 4)	0			ns
t _H	Hold Time (Note 3)(Note 4)	5			ns
T _A	Free Air Operating Temperature	0		70	°C

Note 2: C_L = 15 pF, R_L = 2 kΩ, T_A = 25 °C and V_{CC} = 5V

Note 3: C_L = 50 pF, R_L = 2 kΩ, T_A = 25 °C and V_{CC} = 5V

Note 4: The symbol (↓) indicates the falling edge of the clock pulse is used for reference.

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 5)	Max	Units
V_I	Input Clamp Voltage	$V_{CC} - \text{Min. } I_I = -18 \text{ mA}$			-1.5	V
V_{OH}	HIGH Level Output Voltage	$V_{CC} - \text{Min. } I_{OH} - \text{Max}$ $V_{IL} - \text{Max. } V_{IH} - \text{Min}$	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	$V_{CC} - \text{Min. } I_{OL} - \text{Max}$ $V_{IL} - \text{Max. } V_{IH} - \text{Min}$ $I_{OL} = 4 \text{ mA, } V_{CC} - \text{Min}$		0.36 0.25	0.5 0.4	V
I_I	Input Current @ Max Input Voltage	$V_{CC} - \text{Max}$ $V_I = 7 \text{ V}$			0.1 0.3 0.4	mA
I_{IH}	HIGH Level Input Current	$V_{CC} - \text{Max}$ $V_I = 2.7 \text{ V}$			20 60 80	μA
I_{IL}	LOW Level Input Current	$V_{CC} - \text{Max}$ $V_I = 0.4 \text{ V}$			-0.4 -0.8 -0.8	mA
I_{OS}	Short Circuit Output Current	$V_{CC} - \text{Max (Note 6)}$	-20		-100	mA
I_{CC}	Supply Current	$V_{CC} - \text{Max (Note 7)}$		4	6	mA

Note 5: All typicals are at $V_{CC} = 5 \text{ V, } T_A = 25^\circ\text{C}$.

Note 6: Not more than one output should be shorted at a time, and the duration should not exceed one second. For devices with feedback from the outputs, where shorting the outputs to ground may cause the outputs to change logic state, an equivalent test may be performed where $V_{OS} = 2.125 \text{ V}$ with the minimum and maximum limits reduced by one half from their stated values. This is very useful when using automatic test equipment.

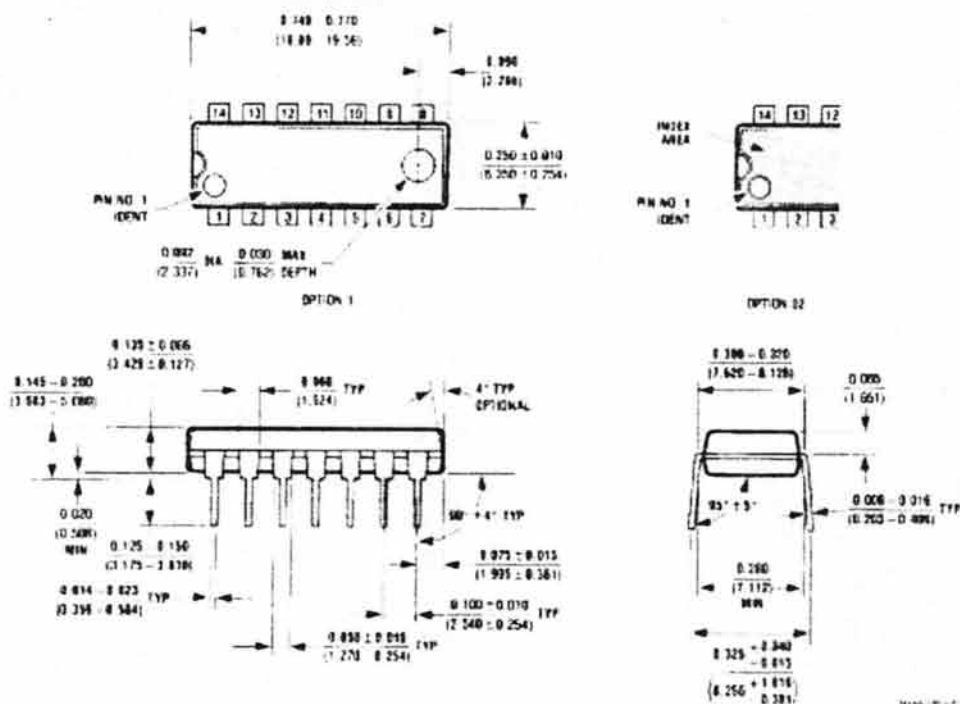
Note 7: With all outputs OPEN, I_{CC} is measured with the Q and \bar{Q} outputs HIGH in turn. At the time of measurement, the clock is grounded.

Switching Characteristics

at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$

Symbol	Parameter	From (Input) To (Output)	$R_L = 2 \text{ k}\Omega$				Units
			$C_L = 15 \text{ pF}$		$C_L = 50 \text{ pF}$		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		30		25		MHz
t_{PHL}	Propagation Delay Time HIGH to LOW Level Output	Clear to Q		20		28	ns
t_{PLH}	Propagation Delay Time LOW to HIGH Level Output	Clear to \bar{Q}		20		24	ns
t_{PLH}	Propagation Delay Time LOW to HIGH Level Output	Clock to Q or \bar{Q}		20		24	ns
t_{PHL}	Propagation Delay Time HIGH to LOW Level Output	Clock to Q or \bar{Q}		20		28	ns

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



14-Lead Plastic Dual-in-Line Package (PDIP), JEDEC MS-001, 0.300 Wide Package Number N14A

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

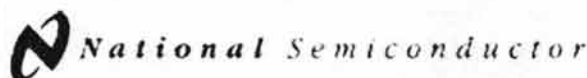
LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

Anexo 8



December 1994

LM35/LM35A/LM35C/LM35CA/LM35D

Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in $^{\circ}$ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^{\circ}$ C at room temperature and $\pm 3/4^{\circ}$ C over a full -55 to $+150^{\circ}$ C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μ A from its supply, it has very low self-heating, less than 0.1° C in still air. The LM35 is rated to operate over a -55° to $+150^{\circ}$ C temperature range, while the LM35C is rated for a -40° to $+110^{\circ}$ C range (-10° with improved accuracy). The LM35 series is

available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-202 package.

Features

- Calibrated directly in $^{\circ}$ Celsius (Centigrade)
- Linear $+10.0$ mV/ $^{\circ}$ C scale factor
- 0.5° C accuracy guaranteeable (at $+25^{\circ}$ C)
- Rated for full -55° to $+150^{\circ}$ C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μ A current drain
- Low self-heating, 0.08° C in still air
- Nonlinearity only $\pm 1/4^{\circ}$ C typical
- Low impedance output, 0.1Ω for 1 mA load

LM35/LM35A/LM35C/LM35CA/LM35D
Precision Centigrade Temperature Sensors

Connection Diagrams

TO-46
Metal Can Package*

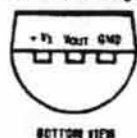


TU115516-1

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH,
LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

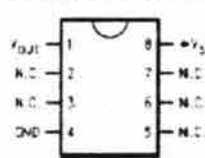
TO-92
Plastic Package



TU115516-2

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

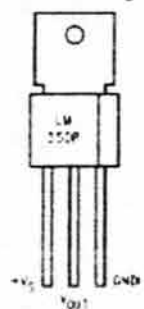
SO-8
Small Outline Molded Package



TU115516-21

Top View
N.C. = No Connection
Order Number LM35DM
See NS Package Number M08A

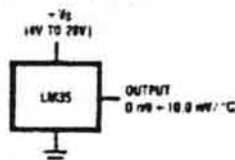
TO-202
Plastic Package



TU115516-24

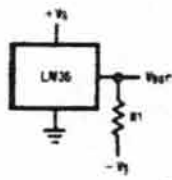
Order Number LM35DP
See NS Package Number P03A

Typical Applications



TU115516-3

FIGURE 1. Basic Centigrade
Temperature
Sensor ($+2^{\circ}$ C to $+150^{\circ}$ C)



TU115516-4

Choose $R_1 = -V_S/50 \mu$ A

$V_{OUT} = +1500$ mV at $+150^{\circ}$ C
 $= +250$ mV at $+25^{\circ}$ C
 $= -950$ mV at -55° C

FIGURE 2. Full-Range Centigrade
Temperature Sensor

LM35 is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp., TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-202 Package,	-65°C to +150°C

Lead Temp.:

TO-46 Package, (Soldering, 10 seconds)	300°C
TO-92 Package, (Soldering, 10 seconds)	260°C
TO-202 Package, (Soldering, 10 seconds)	+230°C

SO Package (Note 12):

Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V

Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)

LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics (Note 1) (Note 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	+0.2	+0.5		+0.2	+0.5		°C
	$T_A = -10^\circ\text{C}$	+0.3			+0.3		+1.0	°C
	$T_A = T_{MAX}$	+0.4	+1.0		+0.4	+1.0		°C
	$T_A = T_{MIN}$	+0.4	+1.0		+0.4		+1.5	°C
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	+0.18		+0.35	+0.15		+0.3	°C
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/°C
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	+0.4	+1.0		+0.4	+1.0		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	+0.5		+3.0	+0.5		+3.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	+0.01	+0.05		+0.01	+0.05		mV/V
	$4V \leq V_S \leq 30V$	+0.02		+0.1	+0.02		+0.1	mV/V
Quiescent Current (Note 9)	$V_S = +5V, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5V$	105		131	91		114	μA
	$V_S = +30V, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30V$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4V \leq V_S \leq 30V$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	μA/°C
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	$T_J = T_{MAX}$, for 1000 hours	+0.08			+0.08			°C

Note 1: Unless otherwise noted, these specifications apply: -55°C $T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; -40°C $T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA and 0°C $T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5V$ and $I_{LOAD} = 50 \mu\text{A}$ in the circuit of Figure 2. These specifications also apply from -2°C to T_{MAX} in the circuit of Figure 1. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W (junction to ambient) and 24°C/W (junction to case). Thermal resistance of the TO-92 package is 180°C/W (junction to ambient). Thermal resistance of the small outline molded package is 230°C/W (junction to ambient). Thermal resistance of the TO-202 package is 85°C/W (junction to ambient). For additional thermal resistance information see table in the Applications section.

Electrical Characteristics (Note 1) (Note 6) (Continued)								
Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	+0.4	+1.0		+0.4	+1.0		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	+0.5			+0.5		+1.5	$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$	+0.8	+1.5		+0.8		+1.5	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$	+0.8		+1.5	+0.8		+2.0	$^\circ\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				+0.6	+1.5		$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$				+0.9		+2.0	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$				+0.9		+2.0	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+0.3		+0.5	+0.2		+0.5	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.8, +10.2		+10.0		+9.8, +10.2	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	+0.4	+2.0		+0.4	+2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+0.5		+5.0	+0.5		+5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	+0.01	+0.1		+0.01	+0.1		mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	+0.02		+0.2	+0.02		+0.2	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	80		56	80		μA
	$V_S = +5\text{V}$	105		158	91		138	μA
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	82		56.2	82		μA
	$V_S = +30\text{V}$	105.5		161	91.5		141	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		3.0	0.5		3.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.7	+0.39		+0.7	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	+0.08			+0.08			$^\circ\text{C}$

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the normal dissipation by the thermal resistance.

Note 4: Tested limits are guaranteed and 100% tested in production.

Note 5: Design limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate shipping quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error or between the output voltage and 10mV/ $^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output voltage versus temperature curve from the best-fit straight line, over the device's rated temperature range.

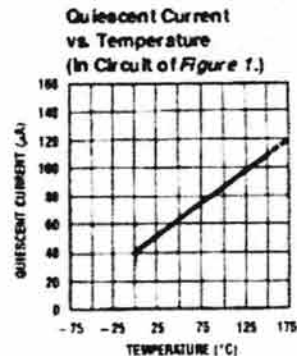
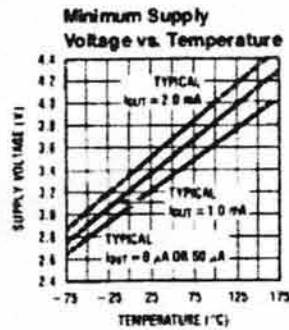
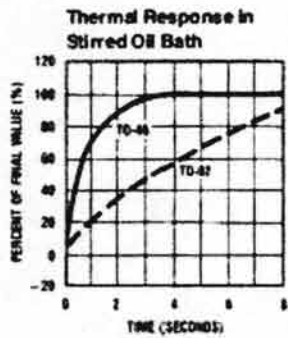
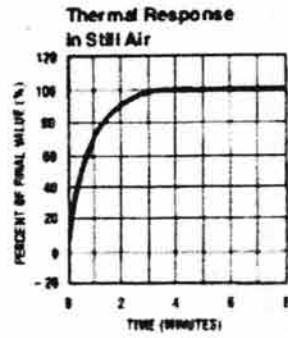
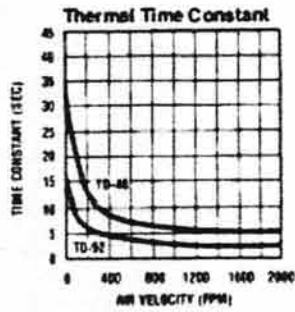
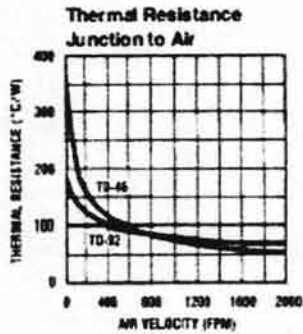
Note 9: Quiescent current is defined in the circuit of Figure 1.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

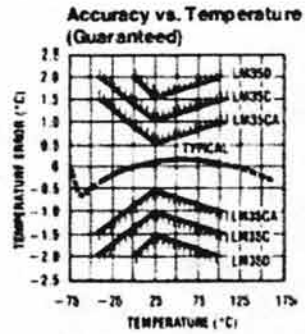
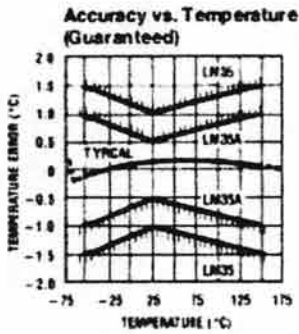
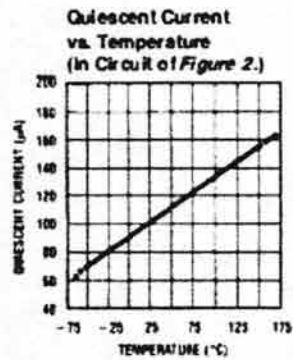
Note 11: Human body model: 100 pF discharged through a 1.5 k Ω resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

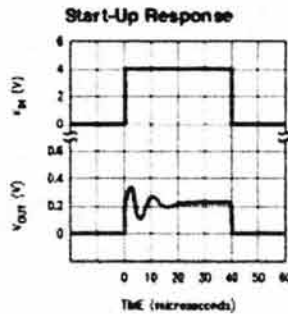
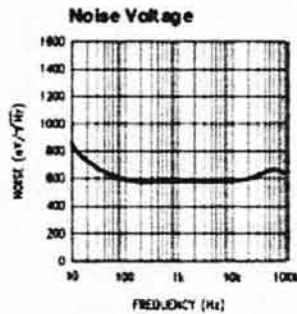
Typical Performance Characteristics



TLH/9516-17



TLH/9516-18



TL/9516-22

Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature, if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

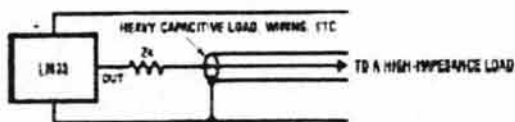
Temperature Rise of LM35 Due To Self-heating (Thermal Resistance)

	TO-46, no heat sink	TO-46, small heat fin*	TO-92, no heat sink	TO-92, small heat fin**	SO-8 no heat sink	SO-8 small heat fin**	TO-202 no heat sink	TO-202*** small heat fin
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	85°C/W	60°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	25°C/W	40°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W				
Stirred oil	50°C/W	30°C/W	45°C/W	40°C/W				
(Clamped to metal, infinite heat sink)	(24°C/W)				(55°C/W)		(23°C/W)	

* Washford type 201, or 1" dia of 0.020" sheet brass, soldered to case, or similar.

** TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications (Continued)



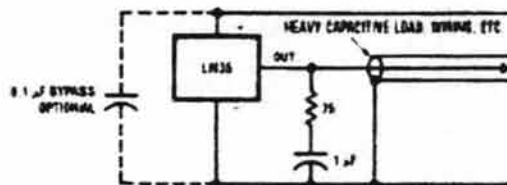
TLH/5516-19

FIGURE 3. LM35 with Decoupling from Capacitive Load

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor, see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground, see Figure 4.

When the LM35 is applied with a 200Ω load resistor as shown in Figure 5, 6, or 8, it is relatively immune to wiring

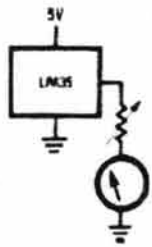


TLH/5516-20

FIGURE 4. LM35 with R-C Damper

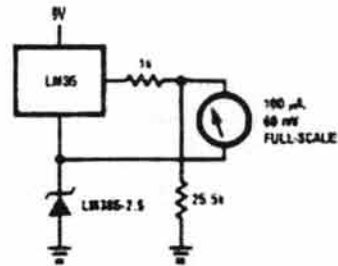
capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc., as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in Figures 13, 14, and 16.

Typical Applications (Continued)



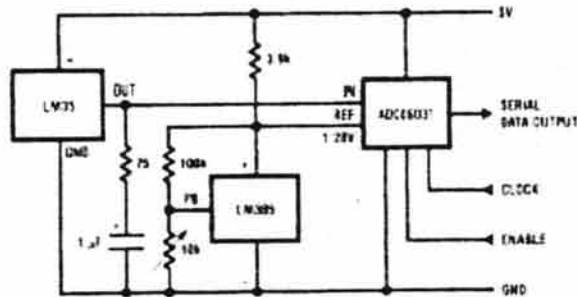
TLH/5576-11

FIGURE 11. Centigrade Thermometer (Analog Meter)



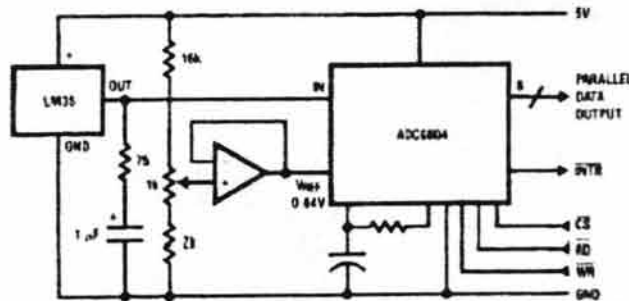
TLH/5576-12

FIGURE 12. Expanded Scale Thermometer (50° to 80° Fahrenheit, for Example Shown)



TLH/5576-13

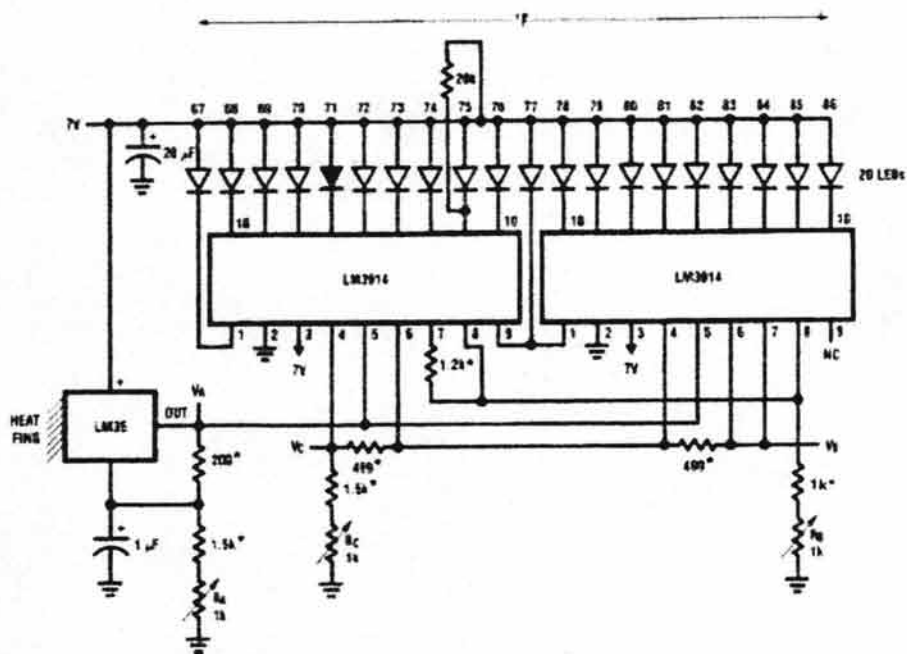
FIGURE 13. Temperature To Digital Converter (Serial Output) (+ 128° C Full Scale)



TLH/5576-14

FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE® Outputs for Standard Data Bus to μ P Interface) (128° C Full Scale)

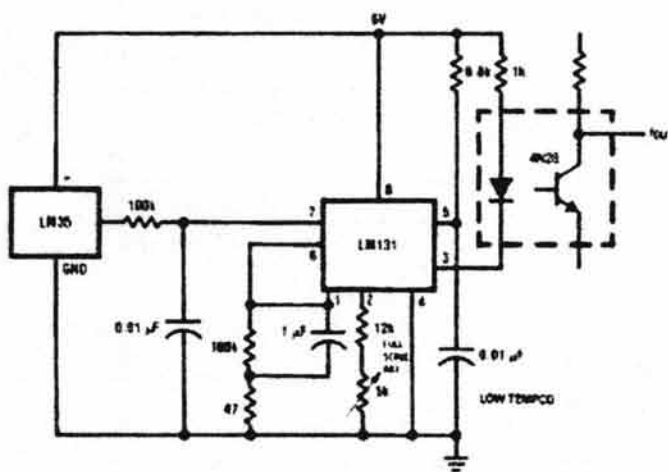
Typical Applications (Continued)



TLU 1125 16-10

- * 1% or 2% film resistor
- Trim R_{10} for $V_{10} = 3.075V$
- Trim R_{11} for $V_{11} = 1.995V$
- Trim R_{12} for $V_{12} = 1.995V$
- Trim R_{13} for $V_{13} = 0.075V - 100mV/°C - T_{ambient}$
- Example: $V_{13} = 2.275V$ at $22°C$

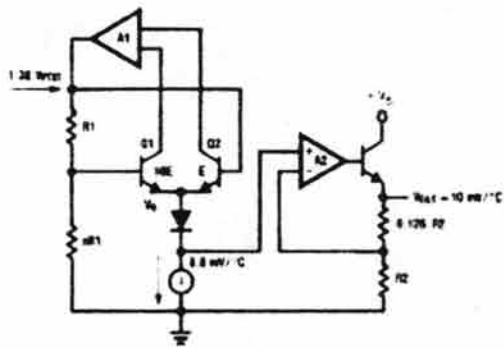
FIGURE 15. Bar-Graph Temperature Display (Dot Mode)



TLU 1125 16-11

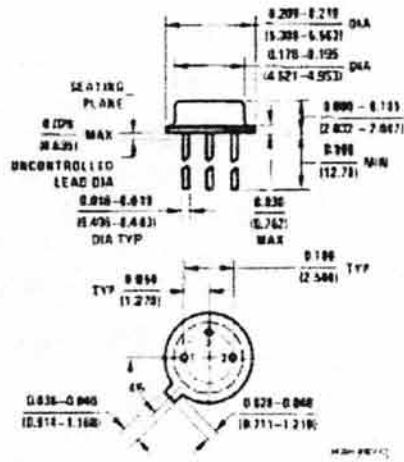
FIGURE 16. LM35 With Voltage-To-Frequency Converter And Isolated Output (2°C to +150°C; 20 Hz to 1500 Hz)

Block Diagram

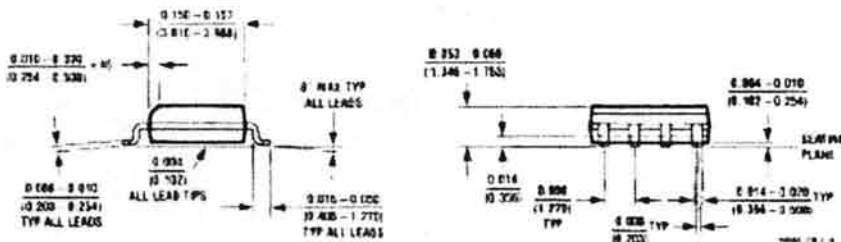
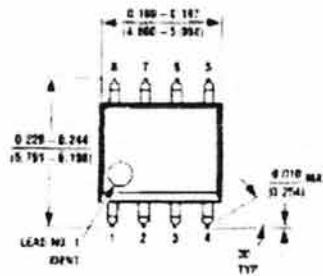


IL-14-125-16-21

Physical Dimensions inches (millimeters)

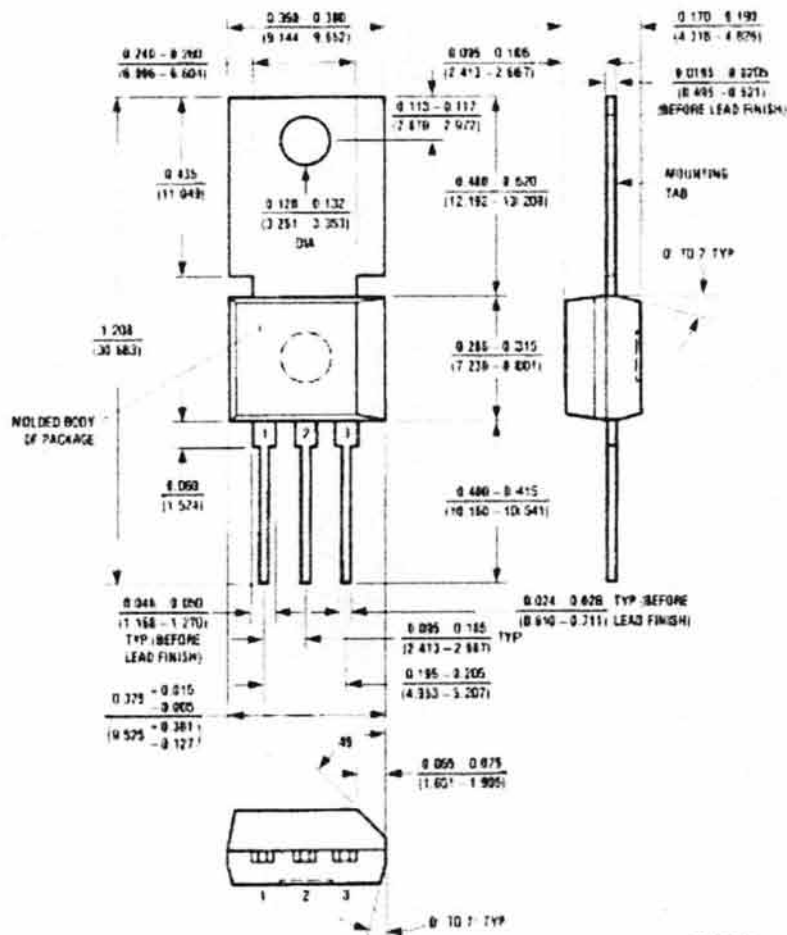


TO-46 Metal Can Package (H)
 Order Number LM35H, LM35AH, LM36CH,
 LM35CAH, or LM35DH
 NS Package Number H03H



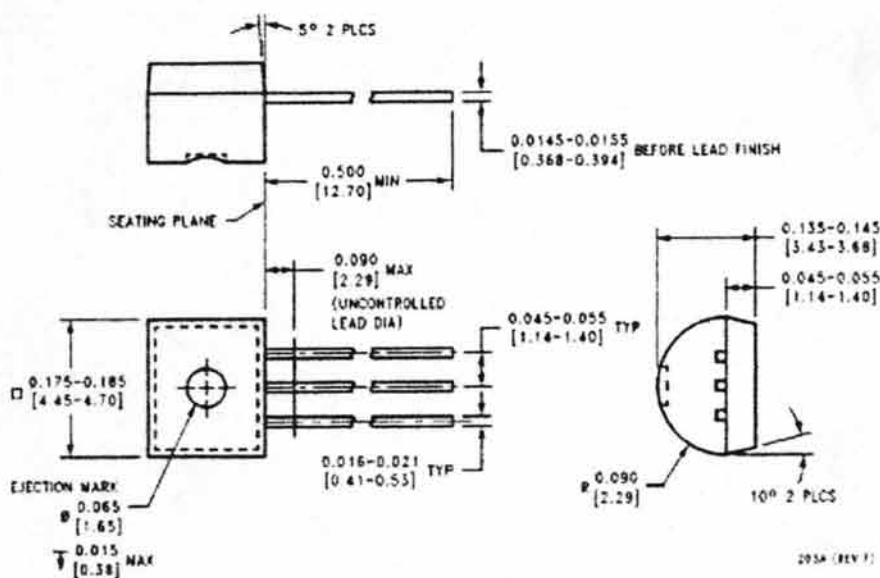
SO-8 Molded Small Outline Package (M)
 Order Number LM35DM
 NS Package Number M06A

Physical Dimensions inches (millimeters) (Continued)



Power Package TO-202 (P)
Order Number LM35DP
NS Package Number P03A

Physical Dimensions inches (millimeters) (Continued)



TO-92 Plastic Package (Z)
Order Number LM35CZ, LM35CAZ or LM35DZ
NS Package Number Z03A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which: (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

<p>National Semiconductor Corporation 2000 Semiconductor Drive P.O. Box 58060 Santa Clara, CA 95052-0000 Tel: (408) 735-2000 TWX: (910) 334-4440</p>	<p>National Semiconductor GmbH Lery-Gesler-Str. 10 D-82026 Planseebruck Germany Tel: (9141) 35-0 Telex: 527992 Fax: (9141) 35-1</p>	<p>National Semiconductor Japan Ltd. Suntory Building Engineering Center 5th Fl. 1-7-1, Fukuoka, Minami-Ku Chikuoh-cho, Chikuoh-ku, Fukuoka 815 Tel: (042) 234-2000 Fax: (042) 234-2000</p>	<p>National Semiconductor Hong Kong Ltd. 12th Floor, Straight Street Overseas Centre, 2 Canton Road Tsimshatsui, Kowloon Hong Kong Tel: (852) 2737-1600 Fax: (852) 2735-9900</p>	<p>National Semiconductor de México, S.A. de C.V. Rue Desoladoro Lázaro Franco 125-A San Pedro de las Cabañas C.P. 40180-000 Tel: (52-15) 212-6000 Telex: 3014131231 12288 BR Fax: (52-15) 212-1161</p>	<p>National Semiconductor (Australia) Pty. Ltd. Building 10 Burrumbeet Park Drive Melbourne 3106 Australia Tel: (61-3) 958-0000 Fax: (61-3) 958-0246</p>
---	--	--	---	--	---

National does not assume any responsibility for use of any circuits described in this document where an implied and National reserves the right to change specifications and specifications.