

Contenido

3

Diseño de un Termómetro Digital para Análisis Clínico en Animales de Trabajo

José Rodolfo Martínez y Cárdenas, Jaime Díaz Altamirano, Fidel Diego Nava (CIDIIR-OAXACA)

Interfaces para el Puerto Paralelo de la PC, en Modo Bidireccional

M. en C. Juan Carlos Herrera Lozada, M en C. Agustín Cruz Contreras, M. en C. Juan Carlos González Robles (CIDETEC-IPN)

9

17

Tecnologías xDSL

Ing. Patricia Pérez Romero (CIDETEC-IPN).

Programación de Sistemas de Cómputo Móviles

M. en C. Juan Carlos Herrera Lozada, M. en C. Israel Rivera Zárate (CIDETEC - IPN);
Rodrigo Morales Alvarado (Alumno de la Maestría en Tecnología de Cómputo del CIDETEC - IPN)

23

30

Algoritmo de Escape para Crear un Fractal Utilizando MPI

Ing. Jesús Antonio Álvarez Cedillo. (CIDETEC - IPN).

Implementación de Hardware / Software para Control Difuso

M. en C. Romeo Urbieto Parrazales, M. en C. Pablo Manrique Ramírez,
Ing. Antonio Hernández Zavala (CIC-IPN).

38

Oxímetro de Pulso Basado en una Palm Parte I: Fundamentos

45

M. en C. Israel Rivera Zárate, M. en C. Juan Carlos Herrera Lozada (CIDETEC-IPN);
Victor Jalil Ochoa (Estudiante UPIICSA-IPN).

Una Técnica para la Localización de Ojos Humanos en una Imagen Bidimensional

Luna Esteban Bruce Leroy, Mendoza Campa Juan Manuel,
Parra Bautista Roberto (ESIME Culhuacan - IPN); M. en C. Aguilar Jáuregui María Elena (CIC-IPN).

49

Diseño de un Termómetro Digital para Análisis Clínico en Animales de Labor

**José Rodolfo Martínez y Cárdenas,
Jaime Díaz Altamirano,
Fidel Diego Nava**
rodolfo_myc@hotmail.com
diaz_aj@engineer.com
fdiego@ipn.mx
Profesores del CIIDIR Oaxaca-IPN

Los métodos empleados en el sureste mexicano para la producción en el campo son obsoletos; no es conveniente introducir maquinaria pesada, por razones económicas y orográficas, lo que obliga a realizar un refinamiento y rediseño de las técnicas y herramientas actualmente empleadas, para poder mejorar la producción. El CIIDIR Oaxaca ha implementado un proyecto cuyo objetivo es el diseño y evaluación de un apero de tracción animal, para mejorar la labranza, conservando fértil y útil la tierra para siembras posteriores. Para evaluar el desempeño del apero se requieren instrumentos de medición que valoren el esfuerzo de los animales de tiro, monitoreando su temperatura corporal y otras variables físicas. El equipo apropiado es inexistente o escaso, y muchas veces sólo adquirible en el extranjero. Para poder medir la temperatura de los animales de tiro, mientras trabajan arrastrando un apero, se desarrolló un sistema de tele termometría que, de manera inalámbrica, puede monitorear la temperatura corporal de las bestias empleadas para el arrastre. Dicho sistema está constituido por dos partes: la primera contiene al sensor, un acondicionador de señal, un convertidor analógico-digi-

tal, un microcontrolador y un transmisor de RF. En esta etapa el sensor toma la temperatura de la oreja del animal, y entrega una señal eléctrica proporcional a ésta, la cuál es digitalizada, procesada y enviada vía RF a la segunda sección, que contiene, un receptor de RF, un microcontrolador, y un Despliegue de Cristal Líquido (LCD). En esta sección se recibe y procesa la información enviada por la primera etapa, para finalmente desplegar el dato en el LCD.

INTRODUCCIÓN

Uno de los sectores económicos de México que más necesitan de tecnologías propias es la agricultura de los estados del sureste del país, en donde aún se aplican con éxito relativo los arados de madera de tracción animal. Desarrollados por los egipcios en la antigüedad y los europeos durante la edad media, no permiten un aprovechamiento máximo de la potencia de los animales de trabajo.

La renovación de las técnicas de labranza de estos campesinos no significa que se deba planear la introducción de tractores y maquinaria pesada, pues la economía y orografía del estado no lo permiten; es por ello que en el CIIDIR-Oaxaca, se realizó el proyecto "Desarrollo y evaluación de un apero de tracción animal para labranza de conservación", el cual tiene como objetivos el mejo-

ramiento de la técnica de labranza de los campesinos para elevar la eficiencia de los animales de trabajo y, al mismo tiempo, permitir la conservación de la fertilidad y utilidad del suelo de siembra. Para tal efecto se necesita evaluar las labranzas con el arado de madera y el apero diseñado, con el fin de comparar la respuesta de los animales de trabajo. En ambos casos se debe vigilar que el esfuerzo realizado por los animales de tiro al arrastrar el artefacto sea moderado, y no les provoque agotamiento excesivo o lesiones cardiovasculares. Para ello, se hacen necesarios el monitoreo y vigilancia de las principales variables fisiológicas de los animales en plena actividad, tales como: ritmo cardíaco, ritmo respiratorio y temperatura corporal.

Para lograr estas mediciones no se pueden emplear las técnicas convencionales utilizadas por los veterinarios, puesto que estas exigen que el animal se encuentre estático. Tampoco se pueden usar para mediciones continuas porque se puede alterar el comportamiento del animal. Pero no existe en el mercado nacional un instrumento adecuado a las necesidades que plantea la medición de dichas variables; en respuesta a esa falta de instrumentación, se diseñó un sistema de teletermetría de temperatura.

El proyecto tiene la finalidad de desarrollar un dispositivo que, mediante un sensor de temperatura,

permita monitorear a distancia la temperatura corporal de los animales de labranza, mientras éstos arrastran un apero. Estas mediciones tienen la finalidad de indicar qué tan grande es el esfuerzo realizado por las bestias al arrastrar determinado modelo de apero. Estos datos permitirán rediseñar el apero, con el fin de buscar aquel que proporcione un mejor rendimiento.

OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL

Desarrollar un sistema automático que permita realizar mediciones, a distancia y de manera inalámbrica, de la temperatura corporal de los animales de labranza mientras arrastran un apero, con el fin de registrar dichas mediciones para su uso posterior.

OBJETIVOS ESPECÍFICOS

- Crear un instrumento de medición de temperatura, útil y de costo razonable.
- Realizar un sistema inalámbrico de transmisión de datos, para poder conocer a distancia la temperatura registrada por el instrumento de medición.
- Diseñar el sistema para que pueda ser adaptado para monitorear otras variables (ritmo cardíaco, ritmo respiratorio, etc.), usando como base el medidor de temperatura.

MATERIALES Y PROCEDIMIENTOS

Los mamíferos son los vertebrados de organización más elevada. No obstante la variabilidad en su

forma, organización y género de vida, que dan origen al establecimiento de grupos u órdenes muy distintos unos de otros, estos animales pueden caracterizarse por tres rasgos fundamentales: la piel revestida de pelos, la presencia de glándulas mamarias, y la constancia de la temperatura de su cuerpo. Un cambio en la temperatura cambia también el carácter de procesos biológicos más complejos. Esto hace que una temperatura relativamente constante sea una necesidad para el eficaz funcionamiento del complicado encéfalo de los animales superiores (aves y mamíferos) adultos.

Durante la evolución, los animales superiores han desarrollado un dispositivo termorregulador que los capacita para mantener bajo condiciones ordinarias una temperatura orgánica interior constante, independientemente de la temperatura del ambiente. Estos animales, *homeotermos* o de sangre caliente, son capaces de llevar a cabo sus actividades usuales bajo límites amplios de temperatura externas, mientras que los animales *poiquilotermos* o de sangre fría, cuyas temperaturas varían directamente con las del ambiente, dependen totalmente de la temperatura externa.

Los mamíferos muy jóvenes en los que la función integradora del encéfalo no se ha desarrollado todavía, muestran una gran tolerancia a los cambios de temperatura orgánica. Esto parece confirmar la idea de que es la complejidad de organización lo que hace necesaria la homeotermia.

Muchas son las condiciones que pueden **causar variaciones** normales en la temperatura orgánica de los **homeotermos**; entre otras, están la edad, el sexo, la estación, la hora del día, la temperatura ambiente, **el ejercicio**, la ingestión de alimentos, la digestión y la bebida de agua.

En el hombre y los animales existen gradientes de temperatura en la sangre, los tejidos y el recto; las temperaturas son más bajas hacia el exterior del cuerpo. También existen variaciones considerables en las diferentes partes del núcleo central y partes periféricas del cuerpo, por lo que los miembros pueden estar a 10°C o más por debajo de la temperatura central.

En los animales se obtiene con mayor facilidad un índice de la temperatura orgánica profunda por inserción de un termómetro en el recto. Aunque la temperatura rectal no representa siempre un promedio de la temperatura orgánica profunda, generalmente deberá considerarse mejor medir la temperatura en este sitio, que utilizar diversos puntos.

La temperatura rectal en los animales domésticos más comunes se encuentra dentro de un rango reducido que va desde los 37.5°C hasta los 41°C en promedio. Es evidente que sería muy poco práctico tratar de tomar la temperatura rectal a un espécimen que se encuentra en plena actividad física, que es como estarán los animales que se deberán monitorear en este proyecto: el sensor no podría hacer un buen contacto con la mucosa rectal del animal, dando lecturas erróneas; además, el cuerpo extraño activaría un reflejo excretorio en el animal.

Por otra parte, la temperatura no debe tomarse sobre ningún músculo del animal durante la actividad física de éste, pues la temperatura del músculo tiene la tendencia a elevarse considerablemente, reduciendo la fiabilidad de las mediciones obtenidas de este modo. La temperatura podría tomarse también de manera oral. El único inconveniente de hacerlo de este modo, es que sería muy difícil mantener el sensor dentro del hocico del animal sin que éste

mastique los cables, escupa o trague el dispositivo.

Es bien sabido que en la mayor parte de los mamíferos las orejas, además de cumplir con la función auditiva, son un elemento de gran importancia en el mecanismo de control de temperatura corporal de los animales. Dicho en otras palabras, las orejas son una ventana a la temperatura corporal del animal y, por lo tanto, una buena opción para tomar las mediciones necesarias. Además, la medición de la temperatura en las orejas del animal, si bien no es tan precisa, es una alternativa a la medición de la temperatura rectal. Tomando esta idea, se determinó que el sensor se colocaría en una oreja del espécimen.

En cuanto al sensor a utilizar, se deberá utilizar uno de los llamados "sensores eléctricos", es decir, que su funcionamiento se basa en las propiedades eléctricas de los materiales de que están fabricados.

La banda de temperaturas a medir en el proyecto está determinada por las temperaturas corporales de los animales de interés; el rango es realmente estrecho, lo cual hace adecuado el uso de un sensor basado en semiconductores. El más adecuado es el denominado sensor de circuito integrado: los sensores de circuitos integrados son muy lineales, ofrecen altos niveles de rendimiento, son relativamente económicos y bastante precisos en temperaturas cercanas a la temperatura ambiente, aunque requieren de una fuente de alimentación.

Por sus características, el sensor seleccionado fue el LM35D, ya que es fácil de adquirir en el mercado nacional.

Este sensor entrega a su salida un voltaje proporcional a la temperatura

que registra, a una razón de 10mV por cada grado Celsius, teniendo por lo tanto una relación lineal de voltaje y temperatura; la gráfica es una línea recta. La ecuación que describe el comportamiento del sensor con respecto a la temperatura es:

$$V_s = T_c(10mV)$$

En donde V_s es la salida del sensor, en milivolts, y T_c es la temperatura en grados Celsius. El LM35 es capaz de medir temperaturas que van de los -55°C , hasta los $+150^{\circ}\text{C}$, como un rango máximo de operación.

Una de las mayores ventajas del LM35 es que sus conexiones son extremadamente sencillas, además de ser muy versátil, pues puede ser configurado de distintas maneras. En este caso, como el rango de temperatura a medir es relativamente pequeño, se eligió una configuración sencilla: las terminales de alimentación y tierra se conectan a la fuente de manera directa, y la terminal de la salida de la señal, se conecta a tierra por medio de una resistencia tal que la corriente que circula por ella debido al voltaje de alimentación sea de 50 microamperes.

Cuando se trabaja con sensores, es muy común que la información que envía el sensor sea demasiado débil para manipularla en las siguientes etapas, por lo que se debe de **condicionar** para ello. Para la implementación del acondicionamiento de señal, lo más adecuado es el uso de **amplificadores operacionales** (circuitos integrados lineales), los cuales facilitan y hacen más efectivo el proceso de acondicionamiento; el diseño de circuitos con amplificadores operacionales presenta grandes bondades, pues permite obtener sistemas estables, con buena fidelidad

a la salida, que consumen poca energía y ocupan poco espacio; además, no son difíciles de conseguir en el mercado, y su costo es bajo. Hay modelos de amplificadores operacionales que se han especializado y se les llama **amplificadores de instrumentación**. Estos tienen básicamente las mismas características de los operacionales en general, pero son capaces de detectar y manipular señales muy pequeñas. La única desventaja del uso de amplificadores operacionales en circuitos portátiles es que exigen el uso de más de una pila, o bien la construcción de un simetrizador de fuente. Para el acondicionamiento de la señal fue elegido el amplificador operacional cuádruple TL064.

Se sabe que el rango de temperaturas a medir va de 37.5°C a 43°C . A este rango se le debe agregar una tolerancia mínima y máxima. Dadas las características de la circuitería a utilizar (Microcontroladores con datos de 8 bits), es conveniente que el rango no rebase una temp. de 25.5°C ; es por esto que se decidió que la temperatura mínima manejable para el sistema fuera de 30.0°C , y la máxima, de 55.5°C , lo que deja un margen de tolerancia inferior de 7.5°C , y un margen superior de 12.5°C ; de esto, se deduce que los voltajes máximo y mínimo que se manejarán serán:

$$V_{30^{\circ}\text{C}} = 30^{\circ}\text{C}(10mV/^{\circ}\text{C}) = 300mV$$

$$V_{55.5^{\circ}\text{C}} = 55.5^{\circ}\text{C}(10mV/^{\circ}\text{C}) = 555mV$$

$$\Delta V = (55.5^{\circ}\text{C} - 30^{\circ}\text{C})(10mV/^{\circ}\text{C}) \\ = 25.5^{\circ}\text{C}(10mV/^{\circ}\text{C}) = 255mV$$

El convertidor Analógico-Digital, por su parte, será ajustado para que a la entrada un voltaje de cero volts corresponda a un valor de cero a la salida (000000000₂₇), y para un valor igual al voltaje positivo de alimentación del PIC16F874, de un valor de 1023 (111111111₂).

MICROCONTROLADORES Y ELECTRÓNICA DIGITAL

El uso de señales digitales permite realizar procesos complejos con la información, en los cuales se produce un mínimo de errores, y es por eso que en este caso se debe digitalizar la señal acondicionada. Para lograrlo se requiere un **Convertidor Analógico Digital (ADC)**; los ADC más comunes entregan a su salida números compuestos por 8 bits (un Byte), con lo cual dan un total de 256 estados posibles. Se usará el ADC interno del PIC16F874, de esta manera el PIC realizará dos funciones: digitalización y procesamiento de información, volviéndose el sistema más compacto y menos propenso a errores.

Dicho ADC empleará como voltajes de referencia los voltajes de alimentación del PIC (0-5V). Debe hacerse notar que este ADC entrega a la salida números de 10 bits; en el proyecto se utilizarán sólo los 8 bits menos significativos, por lo que la salida del acondicionador de señal correspondiente a 55.5V debe de ser mucho menor a 5V.

Los cálculos para dicho voltaje son:

$$\frac{5V}{1023} = \frac{xV}{255}$$

$$xV = \frac{5V(255)}{1023} = 1.2463343V$$

Para que el sistema proporcione siempre lecturas correctas, la alimentación debe estabilizarse con un regulador de voltaje, pues de otra forma, el voltaje de referencia superior sería inestable.

Cuando el ADC ha realizado su función, el PIC16F874 inicia una

transmisión serial de los 8 bits menos significativos del resultado obtenido, siendo éstos anteceditos por un doble bit de inicio.

Para realizar la transmisión serial inalámbrica se usaron módulos de RF para el enlace. Los módulos usados son:

Transmisor: TXM-433-LC-R
Receptor: RXM-433-LC-S

Estos módulos tienen una modulación ASK; a pesar de esto, al tener una transmisión digital se obtiene una buena calidad en el enlace inalámbrico, y un alcance de aproximadamente 40 metros. Los módulos de RF usados también son muy flexibles en cuanto a la velocidad de transmisión, pues pueden ser usados con velocidades que van desde unos cuantos bits hasta 5 kilobits por segundo.

El receptor ASK detecta los bits enviados por el transmisor y los entrega en forma de niveles de voltaje (dos niveles lógicos), pues es una transmisión digital. Un PIC16F84 los procesará para visualizar la información sobre un Despliegue de Cristal Líquido, modelo **DMC16249** de la marca **Optrex**. Generalmente, los despliegues de este tipo tienen un microcontrolador del tipo HD44780, de la marca japonesa Hitachi, o uno muy similar, y ésta no es la excepción.

El PIC16F84 realizará el proceso

de la información para su posterior visualización de esta. Se usará este modelo de PIC por ser uno de los más económicos y manejables; el número de terminales destinadas a entrada/salida, es ideal para la función que realizará. El PIC16F84 debe ser programado para que, cuando detecte en bit de inicio en la terminal usada como entrada serial, inicie un proceso de captura de los 8 bits portadores de información. Después de haber obtenido esta información, lo único restante es mandar a imprimir los caracteres; para ello, se emplea la última etapa del programa del PIC16F84. Es necesario incluir en el programa rutinas de control para el LCD; estas rutinas deben de gobernar las acciones del LCD por medio de las tres líneas de control de éste.

El LCD usado en este proyecto es uno de los más comunes, y no es difícil de conseguir en el mercado nacional; se trata de un LCD de caracteres matriciales, con dos líneas y 16 caracteres por línea.

DIAGRAMAS GENERALES

A continuación se anexan los diagramas generales, con sus valores y características, correspondientes al proyecto tratado en este informe. La **Figura 1** es el diagrama a bloques del sistema en su totalidad; el **Diagrama 1** corresponde a la parte del sistema que realiza la medi-

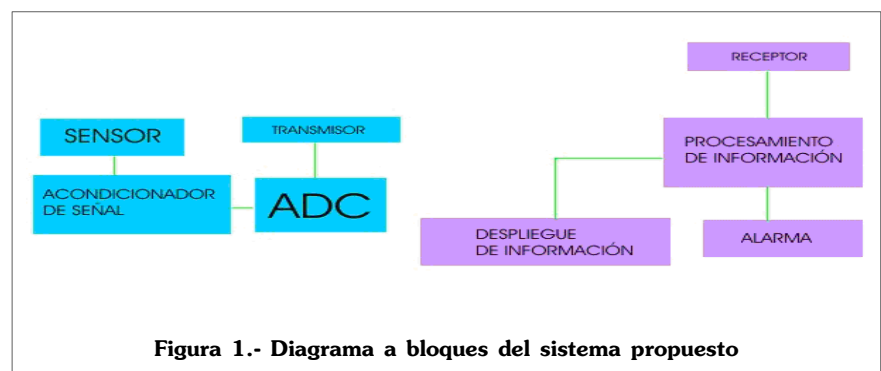
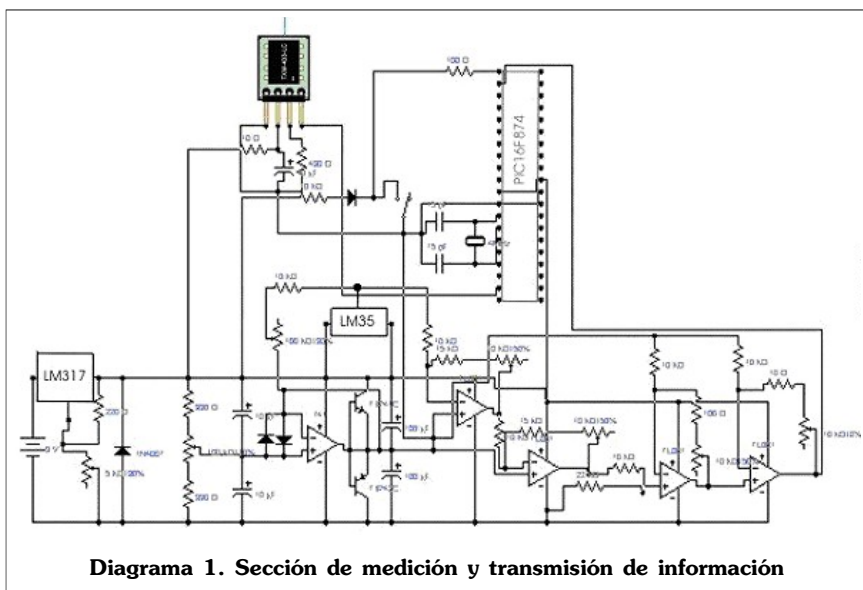


Figura 1.- Diagrama a bloques del sistema propuesto



ción y la transmisión de la información, y el **Diagrama 2** muestra al subsistema encargado de recibir los datos transmitidos y desplegar la información.

CONCLUSIONES

1°.- El sistema desarrollado en el presente proyecto se apega a las necesidades reales.

2°.- El desarrollo de un sistema de la naturaleza del aquí descrito, requiere de una inversión regular, debido a la importación de los módulos de RF empleados, que no se pueden adquirir en el mercado nacional. Si éstos fueran comerciales en el país, el costo total del sistema sería muy bajo, especialmente si se le compara con el costo de un sistema similar de presentación comercial.

3º.- El sistema descrito a lo largo del presente reporte, es funcional en el trabajo de campo, y cuenta con un gran número de posibles aplicaciones adicionales a aquella para la cuál fué creado; además, es fácilmente adaptable a otras situaciones que requieran modificarlo. Una de sus mejores características, es que con

una reducida inversión se puede ampliar su funcionalidad, para emplearlo en más de una aplicación a la vez.

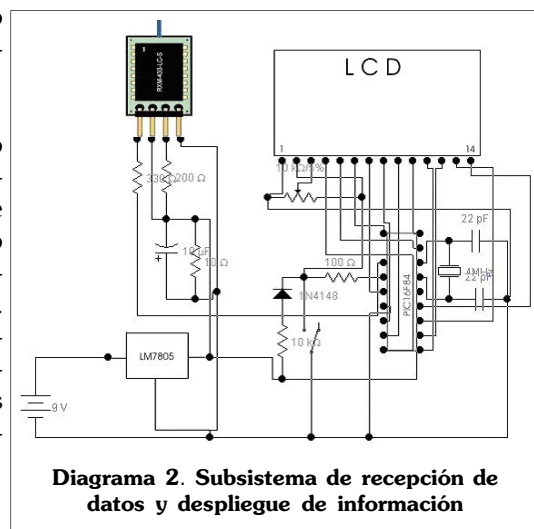
Con base en las experiencias personales adquiridas a lo largo de la realización del presente proyecto, y observando las carencias generalizadas en los laboratorios escolares y de instituciones de investigación, se recomienda que:

1º.- Las instituciones de educación superior e investigación adquieran el equipo necesario para apoyar el trabajo que se realice en sus laboratorios, pues la carencia de éstos impide el correcto desarrollo de las investigaciones y prácticas escolares y profesionales.

2º.- Se implemente el apoyo económico a los residentes e investigadores para la compra de materiales, en especial cuando éstos sean de alto costo y/o deban ser adquiridos fuera del país. Asimismo, que se ayude al investigador para que, con el respaldo de la institución le sea más fácil la adquisición de materiales de importación.

3º.- Se hagan llegar los resultados de las investigaciones, sean prototipos o conocimientos, a aquellos a los que sea de utilidad, para que de este modo el trabajo realizado tenga continuidad y utilidad.

4º.- Se instituyan en los centros de investigación y las escuelas superiores, áreas especializadas en el desarrollo de equipos de instrumentación para su empleo en las actividades científicas y/o didácticas de la misma institución.



BIBLIOGRAFÍA

- [1] Angulo Usategui, José Ma. y Co. *Microcontroladores PIC 1ª parte*. 2ª ed. México. Ed. Mc. Graw-Hill. 2000.
- [2] Angulo Usategui, José Ma. y Co. *Microcontroladores PIC 2ª parte*. 3ª ed. México. Ed. Mc. Graw-Hill. 2000.
- [3] Boylestad, Robert L. y Co. *Electrónica: teoría de circuitos*. 6ª Ed. México. Ed. Prentice Hall. 1997.
- [4] Coughlin, Robert F. y Co. *Amplificadores operacionales y circuitos integrados lineales*. 4ª ed. México. Ed. Prentice Hall. 1993.
- [5] Fanjul, María Luisa y Co. *Biología funcional de los animales*. 1ª ed. México. Ed. Siglo XXI. 1998.

- [6] Dukes, H. H. y Co. *Fisiología de los animales domésticos Tomo II; funciones de integración y reproducción*. 4ª ed. México. Ed. Aguilar. 1981.
- [7] Floyd, T. L. *Fundamentos de sistemas digitales*. 6ª ed. España. Ed. Prentice Hall. 1997.
- [8] García Molina, José Carlos. *Seminario de proyecto sistema automático de medición de la temperatura cutánea promedio*. México. 1982
- [9] Rioja Lo Bianco, Enrique y Co. *Tratado elemental de zoología*. 8ª ed. México. Ed. E.C.L.A.L.S.A. 1968.

PÁGINAS EN INTERNET

- [1] www.microchip.com
- [2] www.rentron.com
- [3] members.fortunecity.es/telectronica/
- [4] www.miguelo.com.ar

Interfaces para el Puerto Paralelo de la PC, en Modo Bidireccional

M. en C. Juan Carlos Herrera Lozada,
M. en C. Juan Carlos González Robles,
Ing. Agustín Cruz Contreras;
Profesores del CIDETEC - IPN

Este es el primero de una serie de artículos enfocados al uso del puerto paralelo de la PC, para propósitos especiales y con tendencias actuales. En esta primera entrega se explica de manera concreta, cómo manejar el puerto de impresión de la PC en modo bidireccional, con la intención de interactuar con un hardware externo. Se aportan soluciones mínimas en software bajo Windows en todas sus versiones, incluyendo Windows XP. La metodología expuesta para configurar el puerto se hace extensiva para aplicaciones diversas con requerimientos similares.

INTRODUCCIÓN

El puerto paralelo se apega al estándar IEEE 1284, liberado en 1994 y que define 4 modos de operación soportados aún en la actualidad:

1. Puerto Paralelo Estándar (SPP)
2. Puerto Paralelo PS/2 (Bidireccional)
3. Puerto Paralelo Mejorado (EPP)
4. Puerto Paralelo con Capacidades Extendidas (ECP).

La mayoría de las computadoras personales recientes, tanto de escritorio como portátiles¹, presentan por omisión una configuración del puerto paralelo en dos direcciones de datos (bidireccional) para cualquier sistema operativo. Los sistemas operativos menos recientes, Windows 98 y anteriores, también son capaces de soportar este tipo de esquema para recibir y enviar datos por el puerto de impresión, siempre y cuando se configure manualmente dicha característica, preferentemente desde el *SETUP*.

El motivo central de este artículo es discutir el modo de operación bidireccional. Para el análisis mostrado se consideran dos vertientes: la programación del puerto bajo el modo MS-DOS (*Microsoft Disk Operating System – Sistema Operativo en Disco*), y como segunda derivación, la programación en Modo Windows. En ambos casos se revisan interfaces unidireccionales y bidireccionales.

PROGRAMACIÓN DEL PUERTO PARALELO EN MODO MS - DOS

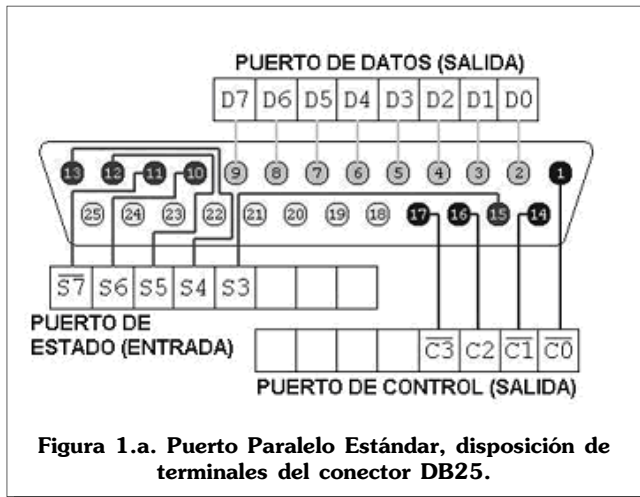
El modo MS- DOS es válido en Windows 98 y versiones anteriores (95, 3.1, etc.). En esta condición es posible escribir directamente a los registros del puerto.

PUERTO PARALELO UNIDIRECCIONAL

Considerando el modo de una sola dirección, comúnmente llamado Puerto Paralelo Estándar (*SPP*), existen tres direcciones consecutivas asociadas con un puerto paralelo; estas direcciones pertenecen al registro de datos (*Data Register*), el registro de estado (*Status Register*) y el registro de control (*Control Register*). Se le denomina dirección base a la que indica la propia del registro de datos, por lo general 0x378; así se tendría para el registro de estado la dirección inmediata siguiente 0x379 y para el registro de control la dirección 0x37A.

Existen alternativas diferentes para encontrar la dirección de los puertos, dado que ésta puede cambiar dependiendo de la arquitectura y organización interna de la PC. Es posible acceder directamente al *panel de control* de Windows y verificar el *sistema*; dentro de los recursos *hardware* se encuentra el *administrador de dispositivos*. El puerto paralelo se utiliza para la conexión de impresoras, por lo que aparecen etiquetados como LPT1, LPT2, LPT3 ó LPT4, según las características de la PC.

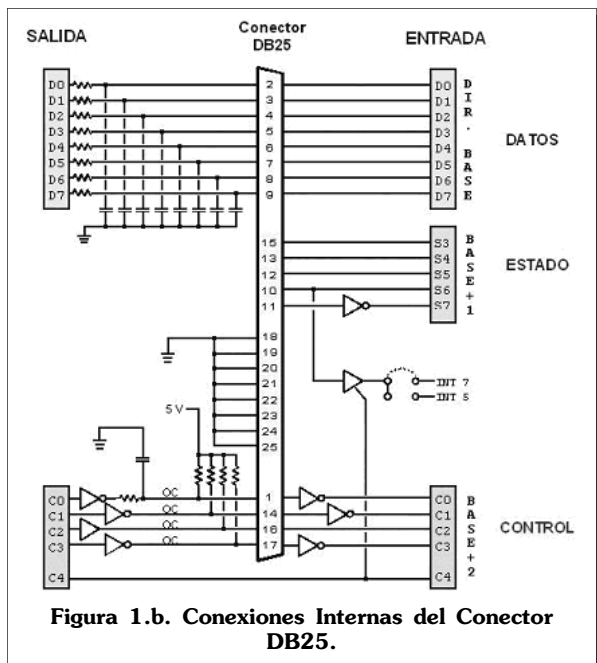
¹ En las portátiles, se tiende a encaminar todos los periféricos hacia USB, eliminando en algunos casos el puerto paralelo y el puerto de juegos; así como la unidad de disco flexible, obligando a que ésta sea externa, o bien, utilizar la alternativa del disco compacto.



Es importante recordar que en el modo estándar, el puerto de datos sólo es de salida, de ahí que se le conozca como *unidireccional*, y es de 8 bits. El puerto de estado es de sólo entrada con 5 bits referidos en el conector y el propio de control tiene 4 bits de sólo salida. En resumen, bajo este modo se tienen 12 líneas de salida (de las cuales, 3 son de tipo activo bajo) y sólo 5 de entrada (con una sola línea de tipo activo bajo), tal y como se aprecia en la **Figura 1.a**.

El puerto paralelo utiliza un conector hembra clase D de 25 terminales (DB-25), definido como *TIPO A* por el estándar IEEE 1284 (obsérvese la **Figura 1.b**). Este conector es el de interés en este artículo.

En las **Figuras 1.a y 1.b**, se aprecia la distribución física de los pines en el conector DB-25. Para fines de



análisis, se considera que los tres registros del puerto son de 8 bits, por lo que se tiene un orden significativo que es necesario respetar cuando se forma una palabra de configuración; por ejemplo, en el caso del registro de estado, se tiene disponible a partir del bit 4 y hasta el bit 8 (S7, S6, S5, S4, S3), los demás están comprometidos o reservados para otros propósitos. De acuerdo al diagrama interno aproximado mostrado en la **Figura 1.b**, el bit más significativo del registro de estado (S7), trabaja con lógica negativa y está físicamente ubicado en la terminal 11 del conector. Si se requiere leer una palabra de entrada a través de este registro, es importante considerar con qué lógica funciona cada línea.

El registro de control, para fines prácticos en el diseño de interfaces en modo estándar, es sólo de salida y utiliza los primeros cuatro bits (C3, C2, C1, C0) del registro, los restantes cuatro están reservados. En este registro, los bits C3, C1 y C0, trabajan con lógica invertida y están localizados físicamente en las terminales 17, 14 y 1 del conector.

Como ya se comentó, los ocho bits del registro de datos se utilizan sólo como salidas y todos trabajan con lógica positiva, ubicándose en orden significativo de la línea 2 a la 9 del conector, tal y como se aprecia en la **Figura 1.a**.

El siguiente programa escrito en Lenguaje C, muestra de manera sencilla cómo se envían y reciben datos. El circuito secuenciador de la **Figura 2** se activa con el botón externo conectado al bit 4 del registro de estado (S3), ubicado físicamente en la terminal 15 del conector.

```
/* {jcris, acruz, jgrobles}@ipn.mx */
```

```
/*Se incluyen las bibliotecas generales, No se utilizan todas en este ejemplo*/
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
```

```
void main(void)
```

```
{
    int entrada;
    int leds[9] = {0,1,2,4,8,16,32,64,128}; /*Datos de la secuencia, en decimal*/
```

```
    int i;
```

```
    while(1)
```

```
    {
        entrada = inportb(0x379);
        if (((entrada)&0x08)==0) /*Si el bit S3 está en 0, el botón está presionado*/
```

```
    {
```

```
for(i = 0; i < 9; i++)
{
    outputb(0x378, leds[i]);
    /*Recorre uno a uno los bits de izquierda
    a derecha*/
    sleep(1);
}
else
    outputb(0x378, 0x00);
    /*Si el botón no es presionado, los LED
    se apagan*/
}
```

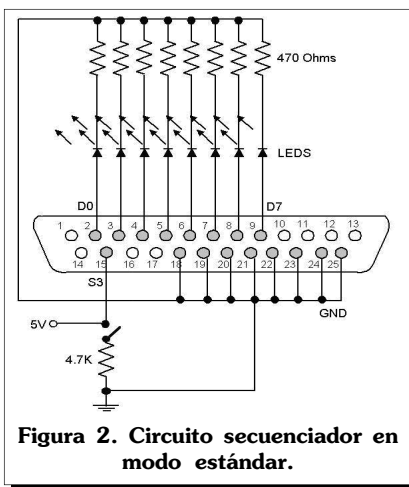


Figura 2. Circuito secuenciador en modo estándar.

Como ya se indicó, este programa sólo funciona en modo MS- DOS. Para que funcione también sobre sistemas operativos superiores a Windows 98, es necesaria una biblioteca de enlace dinámico (dll) que declare nuevas funciones para acceder al puerto paralelo, esta cuestión se detallará más adelante en este mismo documento.

PUERTO PARALELO BIDIRECCIONAL

En algunas aplicaciones prácticas se requieren más líneas de entrada que las disponibles en el puerto de estado; por ejemplo, leer los 8 bits de un convertidor analógico – digital paralelo o interactuar con una pantalla de LCD. Dado que las entradas del puerto de estado están restringidas a sólo 5 bits, es necesario adecuar el programa escrito hacia una lógica de multiplexaje que lea un dato de 4 bits, se almacene en una localidad de memoria y después de un tiempo se lea la otra parte del dato, pensando en una entrada de 8 bits ó más.

Es admisible configurar el puerto de datos para que sus ocho terminales puedan ser también entradas. Esto se logra accediendo al puerto de control y cambiando el

bit número 6 del registro de un estado natural bajo a un estado alto. Cuando C5 está a 0 lógico, las 8 líneas del puerto de datos son salidas y cuando C5 está a 1 lógico, se comportan como entradas. La **Figura 3**, muestra la disposición física de los pines del registro del puerto de control.

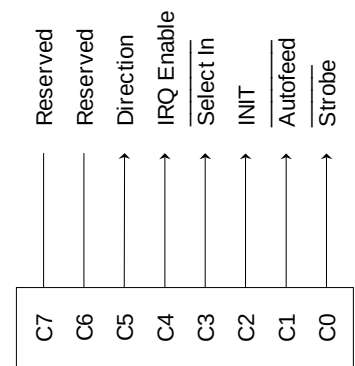


Figura 3. Puerto de Control

Considerando una dirección base 378H para el puerto de datos, se lista el siguiente fragmento en lenguaje C para explicar de manera más concreta la idea anterior.

```
{
    unsigned int Valor, temp;
    outputb(0x37A, 0x20);
    Valor=inport(0x378);
    printf («Valor Leído: %u \n», Valor);
    getch();
}
```

Obsérvese que en la primera instrucción `outputb(0x37A, 0x20)` se escribe al puerto de control con la dirección 0x37A un valor hexadecimal 0x20, traducido a binario de 8 bits como 00100000, especificando que el bit número 6 se establece a un nivel lógico alto, por lo que el puerto de datos (0x378) está configurado como entrada. En la siguiente instrucción `Valor=inport(0x378)` se leen las 8 líneas de datos y se asignan a una variable sin signo previamente definida.

Como ejemplo práctico, se considera un ADC0804 supervisado a través del puerto paralelo en modo bidireccional, como lo expone el diagrama de la **Figura 4**. En este artículo no se expone a detalle el funcionamiento del convertidor por lo que se recomienda consultar la hoja de especificaciones del dispositivo. Para este diseño en particular, la entrada analógica a convertir proviene directamente de una resistencia variable; sin embargo, ésta se puede sustituir (con las adecuaciones necesarias) por algún sensor, por ejemplo, un dispositivo LM35 para concebir un termómetro digital.

Para las señales de control *WR* y *RD*, se consideran los bits 2 y 0, respectivamente del puerto de control (0x37A).

Como ya se mencionó, C5 es el bit que permite configurar el puerto de datos como entrada o salida, por lo que no tiene una línea exterior.

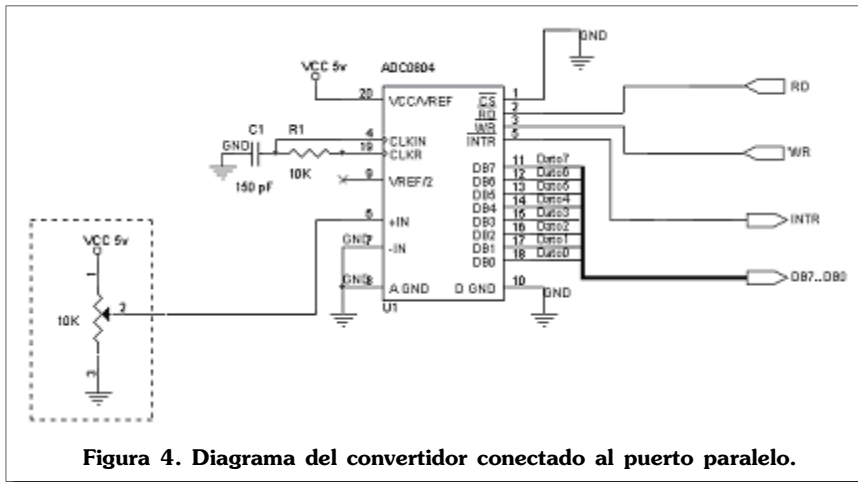


Figura 4. Diagrama del convertidor conectado al puerto paralelo.

En el programa en lenguaje C listado a continuación, se decidió utilizar C2 para Write y C0 para Read en el ADC. Físicamente, y con referencia a la Figura 3, C2 está asignado a la terminal 16 del conector y C0 a la línea 1 del mismo. Nótese que el hardware interno del puerto de control especifica que la línea 1 del conector tiene lógica negativa, por lo que se debe considerar este aspecto en el momento de generar las señales de escritura y lectura.

Para leer el bit correspondiente a la señal INTR que indica el fin de la conversión del ADC, se utiliza el bit S3 del puerto de estado (0x379), ubicado en la terminal 15 del conector.

Para capturar un dato convertido, se utilizan los 8 bits del puerto de datos (0x378) conectados de manera tradicional, del menos significativo al más significativo; en otras palabras, DB0 corresponderá al D0 del puerto (terminal 2), DB1 a D1 (terminal 3) y así, sucesivamente, hasta DB7 que corresponderá a D7 (terminal 9).

Al principio del programa listado se solicita el número de muestras a convertir (cuántas veces se repetirá el programa). Es una manera sencilla de terminar el programa después de "n" lecturas. También es posible asignar una tecla para salir. Se recomiendan pocas muestras cuando el retardo es de un valor alto.

/* ADC0804 por el puerto paralelo, {jcrts,acruz,jgrobles}@ipn.mx
8-bits utilizando modo bidireccional (ECP).*/

```
#include <stdio.h>
#include <dos.h>
```

```
main()
```

```
{
    unsigned int pdatos, pestado, pcontrol, temp, dato_leido;
```

```
int contador, muestra;
pdatos=0x378;
pestado=pdatos+1; pcontrol=pdatos+2;
temp=inportb(pcontrol); /*permite restaurar el
valor original del puerto*/
```

```
clrscr();
/* «muestra» indica el número de lecturas que
deseas realizar*/
for (muestra=0; muestra<50; muestra++)
{
    /* Comienza la conversión poniendo en bajo
«write» y manteniendo C5 en alto para la
bidireccionalidad.
Posteriormente pondremos en alto «write» para
deshabilitarlo; así se genera el pulso negativo
para accionar «write»*/
    outportb(pcontrol, 0x20);
    delay(10000);
    outportb(pcontrol, 0x24);
```

```
/* Espera hasta que la conversión haya concluido recibiendo el bit INTR
proveniente del ADC*/
/* «contador» permite terminar el programa después de 256 ciclos si
no se ha conectado el ADC al puerto*/
contador=0;
do { contador++; }
while (((inportb(pestado) & 0x08)==0) && (contador!=256));
if (contador==256) printf («No hay convertidor!!!\n»);
else
{
    delay(10000); /*Es posible omitir el retardo*/
    outportb(pcontrol, 0x25);
    delay(10000);
    dato_leido=inportb(pdatos);
    outportb(pcontrol, 0x24);
    clrscr();
    printf («ADC value:%i»,dato_leido);
    getch(); /* Únicamente espera por una tecla, una vez que concluyó
el programa*/
}
}
outportb(pcontrol, temp); /*restaura puerto de control a su valor
original*/
return 0;
}
```

De igual forma que en el modo unidireccional, este código funciona correctamente sólo en modo MS-DOS.

PROGRAMACIÓN DEL PUERTO PARALELO EN MODO WINDOWS

Windows NT, 2000 y XP no permiten manejar el puerto paralelo en modo MS-DOS, como sucedía con las versiones anteriores de este mismo sistema operativo. Para solucionar este inconveniente, y dar acceso al puerto en cualquier versión de Windows (en modo Windows) se requieren bibliotecas de enlace dinámico (dll) que gestionan ante el sistema operativo el manejo del puerto.

Una dll se diseña, por lo general, en C++ y se utiliza en lenguajes de alto nivel como el mismo C++, Delphi o Java, entre otros. En este artículo se utiliza una dll *freeware* (uso libre, no comercial) para crear interfaces en *Visual Basic 6.0*.

Ninguno de los compiladores para Windows ha incluido un componente para el manejo de puertos en forma general sólo para comunicaciones e impresión. La ventaja que tiene Visual.

Basic sobre otros lenguajes es la facilidad del entorno visual. La biblioteca *io.dll* se descarga gratuitamente de <http://www.geekhideout.com> y se copia al directorio ubicado en C:/Windows/System32/. En Internet existen disponibles otras bibliotecas similares; la que aquí se indica es la que se ha probado con éxito en todos los diseños presentados.

El siguiente paso consiste en crear un proyecto estándar en Visual Basic y adicionar un módulo que incluya los prototipos de la dll para Visual Basic, listados en la misma página web de donde se descargó dicha dll (referirse a la **Tabla 1**).

Lo anterior se logra activando la opción del menú *Proyecto* y seleccionando *Adicionar Módulo*. En el espacio de edición del mismo módulo se escriben los prototipos de la **Tabla 1**. Para salvar el proyecto completo se requiere nombrar primeramente la *Forma*, después el *Proyecto* y finalmente el *Módulo*. Obsérvese la pantalla de la **Figura 5**.

Básicamente y en correspondencia con los prototipos de la dll, se tienen 13 funciones para manejo del puerto paralelo; nos enfocaremos sólo a cuatro: *PortOut*, *PortIn*, *SetPortBit* y *ClrPortBit*. Las dos primeras permiten enviar y recibir respectivamente un byte por el puerto indicado. *SetPortBit* y *ClrPortBit*, manipulan sólo un bit del registro, ya sea para establecerlo o para limpiarlo.

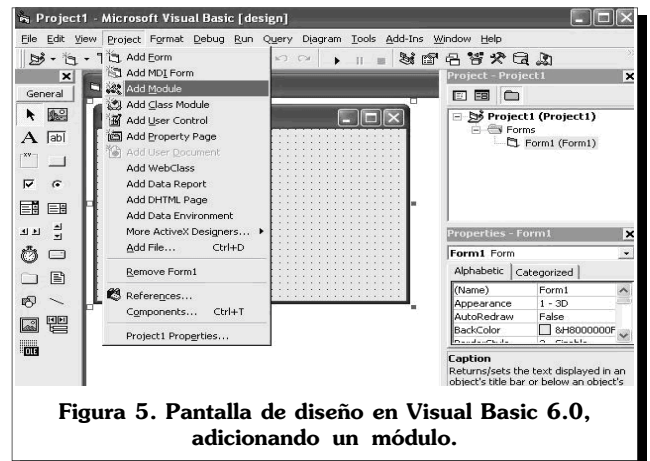


Figura 5. Pantalla de diseño en Visual Basic 6.0, adicionando un módulo.

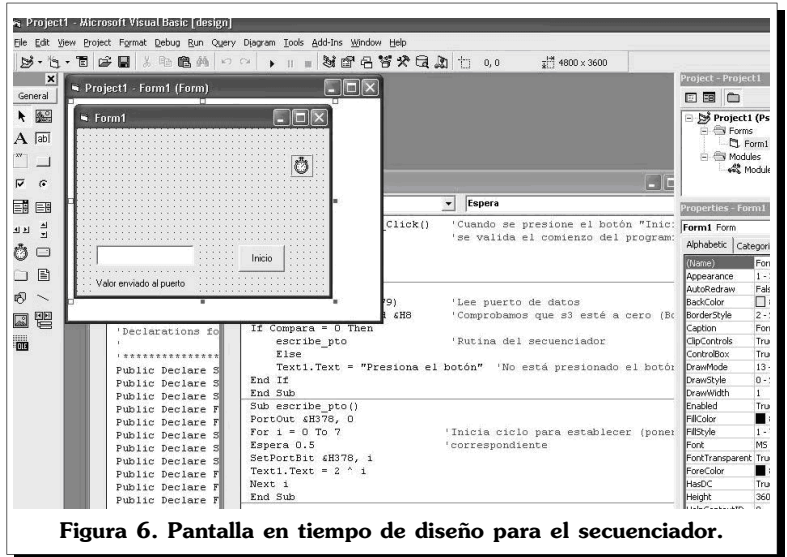


Figura 6. Pantalla en tiempo de diseño para el secuenciador.

PROGRAMACIÓN UNIDIRECCIONAL EN VISUAL BASIC

Considerando el mismo ejemplo en modo estándar propuesto por el diagrama de la **Figura 2**; en Visual Basic no se tienen las directivas *Sleep* ni *Delay*, propias de C, por lo que para temporizar se requiere un *Timer* sincronizado en milisegundos. La pantalla mostrada en la **Figura 6**, es la que se utilizó para concretar el *secuenciador* de ejemplo.

Una vez creado el proyecto, y después de haber adicionado el módulo de declaraciones de la dll, se procede a colocar una caja de texto (*TextBox*) que por omisión llevará el nombre *Text1*.

Después se colocará debajo de la caja una etiqueta (*Label*) que por default está referida como *Label1*. Dentro de las propiedades de este

```
Public Declare Sub PortOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Byte)
Public Declare Sub PortWordOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Integer)
Public Declare Sub PortDWordOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Long)
Public Declare Function PortIn Lib "IO.DLL" (ByVal Port As Integer) As Byte
Public Declare Function PortWordIn Lib "IO.DLL" (ByVal Port As Integer) As Integer
Public Declare Function PortDWordIn Lib "IO.DLL" (ByVal Port As Integer) As Long
Public Declare Sub SetPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Sub ClrPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Sub NotPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Public Declare Function GetPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte) As Boolean
Public Declare Function RightPortShift Lib "IO.DLL" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Public Declare Function LeftPortShift Lib "IO.DLL" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Public Declare Function IsDriverInstalled Lib "IO.DLL" () As Boolean
```

Tabla 1. Prototipos de io.dll a escribirse en un módulo de Visual Basic.

objeto se debe buscar la que se refiere a *Caption* (Texto de etiqueta) y escribir "Valor enviado al puerto".

Se coloca un botón de acción (*CommandButton*) que por omisión se llamará *Command1*. Dentro de sus propiedades, y en *Caption*, se escribe "Inicio".

Por último, se inserta un *Timer*, con el nombre *Timer1*. Este objeto sólo se visualiza en tiempo de diseño, es decir, en la aplicación ejecutable no se verá.

En el editor de la forma (*View Code*) se copia el código siguiente y se presiona el botón *Run* para ejecutar la aplicación.

```
{jcrsl;acruz;jgrobles}@ipn.mx
Option Explicit 'Programa que envía y recibe datos del puerto
paralelo de la PC. Se utiliza io.dll.'

Dim i As Integer
Dim ValorIn As Byte
Dim Compara As Byte

Private Sub Form_Load()
'no hagas nada, cuando se abra la aplicación'
End Sub

Private Sub Command1_Click() 'Cuando se presione el
                             botón «Inicio» se val-
                             da el comienzo del
                             programa'

While (1)
Comienza
Wend
End Sub

Sub Comienza()
ValorIn = PortIn(&H379) 'Lee puerto de datos'
Compara = ValorIn And &H8 'Comprobamos que s3 esté
                           a cero (Botón presionado)'

If Compara = 0 Then
escribe_pto 'Rutina del secuenciador'
Else
Text1.Text = «Presiona el botón» 'No está presionado el botón'
End If
End Sub

Sub escribe_pto()
PortOut &H378, 0
For i = 0 To 7 'Inicia ciclo para establecer (poner a 1) el bit
                'correspondiente'

Espera 0.5
SetPortBit &H378, i
Text1.Text = 2 ^ i
Next i
End Sub

Sub Espera(t As Double) 'Rutina de tiempo'
Timer1.Interval = t * 1000 'Milisegundos'
Timer1.Enabled = True 'Habilitamos timer'
```

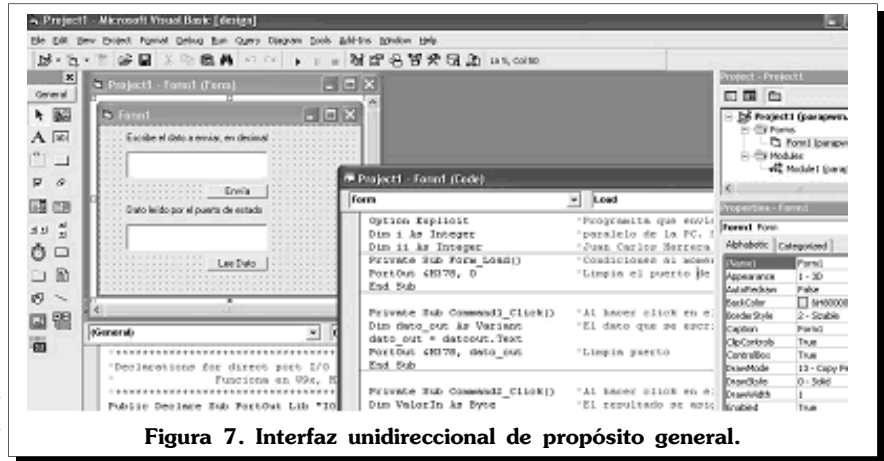


Figura 7. Interfaz unidireccional de propósito general.

```
Do While Timer1.Enabled
DoEvents
Loop
End Sub
```

```
Private Sub Timer1_Timer() 'Habilitación del timer1, relacionado con
                           la rutina Espera'
```

```
Timer1.Enabled = False
End Sub
```

```
Private Sub Form_OKClick() 'Salir de la aplicación'
App.End
End Sub
```

El siguiente código, diseñado para una interfaz en modo unidireccional, permite enviar un dato escrito en la caja de texto correspondiente a "Enviar dato" directamente al puerto de datos (0x378). También es posible leer un dato del puerto de estado (0x379) en la caja de texto correspondiente a "Leer dato". La pantalla en tiempo de diseño se muestra en la **Figura 7**.

```
Option Explicit 'Programa que envía y recibe datos del puerto
{jcrsl;acruz;jgrobles}@ipn.mx'
```

```
Private Sub Form_Load() 'Condiciones al momento de abrir la
                           aplicación'
PortOut &H378, 0 'Limpia el puerto de datos'
End Sub
```

```
Private Sub Command1_Click() 'Al hacer click en el botón Command1
                             («Envía») El dato que se escriba en
                             la caja de texto, en decimal'
```

```
Dim dato_out As Variant
dato_out = datoout.Text
PortOut &H378, dato_out 'Limpia puerto'
End Sub
```

```
Private Sub Command2_Click() 'Al hacer click en el botón Command2
                             («Lee Puerto») El resultado se asig-
                             nará a la variable ValorIn'
```

```
Dim ValorIn As Byte
ValorIn = PortIn(&H379)
dato_in.Text = ValorIn & « , 'está en decimal»
End Sub
```

```
Private Sub Form_OKClick() 'Salir de la aplicación'
App.End
End Sub
```

PROGRAMACIÓN BIDIRECCIONAL EN VISUAL BASIC

Para la programación bidireccional del puerto de datos se sigue la misma lógica descrita en los programas en Lenguaje C anteriores, es decir, se establece el bit 6 (C5) del puerto de control para que el registro de datos acepte un byte proveniente del exterior y se limpia el mismo bit si se desea que el registro de datos sea sólo de salida.

Retomando la interfaz para monitorear el ADC0804, algunos autores desestiman utilizar todas las señales del ADC. Con una frecuencia sustentable, mucho menor que los 8KHz naturales del dispositivo, es posible obtener circuitos más simples como el denominado *Free Running* que propone utilizar un inversor hacia RD de la señal WR, obligando a que RD siempre sea el complemento de WR y viceversa (ver **Figura 8**). El inversor se puede implementar dentro de código, y dado que la frecuencia de trabajo para reportar lecturas se considera lenta, es posible omitir la espera de la respuesta de la señal INTR. Se recomienda reestablecer el puerto de control a su estado original, antes de salir de la aplicación.

La preparación del puerto desde el *SETUP* es fundamental para que funcione correctamente este programa.

El siguiente código en Visual Basic muestra la solución propuesta. Obsérvese que sólo se envía la señal de reloj por el puerto de control, a la vez que se reciben de forma paralela los 8 bits del ADC por el puerto de datos de la PC.

```
'fjcris;acruz; jgrobles}@ipn.mx
```

```
Option Explicit 'Programa que recibe datos de un ADC0804
Dim DatoPuerto As Byte 'con conexiones mínimas Free Running
Dim Temporal As Byte 'Modo del puerto paralelo: ECP (Bidireccional)
```

```
Private Sub Form_Load() 'Limpia puerto de datos al cargar la aplicación
PortOut &H378, 0
```

```
End Sub
```

```
Private Sub Command1_Click() 'Con el botón «Inicio»
Temporal = PortIn(&H37A) 'Almacena el valor del puerto de control
While (1) 'se valida el comienzo de la adquisición
Comienza
Wend
End Sub
```

```
Private Sub Command2_Click() 'Botón «Restaurar Puerto»
PortOut &H37A, Temporal 'Regresa a la configuración original pto. de control
PortOut &H378, 0 'Limpia nuevamente el puerto de datos; detiene programa
End Sub
```

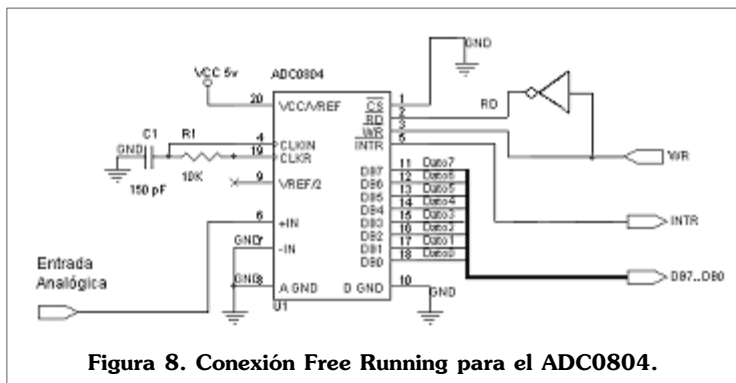
```
Sub Comienza()
PortOut &H37A, &H20 'Configura C5, bidireccional y genera flanco negativo
Espera 0.5 'Rutina de espera (1/2 segundo)
PortOut &H37A, &H24 'Mantiene bidireccionalidad y genera flanco positivo
Espera 0.5
DatoPuerto = PortIn(&H378) 'Lee puerto de datos y escribe equivalente en Volts
Text1.Text = DatoPuerto * 0.0196 & « Volts» '255 binario = 5 Volts analógicos
End Sub
```

```
Sub Espera(t As Double) 'Rutina de tiempo
Timer1.Interval = t * 1000 'Milisegundos
Timer1.Enabled = True 'Habilitamos timer
Do While Timer1.Enabled
DoEvents
Loop
End Sub
```

```
Private Sub Timer1_Timer() 'Habilitación del timer1, relacionada
Timer1.Enabled = False 'con la rutina Espera
End Sub
```

```
Private Sub Form_OKClick() 'Salir de la aplicación
App.End
End Sub
```

Las **Figuras 9a** y **9b**, muestran de manera respectiva, las pantallas correspondientes al tiempo de diseño y al tiempo de ejecución de la interfaz en dos direcciones para el ADC0804. Particularmente en **9b** se aprecia el resultado arrojado por la aplicación para un dato de 255₁₀ equivalente a 5 Volts, enviado por el ADC trabajando sobre Windows XP.



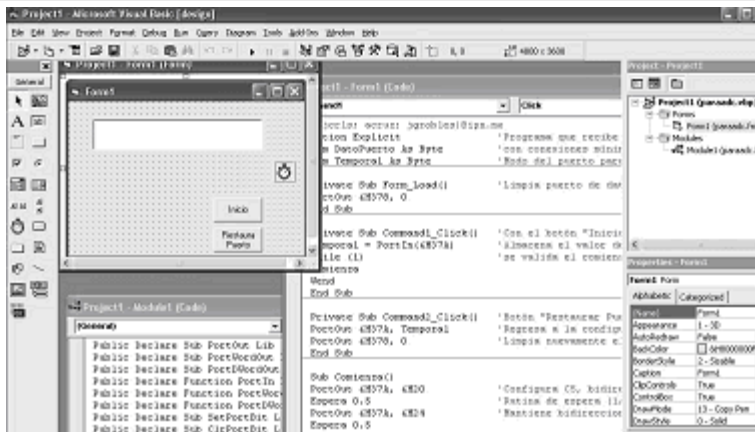


Figura 9a. Pantalla en modo de diseño para la conexión Free Running del ADC0804.

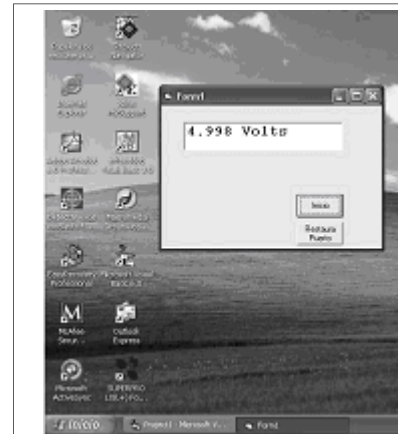


Figura 9b. Aplicación bajo Windows XP.

CONCLUSIONES

Se presentó una metodología sencilla para crear interfaces para el puerto paralelo de la PC, utilizando Lenguaje C para modo MS-DOS y Visual Basic 6.0 para modo Windows. Las soluciones aportadas son adaptables sin cambios drásticos en el código a cualquier aplicación similar o más sofisticada, que requiera intercambio de datos en una o dos direcciones.

Todos los ejercicios mostrados se implementaron físicamente en dos PC con distinto sistema operativo. Las aplicaciones creadas en Lenguaje C se ejecutaron en modo MS-DOS bajo Windows 98 y las de Visual Basic bajo Windows XP. En ambos casos, para la bidireccionalidad de los datos, se configuró el modo ECP en el SETUP.

Cabe mencionar que es posible encontrar en Internet bibliotecas de enlace dinámico para diferentes lenguajes, como se puede advertir en el mismo sitio de donde se descargó io.dll (www.geekhideout.com) y en otros como www.zealsoftstudio.com, por mencionar algunos.

Se presentó el caso particular de un ADC0804 en un esquema bidireccional. La solución disminuye la cantidad de hardware y aumenta la velocidad de la respuesta, tomando en consideración que para plataformas superiores a Windows 98 existen dos modos de velocidad para acceso a puertos: *Normal* y *Fast*.

En un próximo artículo se detallará cómo utilizar el puerto paralelo como una alternativa al extinto bus ISA de la PC.

REFERENCIAS BIBLIOGRÁFICAS E INTERNET

- [1] Hans-Peter Messmer. *The indispensable PC hardware book*. Ed. Addison-Wesley, 1999.
- [2] Dhananjay V. Gadre. *Programming the Parallel Port: Interfacing the PC for Data Acquisition & Process Control*. Ed. CMP Books, 1999.
- [3] David I. Schneider. *An Introduction to Programming with Visual Basic 6.0*. Ed. Prentice – Hall, 2000.
- [4] Francisco J. Ceballos. *Visual Basic 6, Curso de Programación*. Ed. AlfaOmega-Rama, 2002.
- [5] <http://www.geekhideout.com>.
- [6] Datasheet, National Semiconductor. <http://www.national.com/pdf/AD/ADC0804.html>.
- [7] <http://www.doc.ic.ac.uk/~ih/doc/par/> : Interfacing to the IBM-PC Parallel Printer Port.
- [8] <http://www.lvr.com/parport.htm> : Parallel Port Central
- [9] <http://www.beyondlogic.org/ecp/ecp.htm>
- [10] http://www.epanorama.net/links/pc_interface.html
- [11] <http://www.logix4u.net/parallelport1.htm>
- [12] <http://www.zealsoftstudio.com>

Tecnologías xDSL

Ing. Patricia Pérez Romero
Profesor del CIDETEC-IPN

Los sistemas de telecomunicaciones actuales están constituidos por un conjunto variado de redes que abarcan todos los servicios de Voz, Datos y Video, mismos que ahora son posibles gracias a la digitalización.

Sin embargo, se sigue teniendo una limitante en el lado del cliente, ya que entre éste y las centrales telefónicas existe un par de cobre que efectúa su comunicación en forma analógica. Pero con la introducción de la fibra óptica hasta el cliente y la nueva era de tecnología digital la situación está cambiando rápidamente.

INTRODUCCIÓN

Los volúmenes de tráfico de datos ya están excediendo el tráfico de voz en muchos países y se observa que ésta es la tendencia mundial. Por ello, los usuarios requieren actualmente de servicios que necesitan gran ancho de banda: para el acceso a Internet, Intranets, y acceso remoto a Redes de Área Local.

Esta demanda de servicios evoluciona a dos claras tendencias: la necesidad de mayor capacidad y la búsqueda de mayor sencillez en el transporte y gestión de la variedad de tipos de información.

Los proveedores de equipo de telecomunicaciones están desarrollando arquitecturas de red que proporcionan una fácil evolución de las redes telefónicas públicas actuales, a una red centrada en datos y basada en conmutación de paquetes no orientados a conexión, los cuales garantizan la calidad de servicio, además de incluir otras ventajas como el uso eficiente de los dispositivos instalados, escalabilidad y la habilidad de implementar nuevos protocolos de señalización de llamadas y servicios. También el control y sistema de administración tendrán un papel importante, pero sobre todo el hacer llegar estas facilidades al cliente a través de la red de cobre existente y de fibra óptica empleando para ello tecnología xDSL (x Digital Subscriber Line) que permite accederse a la red con mayor velocidad.

¿QUÉ ES xDSL?

xDSL (x Digital Subscriber Line, la "x" se utiliza para diferenciar los tipos de servicios y/o tecnologías DSL) se refiere a un grupo con tecnología similar que provee gran ancho de banda y alta velocidad de acceso sobre circuitos locales de cable de cobre, sin amplificadores o repetidores de señal a lo largo de la ruta del cableado, entre la conexión del cliente y el primer nodo en la red. xDSL se proporciona sobre circuitos locales de cobre no cargados (cables sin nin-

gún tipo de inducción de voltaje o señal).

La Tecnología xDSL soporta formatos y velocidades de transmisión especificados, como lo son T1 (1.544 Mbps) y E1 (2.048 Mbps), y es lo suficientemente flexible para soportar velocidades y formatos adicionales como sean especificados (ej. 6 Mbps asimétricos para transmisión de alta velocidad de datos y video). xDSL puede coexistir en el circuito con el servicio de voz; como resultado, todos los tipos de servicios, incluyendo el de voz, video, multimedia y datos pueden ser transportados sin el desarrollo de nuevas estrategias de infraestructura. xDSL es una tecnología «Modem-Like» (muy parecida a la tecnología de los módem), donde se requiere un dispositivo xDSL terminal en cada extremo del circuito de cobre. Estos dispositivos aceptan flujo de datos, generalmente en formato digital, y lo sobreponen a una señal analógica de alta velocidad. Las dos técnicas de modulación más usadas actualmente para xDSL son:

- CAP-Carrier-less Amplitude Phase Modulation
- DMT-Discrete Multitone Modulation.

CAP (CARRIER-LESS AMPLITUDE MODULATION)

Esta modulación está basada en la Modulación en Amplitud en Cuadratura QAM. El receptor de QAM

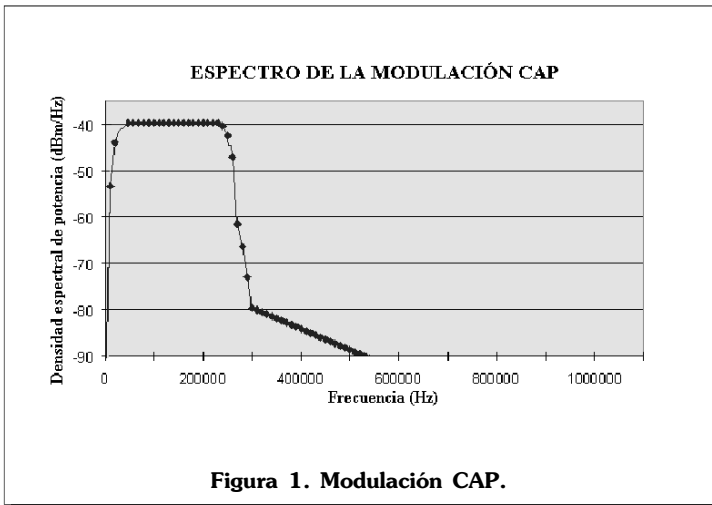


Figura 1. Modulación CAP.

necesita una señal de entrada que tenga la misma relación entre espectro y fase que la señal transmitida, pero las líneas telefónicas instaladas no garantizan esta calidad. La CAP es una implementación de QAM para xDSL, de bajo costo debido a su simplicidad y con una velocidad de 1.544 Mbps (ver Figura 1).

La CAP divide la señal modulada en segmentos que después almacena en memoria. La señal portadora se suprime, puesto que no aporta ninguna información. La onda transmitida se genera al pasar cada uno de los segmentos por dos filtros digitales transversales con igual amplitud, pero con una diferencia de fase de $\pi/2$. En la recepción se reensamblan los segmentos, volviendo a obtener la señal modulada. De este modo, obtenemos la misma forma del espectro que con QAM, siendo la CAP más eficiente que QAM en implementaciones digitales.

DMT (DISCRETE MULTI-TONE MODULATION)

Es un tipo de modulación multi-portadora, que elimina el problema de las altas frecuencias que aumentan considerablemente las pérdidas debido al ruido en las líneas de cobre, dividiendo el ancho de banda disponible en 256 subcanales, que son comprobados para determinar su capacidad portadora (Figura 2).

PROCESO DE MODULACIÓN

- La modulación DMT emplea la transformada discreta de Fourier para crear y demodular cada una de las 256 portadoras individuales, dividiendo el ancho de banda disponible en unidades más pequeñas.
- La línea se comprueba para determinar qué banda de frecuencias es posible y cuántos bits pueden ser transmitidos por unidad de ancho de banda.
- Los bits se codifican en el transmisor mediante la transformada rápida inversa de Fourier y después pasan a un conversor analógico/digital.

- Al recibirse la señal, ésta se procesa mediante una transformada rápida de Fourier para decodificar la trama de bits recibida.

La DMT ha sido seleccionada en un primer borrador del grupo de estandarización T1E1 de ANSI, sobre la CAP y QAM.

Las últimas modificaciones a los estándares sobre ADSL han llevado al desarrollo de una nueva generación de módems capaces de transmitir hasta 8,192 Mbps en sentido descendente y hasta 0,928 Mbps en sentido ascendente. La separación de los trayectos en ADSL se efectúa por Multiplexación por División en Frecuencias (FDM) o por Cancela-

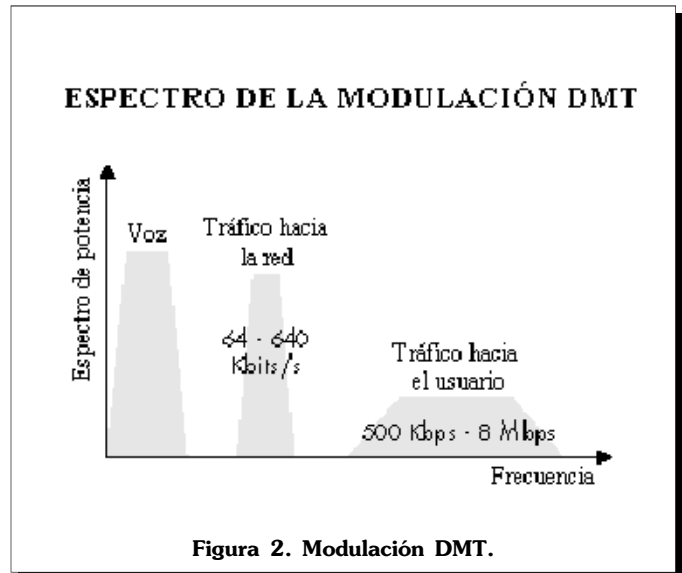


Figura 2. Modulación DMT.

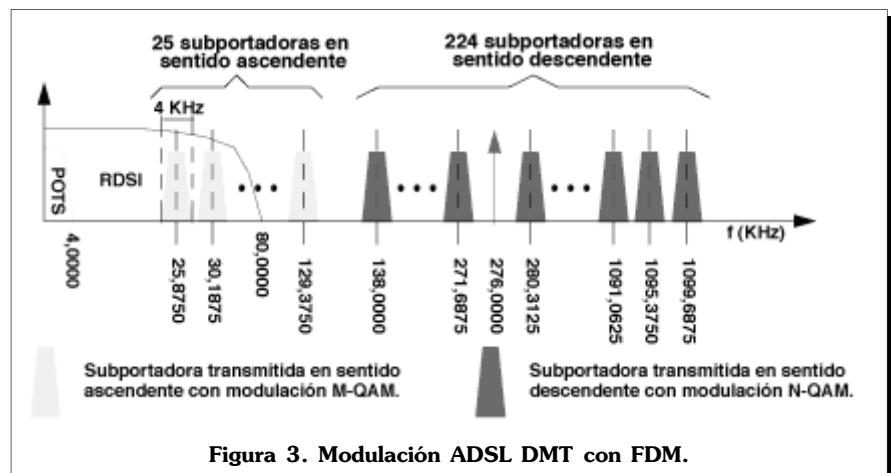
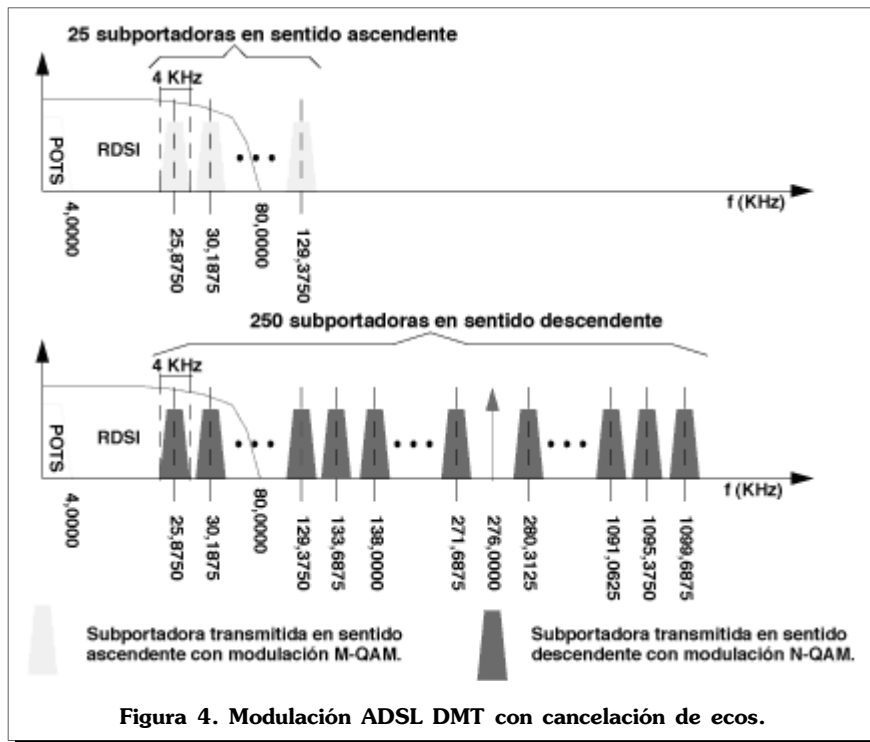


Figura 3. Modulación ADSL DMT con FDM.



ción de Eco, siendo esta última la más empleada (Figuras 3 y 4).

En las dos figuras anteriores se han presentado las modalidades existentes dentro del ADSL con modulación DMT: FDM y cancelación de ecos. En la primera, los espectros de las señales ascendente y descendente no se traslapan, lo que simplifica el diseño de los módems, aunque reduce la capacidad de transmisión en sentido descendente, no tanto por el menor número de subportadoras disponibles sino por el hecho de que las de menor frecuencia, aquellas para las que la atenuación del par de cobre es menor, no están disponibles. La segunda modalidad, basada en un cancelador de ecos para la separación de las señales correspondientes a los dos sentidos de transmisión, permite mayores caudales a costa de una mayor complejidad en el diseño.

En la Figura 3 (Modulación ADSL DMT con FDM) y en la Figura 4 (Modulación ADSL DMT con cancela-

ción de ecos) se muestran los espectros de las señales transmitidas por los módems ADSL tanto en sentido ascendente como descendente. Como se puede ver, los espectros nunca se traslapan con la banda reservada para el servicio telefónico básico (POTS o «Plain Old Telephone Service»), y en cambio sí lo hacen con los correspondientes al acceso básico RDSI. Por ello el ADSL y el acceso básico RDSI son incompatibles.

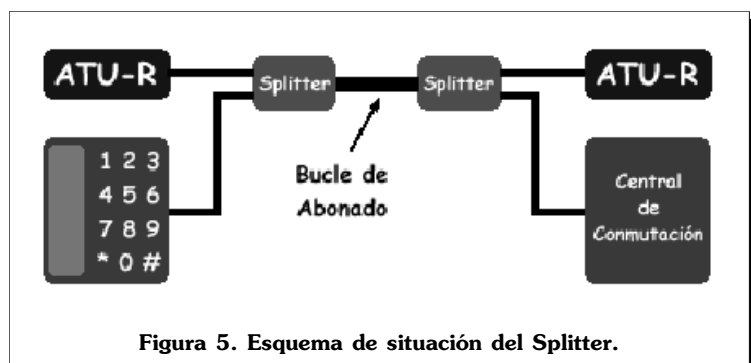
ESTRUCTURA Y CARACTERÍSTICAS TÉCNICAS

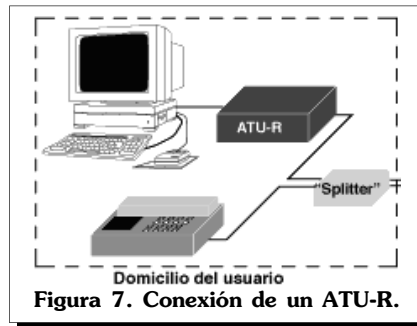
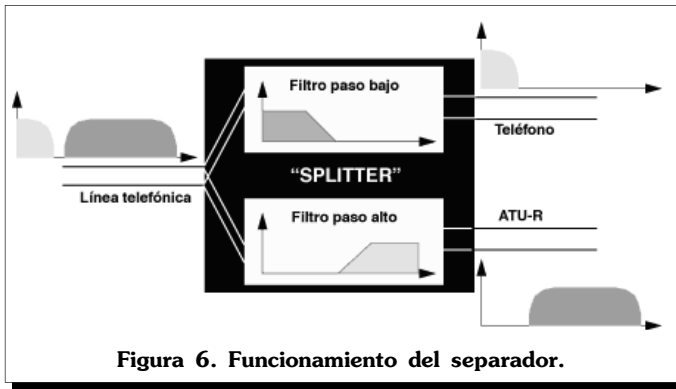
La línea sobre la cual se soporta el ADSL es la línea de cobre; para que sobre esta línea analógica puedan prestarse servicios DSL es necesario introducir cambios tanto en el domicilio del usuario como en la central telefónica. A continuación se enumeran los diferentes equipos y dispositivos que componen la estructura del DSL desde el usuario hasta la central.

EL SEPARADOR

La tecnología ADSL permite la simultaneidad, por el mismo medio, de los servicios tradicionales de voz y de los nuevos servicios de transmisión de datos a alta velocidad. Como es ya sabido, los servicios de voz operan en la banda baja de frecuencia que va desde los 0 KHz hasta los 4000 KHz. A partir de los 4000 KHz queda a disposición de la transmisión de datos. Nos encontramos por tanto con dos bandas de frecuencias de transmisión claramente diferenciadas y no traslapadas que se transmiten por el mismo medio físico: el hilo de cobre.

El separador es un pequeño aparato que se instala en el domicilio del cliente (generalmente cerca del PTR) y que discrimina entre un tipo de señal u otro, es decir identifica la





R como líneas ADSL hubiese, ya que la infraestructura dedicada a ello resultaría excesiva y no sería práctico. Por ello en las centrales están las llamadas DSLAM (Figura 8).

señal de voz (0 - 4000 KHz) y la de datos, y las diferencia para su posterior tratamiento. Al Separador se conecta el hilo de cobre que sube al domicilio desde el registro de suscriptores del edificio y de él sale otro par de hilos de cobre que irán a una toma de teléfono a la cual se conectará el modem ADS o ATU-R (Figura 5).

El separador está compuesto básicamente por dos filtros: un filtro paso bajo que identifica las señales que no superen la frecuencia de los 4000 KHz y un filtro paso alto que realiza la misma función pero con señales que superan los 4000 KHz, es decir con las señales de datos (Figura 6). El separador lo instala y es responsabilidad de la compañía operadora que explote el circuito de suscriptor.

En la central se coloca también un splitter que realiza las mismas funciones para realizar el filtro en la central y discriminar entre lo que es comunicación de voz y de datos, ya que no se les da el mismo tratamiento ni acceden a la misma red de transporte.

otro extremo al separador (Figura 7). Como cualquier modem, puede ser externo o interno, comercializándose mayoritariamente los externos.

En la central, al igual que ocurría con el separador hay un equipo similar llamado ATU-C (ATU-C o «ADSL Terminal Unit-Central»). Todos los ATU-C de la central se concentran en un equipo llamado DSLAM del cual hablaremos a continuación.

DSLAM

(Digital Subscriber Line Access Multiplexer). Como ya se vió, en el domicilio del usuario es necesario instalar, aparte del separador, el modem ADSL o ATU-R. Además, en la central por cada línea ADSL que le llegue es necesario un ATU-C cuyas funciones son similares (no idénticas) a las del ATU-R. Resulta evidente pensar que sería complicado tener en una central telefónica tantos aparatos ATU-C del tamaño de los ATU-

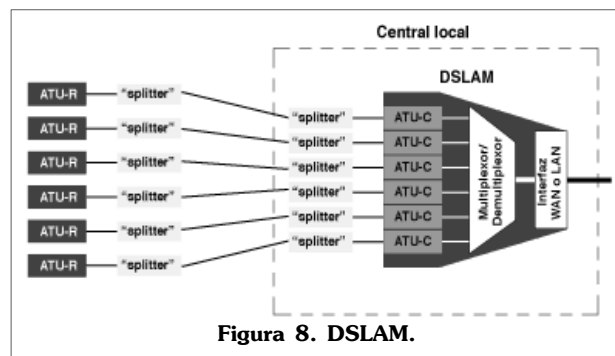
El DSLAM agrupa todas las tarjetas ATU-C de la central, optimizando su uso y gestión. Sin embargo la función del DSLAM no es solo esa. El DSLAM se compone además de un conmutador ATM, las interfaces para las líneas ADSL y la interface para la conexión a la red de transporte ATM. Por tanto, se deduce que la función principal del DSLAM, aparte de agrupar la ATU-C, es la de enrutar todo el tráfico de esas tarjetas (tráfico que será exclusivamente de datos, ya que previamente el separador de la central se habrá encargado de separar las señales) hacia la red de transporte.

SERVICIOS QUE SE PUEDEN OFRECER CON UN SISTEMA DE COMUNICACIÓN xDSL:

- Navegación Internet
- Intranet
- Video Conferencia
- Servicios Transparentes LAN para Clientes Corporativos
- Acceso Remoto LAN para Clientes Corporativos
- Educación a Distancia
- Video bajo Demanda/Televisión Interactiva
- Juegos Interactivos

El ATU-R

(ATU-R o «ADSL Terminal Unit-Remote») El ATU-R es el equipo que se encarga de realizar la conexión en casa del cliente. También es conocido como módem ADSL y se conecta por un lado al PC del usuario, y en el



Considerando la necesidad de soportar el incremento en la demanda del acceso a Internet combinada con telecomunicación e interconectividad de las Redes LAN, podemos ver que xDSL ofrece a los proveedores de ser-

vicios Internet (ISP's) y proveedores de acceso competitivo, una oportunidad de ampliar sus recursos.

Los accesos a Internet y a Redes LAN, pueden ser soportados por la compatibilidad de xDSL con los estándares tradicionales de comunicación. Dados esos desarrollos importantes y difíciles de alcanzar, esta claro que la tecnología xDSL será el mayor componente de la infraestructura del proveedor de servicios. Usando estas capacidades, los proveedores podrán ofrecer un rango completo de servicios, organizándolos rápidamente, y asegurándose un servicio excelente.

La UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) ha elegido la tecnología ATM como base de las redes de banda ancha. Esta elección se debe a la flexibilidad que ofrece para el manejo de servicios que exigen una gama heterogénea de velocidades de tráfico binario, variando desde varios Kbps hasta cientos de Mbps. Sin embargo, del lado del cliente con una línea convencional debe utilizarse la tecnología xDSL para cumplir con los requerimientos de ancho de banda.

VARIANTES DE xDSL

A continuación se enlistan algunas variantes del xDSL; además, en las Figuras 9a y 9b se muestran dos tablas indicando otros datos técnicos.

ADSL.- Es la más popular forma de tecnología xDSL. La clave de ADSL es que el ancho de banda «upstream» y «downstream» es asimétrico. En la práctica, el ancho de banda del proveedor al usuario (downstream) será el camino para mayor velocidad. Downstream corre típicamente en un rango de 64 Kbps a 1.5 Mbps.

Cuadro de la familia xDSL

Tipo de DSL	Descripción	Downstream/Upstream	Límite distancias	Aplicaciones
ISDN	ISDN Digital Subscriber line	128 Kbps	5,5 Km con 24 AWG	Similar al ISDN BRI, solamente para servicios de datos (sin voz sobre la misma línea)
CDSL	Consumer DSL from Rockwell	1 Mbps downstream sin Upstream	5,5 Km con 24 AWG	Sin filtro, para hogares y pymes (SOHO), similar al DSL Lite
DSL Lite	DSL sin filtro	1.544 - 6 Mbps según servicio	5,5 Km con 24 AWG	El estándar ADSL, sacrifica velocidad a condición de no instalar el filtro en el usuario final
G. Lite	DSL sin filtro	1.544 - 6 Mbps según servicio	5,5 Km con 24 AWG	El estándar ADSL, sacrifica velocidad a condición de no instalar el filtro en el usuario final
HDSL	High bit-rate digital subscriber line	1.544 Mbps dúplex sobre 2 pares trenzados; 2.048 Mbps dúplex sobre 3 pares trenzados	3,6 Km con 24 AWG	T1/E1- Servicio entre servidores WAN, LAN, Servidores de acceso

Figura 9a. Variantes de xDSL

Cuadro de la familia xDSL

Tipo de DSL	Descripción	Downstream/Upstream	Límite distancias	Aplicaciones
SDSL	Symmetric DSL	1.544 Mbps dúplex (U.S. y Canadá) 2.048 Mbps (Europa) Línea dúplex downstream y upstream	3,6 Km con 24 AWG	Similar a HDSL pero que requiere solamente un único par trenzado
ADSL	Asymmetric Digital Subscriber Line	1.544 a 6.1 Mbps downstream, 16 a 640 Kbps upstream	1.544 Mbps - 5,5 km 2.048 Mbps - 4,8 km 6.312 Mbps - 3,6 km 8.448 Mbps - 2,7 km	Usado para Internet, video, full motion, videoconferencia, telemedicina, teleducación, etc.
RADSL	Rate-Adaptive DSL from Westell	Adaptado a la línea, 640 Kbps, a 2.2 Mbps downstream; 272 Kbps a 1.088 Mbps upstream	Desconocido	Similar al ADSL
UDSL	Unidirectional DSL propuesto por una compañía Europea	Desconocido	Desconocido	Similar al ADSL
VDSL	Very High Digital Subscriber Line	12.9 a 52.8 Mbps downstream; 1,5 a 2.3 Mbps upstream	1,35 Km - 12.96 Mbps 0,9 Km - 25.82 Mbps 300 mts. - 51.84 Mbps	Red ATM, Fibra

Figura 9b. Variantes de xDSL

tancias moderadas, vía par de cables trenzados convencionales.

ADSL Lite.- Es una versión de velocidad más baja de ADSL, ha sido propuesta como una extensión para el estándar ANSI T1.413 por el Grupo de Trabajo ADSL Universal (UAWG) dirigido por Microsoft, Intel y Compaq. Utiliza el mismo esquema de modulación de ADSL, pero elimina el sistema telefónico viejo (POTS) de las premisas del usuario.

HDSL.- Es generalmente usada como un sustituto de T1/E1. Es una solución para proveer comunicación de datos simétrica full-duplex con velocidades arriba de 1.544 Mbps (2.048 Mbps en Europa) sobre dis-

HDSL.- Tiene un rango mayor que T1/E1, sin el uso de repetidores para permitir transmisión en distancias mayores a 4 Km. Utiliza modulación de amplitud de pulso (PAM) y es una técnica para ancho de banda alto, bidireccional sobre cableado de cobre para servicios T1 y E1.

RADSL.- Es una variación de velocidad de módem xDSL, pero se refiere específicamente a un estándar de modulación propietario designado por Semiconductores Globespan. Utiliza modulación de fase y ampli-

tud menor de portadora (CAP). Los módem DMT T1.413 estándar son también técnicamente RADSL.

Es una tecnología de transmisión que tiene ambas aplicaciones simétrica y asimétrica sobre una línea de par de cobre trenzado simple y con velocidades de datos arriba de 7 Mbps.

SDSL.- Es una implementación de HDSL de 2 cables. Soporta T1/E1 en un par simple a una distancia de casi 4 Km. El nombre se refiere a un servicio simétrico en una variedad de velocidades sobre un simple enlace. Es simétrico porque la velocidad de datos es la misma en ambas direcciones.

VDSL.- Es propuesto para enlaces locales más cortos, quizá hasta 1 Km. La velocidad de datos es de 10 Mbps.

CONCLUSIONES

La tecnología xDSL se ha convertido en un instrumento que está permitiendo la posibilidad de un resurgimiento a las redes telefónicas de cobre, en el preciso momento que la industria de multipares telefónicos comenzaba a cerrar sus puertas, y la de cables de fibra óptica y coaxial incrementaba día a día. xDSL permite a los proveedores ofrecer nuevos servicios a través de los pares de cobre, como son el acceso de banda ancha a Internet, transmisión de datos a alta velocidad y servicio de video en demanda. Para implementar esta tecnología sólo se necesitan los terminales xDSL apropiados, tanto en la oficina, casa o empresa y la central telefónica, y que las líneas de multipares se encuentren en buen estado. El acceso a Internet, el acceso a Redes LAN y otras necesidades de

servicio, son soportados debido a la compatibilidad de xDSL con los estándares tradicionales de los equipos de comunicaciones instalados. Usando estas capacidades, los proveedores podrán ofrecer un rango completo de servicios, organizándolos rápidamente, y asegurándose de un servicio excelente. Las soluciones xDSL también ofrecen a los proveedores de servicios la habilidad de maximizar los recursos de personal, utilizando empleados y habilidades existentes con gran eficiencia. Consecuentemente, sus clientes tendrán alto nivel de satisfacción y los proveedores podrán potencialmente experimentar una ganancia saludable sobre su inversión.

BIBLIOGRAFÍA

- [1] "Seminario de Nuevas Tecnologías", ALCATEL UNIVERSITY MÉXICO, Junio 2000.
- [2] Gerald P. Ryan. "ADSL, SDSL Copper Phone Line Technologie for Multimedia". Applied Technologie Group. 1995.
- [3] Daniel Minoli and Robert Keinath, "Distributed Multimedia Through Broadband Communications", Artech House, 1994.
- [4] Alan Steward, "ADSL: Facing up to the Future", Communication International, Septiembre de 1994.
- [5] T. VanLandegem, M. de Prycker, "Comunicaciones Eléctricas, 2005, Una visión de la red del futuro".
- [6] K. Sistanizadeh, P. S. Chow, J. M. Cioffi, "Multi-Tone Transmission for Asymmetric Digital Subscriber Lines", (ADSL) ICC93.
- [7] V. K. Bhagavath and B. Khasnabish. Emerging, "High-Speed xDSL Access Services: Architectures", Issues, Insights, and Implications, IEEE Communications Magazine, Nov. 1999
- [8] Marcelino Llano. "ADSL a Punto de su Implantación", Byte (Windows Magazine), Mayo. 1999
- [9] DSL and xDSL, <http://whatis.com/dsl.htm>
- [10] Ildefonso M. "Testing xDSL Lines", Polo Sunrise Telecom. Inc. Oct. 1999
- [11] "Ultima versión de la serie xDSL: Very High Speed Digital Subscriber Lines (VDSL)". Janse G. Judith T.
- [12] <http://neutron.ing.ucv.ve/revista-e/No4/VDSL.html>

Programación de Sistemas de Cómputo Móviles

M. en C. Juan Carlos Herrera Lozada,
M. en C. Israel Rivera Zárate,
Profesores del CIDETEC - IPN
Rodrigo Morales Alvarado,
Alumno de la Maestría en Tecnología de
Cómputo del CIDETEC - IPN

e-mails: [jcrs; irivera]@ipn.mx;
 moar82@yahoo.com

Este es el primero de una serie de artículos enfocados a la programación de sistemas de cómputo móviles, en particular los PDA. En esta entrega inicial se realiza un compendio de los términos comunes involucrados en este contexto, así como una aproximación a la programación de aplicaciones bajo Windows CE utilizando Embedded Visual Basic.

INTRODUCCIÓN

En la actualidad, los dispositivos móviles representan un campo tecnológico muy fuerte en cuanto a infraestructura y aplicaciones. La telefonía móvil y sobre todo las computadoras de bolsillo, también conocidas como PDAs (*Personal Digital Assistant* – Asistente Digital Personal), resultan elementos indispensables en la informática contemporánea para el intercambio y proceso de información.

De manera formal, considerando la entrada de datos hacia el sistema, la mayoría de los conocedores coin-

ciden que los PDA son las computadoras de bolsillo que admiten datos a través de un teclado gráfico estimulado a través de una pantalla táctil (*touchscreen*). Cuando el teclado es físico se dice que se trata de un *handheld*. Dada la proliferación actual, se tiende a hacer referencia de forma indistinta cuando se trata con dispositivos de estas características.

Para algunos otros autores, se debe estimar a las agendas electrónicas y a las calculadoras convencionales como otro tipo primitivo de *handhelds* o PDA, con la particularidad de que no tienen sistema operativo.

Aunque existen otras alternativas, como la posibilidad de instalar Linux, la competencia en el mercado de los PDA se libra actualmente entre las plataformas hardware que utilizan *Palm OS* y *Windows CE*.

Palm OS es un sistema operativo diseñado y actualizado por *Palm*., para computadoras de mano de este fabricante y algunos otros como *Sony*, *Kyocera*, *Samsung* y *Handspring*, que han adoptado este mismo administrador de recursos. Surgió en 1996 de la mano de la primera máquina *Palm Pilot*. La versión que más se está empleando hoy día es *Palm OS 5*, también referida como *Garnet* y, entre otras mejoras, soporta un mayor rango de resoluciones de pantalla y transmisión *bluetooth*.

En unos meses se liberará *Palm OS 6.1*, también denominada *Cobalt*, un sistema operativo de 32 bits completamente nativo y funcional dirigido a procesadores portátiles ARM (herederos de los antiguos procesadores *Dragonball*, más rápidos y eficientes); su principal novedad es que el sistema funcionará sobre *Linux*.

Windows CE (Compact Edition) representa la evolución de la experiencia de Microsoft sobre dispositivos móviles. Desde sus inicios, éste ha sido la base para algunas actualizaciones o estandarizaciones dependientes del fabricante del PDA, como por ejemplo *Windows Pocket PC*. En la actualidad Microsoft dirige este Sistema Operativo no sólo hacia los PDA, sino también a *Smartphones*, videojuegos de última generación y computadoras de automóviles; por tal motivo, ha denominado *Windows Mobile* a la última versión de *Windows CE (NET)* cuya característica radica en unificar el sistema operativo de los dispositivos portátiles. Los principales fabricantes de PDA bajo este administrador de recursos son *HP* (fusionado con *Compaq*), *Dell*, *Toshiba*, *Viewsonic* y *Casio*.

El sistema operativo *Windows CE* está basado en la interfaz de programación para aplicaciones *Win32 (API)*. Como en las otras versiones de *Windows*, *Windows Pocket* posee un modelo de programación orientado a eventos: recibe mensajes, interpreta mensajes y actúa sobre los

mensajes. Estos programas tienen una o más ventanas que reciben y procesan mensajes en un ciclo.

De acuerdo a diversos estudios comparativos publicados en línea, Palm OS es más compacto y estable que Windows CE, pero menos versátil en cuanto a soporte de comunicaciones, manejo de recursos y complejidad de procesamiento. Tomando en consideración que los PDA no tienen disco duro y que, por consiguiente el sistema operativo se almacena en una memoria en circuito integrado, Windows Mobile requiere utilizar más memoria que Palm OS, tanto al iniciar como al ejecutar aplicaciones.

En nuestro país en particular, existe más soporte de software y hardware para Palm OS que para Windows CE. A nivel mundial, según datos de *International Data Consulting*, PALM OS acapara un 50% del mercado tecnológico basado en PDA, Windows CE tiene sólo un 25% y el resto se divide entre otros competidores.

PRESTACIONES DE LOS PDAs

Indistintamente del sistema operativo, los PDAs incluyen software básico de manera precargada: correo electrónico, agenda, editores de texto, hojas de cálculo, navegador web, sincronizador (intercambio de archivos entre la Computadora de escritorio y el PDA), entre otros. Para comunicaciones poseen varios puertos, entre los que destacan el puerto serie con estándar RS-232, el puerto de infrarrojo y la ranura de expansión para tarjetas SD (*Secure Digital Memory Card*). En la actualidad se han incorporado puertos para tarjetas PCMCIA (*Personal Computer Memory Card International Association*) y *Compact Flash*.

El soporte en software opcional incluye editores de presentaciones, juegos, visualización de aplicaciones Flash y Java, visor de archivos tipo *pdf* y reproducción de música y videos directamente en formatos multimedia convencionales, entre una gran variedad. Algunos de los más recientes y sofisticados incluyen hardware integrado para conexión *Wireless* (inalámbrica), *Bluetooth* (inalámbrico con enlace a través de radiofrecuencia), *GSM* para telefonía celular y cámara fotográfica.

A continuación se tiene un glosario de términos comunes que se utilizan alrededor de este tipo de recursos de cómputo. En la **Tabla 1**, es posible advertir con mayor detalle dónde aplican los recursos definidos.

Memoria RAM. Es la memoria donde se pueden almacenar archivos de datos del usuario y correr programas. En la mayoría de PDA se trata de memorias *SDRAM* (RAM Dinámica Síncrona), donde a mayor cantidad de memoria es posible tener más archivos abiertos al mismo tiempo o guardar más información.

Memoria ROM. Es la memoria principal del PDA, donde normalmente se almacena el sistema operativo. En la mayoría de PDA es de tipo *FLASH ROM*, que permite actualizarse vía software sin necesidad de desprender el circuito de la tarjeta madre.

Multimedia. Se refiere a la reproducción y/o grabación de música, voz, imágenes y video, así como a la posibilidad de utilizar juegos electrónicos. Las PDA actuales reproducen archivos *MP3* y permiten visualizar gráficos en diferentes formatos, entre éstos el *JPEG*. La *Pocket PC* soporta *Windows Media Player* y la reproducción de videos *WM* y *MPEG*.

Navegación. Este aspecto se refiere a los pasos que se deben realizar con el lápiz óptico sobre la pantalla táctil (*touchscreen*) para pasar de un programa a otro, administrar y visualizar archivos, etc. También es posible realizar algunas opciones de navegación con las teclas de acceso directo que tiene el PDA.

Puertos de comunicación (Conectividad). Son los circuitos electrónicos que permiten el intercambio de información entre el PDA y otros dispositivos como computadoras, teléfonos celulares, impresoras, cámaras digitales, otro PDA, etc.

A continuación se explican los principales puertos:

USB. El *Bus Serial Universal* es el estándar más comúnmente utilizado para sincronizar el PDA con la computadora.

Serial. Soporta el protocolo RS-232 para la transmisión serie asíncrona. Este puerto se detallará en un artículo posterior.

Infrarrojo o IrDA. Validado por la Asociación de Datos Infrarrojos (*Infrared Data Association*), se trata de un estándar desarrollado por un grupo de fabricantes de dispositivos para transmitir datos a través de ondas de luz. Esta conexión inalámbrica se transmite a muy baja velocidad y, al igual que un control remoto de TV, se debe tener línea de vista con el receptor ya que cualquier desviación u obstáculo impide la conexión.

Bluetooth. Es una nueva forma de transmisión por radiofrecuencia entre aparatos electrónicos que cuenten con tarjeta de este mismo tipo; es más rápida, tiene un alcance de hasta 10 metros y no requiere de línea de vista. Se

trata de un estándar Wireless LAN (inalámbrico para redes tipo LAN) que surge con la finalidad de simplificar el mecanismo para intercambiar información en una forma serial; además, está optimizado para aplicaciones en Internet.

Wi-Fi (Fidelidad Inalámbrica - *Wireless Fidelity*) (estándar 802.11b). Es una conexión inalámbrica por radiofrecuencia con un alcance de hasta 100 metros que se emplea principalmente para tener acceso a una red de cómputo LAN o a una conexión inalámbrica a Internet. En México ya hay puntos de acceso Wi-Fi (llamados *hot spots*) en algunas escuelas, tiendas y cafés, que permiten conectarse a Internet dentro de sus instalaciones, sin mayor complicación.

Ranuras (slot) y memorias de expansión. Es el espacio físico o *connector* que ofrece el PDA para insertar una tarjeta de memoria electrónica y disponer de mayor espacio para almacenar información. Otra función, es la de adecuar una conexión (interfaz hardware) hacia un dispositivo externo, como podría ser una cámara digital, un módem, una tarjeta de red, un lector de código de barras, etc.

En el caso de insertar una tarjeta de memoria u otro dispositivo, éste debe coincidir con el tipo de ranura en la que se insertará. Las memorias y las ranuras correspondientes se indican a continuación:

CF (*CompactFlash tipo I y tipo II*). Es grande en dimensiones respecto a los otros tipos, a pesar de ello es la más rápida en su funcionamiento y tiene capacidades que van de los 8 MB hasta los 4 GB. Es la más barata, aunque en algunos casos se requiere de una tarjeta de expansión, propiciando que se vea sacrificada la portabilidad del PDA.

MS (*Memory Stick* estandarizada por Sony). Es pequeña y delgada, aunque no tan rápida como la CF, y con capacidad máxima de 256 MB. La versión MS PRO alcanza hasta 1 GB de capacidad, pero no es compatible con las primeras ranuras MS originales. Se trata de las memorias que utilizan la mayoría de las cámaras digitales.

SD (*Secure Digital*). Es la más delgada –del tamaño de un timbre postal–, pero no es tan rápida como la CF, y su capacidad máxima actual es de 1 GB. Este tipo de

memoria, así como su respectiva ranura, son las más utilizadas en el mercado actual de las PDA.

MMC (*Multimedia Card*). Es similar en dimensiones a la SD, por lo que se puede insertar en ranuras SD; sin embargo, es más lenta y su capacidad puede ser de 128 MB o 512 MB. Su propósito específico es aligerar el trabajo del procesador, con respecto a aplicaciones multimedia, incluidas en éstas.

SDIO Card (*Secure Digital Input/Output*). Son tarjetas SD que agregan otras funciones como acceso a redes inalámbricas (Wi-Fi o Bluetooth), receptores GPS (Global Positioning System) o cámaras digitales. Algunas tarjetas SDIO son compatibles con las ranuras SD convencionales.

Resolución de la pantalla. Es la cantidad de puntos de imagen (*píxeles*) que pueden visualizarse en la pantalla del PDA; se indica como la cantidad de puntos horizontales

PDA	Sistema Operativo	Procesador @ velocidad	Conectividad Integrada	Expansión	SDRAM	FLASH ROM	Duración de la Batería	Resolución de Pantalla Táctil	Adicionales	
iPAQ h3950 Pocket PC	Microsoft Windows Pocket PC 2002	Intel PXA250 @400MHz	IrDA, USB, serial	SD, PCMCIA	64MB	32MB	10 hrs	240 x 320 pixels, Transflective TFT, 65000 colores		
HP iPAQ rx3417	Microsoft Windows Mobile 2003 Second Edition	Samsung S3C 2440 @300MHz	WLAN 802.11b, Bluetooth, IrDA, USB, serial	SD, SDIO	128MB	32MB	10 hrs	240 x 320 pixels, Transflective TFT, 65000 colores	Cámara Digital, Photosmart, 1.3 MP 1280 x 960, Zoom digital 4x	
iPAQ h5550 Pocket PC	Windows Mobile 2003	Intel XScale @400MHz	WLAN 802.11b, Bluetooth, IrDA, USB, serial	SD, PCMCIA	128MB	48MB	10 hrs	240 x 320 pixels, Transflective TFT, 65000 colores	Sistema de seguridad, identificación huella dactilar	
iPAQ h2210 PPC	Microsoft Windows Mobile 2003 Premium Edition	Intel XScale @400 MHz ARM	Bluetooth, IrDA, USB, serial	SD, SDIO, MMC, Compact Flash tipo I y II	64MB	32MB	10 hrs	240 x 320 pixels, Transflective TFT, 65000 colores		
palmOne Tungsten C	Palm OS 5.2.1	Intel XScale @ 400 MHz ARM	WLAN 802.11b, IrDA, USB, serial	SD, MMC	64MB	16MB	20 hrs	320 x 320 pixels, Transflective TFT, 65000 colores		
palmOne Zire 72s (Silver)	Palm OS 5.2.8	Intel PXA270 @312MHz	Bluetooth, IrDA, USB, serial	SD, MMC, SDIO	32MB	8MB	20 hrs	320 x 320 pixels, Transflective TFT, 65000 colores	Cámara Digital, 1280 x 960, Zoom digital 2x	
Dell Axim X50 Pocket PC	Microsoft Windows Mobile 2003 Second Edition	Intel XScale PXA270 @520MHz	WLAN 802.11b, Bluetooth, IrDA, USB, serial	SD, SDIO, MMC, Compact Flash tipo I y II	64MB	128MB, Intel StrataFlash	10 hrs	240 x 320 pixels, Transflective TFT, 65000 colores		

Tabla 1. Comparación de prestaciones entre diferentes PDAs.

contra la cantidad de puntos verticales.

PROGRAMACIÓN DE APLICACIONES

Como ya se mencionó con anterioridad, en dependencia a la plataforma del PDA, ya sea Palm OS o Windows CE (en cualquiera de sus estandarizaciones), es necesario elegir la herramienta de programación adecuada considerando la versión correcta del sistema operativo.

Para complementar este artículo, se realiza la programación de dos aplicaciones simples bajo Windows CE. El PDA utilizado es una *iPAQ Pocket PC* de *Compaq*, modelo 3950 (primer PDA de la *Tabla 1*) con sistema operativo *Windows Pocket PC 2002* precargado de fábrica.

Para el sistema operativo especificado se encontraron las siguientes herramientas de desarrollo: *Embedded Visual Tools* (utilizada en el desarrollo de este artículo), *Mobility Java Technology*, *PocketC*, *Visual CE*, *CodeWarrior*, *CASL*, entre otras.

La decisión de utilizar *Embedded Visual Tools* estuvo determinada por cinco puntos favorables:

1. Su compatibilidad directa con Windows, al pertenecer al mismo fabricante.
2. Es de distribución gratuita, previo registro.
3. En el caso de *Embedded Visual Basic*, se tiene un lenguaje visual que es de fácil manejo.
4. No tiene dificultades para realizar la sincronización entre el PDA y la PC de escritorio, permitiendo la evaluación y depuración directa de programas.

5. Incluye un emulador del PDA, que funciona correctamente siempre y cuando la aplicación no requiera mostrar imágenes. En esta situación, lo recomendable es comprobar el funcionamiento del programa directamente en la *iPAQ*.

Este ambiente de desarrollo contiene *Microsoft Embedded Visual C++ 3.0* y *Microsoft Embedded Visual Basic 3.0*, con los kits de desarrollo (*SDK*) para *Pocket PC 2002* y *Smartphone 2002*. Permite crear, probar y depurar una aplicación diseñada para Windows CE o Pocket PC; los diseñadores que estén familiarizados con Visual Basic o Visual C, se encontrarán con interfaces muy similares a las respectivas versiones para PC, pero con las limitaciones lógicas del dispositivo portátil, razón por la cual es una de las herramientas más populares.

Para programar sobre Windows Pocket 2002, la versión apropiada de *Embedded Visual Tools* es la 3.0, que se puede obtener de los siguientes sitios:

<http://www.microsoft.com/windows/embedded/ce/tools/factsheet.asp>

<http://www.microsoft.com/mobile/downloads/emvt30.asp>

De acuerdo al fabricante, los requerimientos mínimos especifican tener instalado Windows 98 ó superior. Para poder acceder al emulador de PDA, se requiere Windows 2000 ó superior, así como una conexión activa TCP/IP.

EMBEDDED VISUAL BASIC

Pequeño y ligero, *Embedded Visual Basic (EVB)* es un subconjunto de la versión de escritorio, combinando el poder y la robustez de Visual Basic con la portabilidad y fácil manejo, de VBScript.

El *IDE (Integrated Development Environment - Ambiente Integrado de Desarrollo)* provee muchas opciones para la generación de aplicaciones: se pueden usar los paquetes de formularios estándares, o se pueden diseñar aplicaciones sin formularios o interfaces visibles. Adicionalmente, se pueden usar controles *ActiveX* para extender las capacidades básicas de una aplicación.

EVB se instala en una PC y desde ésta se programa la aplicación que posteriormente se descargará al PDA a través del cable de sincronización.

Generalmente, desarrollar en *Embedded Visual Basic* es similar a desarrollar aplicaciones de Visual Basic estándar, es decir, no se generan programas ejecutables directamente (.exe), sino que se generan archivos con la extensión *.vb, estos archivos son una especie de *script* muy optimizado con los datos exactos para ser mostrados en la interfaz gráfica del PDA.

Los archivos *.vb son enlazados a un programa ejecutable (*pvbload.exe*) de manera similar a como se enlazan los archivos *.doc a Word. Este programa se encarga de interpretar los comandos que se encuentran en el archivo *.vb, traduciéndolos y ejecutándolos con ayuda de bibliotecas de enlace dinámico (*.dll) en Windows CE. El *runtime* de *Embedded Visual Basic* está presente en la ROM del sistema operativo.

DESARROLLO DE APLICACIONES

La metodología lógica infiere programar la aplicación en EVB y posteriormente descargarla en el PDA. De manera similar a Visual Basic, el ambiente de desarrollo está basado en la generación de un proyecto. Al acceder a EVB, se obtendrá la pantalla de la *figura 1*. En ésta se declara un

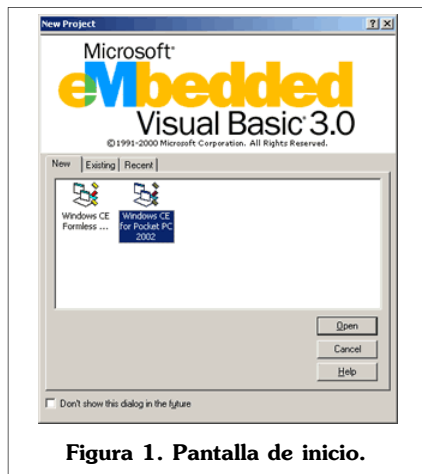


Figura 1. Pantalla de inicio.

nuevo proyecto con la opción *Visual Basic Windows CE for Pocket PC 2002*.

La pantalla de la **figura 2** muestra el IDE en tiempo de diseño para crear la aplicación; se han resaltado cuatro secciones:

1. Explorador de Proyectos.
2. Ventana de Propiedades.
3. Barra de herramientas estándar (Toolbox).
4. Área de trabajo, donde es posible conmutar entre la ventana de código y la ventana del diseño de formulario.

Como una primera aplicación, se propone un cálculo sencillo para estimar las cantidades desglosadas de un recibo de honorarios para efectos de retenciones y pago de impuestos

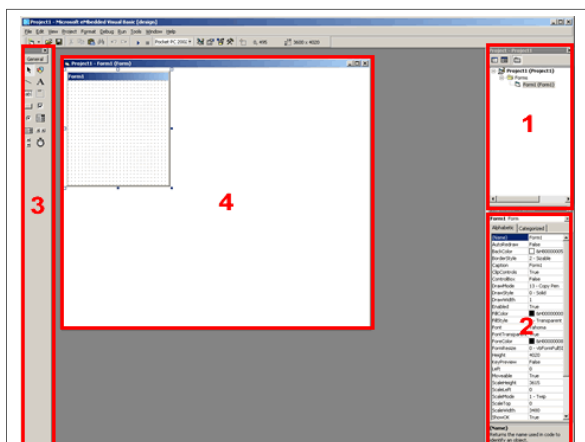


Figura 2. Pantalla en Tiempo de Diseño.

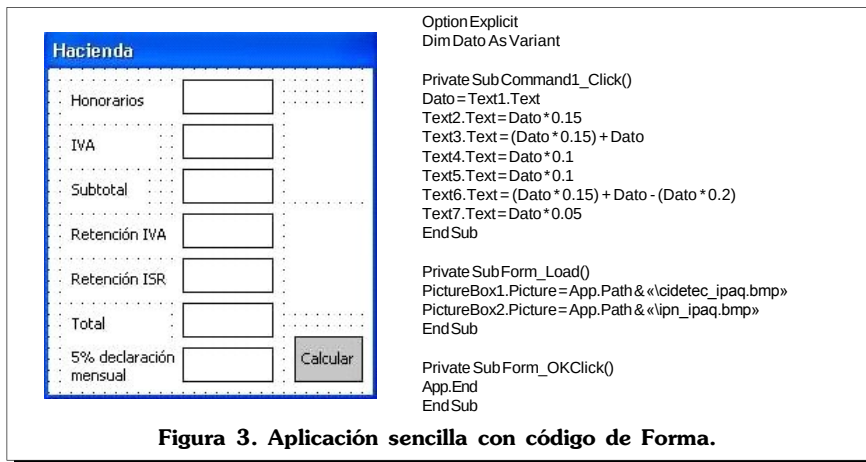


Figura 3. Aplicación sencilla con código de Forma.

causados. El *Formulario* o *Forma* (*Form*) sólo contiene componentes intrínsecos seleccionables de la *Toolbox*: cajas de texto (*TextBox*), etiquetas (*Label*) y un botón de comando (*CommandButton*) para validar el cálculo. La **figura 3** muestra el formulario en tiempo de diseño y el código que lo conforma.

Es importante mencionar que en emulación no resulta sencillo trabajar con imágenes debido a que es necesario configurar el IDE, a la vez de que no es garantía de que en todas las versiones de Windows para PC funcione correctamente, por lo que los controles *PictureBox*, *ImageControl* y similares no se deben indicar en código para emular una aplicación. En la aplicación real descargada al PDA, el uso de estos controles no presenta ningún inconveniente.

En el código EVB listado paralelamente con el formulario de la **Figura 3**, se están utilizando dos controles *PictureBox* para acceder a los logotipos del CIDETEC y del IPN. En la emulación fue necesario suprimir estas líneas del código, así como configurar las propiedades del proyecto, tal y como se aprecia en las **Figuras 4 y 5**.

Obsérvese que la opción *Target* especifica si se trabajará en emulación (*Pocket PC 2002 Emulation*) o si se descargará el diseño hacia un PDA, en este caso, la *Pocket PC 2002 (Default Device)*. La ruta donde se descargará la aplicación se debe especificar en el campo *Remote Path*, consi-

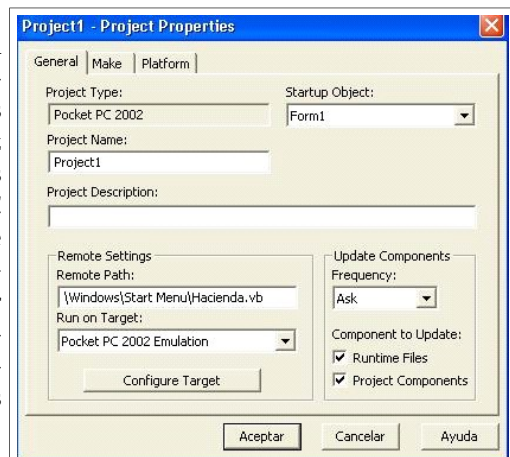


Figura 4. Propiedades del Proyecto.

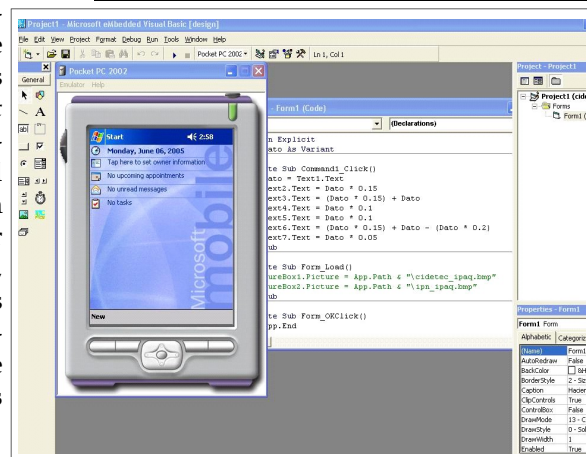


Figura 5. Accediendo al emulador.

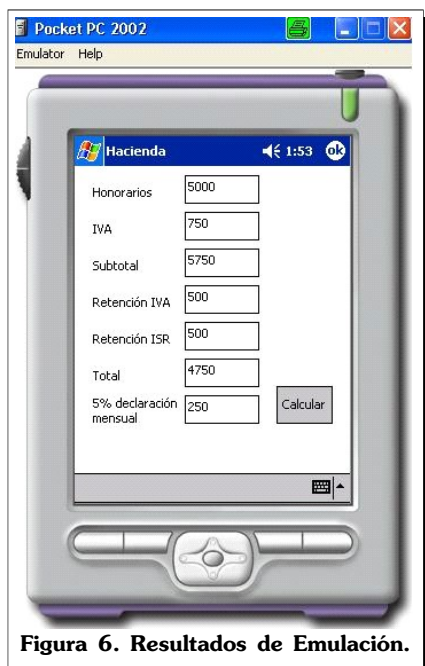


Figura 6. Resultados de Emulación.

derando que será diferente para el dispositivo físico. A través de la opción *Run (Start Debug)* dentro del menú, se inicia la emulación o la descarga del archivo hacia el PDA, según se haya elegido. La **Figura 6** muestra la pantalla de emulación, sin imágenes. En la **Figura 7** se observa a la iPAQ ejecutando la aplicación.

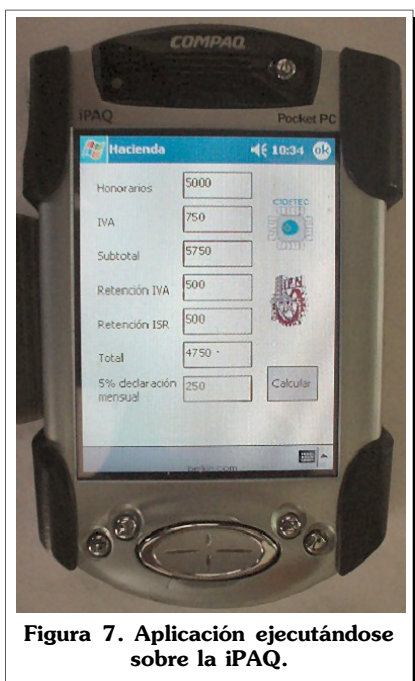


Figura 7. Aplicación ejecutándose sobre la iPAQ.

Con la intención de mostrar una aproximación a los métodos de graficación, se generó una segunda aplicación que permite seleccionar funciones trigonométricas. Dentro del formulario de diseño, se adiciona un control *PictureBox* donde se graficará la función; este componente se debe adecuar con los siguientes valores en el área de propiedades (sección 2 de la **Figura 2**):

Name PBPlano
Height 1575
Width 3000
ScaleHeight -2.4
ScaleWidth 25.2
ScaleLeft -12.3
ScaleTop 1.2

Se requieren tres botones de opción (*OptionButton*), llamados también de radio; dos botones de comando (*CommandButton*) y algunas etiquetas (*Label*). El formulario en tiempo de diseño se muestra en la **Figura 8**. Cabe mencionar que al igual que en la aplicación anterior, las imágenes utilizadas sólo son visibles en tiempo de ejecución y sobre la iPAQ; el formato soportado es mapa de bits de 16 colores.



Figura 8. Forma para la graficación de funciones trigonométricas



Figura 9. Graficación de funciones trigonométricas sobre iPAQ.

Las propiedades *ScaleHeight* y *ScaleTop* determinan la escala del eje de las ordenadas (*Y*) con el propósito de representar los valores -1.2 y 1.2, forzando a utilizar unidades definidas por el diseñador (*PicUser*). De la misma manera se determina la escala del eje de las abscisas (*X*) entre -12.6 y 12.6.

La **Figura 9** exhibe la graficación de funciones trigonométricas en tiempo de ejecución.

CONCLUSIONES

Hoy en día, el campo de los Sistemas de Cómputo móviles presenta un panorama atractivo para los programadores debido a la amplia gama de dispositivos y diferentes tecnologías disponibles que van desde sencillas aplicaciones *stand-alone* hasta aplicaciones más complejas como celulares y bases datos en tiempo real.

La programación de aplicaciones para PDA se facilita cuando se utiliza un lenguaje como Embedded Visual Basic, debido a la similitud que tiene con Visual Basic para PC.

```
Option Explicit
Dim X, Y

Sub Graph_plano()
    PBPlano.DrawLine -12.6, 0, 12.6, 0, vbGreen
    PBPlano.DrawLine 0, 1.2, 0, -1.2, vbGreen
End Sub

Private Sub bt_clear_Click()
    PBPlano.Cls
End Sub

Private Sub Bt_Graficar_Click()
    Graph_plano
    If Op_sen.Value = True Then
        Graph_Sen
    ElseIf Op_cos.Value = True Then
        Graph_Cos
    ElseIf Op_tan.Value = True Then
        Graph_Tan
    Else
        MsgBox «Selecciona una función!»
    End If
End Sub

Private Sub Form_OKClick()
    App.End
End Sub

Sub Graph_Sen()
    'Dibuja la curva
    For X = -12.6 To 12.6 Step 0.05
        Y = Sin(X)
        PBPlano.DrawPoint X, Y, vbYellow
    Next
End Sub

Sub Graph_Cos()
    'Dibuja la curva
    For X = -12.6 To 12.6 Step 0.05
        Y = Cos(X)
        PBPlano.DrawPoint X, Y, vbWhite
    Next
End Sub

Sub Graph_Tan()
    'Dibuja la curva
    For X = -12.6 To 12.6 Step 0.05
        Y = Tan(X)
        PBPlano.DrawPoint X, Y, vbRed
    Next
End Sub

Private Sub Form_Load()
    PictureBox1.Picture = «Windows/Inicio/cidetec_ipaq.bmp»
    PictureBox2.Picture = «Windows/Inicio/ipn_ipaq.bmp»
End Sub
```

Es importante entender que, en lo general, las restricciones principales de los PDA radican en el sistema operativo y en la memoria de usuario, por lo que se debe buscar la manera de comprometer lo menos posible el trabajo del procesador a través de un programa bien estructurado.

Dados los alcances introductorios de este artículo, no se han analizado otras facilidades de EVB para generar aplicaciones más sofisticadas que se ejecuten sobre alguna versión de Windows CE; sin embargo, la metodología aquí expuesta es la misma que se seguiría para programar y descargar una aplicación más completa.

Se deja abierta la posibilidad de probar los programas desarrollados en este documento sobre otro PDA. Sería interesante utilizar una plataforma Windows actual; a la vez, considerando las respectivas diferencias en cuanto a sistemas operativos y herramientas de programación, investigar la plataforma PALM.

En un próximo artículo se detallará cómo diseñar una interfaz hardware que envíe datos hacia la iPAQ a través del puerto serie de la misma.

BIBLIOGRAFÍA

- [1] Fco. Javier Ceballos; *"Curso de Programación de Visual Basic 6"*. Alfaomega Ra-Ma. México. 2003
- [2] *"Diseño de aplicaciones inalámbricas móviles"*. <http://www.intel.com/espanol/update/contents/mo11031.htm>
- [3] *"eMbedded Visual Basic Online Documentation"*. <http://www.microsoft.com/embedded/> <http://www.microsoft.com/windows/embedded/tools/>
- [4] *FORO de usuarios Palm*. <http://www.piensaenpalm.com/foros/>
- [5] *"Introduction to Pocket PC 2002 Programming"*. http://www.devhood.com/tutorials/tutorial_details.aspx?tutorial_id=272
- [6] *"Portal Visual Basic"*. <http://www.portalvb.com/articulos>
- [7] *"Welcome to Trajecto's Win 32 API Programming Tutorial for Pocket PC (Really Fast Graphics Programming)"*. http://www.trajectorylabs.com/win32_proto_type.html

Algoritmo de Escape para Crear un Fractal Utilizando MPI

Ing. Jesús Antonio Álvarez Cedillo.
Profesor del CIDETEC-IPN.

Se define como fractal a aquellos objetos cuya creación depende de las reglas de irregularidad o de fragmentación al ser estudiadas por un proceso matemático determinado, es definida también como un objeto matemático de dimensión no entera.

Según B. Mandelbrot [1] se considera fractal a aquel objeto o estructura que consta de fragmentos con orientación y tamaño variable pero de aspecto similar. Esta característica la da al fractal algunas propiedades geométricas especiales en que dependen de su longitud y la relación entre el área de su superficie y su volumen.

Estas propiedades especiales hacen que se requieran otras herramientas matemáticas diferentes a las comunes para poder explicar sus características. En el cuerpo humano existen estructuras con geometría fractal, como son la red vascular, las ramificaciones bronquiales, la red neuronal, la disposición de las glándulas, etc.

La importancia que tiene esta geometría fractal en el organismo es que permite optimizar la función de los sistemas debido a que en el mínimo espacio tienen la máxima superficie. Al existir estructuras con geometría fractal deducimos que deben ser posibles los fenómenos con características fractales al poder poseer estos fenómenos unos patrones que se repiten constantemente en diferentes escalas de tiempo. Es importante mencionar que cumple con las reglas de paralelización al tener la característica antes mencionada.

Estos fenómenos son analizados por lo general con el uso de herramientas matemáticas de la geometría fractal. De acuerdo a H. P. Koch [2], la teoría fractal puede ser considerada como una herramienta válida y útil para el estudio de fenómenos dinámicos en el cuerpo humano o

en la naturaleza y nos permite una aproximación más acorde con la complejidad y la ausencia de linealidad existente en dichos procesos.

APLICACIONES DE LOS FRACTALES

Aunque puedan parecer simples figuras creadas para uso y entretenimiento de los matemáticos, existen muchas aplicaciones de los fractales tanto a nivel teórico como en el práctico. Considerando la gran variedad de áreas de estudio científico y su gran importancia es imposible hacer una lista de aplicaciones por lo que consideré para este trabajo solo las más importantes.

Su aplicación en el campo de las ciencias abstractas ha sido realmente grande y una de sus aplicaciones en donde se ha comportado como una innovación, es el estudio de las soluciones de sistemas de ecuaciones de segundo grado. Aunque se han hecho consideraciones donde son utilizados los fractales para resolver ecuaciones.

El matemático John Hubbard, de nacionalidad estadounidense, presento un trabajo utilizando el método de Newton [3], para resolver ecuaciones, para realizar esto evaluó desde diferentes puntos iniciales a cada una de las soluciones. Usando su computadora para realizar la exploración y asignando un color a cada cuenca, Hubbard comprobó que los límites de estas regiones del plano no estaban bien definidos. En estos límites se encontraba puntos de un color dentro de otros puntos de color y conforme la rejilla de números y se iba ampliando a medida que más compleja se iba revelando la frontera. En realidad, podía considerarse que no existía tal frontera.

Generalmente las aplicaciones más comunes son generadas por la física o la sismología, pero el campo donde más aplicaciones se han encontrado ha sido en el tratamiento de las imágenes.

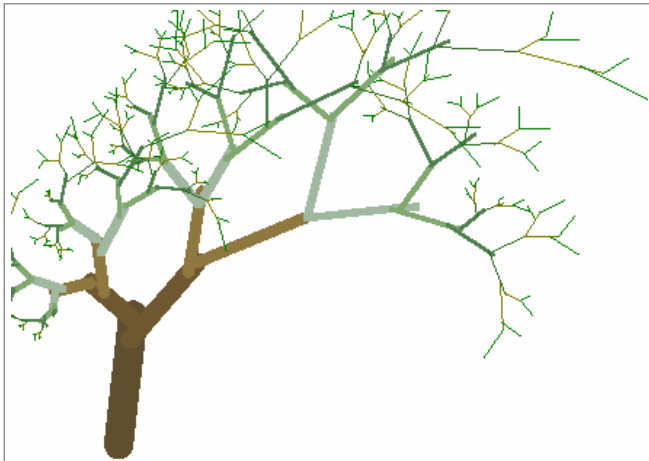


Figura 1. Simulación de un árbol utilizando un fractal.

Michael Barnsley fue el pionero en el tratamiento de imágenes a partir de su denominada transformación fractal [4]. Ésta consiste en el proceso contrario a la formación de un fractal, es decir, en lugar de crear una figura a partir de unas reglas determinadas, se buscan las reglas que forman una figura más determinada. Los fractales se utilizan para comprimir imágenes digitalizadas de forma que ocupen menos espacio y puedan ser transmitidas a una mayor velocidad y con menores recursos de computo.

En la industria del cine es de gran utilidad a la hora de crear los espectaculares efectos especiales de las grandes producciones, ya que es fácil crear todo tipo de paisajes y fondos a través de los fractales.

En la **figura 1** se ve un fractal generado por un programa de alto rendimiento en gráficos que simula un árbol.

En el mundo de la música los fractales juegan un papel muy importante en la composición, en especial, de música tecno o de bases rítmicas para cualquier otro tipo de música. En el mundo de la biología, se pueden apreciar grandes ejemplos de estructuras fractales en el cuerpo humano como la red de venas y arterias. A partir de un vaso sanguíneo grande como la aorta va saliendo vasos más pequeños hasta la aparición de los finísimos capilares de forma que cubran el mayor espacio posible para llevar nutrientes a las células. Por otro, se cree adivinar cierta similitud entre la generación de fractales y el código genético, ya que en ambos casos a partir de una información muy reducida en apariencia, surgen complejas estructuras.

MARCO TEÓRICO

FRACTALES Y CAOS

La dimensión fractal es un índice matemático que es posible calcular y que nos permite cuantificar las características de los objetos o fenómenos fractales. Este índice puede calcularse de varias formas. Por lo general es utilizado un modelo matemático llamado el exponente de Hurst.

La concepción de dimensión que nosotros usamos normalmente en la euclidiana clásica, es la que una dimensión es una recta, dos dimensiones forman un plano y tres dimensiones forman un objeto con volumen. A estas coordenadas les llamamos el eje X o largo, el eje Y o alto y el eje Z profundidad o espesor. Ver **figura 2**.

La geometría fractal y los descubrimientos de la ciencia del Caos se basan en los números complejos. A diferencia del resto de los números, los números complejos no existen sobre la recta numérica en lo absoluto, ni siquiera en aproximación, como en el caso de los números reales. Los números complejos solo existen en un plano de x-y que involucra a los llamados números imaginarios. Tienen únicamente una referencia indirecta con la recta numérica.

LOS NÚMEROS COMPLEJOS

Los números complejos son una combinación de números imaginarios que no tienen lugar físico sobre la

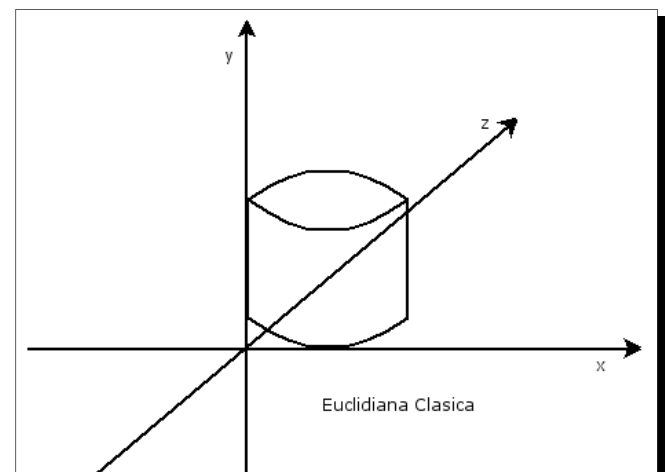


Figura 2. Representación de una dimensión en la euclidiana clásica.

recta numérica, y de cualquier otro tipo de número que si tenga un lugar definido sobre la recta numérica. En las matemáticas los números complejos se representan mediante una letra z , la cual define al número complejo como:

$$z = a + b_i$$

donde:

a = número real

b_i = número imaginario

Tanto las partes real como la imaginaria del número complejo pueden ser positivas o negativas, y cualquiera de los dos puede ser entero o fraccional. Los números complejos pueden fácilmente sumarse, restarse, multiplicarse o dividirse. El uso de x y y proporciona una referencia al plano cartesiano, lo cual ayuda al mejor entendimiento de los números complejos.

El eje x es la representación de la recta numérica, que se representa en forma horizontal. Mientras que el eje y representa la línea de los números imaginarios, de raíces cuadradas negativas. El eje y se muestra verticalmente para crear un plano de números complejos.

como $z = x + Hill$

x = número real

y_i = número imaginario

Todo número no-complejo tiene un lugar sobre la unidimensional recta numérica, mientras que todo número complejo tiene un lugar sobre el plano más amplio de dos dimensiones de los números que se denomina como el plano complejo. De manera que para localizar un número complejo hay que referirse al eje horizontal para localizar su parte numérica real y al eje vertical para ubicar su parte numérica imaginaria. Esto contrasta con todos los números que pueden localizarse en un punto que se encuentra sobre una línea recta unidimensional.

La representación de un numero en el plano por ejemplo $2-3i$ complejo seria mostrado en la **figura 3**.

Si ahora consideramos que existe una línea irregular, esta tiende a formar una superficie, y si se dobla se convierte en un volumen; esta es una técnica muy general de convertir un objeto en un plano en un objeto con tres dimensiones. Muchas estructuras naturales tienen

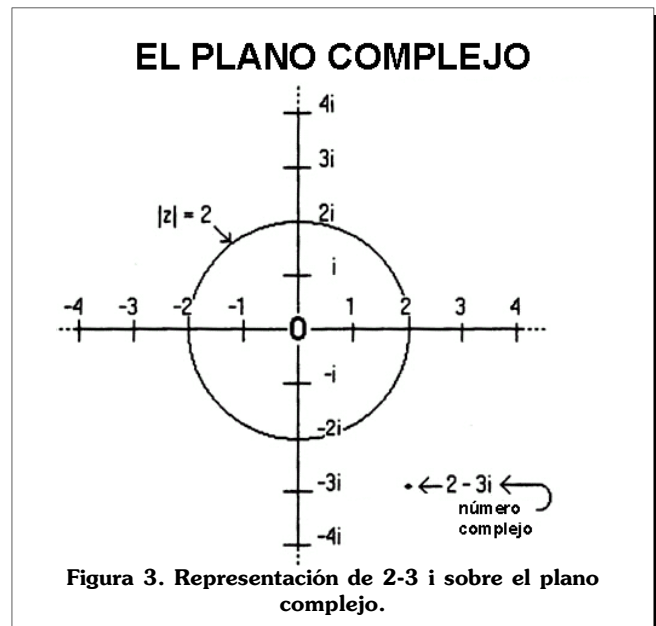


Figura 3. Representación de $2-3i$ sobre el plano complejo.

estas características; por lo que, geoméricamente, estas estructuras podrían tener una dimensión no entera entre dos y tres dimensiones.

De esta forma la dimensión fractal es un índice que nos permite analizar mejor las características geométricas de los objetos usando la geometría fractal. Los fenómenos con comportamiento fractal pueden ser representados por medio de gráficos de líneas, y en estos gráficos se mide su dimensión fractal y de esta forma el análisis de la complejidad de su dinámica caótica. Es posible considerar que la relación más importante entre el caos y los fractales, esta en que los fractales son la representación grafica del caos.

La relación existente entre el caos y los fractales es posible ya que los fractales son figuras geométricas con un cierto patrón que se repite infinitamente y a múltiples escalas, si son observadas detenidamente se podrá descubrir que en ese patrón se encontrarán los mismos componentes hasta el infinito. Esto lo podemos verificar si observamos el fractal a diversas escalas cada vez más pequeñas. Un fractal sin dimensión entera representa la forma grafica en la solución de las ecuaciones caóticas.

De manera simple entonces es posible definir un fractal como una figura geométrica con una estructura muy compleja y pormenorizada a cualquier escala la cual tiene las siguientes características:

1. Los fractales son figuras geométricas, aunque no se ajusten y sea imposible su definición por medio de los conceptos y métodos clásicos vigentes desde Euclides.

2. Se originan a partir de unas situaciones iniciales o reglas muy básicas, que darán lugar a figuras extremadamente complejas.
3. La autosemejanza.. Es definida como la simetría existente dentro de una escala, esta característica los vuelve recurrentes y paralelizables.

ANÁLISIS DEL ALGORITMO Y SU SOLUCIÓN PARALELA

EL FRACTAL DE MANDELBROT

El fractal mas conocido y sin duda él mas analizado es el fractal de Mandelbrot. Para muchos expertos se trata del objeto más complejo de todas las ciencias exactas. Resulta increíble observar su infinita complejidad, y su belleza la cual es indescriptible. Ver **figura 4**.

El descubrimiento y su definición formal fue un proceso muy complejo y fue precedido por los estudios de Gaston Julia y Pierre Gatou durante la I Guerra Mundial. Estos estudios originaron los fractales de Julia y fueron redescubiertos por Benoit Mandelbrot a finales de los 70.

El fractal de Mandelbrot consiste en una serie de números complejos que cumplen una propiedad matemática en la cual, cada numero está compuesto por una parte real y otra imaginaria representada por i, que es igual a la raíz cuadrada de -1, de la siguiente forma:

$$2 + 3i.$$

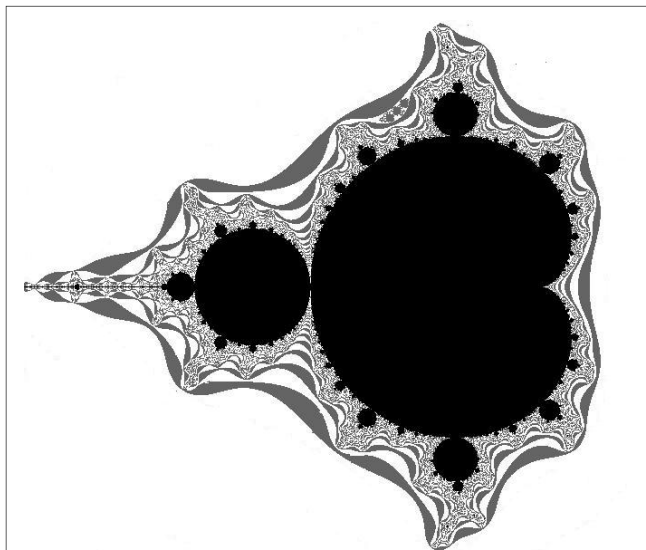


Figura 4. Fractal Mandelbrot realizado en MPI.

Entonces, tomamos, un número cualquiera C y lo elevamos al cuadrado. Al número obtenido le sumamos Cy lo volvemos a elevar al cuadrado y continuamos una y otra vez con el mismo proceso:

Esto genera la siguiente ecuación:

$$z \rightarrow z^2 + C \quad (1)$$

Es posible que puedan existir dos resultados en este proceso iterativo:

- a. El número puede escapar al infinito alejándose del centro del plano y no pertenece al fractal de Mandelbrot.
- b. El numero obtenido y el punto se encuadre se encuentran dentro del fractal.

CONSTRUCCIÓN DEL ALGORITMO

Los puntos que pertenecen al Conjunto de Mandelbrot son aquellos puntos que hacen que partiendo del origen, mantienen al menos de 2 unidades del origen, independientemente del número de puntos a evaluar que se realicen.

Para saber si un punto P(a,b) del plano pertenece al Conjunto de Mandelbrot debemos seguir el siguiente procedimiento:

- a. Siempre se evalúa del origen de coordenadas $Z_0 = (0,0)$
- b. De este punto se dirige al primer punto a evaluar $Z_1 = (a,b)$
- c. El próximo punto a evaluar será el punto $Z_2 = (x_2, y_2)$, cuyas coordenadas están definidas por el siguiente algoritmo:

$$Z_2 = (Z_1)^2 + (a, b)$$

De esto se deduce que para el «*enésimo punto de evaluación n*» :

$$Z_n = (Z_{n-1})^2 + (a,b)$$

$$\text{es decir } x_n = (x_{n-1})^2 - (y_{n-1})^2 + a$$

$$y_n = 2 x_{n-1} y_{n-1} + b$$

Si después de realizar la gráfica, el punto Z_n se encuentra a una distancia menor que 2 unidades del origen,

entonces el punto $P(a,b)$ pertenece al Conjunto de Mandelbrot.

En la práctica puede resultar complicado realizar un número muy grande de iteraciones, de forma que se debe de evaluar el conjunto de Mandelbrot por aproximaciones, existen dos aspectos de este proceso que se debe de saber, cuando se encuentra el punto dentro del conjunto o cuando no lo está.

A continuación veremos dos ejemplos de este proceso, para conocer el funcionamiento correcto del algoritmo, se le darán valores y se solucionará la ecuación cuando:

EL PUNTO PUEDE ESCAPAR AL INFINITO ALEJÁNDOSE DEL CENTRO DEL PLANO Y NO PERTENECE AL FRACTAL DE MANDELBROT

Veamos como se comporta un punto con un color particular dentro del plano, empezaremos con un punto fuera del conjunto del Mandelbrot:

$$C = -0.5 + i$$

Iteracionando la función $Z_n = Z_{n-1}^2 + C$ produce los siguientes valores

Numero de Paso	Valor actual	Distancia del origen
1	$-0.5 + i$	1.12
2	-1.25	1.25
3	$1.06 + i$	1.46
4	$-0.37 + i * 3.1$	3.15

En la tercera iteración se aprecia que la distancia de el punto de origen es mas grande de dos. Esto nos indica

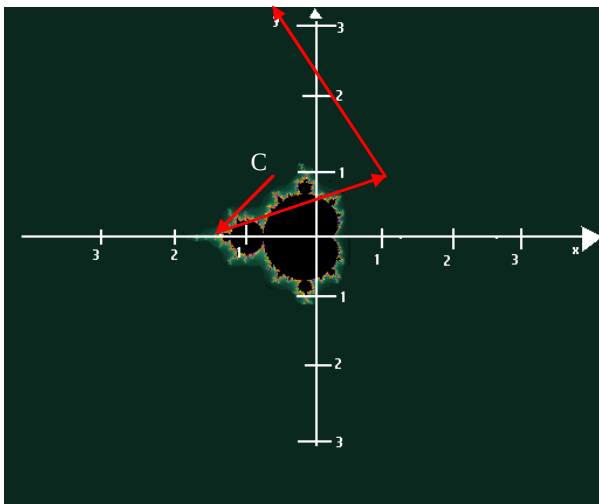


Figura 5. Gráfica que muestra el comportamiento de un punto fuera del Mandelbrot.

que el punto C no se mantiene dentro del conjunto del Mandelbrot, ver figura 5.

EL NUMERO OBTENIDO Y EL PUNTO SE ENCUADRE SE ENCUENTRAN DENTRO DEL FRACTAL

Si se hace el mismo procedimiento con un punto dentro del conjunto del Mandelbrot $C = 0.2 + i * 0.5$. En este caso, la distancia de el punto de origen nunca será mayor a dos. Cuando se alcanza el número máximo de iteraciones por ejemplo 200 es asumido que el punto inicial C se encuentra dentro del conjunto Mandelbrot y es pintado en negro.

Numero de Paso	Valor actual	Distancia del origen
1	$0.2 + i * 0.5$	0.54
2	$-0.01 + i * 0.7$	0.7
3	$-0.29 + i * 0.49$	0.57
4	$0.05 + i * 0.22$	0.22
5	$0.15 + i * 0.52$	0.54
6	$-0.05 + i * 0.66$	0.66
7	$-0.23 + i * 0.44$	0.48
8	$0.06 + i * 0.3$	0.3

Ver figura 6.

La función equivalente en lenguaje C sería la siguiente:

```
void compute(double x, double y, double c_real, double c_imag,
double *ans_x, double *ans_y)
{
    *ans_x = x*x - y*y + c_real;
    *ans_y = 2*x*y + c_imag;
}
```

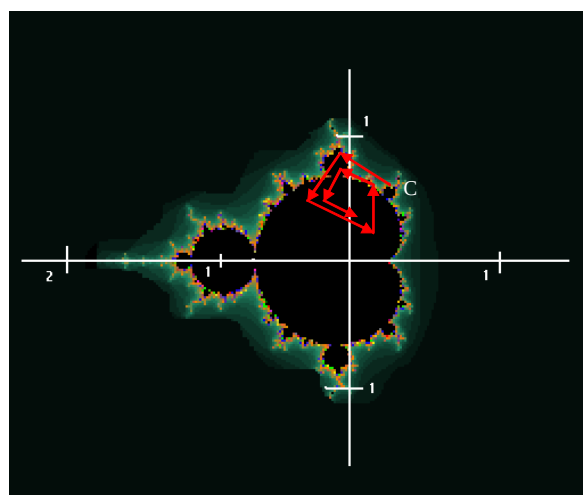


Figura 6. Gráfica que muestra el comportamiento de un punto dentro del Mandelbrot.

Como se puede observar esta función es definida de manera tal que pueda re utilizarse en un proceso recursivo.

RESULTADOS

Un problema que se presento en el desarrollo del proyecto fue la manera de poder almacenar el resultado en un archivo gráfico, es importante reconocer el entorno de programación MPI contiene aun, un limitado entorno gráfico y aunque existe el entorno MPE, no seria posible guardarlo en un archivo.

Investigando la manera de generar este archivo se encontró la manera de crear archivos PGM (Portable greyscale images) ya que es simple, bajo el inconveniente de que genera archivos en escalas de grises.

Para esto se le agrego al algoritmo la librería `pgm.h`.

La función `pgmWriterHeader` es utilizada con la declaración de una variable tipo archivo para generar la cabecera del lectura y la función `converttoworld` convierte cada dato calculado en pixeles que son escritos a continuación de la cabecera.

La funciones descritas son las siguientes :

a. Funcion de creación de cabecera:

```
void pgmWriteHeader ( FILE *f, int width, int height, int maxval )
{
    fprintf ( f, «P5\n%d %d\n%d\n», width, height, maxval );
}
```

b. Función de escritura de datos :

```
void convertToWorld ( int ix, int iy, int imgsize, double xmin, double
xmax, double ymin, double ymax, double *xw, double *yw )
{
    double x, y;
    x = ix;
    y = imgsize - iy;
    x /= imgsize; // normaliza
    y /= imgsize;
    x = ( x * (xmax-xmin) + xmin );
    y = ( y * (ymax-ymin) + ymin );
    *xw = x;
    *yw = y;
}
```

Como resultado de este análisis se deriva el siguiente programa

```
#include <stdio.h>
#include <math.h>
#include <sys/time.h>
#include «mpi.h»
```

Se incluye una función que calcule la distancia entre x y y:

```
double distance(double x, double y)
{
    return(x*x + y*y);
}
```

Por último se hace la integración de mpi sobre la ecuación de donde se tiene el siguiente programa:

```
/* Autor: M.C. Jesús Antonio Alvarez Cedillo
CIDETEC-IPN
Dibujo de un Mandelbrot usando comandos MPI y MPE
*/
```

```
void compute(double x, double y, double c_real, double c_imag,
double *ans_x, double *ans_y)
{
    *ans_x = x*x - y*y + c_real;
    *ans_y = 2*x*y + c_imag;
}
```

```
double distance(double x, double y)
{
```

```
/* calcula la distancia entre x y y.
* Recibe: doubles x y y.
* Regresa x^2 + y^2.
*/
```

```
    return(x*x + y*y);
}
```

```
/* salida de la cabecera del archivo PGM*/
void pgmWriteHeader ( FILE *f, int width, int height, int maxval )
{
    fprintf ( f, «P5\n%d %d\n%d\n», width, height, maxval );
}
```

```
void convertToWorld ( int ix, int iy, int imgsize, double xmin, double
xmax,
double ymin, double ymax, double *xw, double *yw )
{
    double x, y;
    x = ix;
    y = imgsize - iy;
    x /= imgsize; // normaliza
    y /= imgsize;

    x = ( x * (xmax-xmin) + xmin );
    y = ( y * (ymax-ymin) + ymin );

    *xw = x;
    *yw = y;
}
```

```
/* Programa principal */
```

```
int main(int argc, char* argv[])
{
    const int IMAGE_SIZE = 1024;
    const int MAX_ITER = 255;

    int n,
        ix,
        iy;

    double r_min, r_max;
    double i_min, i_max;
    double x,
        y,
        c_real,
        c_imag;

    MPI_Init(&argc,&argv);

    /* Ingreso
    printf(«\nIngresa el rango sobre la exis REAL (ej, -2,2): «);
    scanf («%lf,%lf», &r_min, &r_max );
    printf(«\n Ingresa el rango sobre la exis
    IMAGINARIO(ej, -2,2):»);
    scanf («%lf,%lf», &i_min, &i_max );

    FILE *f = fopen ( «mandel.pgm», «w» );
    pgmWriteHeader ( f, IMAGE_SIZE, IMAGE_SIZE, 255);

    for (iy = 0; iy < IMAGE_SIZE; iy++)
    {
        for (ix = 0; ix < IMAGE_SIZE; ix++)
        {
            convertToWorld ( ix, iy, IMAGE_SIZE, r_min, r_max, i_min, i_max,
                &c_real, &c_imag );
            x = y = 0.0;
            n = 0;

            while (n < MAX_ITER && distance(x,y) < 4.0)
            {
                compute(x,y,c_real,c_imag,&x,&y);
                n++;
            }

            if (n < MAX_ITER )
                fputc ( n%255:100, f );
            else
                fputc ( 0, f );
        }
    }

    fclose ( f );

    MPI_Finalize();
}
```

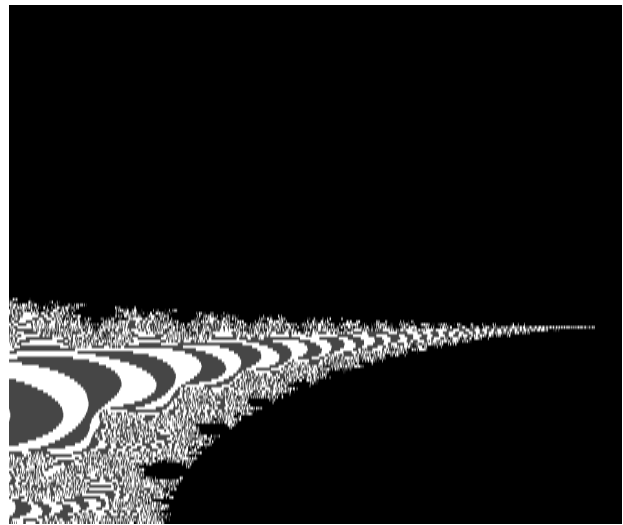


Figura 7. fractal de prueba generado por el algoritmo.

El algoritmo trabajó muy bien para diferentes rangos de evaluación del conjunto de Mandelbrot, a una velocidad muy aceptable y con resultados mas que estupendos, un ejemplo de esto se muestra en la **figura 7** donde se genera un fractal en el rango de x real (-0.25,-0.125) y en el rango de x imaginario (-0.125,-0.62).

Como puede observarse se debe de ingresar un rango de valores en los Reales y en los Imaginarios para que en relación a estos valores se puedan realizar las iteraciones.

CONCLUSIONES

La construcción de un algoritmo paralelo ya es una tarea compleja que involucra la manera como los datos deberán de ser manejados, el manejo de las iteraciones y el desempeño óptimo que mejore la velocidad de procesamiento. Por otro lado la cuarta dimensión es el hogar de los números complejos y de la geometría fractal.. Se trata del continuo espacio-temporal del hombre y la naturaleza, en donde existe un constante cambio basado en la retroalimentación. Como lo descubrió Mandelbrot, la cuarta dimensión incluye no solo a las tres primeras dimensiones, sino también a los intervalos existentes entre ellas, las dimensiones fractales.

La fórmula de Mandelbrot es un resumen de sus descubrimientos en la geometría fractal de la naturaleza [1], en el mundo real de la cuarta dimensión. Esto contrasta con el mundo idealizado de las formas euclidianas de las dimensiones primera, segunda y tercera. Estas formas habían preocupado a la casi totalidad de los matemáticos anteriores a Mandelbrot. La geometría euclidiana estaba relacionada con una perfección abstracta que es prácticamente inexistente en la naturaleza. Esta era incapaz de describir la forma de una nube, de una montaña, de una playa o de un árbol. El uso de nuevos algoritmos que utilicen MPI para resolver problemas de fractales proporcionan al investigador nuevas herramientas de referencia .

En su libro “The Fractal Geometry of Nature” (La Geometría Fractal de la Naturaleza), Mandelbrot menciona [1]: “las nubes no son esferas, las montañas no son conos, las playas no son círculos, y los ladridos no son agradables, ni la luz viaja en línea recta.”

A cual comentario anexo : “Nada en este mundo es lo que aparenta ser”

5. BIBLIOGRAFÍA

- [1] Mandelbrot, B. *The fractal geometry of nature*. Freeman and Co. 1977.
- [2] Prusinkiewicz, P.; Sandness, G. *Koch curves as attractors and repellers*. IEEE Computer, 1988
- [3] Hubbard and Schleicher and Sutherland. *How To Find All Roots of Complex Polynomials by Newton's Meted*. INVMATH: Inventiones Mathematica, 2001
- [4] Michael Barnsley. *Fractals Everywhere*. Academic Press Inc., second edition 1993. BRIGGS, J. PEAT, F. D. *Turbulent Mirror*. Harper & Row Pub. USA. 1989.

Implementación de Hardware / Software para Control Difuso

M. en C. Romeo Urbieto Parrazales,
M. en C. Pablo Manrique Ramírez,
Ing. Antonio Hernández Zavala;
Profesores del CIC-IPN
e-mail: rurbieto@cic.ipn.mx

Este artículo trata el diseño e implementación de hardware/software de control difuso usando tecnología avanzada de microcontroladores. El diseño de control difuso usa un esquema de dos variables de entrada: el error de velocidad y su respectiva derivada, y una salida, el voltaje de armadura del motor. Las variables de entrada fueron convertidos a valores difusos en tres funciones de membresía triangulares cada una, mientras la variable de salida en cinco funciones de membresía puntuales. La base del conocimiento viene expresada mediante el método de la composición max-min, produciendo una matriz de 4x4 reglas. La implementación de hardware viene representada en un diagrama a bloques: un microcontrolador cop8cdr como núcleo del sistema de hardware y sus respectivos periféricos: teclado, despliegue y adquisición, mientras que el software está representado en diagramas de flujo de las rutinas de control y periféricos.

INTRODUCCIÓN

Un sistema es una conjunción de procesos, desde los más simples hasta los más complejos, con entradas, funciones y salidas.^[1]

Con el paso del tiempo surgen diversas teorías que se encargan de mejorar el desempeño de los sistemas; en el caso de sistemas inciertos (ya sean industriales, humanos o naturales) existe la Lógica Difusa, que es una herramienta matemática para modelación o control de los procesos.^[2]

Aunque en este artículo se utilizan la Lógica y el Control Difuso como herramientas para control de procesos eléctricos; no se abordará la base teórica en la que

se sustentan. Sin embargo, se detalla la realización de un Sistema de Control Difuso que maneja un motor de corriente directa implementado en un microcontrolador.^[3]

Todo proceso de control, requiere que el usuario asigne un valor de operación para la variable a controlar, debe existir también un sistema que lleve al proceso a las condiciones indicadas por el usuario de modo fácil y preciso.

Para lograr lo anterior, se utiliza un motor de CD y un tacómetro analógico; en el cual mediante un dispositivo electrónico, el usuario define la velocidad a la que desea opere el motor. Posteriormente, un dispositivo se encarga de manejar el motor de forma automática según lo especificado manteniendo la velocidad hasta que el usuario determine lo contrario.^[4]

DISEÑO

Problema. Se desea controlar la velocidad de un motor de corriente directa, mediante una interfaz de usuario electrónica; para conocer la velocidad actual del motor se cuenta con un tacómetro analógico que mediante un transductor entrega la lectura. Para asignar el voltaje de operación en la entrada del motor se cuenta con un transductor de entrada.

Análisis. Como se puede observar, mediante una señal de entrada en el rango de 0 a 6 Vcc, se puede controlar al motor a velocidades en el rango de 0 a 2000@Rpm. Debido a las especificaciones del motor no se deben rebasar las 1700@Rpm. Por ello, el motor será operado en el rango de 0 a 5 Vcc. El valor equivalente en voltaje a la velocidad actual del motor, es entregado por el driver del tacómetro en el que la salida se encuentra en el mismo rango de voltaje de entrada, por lo que para la entrada y la salida se utilizarán los valores de la **Tabla 1**.^[5]

Vin REF	Motor (@Rpm)	Vout TACH	Valor decimal
0	0	0	0
0.25	0	0.25	12
0.5	0	0.5	25
0.75	36	0.75	38
1	160	1	51
1.25	260	1.25	63
1.5	360	1.5	76
1.75	460	1.75	89
2	560	2	102
2.25	660	2.25	114
2.5	760	2.5	127
2.75	860	2.75	140
3	960	3	153
3.25	1060	3.25	165
3.5	1160	3.5	178
3.75	1260	3.75	191
4	1360	4	204
4.25	1460	4.25	216
4.5	1560	4.5	229
4.75	1660	4.75	242
5	1760	5	255

Tabla 1. Valores de entrada y salida del proceso.

En la tabla, la señal de entrada para el sistema analógico del motor recibe el nombre de OUT, al dato de salida del tacómetro se le asignará el nombre de TACH, y al dato de referencia asignado por el usuario se le llamara REF. Estos son los nombres de variable que utilizará el controlador.

En el diagrama general (Figura 1) se puede notar que el modelo de control es de lazo cerrado, esto es, el dato de salida del controlador depende directamente del dato de

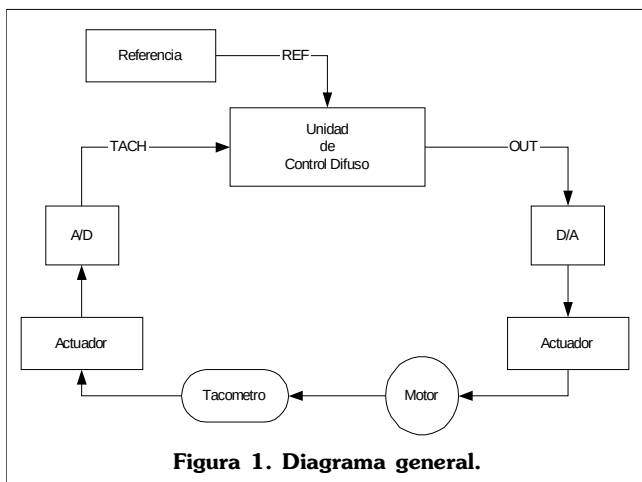


Figura 1. Diagrama general.

entrada; a partir de esta referencia y el dato tomado del tacómetro por el convertidor A/D (TACH) se obtiene un error, el cual se usa en el algoritmo de control difuso para obtener una salida de control que será enviada a través de un convertidor D/A hacia el motor, repitiendo iterativamente el procedimiento anterior.

Cabe señalar que solo será tratada la parte digital, misma que comprende los conversores A/D y D/A, la unidad de control difuso y la referencia de usuario, por lo que el motor, tacómetro y actuadores no se abordarán a detalle; esto es, se presupone su conocimiento.^[6]

El Sistema recibe un par de datos de entrada provenientes del exterior; el primero de ellos es el introducido por el usuario REF y el segundo es la velocidad actual del motor dada por el tacómetro TACH.

Al restar al valor de referencia el dato del tacómetro, se obtiene el valor del ERROR,

$$ERROR = REF - TACH$$

La derivada del error se obtiene de la siguiente operación

$$DERROR = ERROR_i - ERROR_{i-1}$$

Donde $ERROR_{i-1}$ es el error en el instante anterior, que se asigna a la variable AUX; el cálculo de la derivada del error permite al controlador seguir la ruta correcta de evaluación. Lo anterior se puede apreciar en la Figura 2.

Control Difuso. El Control Difuso, esta compuesto por tres etapas: la primera es la fuzzificación, aquí se convierten los valores de entrada a valores difusos; la segunda es la evaluación de los valores fuzzificados en la base de reglas de control; y la última es la desfuzzificación, en la que se obtiene un valor de salida no difuso. Una vez identificado el modelo del control difuso, se puede detallar ampliamente el sistema a realizar.^[7]

La Figura 3, muestra las partes que componen al sistema en su totalidad.

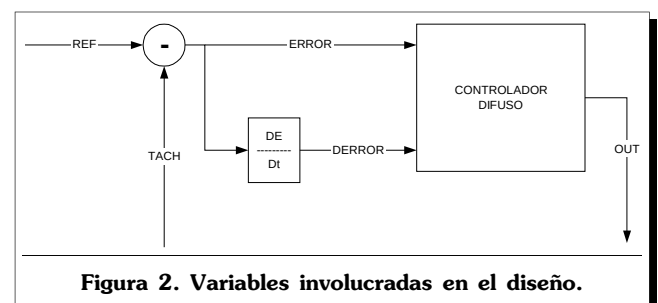


Figura 2. Variables involucradas en el diseño.

Fuzzificación. Un controlador difuso utiliza números difusos de la forma $A_\alpha = (x, \mu_\alpha(x))$, para los que a cada valor de entrada x , le corresponde un valor de $\mu_\alpha(x)$. Para las variables de entrada ERROR y DERROR, se tienen los siguientes números difusos con sus correspondientes variables lingüísticas (ver también las Figuras 4 y 5).

Para ERROR:

ERROR-NEGATIVO

$$EN = 1 - \left(\frac{x + 255}{255} \right), [-255, 0] \quad \dots (E0)$$

ERROR-ZERO

$$EZ = \frac{x + 55}{55}, [-55, 0] \quad \dots (E1) \quad \dots (E2)$$

$$EZ = 1 - \frac{x}{55}, [0, 55]$$

ERROR-POSITIVO

$$EP = \frac{x}{255}, [0, 255] \quad \dots (E3)$$

Para DERROR:

DERIVADA_ERROR-NEGATIVA

$$DEN = 1 - \left(\frac{x + 255}{200} \right), [-255, -55] \quad \dots (DE0)$$

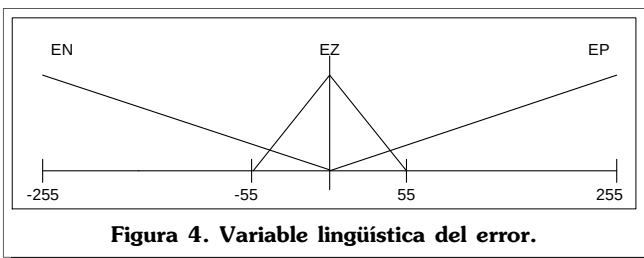


Figura 4. Variable lingüística del error.

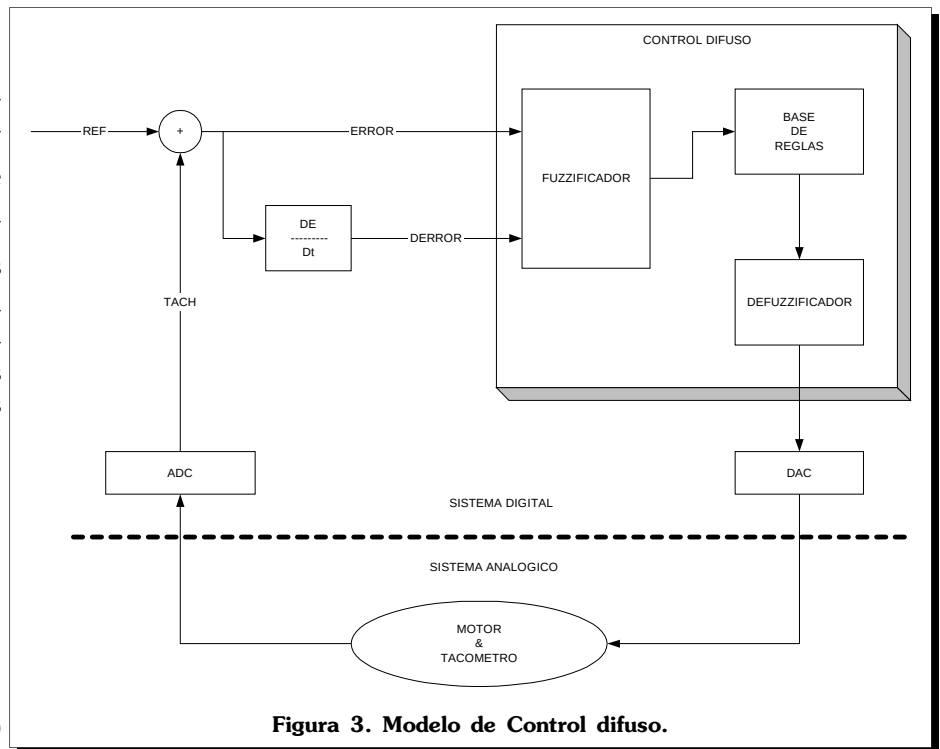


Figura 3. Modelo de Control difuso.

DERIVADA_ERROR-ZERO

$$DEZ = \frac{x + 110}{110}, [-110, 0] \quad \dots (DE1) \quad \dots (DE2)$$

$$DEZ = \frac{x - 110}{-110}, [0, 110]$$

DERIVADA_ERROR-POSITIVA

$$DEP = \frac{x - 55}{200}, [55, 255] \quad \dots (DE3)$$

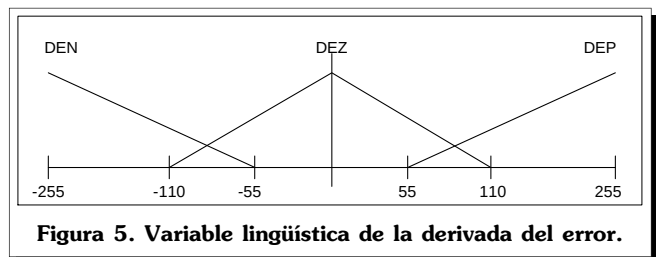


Figura 5. Variable lingüística de la derivada del error.

Se utilizan un par de vectores que contienen los datos calculados en esta etapa, con el propósito de poder realizar el método de composición; cada localidad corresponde a la formula que se indica. Los vectores se muestran a continuación.

ERROR

Posición	0	1	2	3
Formula	EO	E1	E2	E3

DERROR

Posición	0	1	2	3
Formula	DEO	DE1	DE2	DE3

Método de Composición max-min. Para evaluar un dato en la base de reglas, se debe contar con un método que permita elegir de entre el conjunto de datos de entrada, cuál será utilizado; para esto se realiza la composición MAX-MIN entre ERROR y DERROR, el dato obtenido debe indicar qué posición de la base de reglas se debe utilizar.^[8]

Esta composición, mediante un proceso similar a la multiplicación de matrices, produce como resultado una matriz de 4x4 (**Tabla 2**), mediante evaluación de mínimos y máximos de acuerdo a la siguiente formula:

$$E \circ DE = \left\{ (x, z), \max_y (\min(\mu_E(x, y), \mu_{DE}(y, z))) \right\}$$

	0	1	2	3
0	AL00	AL01	AL02	AL03
1	AL10	AL11	AL12	AL13
2	AL20	AL21	AL22	AL23
3	AL30	AL31	AL32	AL33

Tabla 2. Matriz alfa resultante.

Dado que la composición dejará un solo resultado diferente de cero, este corresponde a la localidad en la base de reglas que determinará la salida. La base de reglas utilizada se muestra en la **Tabla 3**.

	0	1	2	3
0	NM	NM	NM	NL
1	PM	Z	Z	NM
2	PM	Z	Z	NM
3	PL	PM	PM	PM

Tabla 3. Base de reglas óptimas.

A cada una de las variables, le corresponde un valor de salida según se muestra:

NL = -64
 NM = -31
 Z = 2
 PM = 31
 PL = 64

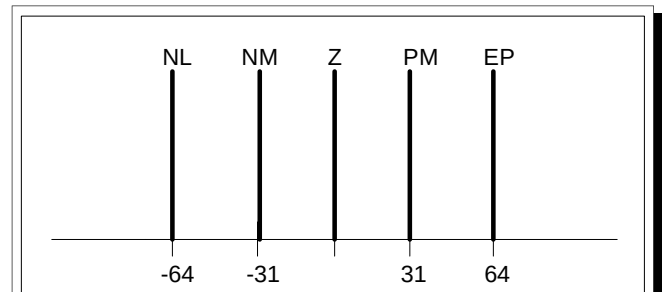


Figura 6. Variable lingüística de salida.

El resultado se carga en la variable de programa RES. La **Figura 6** es la forma gráfica de representación de los resultados de salida.

Desfuzzificación. Esta etapa se encarga de sumar o restar, según el caso, el valor obtenido de la etapa anterior con el valor de la salida actual.

$$OUT = \sum RES$$

IMPLEMENTACIÓN DE HARDWARE/SOFTWARE

El sistema se implementará en un microcontrolador COP8 con convertor A/D integrado, la salida será dada a través de un convertor D/A y los resultados se presentarán en una pantalla LCD.

El diagrama general del sistema controlador difuso en hardware se muestra en la **Figura 7**.

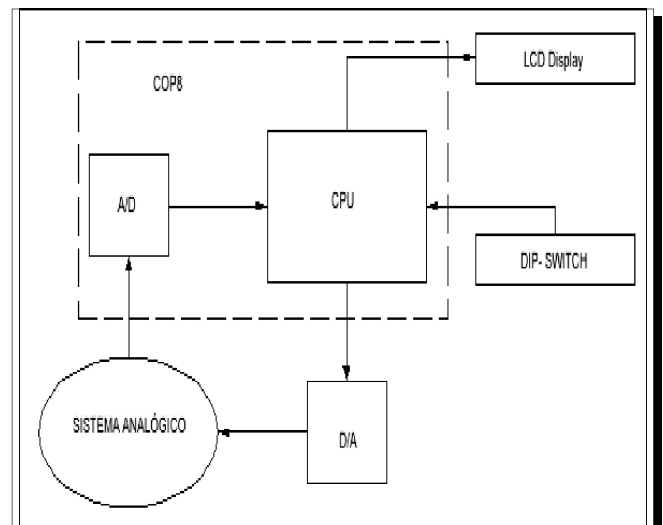


Figura 7. Hardware del sistema de control difuso.

De manera general, el algoritmo que se utiliza es el siguiente:

1. Lee referencia del usuario.
2. Lee dato del ADC.
3. Cálculo de error y derivada del error.
4. Fuzzificación.
5. Composición MAX – MIN.
6. Evaluación de reglas.
7. Defuzzificación.
8. Envío de salida a DAC.
9. Repite desde 1.

En la **Figura 8** se muestra el diagrama de flujo del programa y en la **Figura 9** se tiene el diagrama de conexiones del sistema.

• FUNCIÓN DE LECTURA DE REFERENCIA.

Para esta función se requiere leer un dato que el usuario ha colocado en el puerto de entrada, al momento de presionar un botón. Para la entrada se utilizan los bits de acuerdo a la tabla siguiente:

Puerto	A1	A2	A3	A4	A5	A6	A7	E0
REF	0	1	2	3	4	5	6	7

Para leer el dato se requiere realizar un desplazamiento en A, y probar el bit 0 de E.

Un botón conectado en el puerto G0, sirve como señal de interrupción; al generarse esta, el dispositivo realiza la rutina de servicio a interrupción externa, para obtener la lectura correspondiente.

• FUNCIÓN DE LECTURA DE DATO DE ADC.

El registro de configuración del convertidor integrado, cuenta con el bit ADBSY; el cual al ser establecido inicia la conversión, esto tarda un período de 15 ciclos de reloj, por lo que el algoritmo es el siguiente:

1. Establecer ADBSY.
2. Mientras ADBSY=1, esperar
3. Hacer TACH=ADRSTH
4. Fin.

• FUNCIÓN DE CÁLCULO DE ERROR Y DERIVADA DE ERROR.

El error es igual al dato de referencia menos la lectura del ADC; y la derivada del error es igual al error calculado menos el error anterior. Como el resultado puede ser positivo o negativo, existe un vector de signo (SIG-

NO), en donde el bit 7 corresponde al signo del error y el bit 6 al signo de la derivada de error.

• FUNCIÓN DE FUZZIFICACIÓN.

Dado que esta función será implementada en hardware las operaciones deben ser menos complejas, por lo que se utilizan las siguientes formulas equivalentes a las vistas anteriormente.

Puerto	D4	D5	D6	D7	E4	E5	E6	E7	F0	F1
DAC	D0	D1	D2	D3	D4	D5	D6	D7	WR	CS

E0
255-ERROR=RES
255-RES=RES

E1
55-ERROR=RES
RES x 4 = RES

E2
ERROR x 4 = RES
255 - RES = RES

E3
ERROR=RES

DE0
255 - DERROR=RES
255 - RES = RES

DE1
110 - DERROR=RES
RES x 2 = RES

DE2
DERROR x 2 = RES
255 - RES = RES

DE3
DERROR - 55 = RES

• FUNCIÓN DE COMPOSICIÓN MAX – MIN.

La composición entre los vectores ERROR y DERROR, utiliza apuntadores (EPTR, DEPTR respectivamente) a las localidades donde se encuentran los arreglos, y la matriz de alfa es apuntada por ALPTR. El método realiza las comparaciones entre ERROR X DERROR, colocando el resultado en la matriz de alfa; al final se realiza una segunda vuelta de búsqueda de máximo, para así solo tener un valor diferente de 0, que será el utilizado por la función siguiente.

• FUNCIÓN DE EVALUACIÓN DE REGLAS.

Esta función lee cada localidad de la matriz de alfa, si el contenido de ésta es diferente de cero, realiza la evaluación correspondiente, cargando en la variable RES el resultado obtenido; se utilizan la tabla y los valores de salida descritos anteriormente.

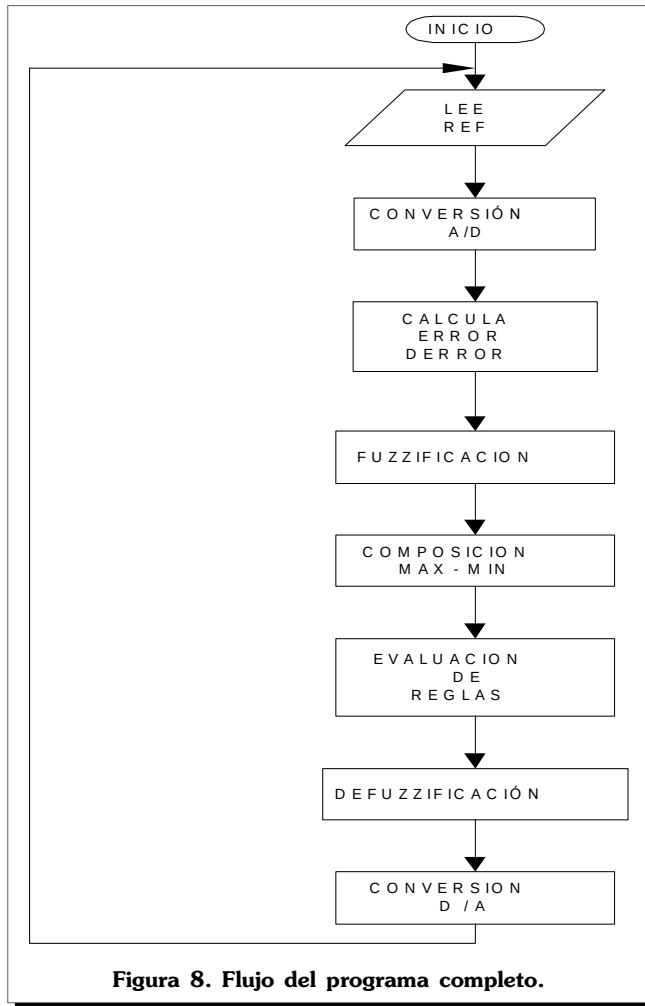


Figura 8. Flujo del programa completo.

• FUNCIÓN DE DESFUZZIFICACIÓN.

Mediante la utilización de RES y el bit de signo de ERROR, se realiza ya sea una suma o una resta de OUT +/- RES, utilizando OUT como un acumulador.

• ENVÍO DE SALIDA A DAC.

El contenido de OUT, se transfiere a puertos, la entrada del DAC, posteriormente se genera la señal de carga y retención de datos, con lo que se genera la señal de control en el voltaje deseado. Los puertos y las entradas al DAC se muestran en la tabla. Se utiliza un DAC TLC7524 configurado en modo voltaje.

CONCLUSIONES

1. El control difuso puede ser aplicado a cualquier problema que requiera la utilización de control.
2. El método MAX - MIN proporciona de manera fácil y precisa el resultado deseado de la base de reglas, pero si éste no está bien definido, conduce a errores.

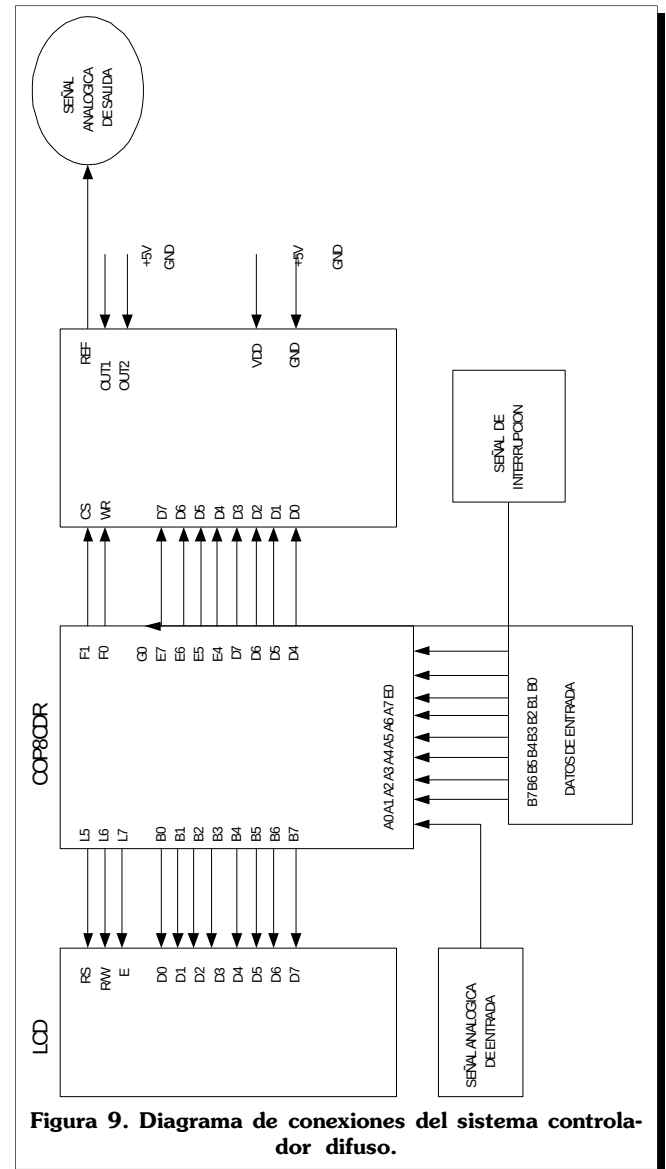


Figura 9. Diagrama de conexiones del sistema controlador difuso.

3. La definición de los números difusos de entrada es labor compleja, por lo cual se hace uso de otras herramientas matemáticas, tales como Redes Neuronales, Algoritmos Genéticos, etc.
4. No es necesario el uso de punto flotante como tal en las operaciones, se puede dar un resultado preciso trabajando con números enteros.
5. El verdadero poder de aplicación del control difuso se encuentra en problemas para los cuales las salidas no requieran un gran número de valores de salida, sino para los cuales exista un rango al cual se le pueda asignar un valor fijo.

REFERENCIAS

- [1] Peter Harriott. *Process Control. Capítulo 3. Open-Loop Response of Simple Systems*. McGraw-Hill Book Company. 1964.
- [2] George Bojadziev & Maria Bojadziev. *Fuzzy Sets, Fuzzy Logic, Applications. Capítulo 11. Fuzzy Logic and Applications*. World Scientific. 1995.
- [3] Jeremy Binfet & Bogdan M. Wilamowsky. *Micro-processor of Fuzzy Systems and Neural Network*. Página 234-239. 2001 IEEE.
- [4] José Ramón Alique López, Elena Agüero Gutiérrez, Leonardo C. Rosa. *A Fuzzy Logic Based Approach to Machine-Tool Control Optimization*. Páginas 524-539 del libro Fuzzy Control Systems. 1994.
- [5] Steve G. Romanink and Lawrence O. Hill. *Learning Fuzzy Control Rules from Examples*. Páginas 376-395 del libro Fuzzy Control Systems. 1994.
- [6] Ramiro Aduviri Velasco. *Comparación de Controladores Convencionales y Controladores Difusos*. Universidad de San Andrés. La Paz Bolivia. Agosto 2003.
- [7] George Bojadziev & Maria Bojadziev. *Fuzzy Sets, Fuzzy Logic, Applications. Capítulo 6 Fuzzy Sets*. World Scientific. 1995.
- [8] George Bojadziev & Maria Bojadziev. *Fuzzy Logic, Applications. Capítulo 7 Fuzzy Relations. Fuzzy Sets*. World Scientific. 1995.

Oxímetro de Pulso Basado en una Palm

Parte I: Fundamentos

M. en C. Israel Rivera Zárate,
M. en C. Juan Carlos Herrera Lozada,
Profesores del CIDETEC-IPN.
Victor Jalil Ochoa;
Estudiante UPIICSA-IPN.

La importancia del estudio de las variables involucradas con la salud de la sangre de algún paciente, tales como los niveles de pH, P_{O_2} , P_{CO_2} , hematocrito, hemoglobina total, saturación de O_2 , etc., ha llevado a desarrollar instrumentos para medir y analizar todos estos parámetros por diferentes métodos. Sin embargo, presentan el inconveniente de ser por lo general métodos invasivos. En el desarrollo de este trabajo se verá un método no invasivo para la medición de la saturación de oxígeno en la sangre por medio de la interpretación de la absorción de luz de longitudes de onda específicas, que dependerá de la proporción existente entre hemoglobina oxigenada (HbO_2) y la hemoglobina desoxigenada (Hb).

TERMINOLOGÍA UTILIZADA EN ESPECTROSCOPIA DE ABSORCIÓN

En la **Tabla 1** se enumeran los términos y símbolos empleados con mayor frecuencia en espectroscopia de absorción. Recientemente se ha hecho un esfuerzo considerable por la *American Society for Testing Materials* para crear una nomenclatura uniforme. Los términos y símbolos que se enumeran en las dos primeras

columnas de la **Tabla 1**, se basan en estas recomendaciones [1]. La tercera columna contiene otros símbolos que podrán encontrarse en la bibliografía más antigua.

TRANSMITANCIA

Al hacer pasar un haz de radiación luminosa a través de una capa de solución con cierto grado de concentración, y que contiene una especie molecular que posee un coeficiente de absorción ante tal longitud de onda radiante, se observa, que como consecuencia de las interacciones entre los fotones y las partículas absorbentes, la potencia del haz disminuye de P_o a P . La transmitancia T de la solución es la fracción de la radiación incidente transmitida por la solución:

$$T = P/P_o$$

Por lo general, la transmitancia se expresa como porcentaje.

ABSORBANCIA

La absorbancia de una solución está definida por la ecuación:

$$A = -\log_{10} T = \log P_o/P$$

Obsérvese que, a diferencia de la transmitancia, la absorbancia de una solución aumenta a medida que aumenta la atenuación del haz.

ABSORTIVIDAD Y ABSORTIVIDAD MOLAR

Como se verá a continuación, la absorbancia es directamente proporcional a la trayectoria de la radiación a través de la solución y a la concentración de la especie molecular que produce la absorción. Es decir:

$$A = abc$$

donde **a** es una constante de proporcionalidad llamada absorptivi-

Término y símbolo	Definición.	Otros nombres y símbolos
Potencia radiante, P , P_o	Energía de la radiación en ergs incide en el detector, por cm^2 de superficie y por segundo.	Intensidad de la radiación, I , I_o .
Absorción, A	$\log P_o/P$	Densidad óptica, D ; extinción, E
Transmitancia, T	P_o/P	Transmisión, T
Trayectoria b de la radiación, en cm.	-	l , d
Absortividad, a	$A/(bc)$	Coefficiente de extinción, k
Absortividad molar, e	$A/(bc)$	Coefficiente de extinción molar

Tabla 1. Símbolos y terminos más importantes utilizados en las medidas de absorción.

dad. Resulta evidente que la magnitud de **a** depende de las unidades utilizadas para **b** y **c**. Cuando se expresa la concentración en moles por litro y la trayectoria a través de la celda en centímetros, la absorptividad se denomina *absorptividad molar* y se representa con el símbolo **e**. En consecuencia cuando **b** se expresa en centímetros y **c** en moles por litro:

$$A = e bc$$

LEY DE BEER-LAMBERT

Esta ley indica que para una cierta concentración de absorbente, la intensidad de la luz transmitida, que previamente se ha logrado que sea paralela plana y que entre al medio absorbente, formando ángulos rectos con el plano, disminuye logarítmicamente a medida que la longitud del trayecto aumenta en forma aritmética.^[1]

La relación entre la intensidad y la concentración de la especie absorbente tiene mucho más interés, por lo que **Beer** determinó que, al aumentar la concentración del absorbente, se produce el mismo efecto que un aumento proporcional en la longitud del trayecto de absorción de la radiación. De esta forma, la constante de proporcionalidad **k** de la ecuación es a su vez, proporcional a la concentración de soluto absorbente, esto es:

$$k = aC$$

Sí la longitud de trayecto de la muestra se expresa en centímetros y la concentración en gramos de absorbente por litro de solución, la constante **a**, llamada *absorbancia relativa* específica o coeficiente de absorción, tiene por unidades litro g⁻¹ cm⁻¹.

Con frecuencia se desea especificar C en términos de concentraciones molares, manteniendo b en uni-

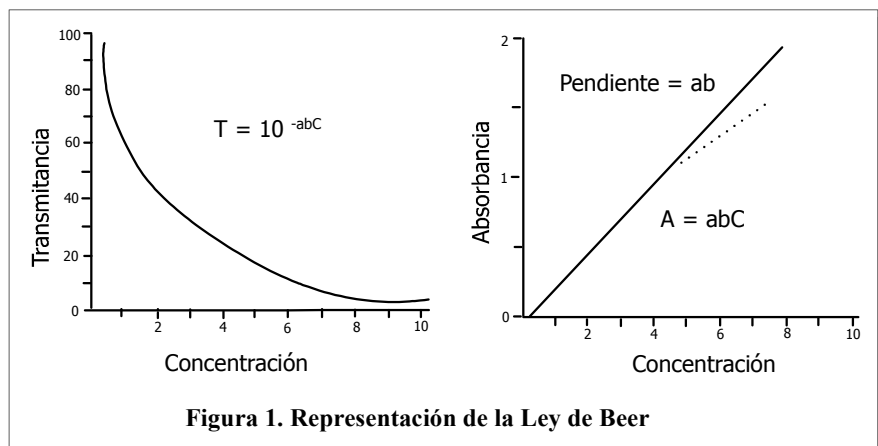


Figura 1. Representación de la Ley de Beer

dades de centímetros, Entonces la ecuación se escribe como:

$$\log \frac{P_0}{P} = \epsilon bC$$

donde ϵ , en unidades de:

$$L \text{ mol}^{-1} \text{ cm}^{-1},$$

se llama coeficiente molar o coeficiente molar de absorción. Una gráfica de la absorbancia en función de la concentración es una línea recta que pasa por el origen, tal como se muestra en la **Figura1**.

Las escalas de lectura y de medición de los espectrofotómetros suelen estar calibradas para leer absorbancia y transmitancia. La sensibilidad de un espectrómetro depende de la magnitud de la absorbancia específica y de la absorbancia mínima que puede medirse con el grado de certidumbre requerido.

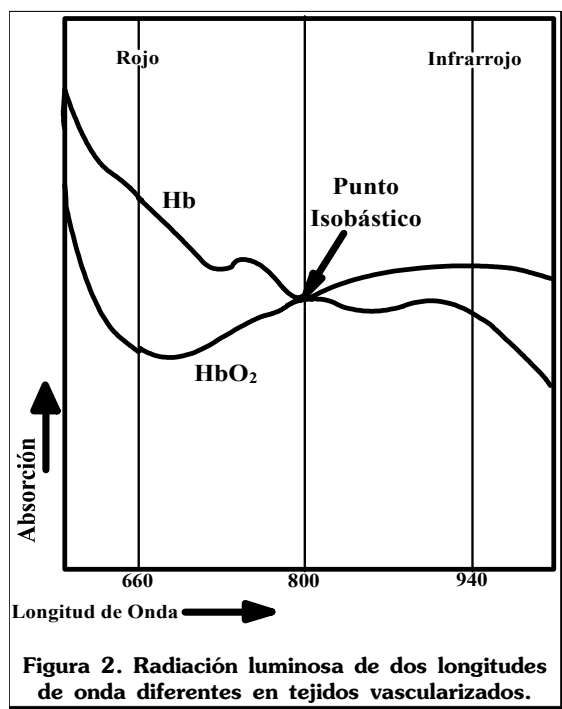
PRINCIPIOS DE LA OXIMETRÍA

Oximetría es un término general relativo o aplicable a las diferentes tecnologías capaces de medir la saturación de la hemoglobina (Hb) por el oxígeno. De manera general, las técnicas oximétricas se pueden dividir en: 1) Espectrofotometría para el análisis de la Hb *in vitro*; 2) Oximetría

de pulso (SpO₂) para medición no invasiva de la saturación de la Hb y 3) Oximetría fibróptica para medición invasiva de la saturación de la oxihemoglobina *in vivo*.^[2]

Todas estas técnicas de oximetría se basan en principios espectrofotométricos que miden las porciones de luz transmitida y/o absorbida por parte de la Hb. Para los fines de este trabajo, nos ocuparemos de la oximetría de pulso que se puede conceptualizar como una técnica de monitoreo no invasivo que determina de manera continua y relativamente confiable la saturación arterial de oxígeno (SaO₂), en el momento preciso en que está sucediendo.

La oximetría básicamente es la interpretación de la coloración sanguínea que depende de la SaO₂. El cambio de color de la sangre al saturarse de oxígeno, se debe a las propiedades ópticas de la molécula de Hb (específicamente de la porción *heme*). A medida que la sangre se desoxigena se vuelve menos permeable a la luz roja, el tejido pierde entonces su apariencia rosada, tomando un tinte azulado; de manera que visto de una manera simplista, el oxímetro sólo tiene que medir lo rojo de la sangre arterial e interpretarlo en términos de saturación, pudiendo entonces establecer que el oxímetro de pulso mide la absorción de luz



de longitudes de onda específicas que dependerá de la proporción existente entre Hb oxigenada y Hb desoxigenada.

Resulta fundamental recordar que, de manera general, existen normalmente dos tipos de Hb en la sangre, las llamadas hemoglobinas funcionales (la oxihemoglobina o Hb ligada al oxígeno, y la hemoglobina reducida (HbR), que si bien se encuentra desoxigenada, tiene la capacidad de unirse al oxígeno transformándose en oxihemoglobina); las hemoglobinas denominadas disfuncionales, las cuales presentan otro tipo de comportamiento no fisiológico cuando interactúan con el oxígeno (carboxihemoglobina, metahemoglobina y sulfahemoglobina). Es importante considerar este último señalamiento, dado que bajo condiciones normales las hemoglobinas denominadas funcionales son las más abundantes en la sangre, por lo que teóricamente se acepta para fines de oximetría de pulso que la sangre se compone solamente por dos absorbedores de luz, la oxihemoglobina (HbO₂) y la HbR.

Partiendo de este fundamento exclusivamente teórico, es que en la oximetría de pulso, se utiliza luz con sólo dos diferentes longitudes de onda.^[3]

Las características del espectro de absorción de la luz de la HbO₂ y de la HbR, presentan diferencias que son máximas en la región roja e infrarroja del espectro como se muestra en la Figura 2.

Así, a una longitud de onda de 660 nm, la luz roja visible se absorbe más por la HbR que por la HbO₂, y a una longitud de onda de 940 nm, la luz infrarroja se absorbe más por la HbO₂ que por la HbR. Estas dos luces de diferente longitud de onda (roja e infrarroja) se hacen pasar a través del árbol arterial y los porcentajes de HbO₂ y HbR se determinan por la medición de la proporción de luz roja e infrarroja transmitida hasta el foto-detector. Existe oxígeno dentro la sangre que circula por las venas y arterias de todo sistema circulatorio.

Este oxígeno es llevado en la sangre en dos estados separados. Normalmente, el 98% del oxígeno, esta combinado con hemoglobina (Hb) en las células rojas de la sangre. El restante 2% esta físicamente disuelto en el plasma. La cantidad de saturación de oxígeno (Saturación, S) adherida a la hemoglobina en la sangre arterial esta definida como el porcentaje de concentración de oxihemoglobina (HbO₂) en la concentración total de Hemoglobina.

Si se asume que la transmisión de la luz a través de la sangre arterial es influenciada exclusivamente por las

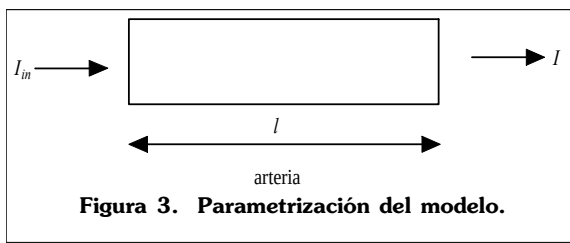


Figura 3. Parametrización del modelo.

concentraciones relativas de HbO₂ y Hb y sus coeficientes de absorción a las dos longitudes de onda medidas, entonces la intensidad de la luz se reducirá logarítmicamente con la longitud de la trayectoria conforme lo establece la ley de Beer – Lambert. Ver Figura 3.

A la longitud de onda λ_1

$$I_1 = I_{in1} 10^{-(\alpha_{o1}C_o + \alpha_{r1}C_r)l}$$

A la longitud de onda λ_2

$$I_2 = I_{in2} 10^{-(\alpha_{o2}C_o + \alpha_{r2}C_r)l}$$

Donde:

C_o es la concentración de oxihemoglobina (HbO₂).

C_r es la concentración de la hemoglobina reducida (HbO).

α_{on} es el coeficiente de absorción de HbO₂ a la longitud de onda λ_n

α_{rn} es el coeficiente de absorción de Hb a la longitud de onda λ_n

De acuerdo con la ley de Beer-Lambert :

$$R = \frac{\log_{10}(I_1 / I_{in1})}{\log_{10}(I_2 / I_{in2})}$$

Se puede observar que:

$$\begin{aligned} \text{SaO}_2 &= \frac{C_o}{C_o + C_r} \\ &= \frac{\alpha_{r2}R - \alpha_{r1}}{(\alpha_{r2} - \alpha_{o2})R - (\alpha_{r1} - \alpha_{o1})} \end{aligned}$$

Lo anterior permite proponer un sistema de procesamiento que permita calcular la SaO_2 con base en la ecuación anterior; sistema que será expuesto en detalle en los artículos siguientes.

BIBLIOGRAFIA

- [1] J. F. Kelleher, Pulse oximetry, *J. Clin. Monit.*, vol. 5, pp. 37–62, 1989.
- [2] J. W. Severinghaus and J. F. Kelleher, Recent developments in pulse oximetry, *Anesthesiology*, vol. 76, pp. 1018–1038, 1992.
- [3] J. T. B. Moyle, *Pulse Oximetry*, 1st ed. London, U.K.: BMJ, 1994.

Una Técnica para la Localización de Ojos Humanos en una Imagen Bidimensional

Luna Esteban Bruce Leroy,
Mendoza Campa Juan Manuel,
Parra Bautista Roberto,
ESIME Culhuacan - IPN
M. en C. Aguilar Jáuregui María Elena
Profesora del CIC-IPN

El presente trabajo propone una técnica para la localización de ojos de personas en una imagen bidimensional. Los vectores característicos del ojo humano se obtuvieron a partir de una base de datos de imágenes de 300 X 300 píxeles.

El sistema fue desarrollado bajo una plataforma C++ e implementado en una PC con un procesador Pentium IV a 2.40 GHz.

INTRODUCCIÓN

Dentro del campo de la visión artificial, la detección y el rastreo de ojos juegan un papel muy importante ya que con estas funciones se puede identificar a una persona y/o interpretar su estado emocional. Por lo tanto, es crucial un sistema de detección y/o rastreo de ojos eficiente para poder tener una mejor interacción humano – computadora.

A pesar de muchos esfuerzos todavía no existen métodos robustos y 100% precisos para detectar y rastrear ojos humanos.

El desarrollo actual se puede dividir en dos categorías: los que se basan sobre imágenes en línea y los de fuera de línea.

Este trabajo se enfoca en la segunda categoría, ya que fue necesario considerar la complejidad de los algoritmos y los tiempos de procesamiento.

Aún cuando existe una gran variedad de tipos de ojos humanos en cuanto a rasgos se refiere, es posible obtener sus características propias o descriptores, los cuales permiten afirmar que en verdad se trata de un ojo y no de algún objeto parecido en cuanto a forma.

EXTRACCIÓN DE CARACTERÍSTICAS

La forma de los ojos brinda características muy importantes para su reconocimiento. Debido a que estas formas son muy variadas, en la metodología propuesta la segmentación se realizó de forma manual, como se muestra en la **figura1**.

Los formatos de imágenes que se evaluaron fueron: bmp (Bitmap), jpeg (Joint Photographic Expert Group) y gif (Graphics Interchange Format) debido a que son los más usados en computación. Bmp consiste en un formato sin pérdida de información; por otra parte, jpeg es un formato de compresión de imagen que no produce pérdida de color aunque la compresión elimina datos, mientras que gif es un formato con un alto grado de compresión y mayor pérdida de color^[2].

Para lograr mejores resultados en cuanto al funcionamiento del sistema se utilizó el formato bmp para la extracción de características.

Aunque el sistema visualiza las imágenes en color verdadero, el trabajo del sistema se basa únicamente en el canal verde que es el que presenta información con menos ruido.

Para la selección de la información

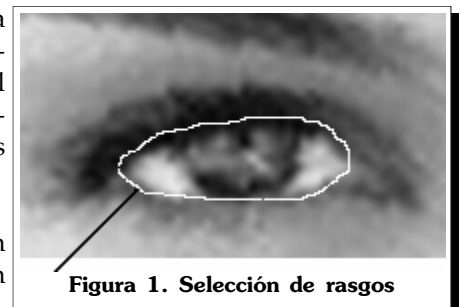


Figura 1. Selección de rasgos

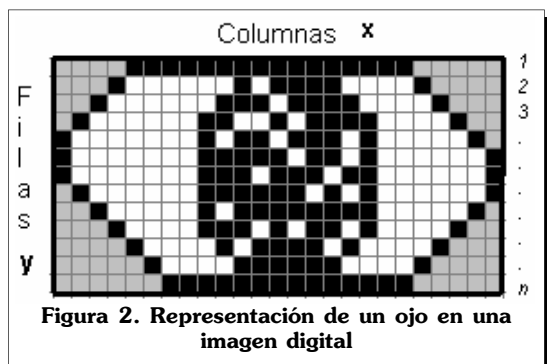


Figura 2. Representación de un ojo en una imagen digital

cuantitativa de nuestro interés se deben obtener los valores de brillo de cada píxel en sus coordenadas (x, y); la manera de realizar esto es mediante el uso de una matriz bidimensional (x, y). Considerando que la forma del ojo es casi elíptica, es difícil representar la intensidad de brillo de los píxeles en forma matricial.

La **figura 2** representa la forma (aproximada) de un ojo en una imagen digital, en la que se muestran las coordenadas (x, y) de la misma. Los píxeles en color gris representan ruido o información inútil para describir un ojo, por consiguiente se consideran como un valor X (sin importancia) en la matriz. Dado esto, fue necesario transformar la matriz resultante en un vector siguiendo los criterios siguientes:

1. Se consideran los píxeles pertenecientes o internos al borde.
2. Se descartan los píxeles externos al borde.
3. Los primeros valores del vector corresponden a los píxeles que radican en la primera fila y que obedecen al criterio número 1.
4. Los valores posteriores del vector corresponden a los valores obtenidos a partir de la n fila y así sucesivamente hasta la última fila que obedezca al criterio 1.

OJO = [a₁₁, ..., a_{1m}, a₂₁, ..., a_{2m}, ..., a_{n1}, a_{nm}]

Donde:

- a → cualquier valor de intensidad de brillo.
 n → las filas pertenecientes o internas al borde.
 m → son las columnas pertenecientes o internas al borde.

Una vez obtenido el vector se obtiene el histograma o función de densidad que corresponde a una función discreta $p(r_k) = n_k/n$, donde r_k es el k-ésimo nivel de gris, n_k es el número de píxeles de la imagen con ese nivel de gris, n es el número total de píxeles de la imagen y k es el rango de niveles de gris [0,255] [1]. Como se mencionó

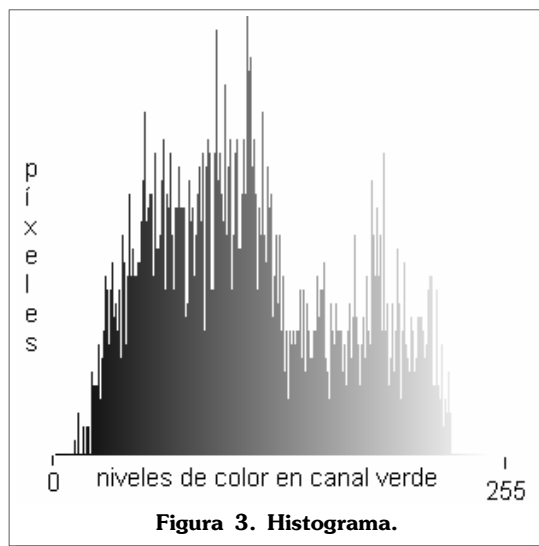


Figura 3. Histograma.

anteriormente, para este trabajo se utiliza el canal verde de las imágenes RGB en lugar de los niveles de gris, pero el rango sigue siendo el mismo.

La **figura 3** muestra el histograma del área del ojo seleccionado.

MOMENTOS INVARIANTES

Estos momentos codifican la superficie del objeto y son invariantes a traslación, rotación y escala. Estos momentos de orden p+q de una imagen I(x, y) se definen como:

$$M_{p,q} = \sum_{x=1}^N \sum_{y=1}^M x^p y^q I(x, y)$$

Para realizar una descripción del objeto independientemente de su posición se utilizan los momentos centrales que se obtienen a partir de la fórmula siguiente:

$$\mu_{p,q} = \sum_{x=1}^N \sum_{y=1}^M (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

Los momentos centrales normalizados se utilizan para describir al objeto, independientemente del tamaño que tengan, y están definidos por la fórmula:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^\alpha} \quad \alpha = \frac{p+q}{2} + 1$$

Posteriormente se extraen los momentos invariantes de Hu por medio de las fórmulas siguientes:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Wu Yan^[3] propone añadir información del borde y del histograma del objeto para obtener características que son invariantes a cambios en el nivel de gris de la imagen^[4]. Estos momentos se obtienen a partir de los momentos unidimensionales de la imagen y se definen como:

$$\beta_1 = \eta_4 / \eta_2^2$$

$$\beta_2 = \eta_5 / \eta_2 \eta_3$$

$$\beta_3 = \eta_6 / \eta_2 \eta_4$$

De esta forma, el vector característico que va a representar un ojo esta formado por 4 momentos de Hu y los 3 momentos de Yan.

En cada uno de los vectores característicos no se consideraron los tres últimos momentos de Hu debido a que tienen valores muy pequeños en comparación con los restantes, por tal motivo se optó por descartarlos.

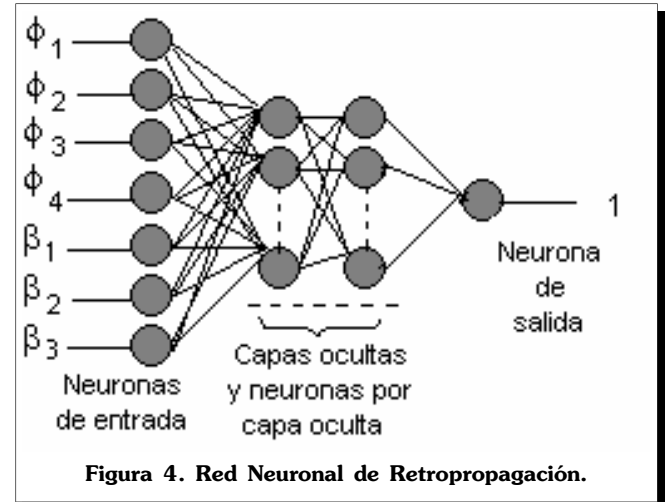
Vector característico = $[\phi_1, \phi_2, \phi_3, \phi_4, \beta_1, \beta_2, \beta_3]$

RED NEURONAL DE RETROPROPAGACIÓN

Una de las aplicaciones más extendidas de la red neuronal de retropropagación (BPN, Back Propagation Network) es el reconocimiento o clasificación de patrones^[5].

Para éste trabajo la red diseñada se compone de siete neuronas de entrada debido a que el vector característico tiene siete momentos y una neurona de salida con valor igual a 1 (describe a un ojo).

Para el entrenamiento de la red neuronal se varió el número de capas ocultas y el número de neuronas por capa con el propósito de verificar que arquitectura es más eficiente para disminuir el error de aprendizaje (figura 4).



En lo que respecta a la fase de prueba de la BPN, se utilizarán imágenes de igual tamaño que las imágenes que sirvieron como muestra para la extracción de características; para ello será necesario realizar un barrido sobre la imagen pixel por pixel hasta que la red encuentre las coordenadas (x, y) de la imagen que describan a un ojo.

Las imágenes a utilizar para la prueba del sistema serán en formato bmp para la evaluación de la eficiencia del mismo; posteriormente se emplearán los formatos de compresión para comparar los resultados obtenidos tomando en cuenta diferentes tamaños de imagen.

CONCLUSIONES

Para el funcionamiento óptimo del sistema se evalúa el número de capas ocultas necesarias que se emplean en la red neuronal, con esto se esperan resultados satisfactorios y una reducción de costos en cuanto a tiempo de procesamiento y complejidad de programación.

El uso de este sistema puede servir como una base para el desarrollo de diversos sistemas que se basen en

la extracción de características de rostros humanos y el reconocimiento de los patrones resultantes en imágenes digitales. Las aplicaciones posibles de este sistema pueden ser el reconocimiento de personas y en un futuro la autenticación de las mismas, esto a partir de sus ojos.

REFERENCIAS

- [1] González, R. C., Woods, R. E.; *"Tratamiento Digital de Imágenes"*, Addison Wesley, 1996.
- [2] Margulis, Dan; *"Professional Photoshop, Color Correction, Retouching and Image manipulation with Adobe Photoshop"*, Wiley, 1995.
- [3] Wu Yan, Ding Mingyue, Ann Peng Jiaxiong; *"Research on using moment invariants in scene matching"*, Institute of Image Recognition and Artificial Intelligence, HuaZhong. University of Science and Technology, WuHan, Hubei, China, 430074).
- [4] Ming-Kuei Hu; *"Visual pattern recognition by moment invariants"*, Information Theory, IEEE Transactions on, Volume: 8, Issue: 2, Feb 1962, Pages:179 – 187.
- [5] José R. Hilera, Víctor J. Martínez; *"Redes Neuronales Artificiales, fundamentos, modelos y aplicaciones"*, Alfaomega Ra-ma, 2000.
- [6] Kennet R. Castleman; *"Digital Image Processing"*, Prentice Hall, 1996.
- [7] Maria Petrou, Panagiota Bosdogianni; *"Image Processing, The fundamentals"*, Wiley, 1999.
- [8] A. Pérez, M. L. Córdoba, A. García, R. Méndez, M. L. Muñoz, J. L. Pedraza, Sánchez; *"A Precise Eye-Gaze Detection and Tracking System"*, Departamento de Arquitectura y Tecnología de Sistemas Informáticos (DATSI), Universidad Politécnica de Madrid.
- [9] D. Davies, P. Palmer, & M. Mirmehdi; *"Detection and Tracking of Very Small Low Contrast Objects"*, School of Electrical Engineering, Information Theory & Mathematics, University Surrey, England.
- [10] T. D'Orazio, M. Leo, A. Distante; *"Eye detection in face images for a driver vigilance system"*, CNR-ISSIA, Italia.
- [11] R. Thilak kumar, S. Kumar Raja and A. G. Ramakrishnan; *"Eye Detection using color cues and projection functions"*, Departament of Electrical Engineering, Indian Institute of Science.
- [12] Aguilar Ma. Elena, Sossa Humberto, *"Detección y Localización de Rostros Humanos en Imágenes Usando descriptores invariantes y Redes Neuronales"*, Informe Técnico, serie Azul, CIC-IPN, 2000.
- [13] Christian Bird and Bhaskar Shrestha; *"Tracking the Human Eye"*, University of Vermont.