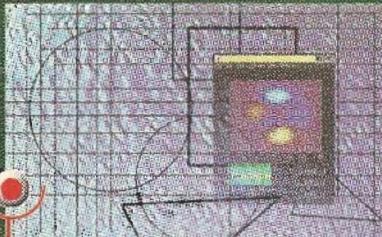


Electrónica & Computadores

• PROYECTOS • TECNOLOGIA • APLICACIONES PRACTICAS • TEORIA

ALARMA PARA MOTOCICLETAS



Construya una interfaz gráfica con MATLAB



Evolución de la arquitectura RISC



Generador de efectos sonoros



Generación digital de formas de onda

• Argentina \$ 4.90 • Bolivia \$ 1.490 • Chile \$ 1.490 • Colombia \$ 5.000 • Ecuador
• México \$ 30.⁹⁰ • Panamá \$ 3.50 • Paraguay \$ 1.490 • Uruguay \$ 1.490 • Venezuela Bs. 2.200

ISSN 0121-9138



0062

970121 913008

CONTENIDO

Editorial	2
Correo de los lectores	4

PROYECTOS

Alarma para motocicleta	6
Generador de efectos sonoros	10

ELECTRÓNICA INDUSTRIAL

Introducción al Control Difuso	17
--------------------------------------	----

ELECTRÓNICA BÁSICA

Leyes básicas de los circuitos	22
--------------------------------------	----

ELECTRÓNICA PRACTICA

Cuaderno de experimentos	27
Circuitos & Aplicaciones	29
Filtros programables	31

ELECTRÓNICA DIGITAL

Manejo de la memoria de datos EEPROM en microcontroladores PIC	36
--	----

TECNOLOGIA

El protocolo 1-Wire de Dallas Semiconductor	43
Manejo de las propiedades dieléctricas en la membrana celular	47
La Arquitectura RISC	52
Qué hay de nuevo en...	57
Nuevos Circuitos Integrados	65

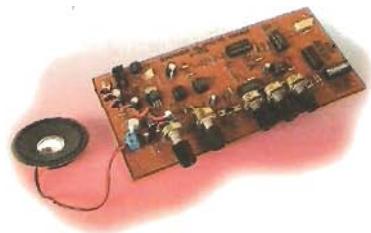
COMPUTADORES

Fundamentos de Redes de Computadoras. Parte 1.....	59
Rutas & Lugares	68
Construya una interfaz gráfica con Matlab	71

PROYECTO

Generador de efectos sonoros	10
------------------------------------	----

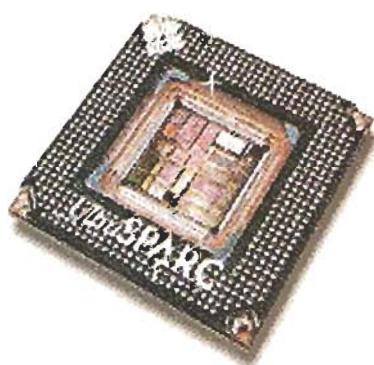
Construyendo este práctico y divertido generador de efectos sonoros, usted podrá simular desde el canto de un pájaro hasta una guerra espacial, pasando por explosiones, ruidos, disparos, sirenas, olas, helicópteros, motocicletas, teletipos, caballos al galope, tormentas, instrumentos y hasta ultrasonidos.



COMPUTADORES

La Arquitectura RISC	52
----------------------------	----

Todos conocemos las especificaciones básicas que caracterizan a las computadoras, como la velocidad y el tipo de procesador. Sin embargo, existe otra igual de importante pero poco conocida por los usuarios comunes. Se trata de la arquitectura, de la cual hay dos tipos principales la CISC y la RISC. Conozcamos un poco acerca de ellas.



EDITORIAL

Las motocicletas, por su gran número en el mercado y su facilidad de arranque y manejo, se han convertido en uno de los blancos favoritos de los amigos de lo ajeno. Seguramente muchos de nuestros lectores o sus amigos y familiares tienen un vehículo de este tipo. Qué mejor oportunidad entonces para practicar y fabricar una versátil y sencilla alarma con microcontrolador para evitar que esto suceda. Además, los lectores pueden complementarla y mejorarla ya que la capacidad del controlador no está totalmente utilizada.

El otro proyecto: "Generador de efectos sonoros", es un excelente circuito en el cual se combinan la electrónica análoga con la electrónica digital y tiene una gran utilidad didáctica así como un uso real muy interesante. Podríamos agregarlo a nuestro sistema de sonido para fiestas o para crear efectos sonoros en programas de radio, de televisión, en videos caseros, estudios de grabación, etc. Aquí se encuentran circuitos osciladores, multivibradores, VCO, PLL, amplificador de audio, mezclador digital, fuente de poder, etc.; una combinación difícil de lograr de lo análogo con lo digital que puede ser muy didáctica o representativa de los principales circuitos electrónicos básicos.



Seguramente Uds. habrán ya leído o escuchado mucho o poco sobre la lógica difusa o Fuzzy Logic, una de las técnicas de control que más se está empleando actualmente pero de la cual en la práctica sólo estamos recibiendo los primeros resultados. En el artículo que incluimos en esta edición, se hace una muy fácil y comprensible "Introducción al Control Difuso", que puede ser muy útil para aquellos que han querido abordar este tema y no han encontrado un punto de partida sencillo y fácil tal como el autor lo explica. De ahí en adelante lo más recomendable es emprender un proyecto real así sea muy simple pero como en toda la electrónica, la práctica hace al maestro. Trataremos de publicar un proyecto práctico sobre este tema a la mayor brevedad posible con el fin de profundizar en esta técnica que se irá haciendo cada vez más popular.

En el resto de la revista encontramos una muy buena cantidad de artículos sobre temas diversos como las "Leyes básicas de los circuitos" lo que será muy útil para los que se inician pero también sirve de "repaso" para los que ya las conocen y para los profesores en su tarea didáctica. Los conceptos básicos y aplicaciones de los filtros programables, el manejo de la memoria EEPROM de datos en los microcontroladores PIC, el protocolo I-wire de Dallas Semiconductor y las propiedades dieléctricas de la membrana celular cubren diferentes áreas de la electrónica tanto teórica como práctica.

En el tema de las computadoras, se hace una introducción a las redes en forma general, se habla de la arquitectura RISC y presentamos un tutorial paso a paso sobre las interfaces gráficas de Matlab, uno de los programas más utilizados actualmente en todo tipo de aplicaciones de ingeniería.

Felipe González G.

ecekit@avan.net

Marzo 2000

Editorial
CEKIT
Avenida 30 de Agosto N° 36 - 79 A.A. 194
PBX: 3291979. Fax Nat. (6) 3360311
Pereira, Colombia, S.A.
<http://www.cekit.com.co>
e-mail: cekit@col2.telecom.com.co

Editor y gerente general
Felipe González G.

Gerente administrativo y financiero
Marcelo Alvarez H.

Director técnico
Felipe González G.

Subdirector
Guillermo Ramos R.

Circulación Internacional
Humberto Real Blanco

Director Creativo
Juan Guillermo Escalante E.

Director de Arte
Albert Cobos P.

Diseño de carátula
Carlos Fernando Escobar

Departamento técnico:
Jorge Eduardo Hernández M.

Juan Andrés Castaño W.

Carlos Fernando Escobar

Fotografía

Héctor H. Ilméneg G.

Representación en Área II

Editorial CONOSUR S.A. Avda. Belgrano 355

Piso 10 - (1092) Cap. Fed.

Tel. (5411) 4342 - 9029 Fax. (5411) 4342 - 9025

Buenos Aires - Argentina

Gerente General: Horacio L. Nitoli

Edición Argentina

Editor responsable: Carlos Alberto Magurno S.

Ventas de Publicidad: 4342-9029

Distribuidores:

Argentina

Capital: Vaccaro Sánchez y Cía.

Moreno 794, 9º (1091) Buenos Aires

Interior: Distribuidora Bertran S.A.C.

Av. Velez Sarsfield 1950 (1285) Buenos Aires

Bolivia - La Paz: Agencia Moderna Ltda.

Colombia - Santa Fe de Bogotá: Distribuidoras Unidas

Chile - Santiago de Chile: Distribuidora Alta S.A.

Ecuador - Quito: Distribuidora Andes

México - Ciudad de México D.F.: Distribuidora INTERMEX

Panamá - Ciudad de Panamá: Distribuidora Panamex

Paraguay - Asunción: Selecciones S.A.C.

Uruguay - Montevideo: Gratia S.A.

Venezuela - Caracas: Distribuidora Continental

©PUBLICACIONES CECIT S.A. - Avenida 30 de Agosto

N° 36-79 - PBX: (6) 3291979 - A.A. 194 - Fax: (6) 3360311

Pereira, Colombia S.A. - 1.999 Pereira-Colombia S.A.

Febrero del 2000 - ISSN 0121-9138. Resolución 0444 de Mayo 18/94 de la Dirección General del Derecho de Autor, Mingobienro.

Prohibida su reproducción parcial o total por cualquier medio sin permiso escrito del editor. La protección de los derechos de autor se extiende no sólo al contenido, sino también a los diagramas y circuitos impresos, incluido su diseño, que en ella se reproducen.

Impresión:

DONNELLY COBRANI ARGENTINA

Panamericana Km. 36.7

Buenos Aires

Alarma para motocicleta

En estos tiempos de tanta inseguridad, ninguna precaución está de más. Construya este proyecto y regálese un poco de tranquilidad. Proyectos de este tipo nos demuestran la utilidad que pueden tener los microcontroladores en nuestra vida diaria.

JUAN ANDRÉS CASTAÑO WELGOS
welgos@telesat.com.co



En esta oportunidad desarrollaremos un sistema sencillo de alarma para motocicleta basado en un microcontrolador PIC. El funcionamiento es el siguiente: la activación de la alarma se hace con la llave de encendido de la motocicleta. Si el usuario da vuelta a la llave de encendido y la apaga dentro de un período de tiempo comprendido entre tres y seis segundos, el sistema se arma o se activa. Si esto ocurre, el pito de la moto se activa dos veces seguidas indicando que el sistema está listo y disponible para sensar cualquier movimiento. Si la llave de encendido se pone en ON por un período mayor a seis segundos, el sistema asumirá que la moto se ha encendido normalmente y por lo tanto no leerá el sensor ni entrará en estado de alerta o armado.

Este proyecto lo puede conseguir bajo la referencia *K-214 ALARMA PARA MOTO*, en cualquier distribuidor de productos CEKIT, en el cual se incluyen todos los elementos, el circuito impreso y la documentación. Si quiere información adicional sobre este proyecto puede escribirme o visitar mi página web www.welgos.homestead.com.

Sensor de movimiento

Una vez armado el sistema, chequeará permanentemente el estado de un sensor de movimiento dispuesto en algún lugar estratégico de la motocicleta. El sensor utilizado es un sensor de mercurio común, figura 1, que se puede encontrar fácilmente en el mercado. Este sensor básicamente consiste de un contacto normalmente abierto, el cual se

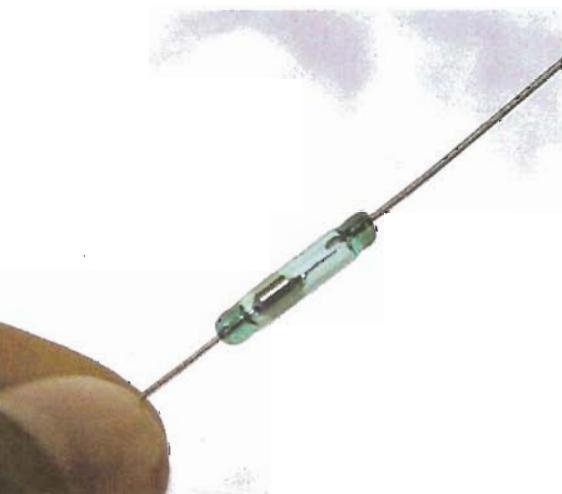


Figura 1. Sensor de movimiento.

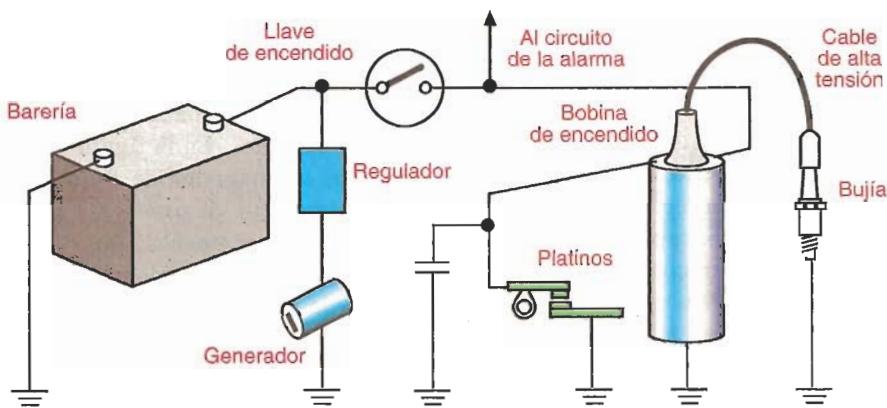


Figura 2. Esquema básico del encendido de una motocicleta

cierre cuando debido al movimiento del sensor, el mercurio líquido se acerca al contacto y sirve de puente entre los dos puntos.

Cuando desarrolle algún proyecto de esta naturaleza, lo más importante que se debe garantizar es que el circuito eléctrico esencial de la moto debe quedar independiente de la nueva conexión realizada de la alarma. Esto quiere decir que se deben evitar problemas como por ejemplo, que la motocicleta no funcione en el caso que el circuito de la alarma se dañe o se moje. Lo ideal es que en caso de emergencia, el circuito pueda ser «deshabilitado» por cualquier persona que no tenga conocimiento alguno en electrónica y la moto funcione normalmente. El esque-

ma de encendido de una moto se puede apreciar en la figura 2, consta de varios elementos como el generador, la bobina y muchos otros elementos que en este documento no se analizarán. Se puede observar que la llave de encendido en realidad funciona como un interruptor normalmente abierto el cual se cierra y permanece así mientras la moto está encendida.

Descripción del circuito

El diagrama esquemático del proyecto se puede ver en la figura 3. La alimentación normal de la alarma se toma de la batería de 12 V de la motocicleta. Debido a que el circuito no presenta requerimientos importantes de corriente, se decidió colocar un diodo zener de 9.1V en serie. Al po-

nerle los 12V, el diodo entra en su región de avalancha y el resultado es un voltaje de aproximadamente 3.05 V, los cuales son suficientes para alimentar el circuito y para que el micro de la salida apropiada. Adicionalmente se han dispuesto un condensador electrolítico y uno cerámico a la salida de la alimentación para evitar problemas de ruido, los cuales son comunes al trabajar en esta clase de fuentes y que pueden ser fatales para el circuito. La entrada al microcontrolador de la señal de la llave de encendido también se rebaja a 3.05 voltios utilizando el mismo principio del diodo zener.

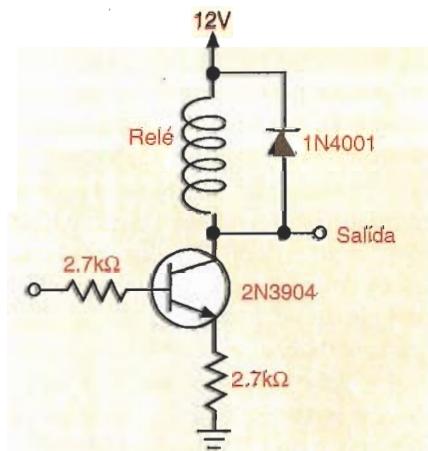


Figura 4. La señal de activación originada en el micro es llevada a un relé.

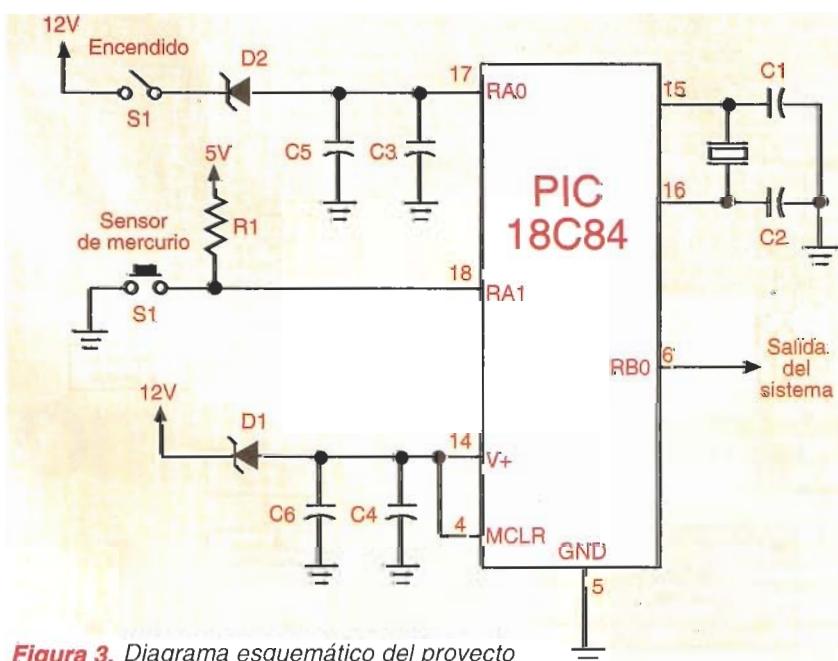


Figura 3. Diagrama esquemático del proyecto

Lista de materiales

Resistencias a 1/4W

- 10KΩ (R1)
- 2.7KΩ (2)

Condensadores

- 20 pF cerámico (C1,C2)
- 100 pF cerámicos (C3, C4)
- 10 µF electrolíticos (C5, C6)

Circuitos integrados

- PIC18F84

Otros

- Base para integrado de 18 pines
- Cristal a 4 MHz
- Diódos zener de 9.1V
- Transistores 2N3904 (Q1)
- Circuito impreso K-214
- Circuito Impreso RELE1
- Diódos IN4001
- 10 µF electrolíticos (C5, C6)
- Relé 12V
- conectores de tornillo tres terminales
- conectores de tornillo dos terminales

Tabla 1.

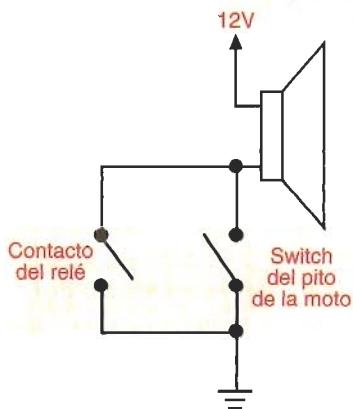


Figura 5. La señal del relé debe activar el pito de la motocicleta

Circuito de salida con relé

Para dar la salida hacia el pito de la moto, se decidió incorporar un circuito de relé en una tarjeta separada, esto es porque posiblemente sea más fácil ocultar en una moto dos tarjetas pequeñas que una sola pero más grande. El microcontrolador entrega sus señales de alarma por medio del pin 6, o pin 0 del puerto B (RB0,0), y dispara el contacto del relé, figura 4. La fotografía del circuito del relé se puede apreciar en la portada de este artículo. Esta señal se debe acoplar al pito de la moto, lo que se puede hacer como se muestra en la figura 5. Cuando la alarma se encuentra en estado de armado y detecta movimiento en el sensor, el pito empezará a sonar.

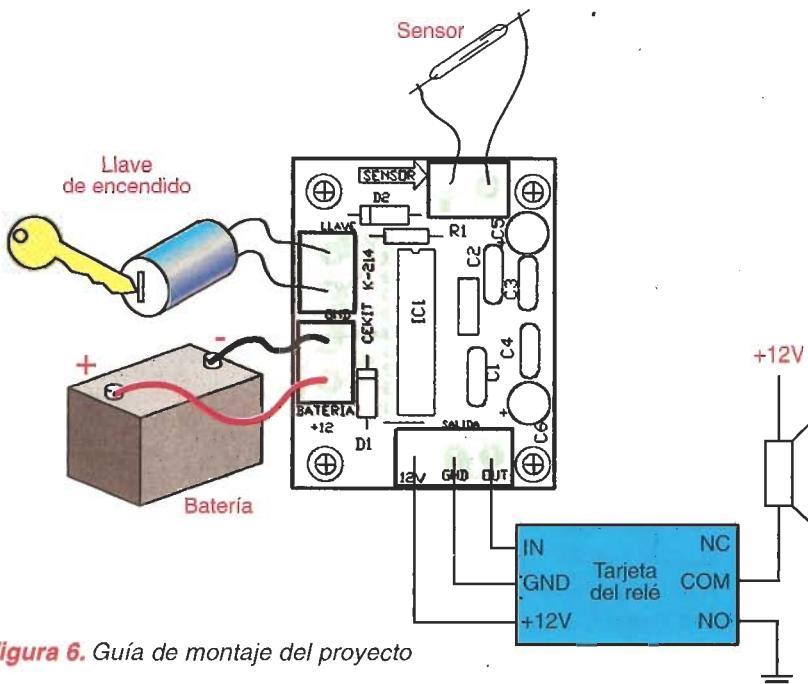


Figura 6. Guía de montaje del proyecto

Círculo impreso del proyecto

Para ocultar la alarma dentro de la moto, esta se diseñó para que quede lo más pequeña posible. Sus dimensiones reales son de 3.7cm x 4.4 cm y sería recomendable también que una vez instalada la recubriera con alguna clase de resina, para evitar que el circuito entre en contacto con el medio ambiente. Para evitar confusiones y dar claridad en el montaje del circuito, puede guiarse por la guía de montaje que se ilustra en la figura 6, en ella se puede ver la ubicación de los distintos componentes dentro del circuito, la fotografía final del proyecto se aprecia en la portada de este artículo y la lista de materiales se entrega en la tabla 1.

Programa del microcontrolador

En realidad el programa es bastante sencillo. Una vez configurado el mapa de memoria de la CPU, los puertos y la rutina de reloj (puede hacerse por tmr0 o por una sencilla rutina de retraso), se debe chequear si se dan las condiciones bajo las cuales la alarma se puede armar. Cuando estas condiciones se han establecido positivamente, se entra al estado de armado y se chequea continuamente el puerto donde se lee el estado del sensor de movimiento. Cuando se lee una señal de activación de ese sensor, se activa la alarma y se envía la señal al pito

de la moto. En realidad el programa lleva toda una serie de banderas cuyo estado se debe leer continuamente para saber en qué parte del proceso estamos ubicados. En la figura 7 observamos un diagrama de flujo global del proyecto. Seguramente se le pueden adicionar muchas otras características de funcionamiento a esta alarma, de manera que si le llama la atención, síntase invitado a profundizar en este proyecto. ☺

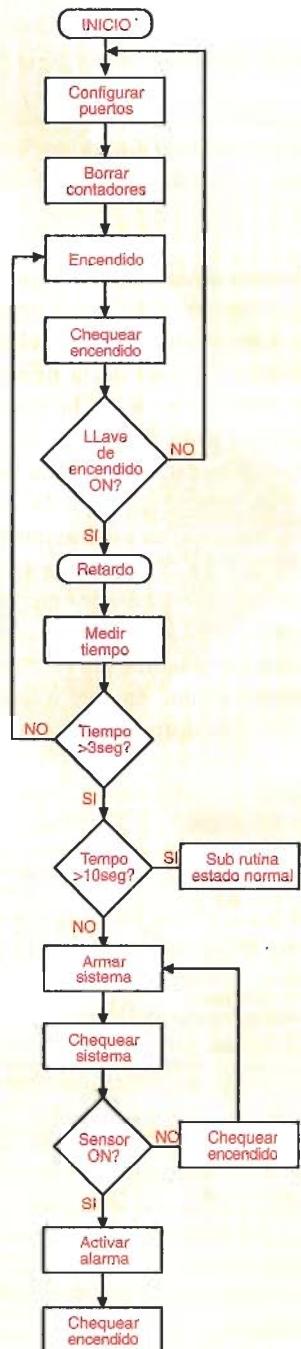
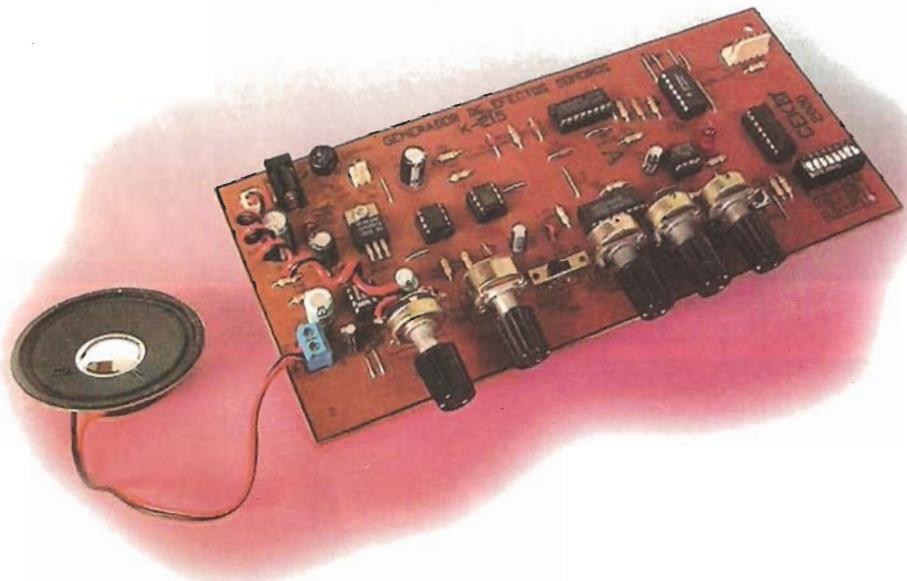


Figura 7. Diagrama de flujo del programa

Generador de efectos sonoros

Construyendo este práctico y divertido generador de efectos sonoros, usted podrá simular desde el canto de un pájaro hasta una guerra espacial, pasando por explosiones, ruidos, disparos, sirenas, olas, helicópteros, motocicletas, teletipos, caballos al galope, tormentas, instrumentos y hasta ultrasonidos.



JORGE E. HERNÁNDEZ M.
edher@telesat.com.co

El generador de efectos sonoros que se describe en este artículo permite simular ruidos y sonidos de muchos tipos, incluyendo pitos, sirenas, disparos, explosiones, revólveres espaciales, autos de carreras, aviones, etc. El mismo está desarrollado alrededor de unos pocos circuitos integrados lineales y CMOS de fácil consecución, incluyendo dos temporizadores 555, un operacional LF353, un PLL 4046B, un multiplexor 4051B, un multiplexor 4052B, un juego de compuertas AND 4081B y un amplificador de audio LM386.

Los diferentes efectos se programan mediante unos pocos potenciómetros e interruptores. También incluye provisiones para conectar un generador de ruido externo y un modulador de tiempos de ataque y de caída.

Descripción y operación

En la **figura 1** se muestra el diagrama general de bloques de nuestro generador de efectos sonoros, identificado con la referencia **K-215** de **CEKIT**. En las **figuras 2** hasta **8** se presentan los diagramas esquemáticos correspondientes a cada bloque. El sistema consta básicamente de los siguientes circuitos:

- Un oscilador de superbaja frecuencia (SLFO)
- Un oscilador controlado por voltaje (VCO)
- Un selector de control de VCO
- Un mezclador digital con modulador AND interno
- Un multivibrador monostable
- Un selector de envolvente
- Un amplificador de audio
- Una fuente de alimentación

La fuente de alimentación, **figura 2**, está desarrollada alrededor de un puente rectificador de onda completa (BR1), un regulador monolítico de 5V (IC9) y sus componentes asociados (C8-C10, R19, LED3). La misma suministra una tensión no regulada de +12V para el amplificador de audio y una tensión regulada de +5V para el resto de la circuitería analógica y digital. El sistema puede ser alimentado con tensiones AC o DC desde 6V hasta 15V.

El oscilador de superbaja frecuencia (SLFO), **figura 3**, está desarrollado alrededor de un temporizador 555 (IC1), un amplificador operacional LF353 (IC2) y sus componentes asociados (R1-R4, C1-C2, D1, P1). El mismo suministra una onda cuadrada para el mezclador y una onda triangular para

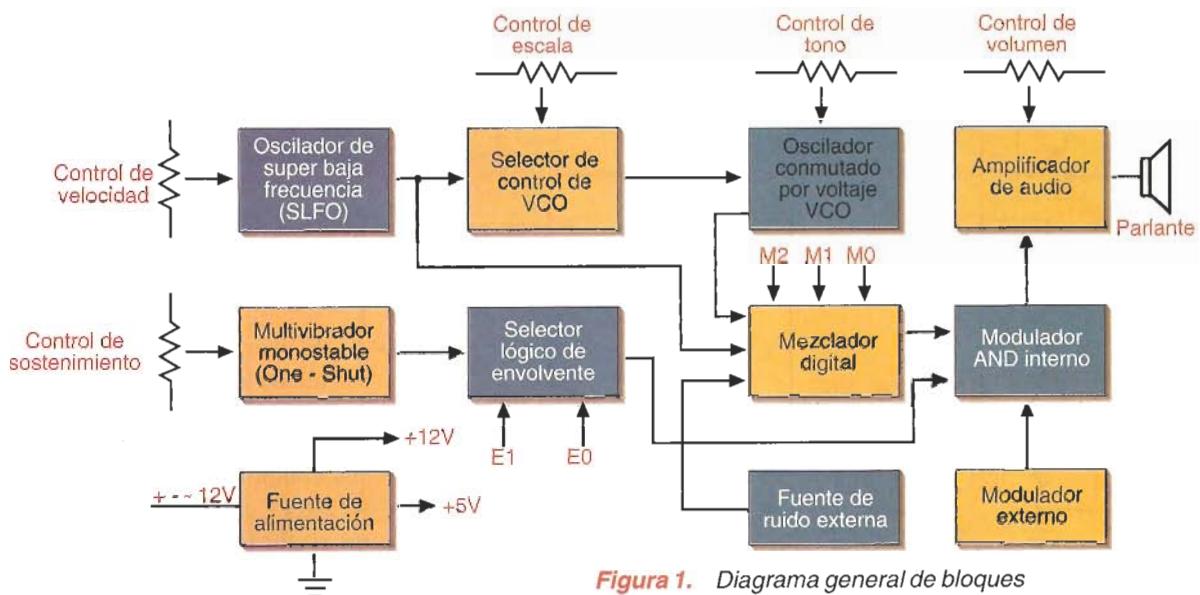


Figura 1. Diagrama general de bloques

el VCO. La frecuencia de estas señales es variable entre 0.1 Hz y 30Hz, aproximadamente, y se controla mediante P1.

El oscilador controlado por voltaje, figura 4, está desarrollado alrededor de un PLL digital 4046B (IC3) y sus componentes asociados (P2, P3, R5-R6, C3, S1). El mismo genera una onda cuadrada que alimenta tanto el mezclador como el selector de envolvente. La frecuencia de esta señal puede ser controlada internamente por la onda triangular del SLFO o externamente por un nivel DC aplicado a través de P3. La selección de uno u otro modo se realiza mediante S1. En cualquier caso, la variación de la frecuencia dentro del rango seleccionado se realiza mediante P2. Combinando los efectos de P2 y P3, el VCO puede proporcionar frecuencias desde 80 Hz hasta 30kHz, aproximadamente.

El monostable, figura 5, está desarrollado alrededor de un temporizador 555 (IC4) y sus componentes asociados (P4, R7-R9, C4, S3, LED1). Su función es producir un pulso de corta duración cuando se acciona S3 y permitir la simulación de sonidos momentáneos como disparos, campanas, explosiones, etc. Este tiempo es variable entre 0.1 y 5 segundos, aproximadamente, y se controla mediante P4.

El selector lógico de envolvente, figura 6, está desarrollado alrededor de un multiplexor análogo de 4 canales 4052B (IC5), una compuerta AND

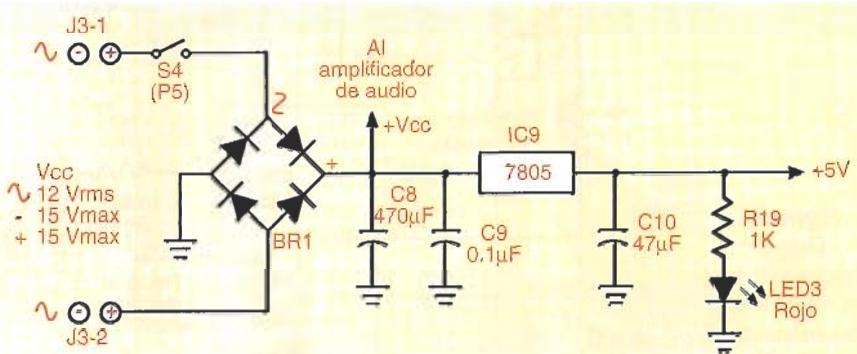


Figura 2. Diagrama esquemático de la fuente de alimentación

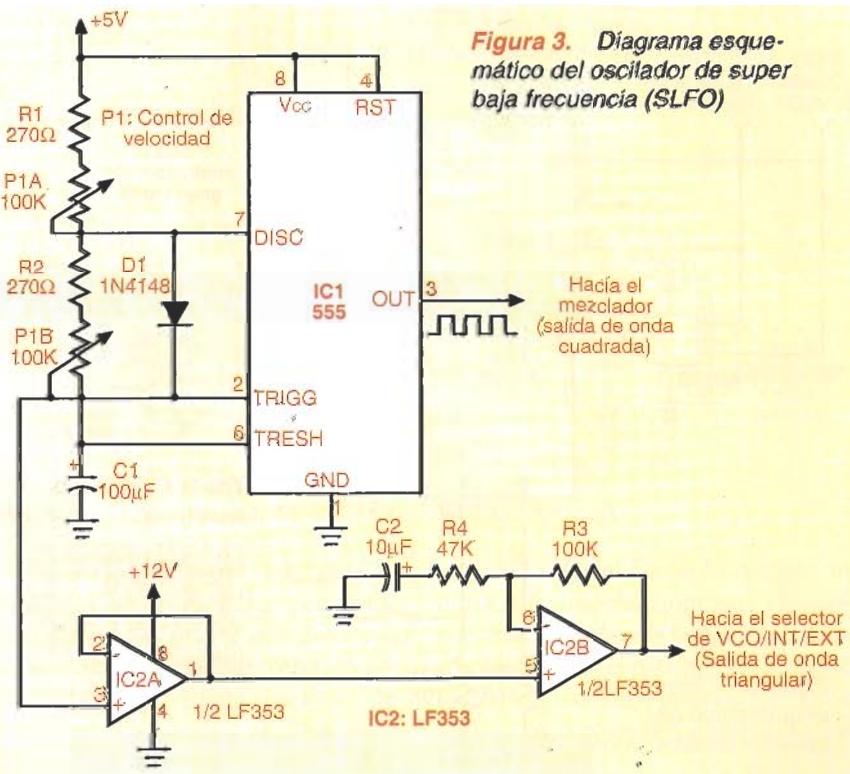


Figura 3. Diagrama esquemático del oscilador de super baja frecuencia (SLFO)

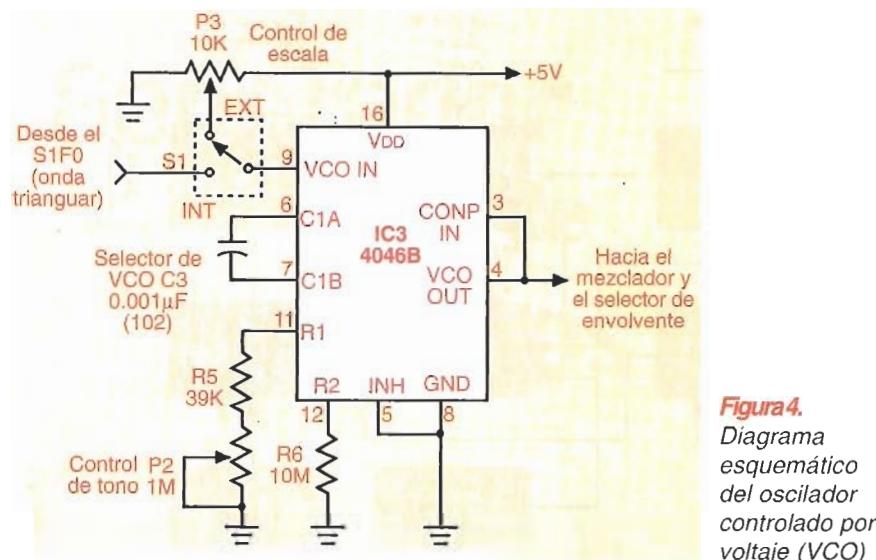


Figura 4.
Diagrama esquemático del oscilador controlado por voltaje (VCO)

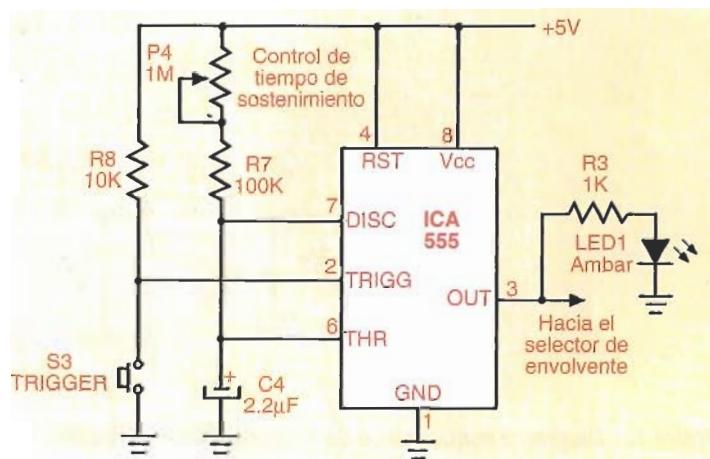


Figura 5.
Diagrama esquemático del multivibrator monostable (One-Shot)

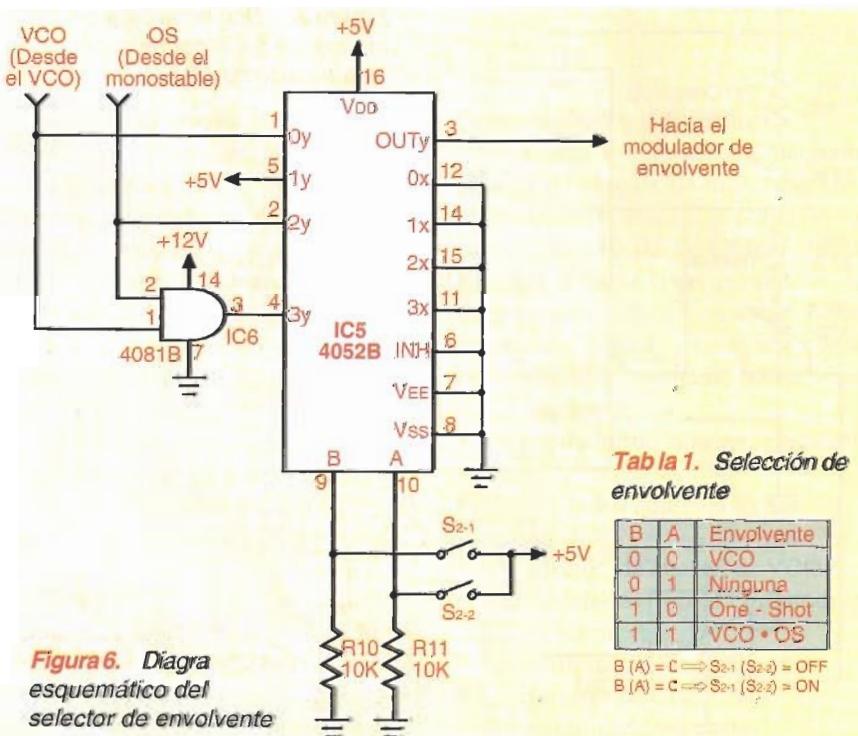


Tabla 1. Selección de envolvente

B	A	Envolvente
0	0	VCO
0	1	Ninguna
1	0	One-Shot
1	1	VCO + OS

B (A) = C \Rightarrow S₂₋₁ (S₂₋₂) = OFF
B (A) = C \Rightarrow S₂₋₁ (S₂₋₂) = ON

4081B (IC6A) y sus componentes asociados (R10-R11, S2-1,2). Su función básica es producir la envolvente o señal moduladora de la onda de salida del mezclador. Dicha envolvente, que puede ser el pulso monostable, los pulsos del VCO, una combinación de ambos o ninguna, se selecciona mediante S₂₋₁ y S₂₋₂, como se indica en la tabla adjunta.

El mezclador, figura 7, está desarrollado alrededor de un multiplexor analógico de 8 canales 4051B (IC7), 3 compuertas AND 4081B (IC6B-D) y sus componentes asociados (D2-D5, R12-R17, J1, J2, S2-3,4,5,6). El mismo recibe las señales provenientes del SLFO, el VCO y un filtro de ruido externo, las mezcla o combina y suministra la señal resultante al modulador interno. Este último (IC6D), a su vez, la combina con la señal entregada por el selector de envolvente para atacar el amplificador. La señal de mezcla se programa mediante S2-3, S2-4 y S2-5, como se indica en la tabla adjunta a la misma figura.

El mezclador incluye provisiones para conectar a una fuente de ruido blanco externa y un modulador externo. Mediante la disponibilidad de una fuente de ruido blanco, aplicada a J1, se pueden simular sonidos agradables y relajantes, como los producidos por las olas del mar o la lluvia, y proporcionarle más realismo a explosiones, galope, etc. Si no se utiliza esta función, J1 puede dejarse al aire.

La idea del modulador externo, por su parte, conectado a J2, es permitir la modificación de los tiempos de ataque y de decaimiento de la señal de salida del modulador interno con el fin de conseguir efectos especiales. Si no se utiliza esta función, el interruptor S₂₋₆ debe mantenerse ON. En futuras entregas de **Electrónica & Computadores** mostraremos algunos diseños prácticos de fuentes de ruido y moduladores adecuados para esta aplicación.

El amplificador de audio, figura 8, está desarrollado alrededor de un LM386 (IC8) y sus componentes asociados (P5, C5, C6-7, S2-7, SP1, R18, LED2). Su función básica es reproduc-

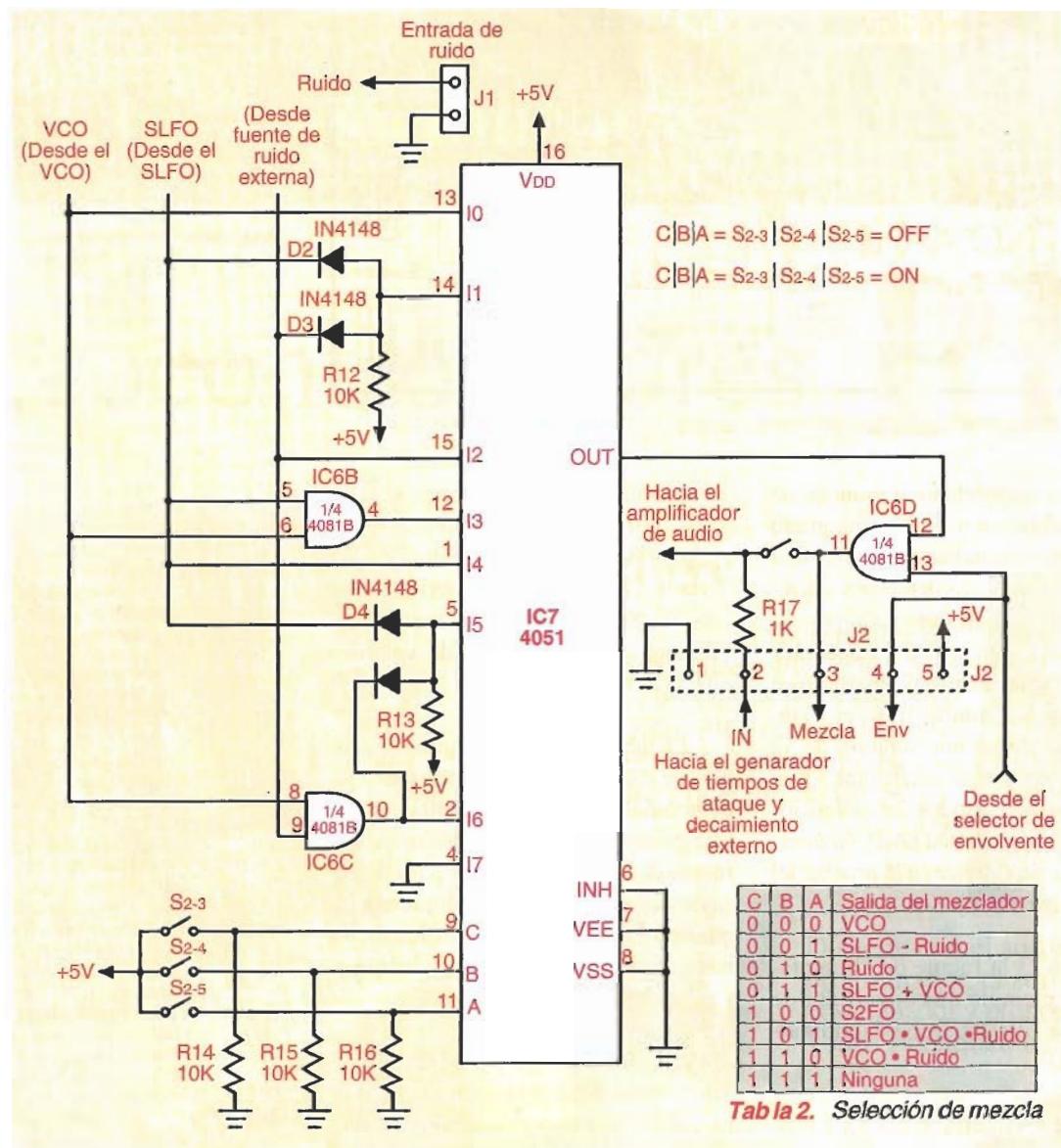


Figura 7. Diagrama esquemático del mezclador digital y el modulador interno

cir la señal de salida del modulador, interno o externo, en un parlante (SP1). El volumen del sonido producido se controla mediante P5. También incluye un selector de ganancia (S2-7) commutable entre 20 (OFF) y 200 (ON). Esta última opción es adecuada para sonidos débiles o de muy baja frecuencia,

Construcción y prueba

En la figura 9 se muestra la guía de localización de componentes del generador de efectos sonoros CEKIT K-215 sobre la tarjeta de circuito impreso suministrada con el kit. La lista completa de materiales requeridos se relaciona en un recuadro. El montaje de la tarjeta se realiza en la forma usual, comenzando por

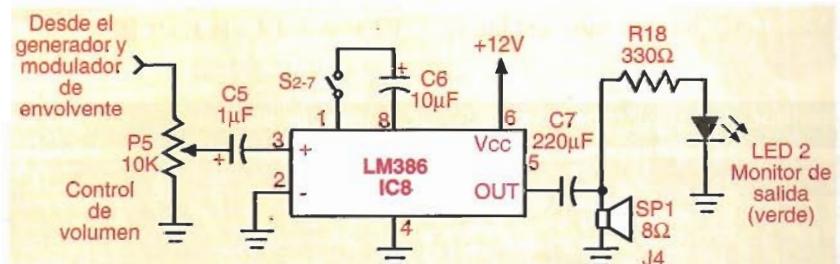


Figura 8. Diagrama esquemático del amplificador de audio

los puentes de alambre que complementan las pistas de cobre. Hecho esto, continúe con las resistencias fijas, los diodos, los condensadores cerámicos y electrolíticos, los LEDs, las bases de los circuitos integrados, el puente rectificador y el regulador de voltaje. Por

ultimo, instale el botón de disparo, el dipswitch, los potenciómetros y los conectores de acceso de la fuente, el interruptor general, el parlante, la fuente de ruido y el modulador externo. Estos dos últimos son opcionales y, por ahora, no los utilizaremos..

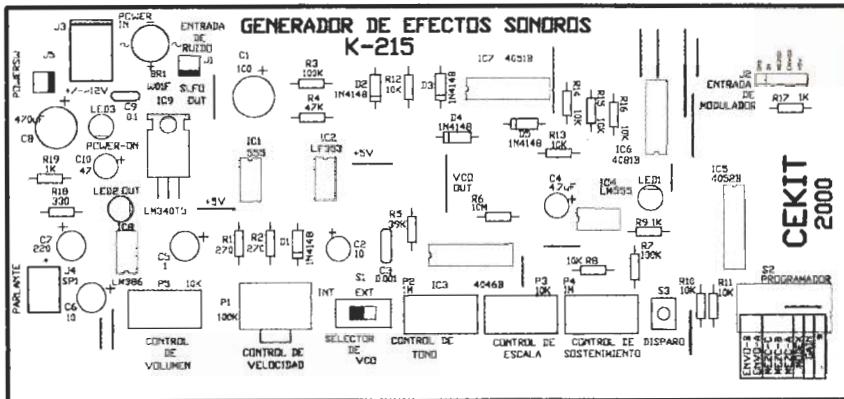


Figura 9. Guía de localización de componentes

Una vez completado el montaje de la tarjeta, instale cada circuito integrado en su base correspondiente y realice una inspección visual para detectar y corregir errores de montaje como componentes mal instalados u orientados, puentes de alambre faltantes, pistas abiertas o en cortocircuito, soldaduras frías, etc. Hecho esto, sitúe todos interruptores de S2 en la posición OFF, el interruptor S1 en la posición EXT y todos los potenciómetros de control en una posición intermedia. Proceda entonces a la prueba del proyecto. Siga estos pasos:

- Conecte a J3 la fuente de alimentación, por ejemplo un convertidor de 120VAC a 12VDC. Deberá iluminarse el LED3
- Programe los interruptores S2-1 hasta S2-5 de acuerdo a la combinación de efectos que desee producir y manipule los controles asociados a cada una hasta obtener sonidos reconocibles. Los sonidos momentáneos,

como disparos, explosiones y acordes, por ejemplo, se obtienen seleccionando el monostable como envolvente (ver figura 6) y activando el botón S3 (trigger). Para mayorrealismo, ajuste el control de volumen (P5) al máximo.

La tabla 3 relaciona algunos ejemplos de sonidos comunes que usted puede simular o sintetizar fácilmente con este generador, disponiendo o no de una fuente de ruido externa. Despues de ensayar con estos efectos, dedíquese a juzgar con los diferentes controles, botones e interruptores para descubrir nuevas posibilidades.

Cuando obtenga un efecto interesante, fíjese bien en la posición final de los controles y anote sus observaciones para poderlo reproducir más tarde. Por ejemplo, puede adoptar una convención horaria para describir la posición de los potenciómetros, digamos P1 a las 11, P2 a las 3, P4 a la 1, etc. \square

CONTROL DE VCO	ENVOLVENTE	SEÑAL DE MEZCLA	CONTROLES ACTIVOS	SONIDOS PRODUCIDOS
EXT	VCO	VCO	P2, P3	Tonos graves hasta ultrasónicos
EXT	OS	VCO	S3, P2, P3	Notas musicales y acordes varios
INT	Cualquiera	VCO	P2, P1, S7	Guerra espacial, silbidos, sirenas
EXT	VCO	SLFO	P1	Teletipos, alarmas monitoras
EXT	VCO*OS	Ruido	R2, (*), P2, P1	Olas, ruido relajante
INT	VCO*OS	SLFO*VCO	P1, S3	Guerra espacial, trinos de pájaros
INT/EXT	Cualquiera	SLFO	P1, P2, (*), P5, P1, P3, P2, (*)	Golpeteos, motocicletas, cascos
INT	VCO*OS	VCO*RUIDO	P2, P1, (*), P3	Lluvia, patrullas, ambulancias
INT/EXT	VCO	VCO*RUIDO	P2, P3, (*)	Locomotora de vapor
INT	VCO	SLFO*Ruido*VCO	P1, P2	Diligencia, caballos al galope
INT/EXT	VCO	SLFO*VCO*Ruido	S3, P1, P5, (*)	Avión bimotor, tractor, lancha
INT	VCO	Ninguna	S3, P1, (*)	Encendido de un carro
INT/EXT	VCO*OS	SLFO*Ruido	Ninguno	Fumigaciones, explosiones
INT	VCO*OS	SLFO*RUIDO	Ninguno	Disparos, guerra terrestre
INT	Ninguna	Ninguna	Ninguno	Silencio total

(*): Control de ancho de banda del filtro de ruido externo. Se recomienda que este ultimo sea del orden de 50 a 70 kHz

Lista de materiales

Resistencias fijas (5%, 1/4 W)

R1,2	270Ω
R3,7	100K
R4	47K
R5	39K
R6	10M
R8,10-16	10K
R9,17,19	1K
R18	330Ω

Potenciómetros rotatorios

P1	100K, doble, log
P2	1M, log
P3	10K, lineal
P4	1M, lineal
P5	10K, log, con interruptor

Condensadores

C1	100μF/16V, electrolítico
C2,6	10μF/25V, electrolítico
C3	0.001μF/50V, cerámico
C4	2.2μF/16V, electrolítico
C5	1μF/25V, electrolítico
C7	220μF/25V, electrolítico
C8	470μF/25V, electrolítico
C9	0.1μF/25V, electrolítico
C10	47μF/16V, electrolítico

Semiconductores

BR1	Puente rectificador WO1F
D1-5	Diodos de conmutación 1N4148
IC1,4	LM555 (timer)
IC2	LF353 (amplificador operacional)
IC3	4046B (PLL digital)
IC5	4052B (mux análogo, 4 canales)
IC6	4081B (compuertas AND)
IC7	4051B (mux análogo, 8 canales)
IC8	LM386 (amplificador de audio)
IC9	LM340T5 (regulador de 5V)

Transductores

SP1	Parlante dinámico, 8Ω, 0.25W
LED1	LED ámbar, 5 mm
LED2	LED verde, 5 mm
LED3	LED rojo, 5 mm

Interruptores para circuito impreso

S1	Deslizante, spdt, miniatura
S2	Dipswitch de 8 secciones
S3	Pulsador normalmente abierto

Conectores para circuito impreso

J1	En línea, 2 pines, polarizado, H/M
J2	En línea, 5 pines, polarizado, H/M
J3	Jack para adaptador AC/DC o AC/AC
J4	Bornera de tornillos, 2 pines

Accesorios

4	Bases DIP de 8 pines (IC1,2,4,8)
1	Base DIP de 14 pines (IC6)
3	Bases DIP de 16 pines (IC3,5,7)
5	Perillas para potenciómetros
1	Tarjeta de circuito impreso CEFIT K-125
1	Tornillo milimétrico de 3x5 con tuerca

Varios

Alambre telefónico #22 o #24 para puentes (2 m)
Soldadura (2 m)

Tabla 3. Ejemplos de sonidos sintetizables con el generador K-125

al Control Difuso



La inteligencia artificial es un campo que ha despertado gran interés entre los científicos de todo el mundo. En los últimos años estas ideas se han venido introduciendo en las aplicaciones industriales. En este artículo analizaremos brevemente como la técnica de la lógica difusa puede usarse para adicionarle cierto grado de inteligencia a las máquinas.

JUAN ANDRÉS CASTAÑO WELGOS

welgos@telesat.com.co

Suponga que quiere medir la temperatura existente en una habitación. Si pudiéramos preguntarle a varias personas sobre esa temperatura en un momento dado, posiblemente encontraríamos un rango variado de respuestas como por ejemplo «está fresco», «cálido», «un poco frío», etc. Esto se debe a que cada persona tiene su propio punto de vista.

Si tuviéramos a nuestra disposición un termostato en esa habitación, ese sensor nos entregaría una respuesta concreta, como «frío» o «caliente». En la

figura 1 vemos la comparación entre el concepto vago del humano y la respuesta concreta del sensor.

La lógica difusa se puede usar para construir sistemas de control que funcionen tratando y clasificando los datos de una manera similar a como lo haría un ser humano.

La lógica difusa categoriza los datos por medio de probabilidades. Imaginemos que se hizo una recopilación de los datos del punto de vista que tienen varias per-

sonas sobre la temperatura en ese cuarto. Esos resultados pueden arrojar, por ejemplo, que cuando la temperatura está por debajo de 10 °C todos dicen que es frío. Que a los 20 °C 50% de los encuestados dicen que es frío y el otro 50% dicen que es caliente y cuando la temperatura está por encima de los 30 °C, todos dicen que es caliente. Estos resultados se pueden plasmar en una gráfica, figura 2. Bajo este esquema, una temperatura de 25 °C puede describirse como «ligeramente frío» (probabilidad 0.25) y «casi caliente» (probabilidad 0.75).

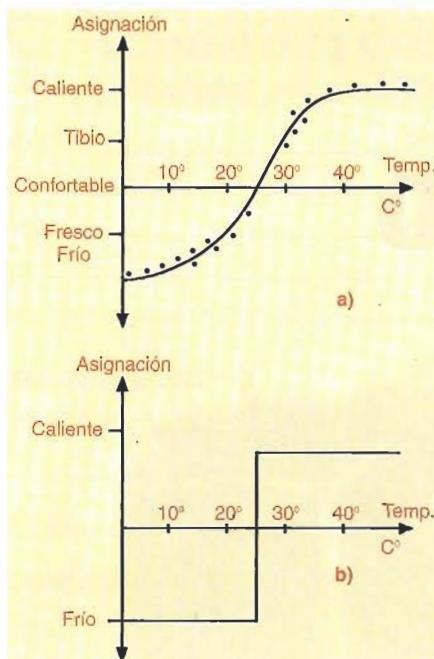


Figura 1. Diferencia entre la asignación difusa y la concreta. a) Punto de vista de las personas. b) Punto de vista del termostato

Clasificación de los datos

Observemos que en la figura 2, se clasificaron los datos en dos grupos, caliente y frío.

Si fuéramos a analizar por ejemplo, la edad de las personas, encontraríamos cinco grupos: niños, jóvenes, adultos jóvenes, adultos de edad intermedia y ancianos. Si quisieramos plasmar estos conceptos en una gráfica, obtendríamos algo como lo que muestra la figura 3 (a).

Esto podría acarrear ciertas incoherencias;

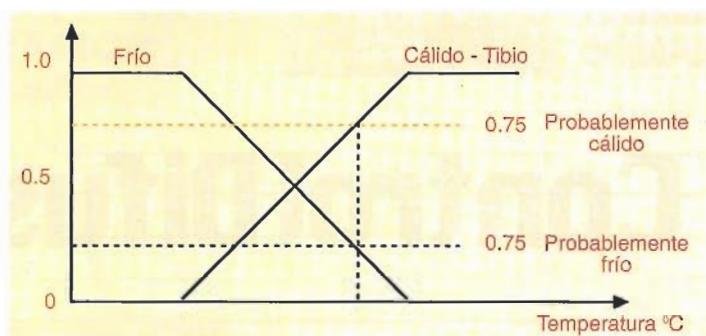


Figura 2. Temperaturas asignadas como probabilidades

por ejemplo que una noche una persona se acostará siendo un adulto de edad intermedia, y despertara al otro día siendo un anciano. Esto como se pude notar, no corresponde fielmente a la realidad. Un arreglo más realista es el que se aprecia en la figura 3(b). En este esquema se definen las probabilidades para cada grupo de edades. Alguien que tenga 27 años puede describirse como «un tanto joven» (probabilidad 0.5), «adulto joven» (probabilidad 0.85) o «adulto de edad intermedia» (probabilidad 0.1).

El primer paso para construir un sistema basado en lógica difusa es considerar cuántos datos se van a clasificar. Una clasificación que es muy conocida es establecer los siguientes grupos:

- PG Positivo grande
- PM Positivo mediano
- PP Positivo pequeño
- ZR Aproximadamente cero
- NP Negativo pequeño
- NM Negativo mediano
- NG Negativo grande

Todos estos conceptos pueden usarse en un esquema de control difuso, clasificando los datos dentro de los grupos. El primer paso consiste en establecer las probabilidades; por lo general siempre se tiende a usar una distribución de probabilidad; entre las más usadas se encuentran la triangular, la campana y la trapezoidal, figura 4.

Por ejemplo, observemos cómo hacer un sistema de control de un automóvil. Se puede pensar en dos variables análogas, la velocidad del vehículo y la distancia con respecto al vehículo de enfrente. Un sistema de control difuso tomará esas dos entradas y las procesará para controlar el carro. La figura 5 es un diagrama de nuestro esquema de control.

Primero clasifiquemos nuestras dos entradas usando cuatro de las siete posibles categorías: aproximadamente cero (ZR) y las tres categorías positivas: positivo pequeño (PP), positivo mediano (PM) y positivo grande (PG).

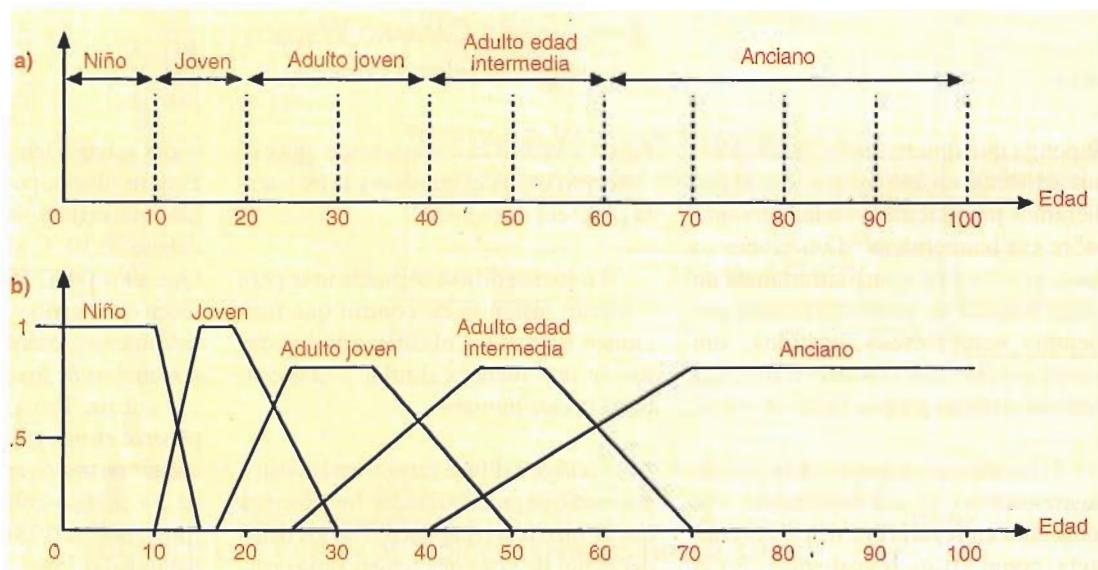
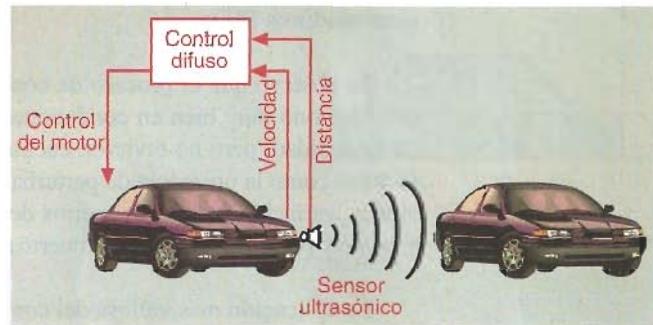
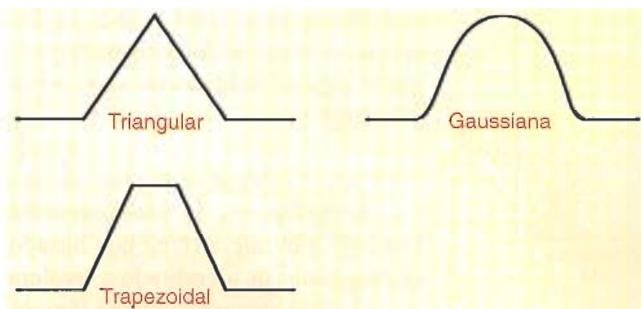


Figura 3. Asignación de la edad en cinco grupos. a) Asignación concreta. b) Asignación probabilística de la edad



Decimos que el vehículo tiene una velocidad próxima a cero cuando tiene un rango de velocidades comprendido entre 0 y 20 Km/h y se puede afirmar que presenta «baja velocidad» cuando está entre 0 y 40 km/h. Las cuatro categorías se pueden representar como se muestra en la figura 6.

Ahora debemos representar la distancia con respecto al carro de enfrente. En esta ocasión haremos esto de la manera más sencilla posible, como se ve en la figura 7. En un sistema real esto puede ser más complejo.

Nuestro sistema de control tiene una sola salida: la fuerza que se le debe aplicar al carro. En los sistemas reales es común que se deban implementar varias salidas de control. En todo caso, esta fuerza debe clasificarse también en conjuntos difusos, figura 8.

Definición de las reglas

Aquí se establece la estrategia de control que puede ser algo como:

Regla 1: Si la distancia entre los carros es corta y la velocidad es un tanto alta, entonces se debe impulsar fuerte el motor.

Regla 2: Si la distancia entre los carros es moderadamente larga y la velocidad

del carro es alta, entonces se debe impulsar el auto moderadamente fuerte.

Si definimos estas señales así:

Velocidad V

Distancia D

Fuerza F

Se pueden escribir estas reglas usando un formato SI/ENTONCES:

Regla 1: SI $D = PS$ y $V = PM$
ENTONCES $F = PG$

Regla 2: SI $D = PM$ y $V = PG$ ENTONCES $F = PM$

La regla 1 puede diagramarse como se ve en la figura 9.

Evaluación de las reglas

Para demostrar como se aplican esas reglas, vamos a imaginar un viaje en el que vamos a 55 km/h y estamos a una distancia de 27 m del carro de enfrente.

Primero observemos los valores de cada término en las reglas. Vamos a hablar sobre la probabilidad de cercanía a cada conjunto, o grado. De la regla 1 el grado de $V = PM$ es 0.25, y el grado de DS es 0.38. Analizando la regla 2 nos damos cuenta que el grado de $V = PG$ es 0.75 y de $D = PM$ es 0.62.

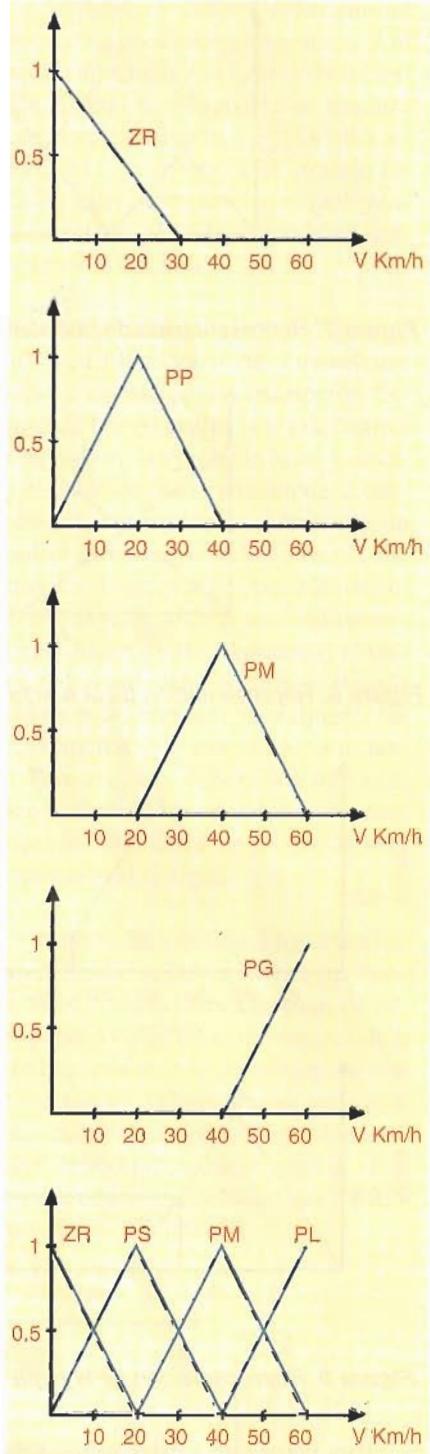


Figura 6. Representación de la velocidad del vehículo

Ahora tenemos cuatro valores para $B=0.25$ y 0.38 por la regla 1, más 0.75 y 0.62 por la regla 2. El paso siguiente es partir de estos valores del conjunto difuso hacia un solo valor el cual pueda ser enunciado como señal de control al carro. Esto es lo que se conoce como **defuzzificación**.

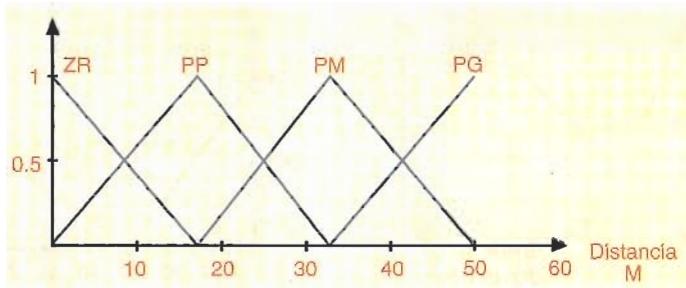


Figura 7. Representación de la distancia del vehículo

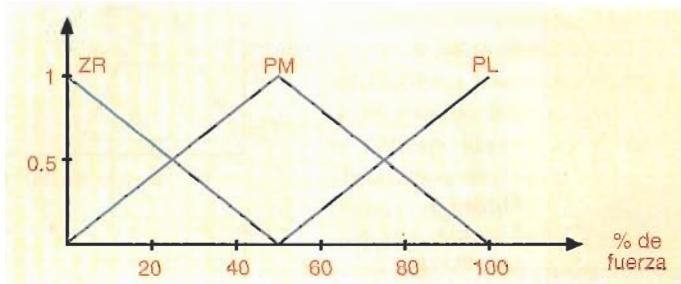


Figura 8. Representación de la fuerza necesaria

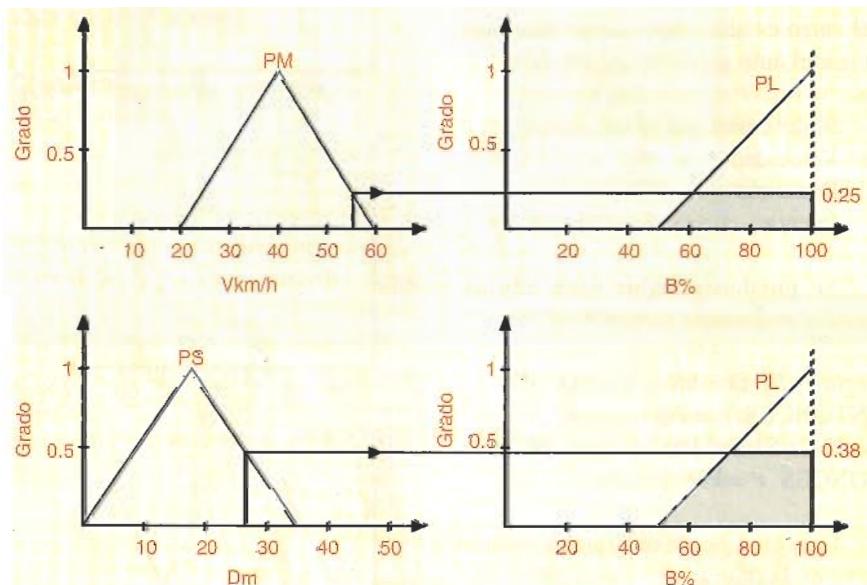


Figura 9. Representación de la regla 1

Para llevar esto a cabo, se toma un simple valor de cada regla. Si la regla usa AND (tal como lo hacen nuestras dos reglas) se toma el menor valor. Si la regla usa OR se toma el valor más alto.

De esta manera, la salida de la regla 1 es 0.25 y la salida de la regla 2 es 0.62, como se muestra en la figura 10. Las áreas de salida A1 y A2 bajo los valores mínimos se combinan. La salida hacia el mundo real es entonces el centro de gravedad de la combinación de las áreas.

La evaluación del proceso operando continuamente, es que trabajando con dos entradas: la información de la velocidad y los datos de la distancia, y sometiéndolos a un sistema difuso, producirán una fuerza sobre el motor segura y efectiva.

Aplicaciones industriales

En los últimos años, la lógica difusa ha probado su potencial al ser aplicada en procesos de automatización. En esta área en particular, los ingenieros han

descubierto su utilidad cuando en sus aplicaciones de control normales encontraron algunas deficiencias en el resultado final.

Para el control de eventos se utiliza normalmente la programación con lógica escalera (lenguaje basado en diagramas de alambrado o escalera y que corren en controladores lógicos programables), y en el caso de control continuo, normalmente se usan controladores PID.

Se observó que el proceso de control funcionó muy bien en condiciones de estabilidad, pero no tuvieron cuenta factores como la presencia de perturbaciones, variación en los parámetros del proceso y presencia de tiempos muertos.

La aplicación más valiosa del control difuso en la automatización industrial se ve en el diseño de controladores multivariados, en los cuales se recurre a la experiencia de los operarios para diseñar el control, evitándose el arduo camino de desarrollar un complejo modelo matemático de los procesos. En 1990, cuando estaba entrando al auge del control difuso en automatización industrial, la compañía alemana *Möller GmbH* y la compañía americana *Inform Software* crearon el *fuzzyPLC*, figura 11, basados en la observación de las necesidades de la integración de la lógica difusa en las técnicas de automatización industrial. Este dispositivo se caracteriza por su bajo costo, conversores de 12 bits, capacidad de extender periféricos y poder adquirir más de 100 señales, con una conexión para bus de campo basado en RS-485, la computación del control difuso se hace en un microcontrolador RISC de 16/32 bits, RAM interna de 256 KB y con la posibilidad de expandirla con tarjetas de memoria flash.

Algunas de las aplicaciones ya estudiadas y que han resultado exitosas en el mundo industrial, abarcan el control de grúas transportadoras, el control de fuego en plantas de incineración, control de dosificación en plantas de tratamiento de agua, control de temperatura en máquinas de moldeo de plásticos y control de conversión de energía eólica.

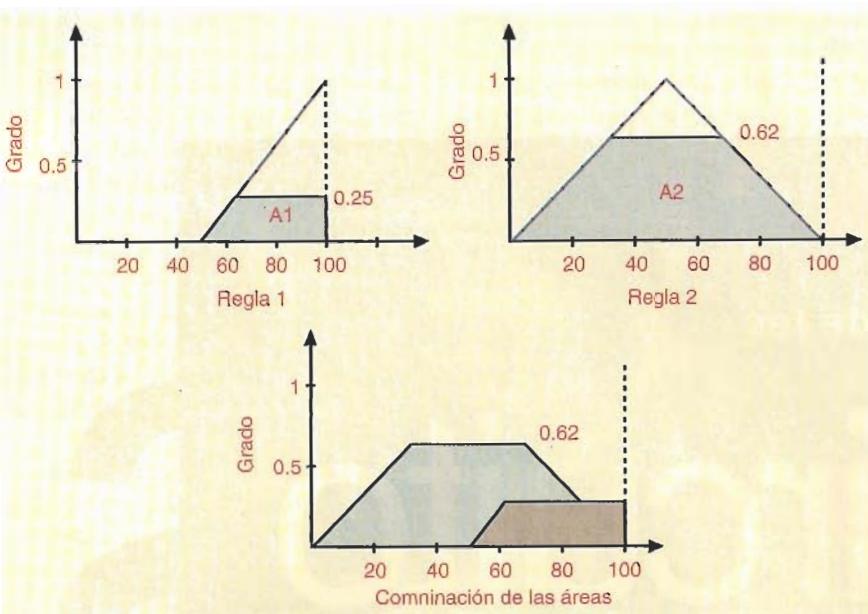


Figura 10. Proceso de de-fuzzificación

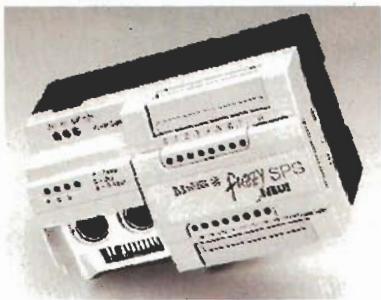


Figura 11. PLC difuso

Experimentación con control difuso

Típicamente las aplicaciones de control difuso se agrupan en dos categorías: La primera consiste en mejorar productos existentes, adicionándole propiedades o funciones que se pueden interpretar como inteligentes. La segunda consiste en desarrollar sistemas que utilicen sensores para responder continuamente a condiciones cambiantes de la entrada. La lógica difusa puede afrontar las alienalidades que se puedan presentar en el sistema. Analicemos ahora una sencilla aplicación que puede realizarse utilizando microcontroladores PIC, la cual apreciamos en la figura 12.

Se trata de controlar la altura de una bomba de aire que flota en el interior de un tubo debido a la acción de un ventilador. El panel de control facilita al usuario entrar la posición deseada de la bomba de aire por medio de un teclado. La en-

trada del teclado se visualiza en el módulo LCD. EL panel de control inicia un ciclo para calcular la altura actual y de la bomba. La altura deseada y la altura real se visualizan continuamente en el LCD. Esta información es transmitida al PC por medio de un enlace RS-232. El algoritmo difuso (que corre en el PC), calcula el ciclo útil apropiado de la señal DC para controlar el motor, y lo transmite al panel de control. Esto en realidad «emula» un entorno de trabajo real, en el cual la PC realiza un trabajo de control en tiempo real. El panel de control maneja el ciclo útil del motor DC basado en la información del PC. Este proceso solo puede ser detenido por el usuario.

Diseño difuso

La lógica difusa convierte entradas concretas a un marco de descripción lingüística. Luego evalúa la ley de control contenida en las reglas difusas y retorna el resultado final a una respuesta concreta. El primer paso en el diseño de un control difuso es la definición del sistema. Las únicas fuentes posibles de entrada al algoritmo de control difuso son el transductor difuso, el usuario y el voltaje del ventilador. La clave es decidir cuál de esas entradas son realmente las más importantes y cuáles no lo son tanto. Para eso, la combinación de variables y términos proporciona una descripción lingüística de lo que le está ocurriendo al sistema.

Cuando las variables lingüísticas se han definido, se deben definir los tipos de datos y sus valores. Los tipos de datos pueden definirse como enteros de 8 o 16 bits; con base en ese número de bits se asignan los valores finales que van a tener cada uno de los valores y toma un valor dentro de un rango digital. Ese valor de código puede estar entre 0 y 255 (para datos de 8 bits). Ω

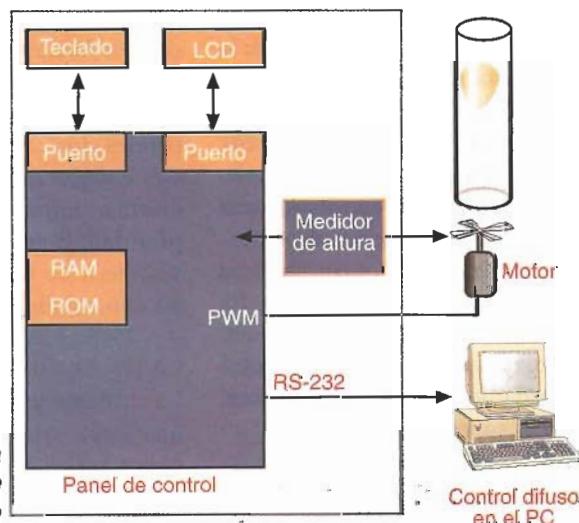


Figura 12. Sistema experimental de control difuso

Leyes Básicas de los Circuitos

A circuit diagram is integrated into the letter C. It shows a 9V battery connected in series with a 18Ω resistor (R1). This combination is in parallel with a 3V battery and a 6Ω resistor (R2). A voltmeter is connected across R1, showing 9V = V/R1. A current source is also shown.

JORGE E. HERNÁNDEZ M.

edher@telesat.com.co

Los nombres Kirchoff, Thévenin, Norton y superposición son familiares para cualquiera que haya estudiado los principios básicos de electricidad porque están relacionados con las leyes que gobiernan el comportamiento de los circuitos eléctricos. Sin embargo, a veces olvidamos que estas mismas leyes pueden ser también aplicadas con éxito al análisis de circuitos electrónicos.

La Ley de Ohm, suficientemente conocida por todos nosotros, es fundamental en el estudio de la electrónica. Con ella podemos calcular la corriente, el voltaje o la impedancia asociada a un componente simple, una sección de un circuito o un circuito completo, sin importar su complejidad o su naturaleza. Sin embargo, muchos circuitos que encontramos en nuestra experiencia corriente no pueden ser analizados con facilidad aplicando únicamente la Ley de Ohm. En estos casos, deben utilizarse otras técnicas sistemáticas de análisis, como las Leyes de Kirchoff y los teoremas de Thévenin, Norton y la superposición .

En este artículo examinaremos los métodos anteriores desde un punto de

vista cualitativo y cuantitativo, tomando como ejemplos circuitos DC resistentes relativamente sencillos y alimentados por fuentes de voltaje o de corriente no controladas. No obstante, tenga en cuenta que se trata de resultados generales, los cuales pueden ser aplicados igualmente a circuitos AC, DC o mixtos, de cualquier grado de complejidad y provistos de fuentes de voltaje o de corriente controladas o no controladas.

La ley de Ohm

La corriente y el voltaje asociados con una carga resistiva se relacionan mediante la Ley de Ohm, la cual establece que la corriente (I) es directamente proporcional al voltaje aplicado (V). La

constante de proporcionalidad es precisamente la resistencia (R) del elemento. Se asume que la temperatura y demás condiciones ambientales no cambian. Esto es:

$$I = V/R$$

Por tanto, si se aumenta o disminuye el voltaje aplicado a una resistencia, la corriente a través de la misma aumenta o disminuye en la misma proporción. Por ejemplo, si se aplican 12 voltios a una resistencia de 100 ohmios, la corriente resultante a través de la misma es $12V/100 \Omega = 0.12 A$. Si el voltaje aumenta o disminuye en una cantidad N , la corriente también aumenta o disminuye en esa misma cantidad.

Las leyes de Kirchoff

El comportamiento cuantitativo de las corrientes y los voltajes en un circuito, está regido por dos reglas universales muy sencillas conocidas como las Leyes de Kirchoff, denominadas así en honor del físico alemán Gustav Robert Kirchoff (1824-1887), quien las postuló en 1847. Las Leyes de Kirchoff proporcionan un método sistemático de análisis de todo tipo de circuitos y son la base de los programas de simulación de circuitos por computadora.

La primera de estas leyes está relacionada con el comportamiento de las corrientes en los nodos y la segunda con el comportamiento de los voltajes alrededor de las mallas.

La Ley de Kirchoff de las corrientes (LKC) establece que la suma algebraica de las corrientes que entran a un nodo en un instante dado, es siempre igual a la suma algebraica de las corrientes que salen en ese instante. En la figura 1 se ilustra este concepto, aplicado a un divisor de corriente resistivo. Este caso, de acuerdo a la LKC, la corriente total que fluye en el punto X ($I_T=6$ A) debe ser igual a la suma de las dos corrientes I_1 (4 A) e I_2 (2 A). Efectivamente, $6 \text{ A} = 4 \text{ A} + 2 \text{ A}$. Un análisis similar puede ser realizado en al punto Y

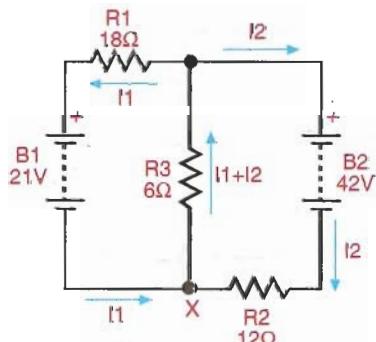


Figura 1. Ilustración de la primera Ley de Kirchoff o Ley de las Corrientes

La Ley de Kirchoff de los voltajes establece que la suma algebraica de las caídas de voltaje alrededor de un

circuito debe ser igual a la suma algebraica de las fuerzas electromotrices o fuentes alrededor de la misma. En la figura 2 se ilustra este concepto, aplicado a un divisor de voltaje resistivo. En este caso, de acuerdo a la LKV, el voltaje total aplicado (12V) debe ser igual a la suma de las caídas de voltaje sobre R_1 y R_2 . (9V y 3V, respectivamente). En efecto, $12V = 9V + 3V$. En las figuras 2 (B) y 2 (C) se ha agregado una segunda fuente de voltaje de 12V. Ahora, el voltaje total aplicado es 24V y el mismo debe repartirse entre R_1 (18V) y R_2 (6V).

Una situación relativamente más complicada se muestra en la figura 3.

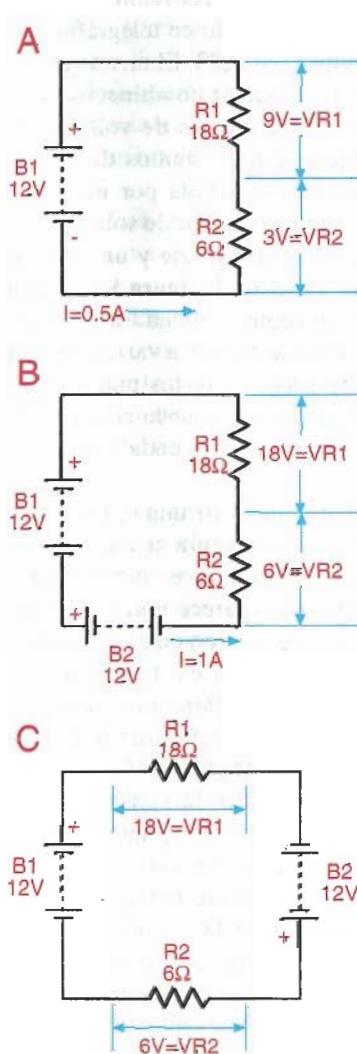


Figura 2. Ilustración de la segunda Ley de Kirchoff o Ley de los Voltajes

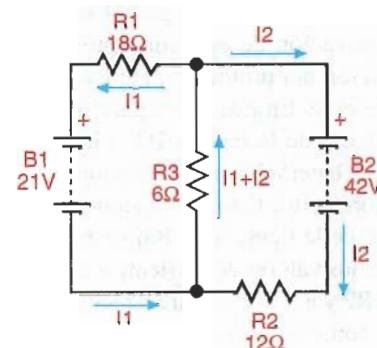


Figura 3. Otro ejemplo de aplicación de la segunda ley de Kirchoff

En este caso, la idea es calcular la corriente que fluye a través de cada una de las tres resistencias. Debido a que las fuentes de voltaje están completamente separadas, se requiere aplicar la segunda Ley de Kirchoff y tener un poco de habilidad en la solución de ecuaciones simultáneas. Inicialmente, note que la corriente I_1 fluye a través de B_1 , R_3 y R_1 , mientras que la corriente I_2 lo hace a través de B_2 , R_2 y R_3 . Asimismo, la corriente a través de R_3 es la combinación de I_1 e I_2 . La aplicación de la LKV y algunas técnicas algebraicas simples a ambas mallas conduce al siguiente sistema de ecuaciones simultáneas:

$$\begin{aligned}-9I_1 - 6I_2 &= -21 \\ -6I_1 - 18I_2 &= -42\end{aligned}$$

Resolviendo este sistema por cualquiera de los métodos algebraicos disponibles (sustitución, determinantes, etc.), se encuentra que $I_1=1$ A e $I_2=2$ A. Por tanto, la corriente a través de R_3 es $I_1+I_2 = 3$ A. En caso de obtener un valor de corriente negativo, significa simplemente que la dirección asumida al comienzo del análisis del problema es contraria a la dirección real.

El Teorema de la superposición

Las Leyes de Kirchoff permiten analizar los efectos de varias fuentes de voltaje o de corriente actuando simultáneamente sobre un circuito. Esto puede conducir a ecuaciones relativamente complejas. Otro enfoque es utilizar el teorema de la superposición, el cual nos permite examinar el efecto de cada

fuent por separado y luego combinar los resultados. En la figura 4 se ilustra la aplicación de este concepto para la solución del problema planteado anteriormente. En este caso, para observar el efecto de la batería B1 sola, retiramos la batería B2, sustituyéndola por un cortocircuito. Esto nos conduce al circuito de la figura 4A. Bajo esta condición, los valores de corriente a través de R1, R2 y R3, con las direcciones indicadas, son, en su orden, 3 A, 1 A y 2 A.

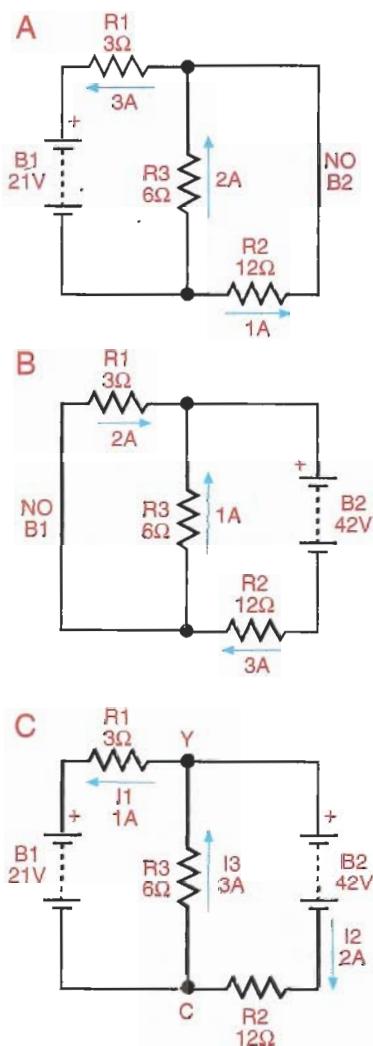


Figura 4. Ilustración del Teorema de la superposición.

A continuación, realizamos el mismo procedimiento con la fuente de voltaje B2, sustituyendo B1 por un cortocircuito, como se ilustra en la figura 4B. Ahora, los valores de corriente asociados con R1, R2 y R3, en las direccio-

nes indicadas, son 1 A, 2 A y 3 A. El paso final es combinar ambas respuestas, siendo especialmente cuidadoso en la observación de las direcciones de las corrientes. Bajo esta condición, se encuentra por superposición que la corriente a través de R1 (I1) es 3 A - 2 A = 1 A, la corriente a través de R2 (I2) es 3 A - 1 A = 2 A y la corriente a través de R3 (I3) es 2 A + 1 A = 3 A. En la figura 4C se muestran estos resultados, los cuales coinciden plenamente con los obtenidos por Leyes de Kirchoff.

El Teorema de Thévenin

Otra herramienta que puede simplificar el análisis de algunos circuitos complejos es el Teorema de Thévenin, atribuido a M. L. Thévenin, un ingeniero francés experto en telegrafía, quien lo enunció en 1883. El mismo establece que cualquier combinación de resistencias y fuentes de voltaje, entre cualquier par de puntos de una red, puede ser sustituida por un circuito equivalente constituido solamente por una fuente de voltaje y una resistencia en serie. En la figura 5A se ilustra este concepto, aplicado a un divisor de voltaje sometido a varias condiciones de carga entre los puntos X e Y. Se desea determinar la corriente y el voltaje asociadas a cada una.

Para convertir una red a su equivalente de Thévenin se requieren dos pasos. El primero es determinar el voltaje que aparece entre los puntos bajo consideración en condiciones de circuito abierto, es decir sin carga. Este voltaje se denomina Voltaje de Thévenin (VTH), figura 5B. En nuestro caso, $VTH=13.33$ V. El segundo paso es calcular la resistencia equivalente entre dichos puntos con todas las fuentes suprimidas, es decir las fuentes de voltaje reemplazadas por cortocircuitos y las fuentes de corriente reemplazadas por circuitos abiertos. Esta resistencia se denomina Resistencia de Thévenin (RTH), figura 5C. En nuestro caso, $RTH=40$ Ohm.

Por tanto, el circuito equivalente de Thévenin es como el mostrado en la figura 5D. Una vez obtenido este modelo, se puede utilizar para deter-

minar la corriente y el voltaje asociados con cualquier carga conectada entre los puntos X e Y. Un mayor beneficio se obtiene cuando el Teorema de Thévenin se aplica al análisis de circuitos más complejos, como se indica en la figura 6. En este caso, se asume que R3 es una carga que puede adoptar muchos valores diferentes y se desea calcular la corriente asociada a cada uno.

Puesto que R3 es la carga, debemos removerla del circuito y calcular el equivalente de Thévenin visto por la misma (puntos X e Y). Para continuar, determinamos la resistencia

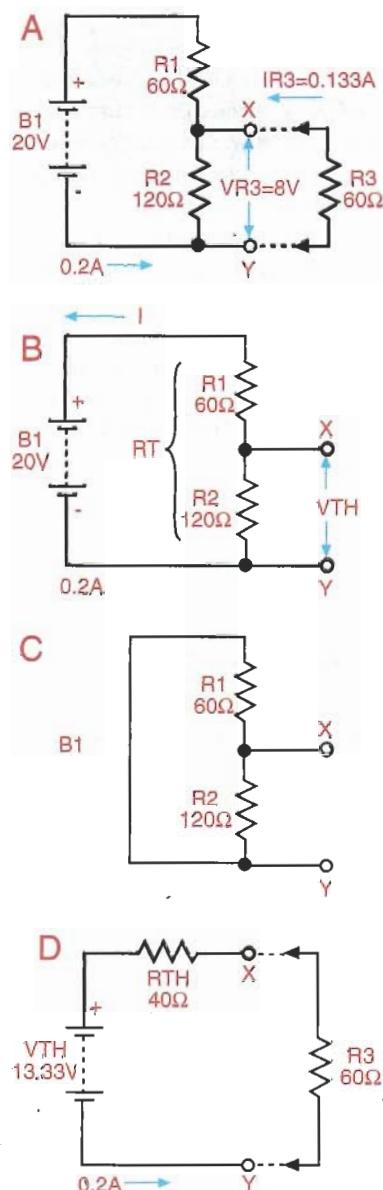


Figura 5. Ilustración del Teorema de Thévenin

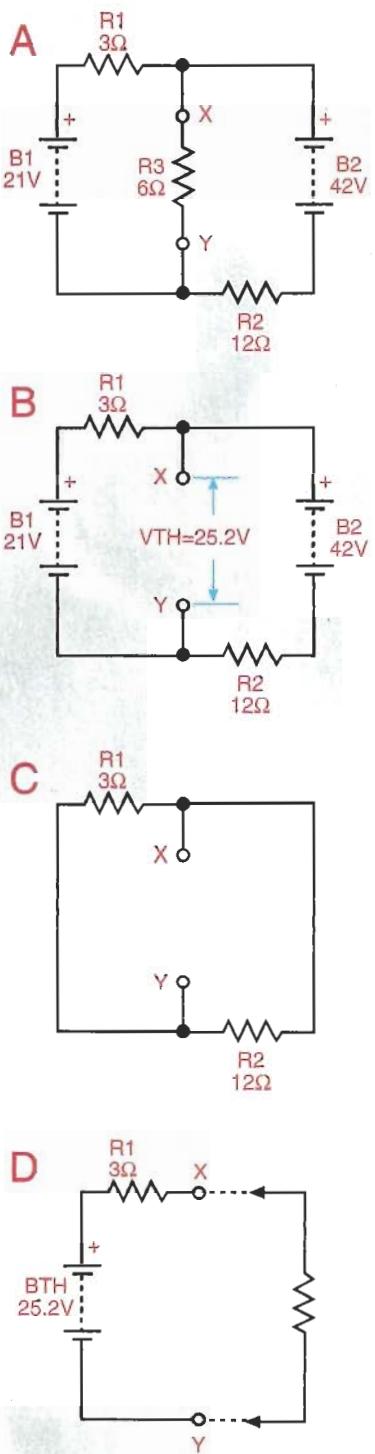


Figura 6. Aplicación del teorema de Thévenin al análisis de un circuito complejo

equivalente de Thévenin (R_{TH}) sustituyendo las fuentes por cortocircuitos, como se indica en la figura 6C. Puesto que R_1 y R_2 aparecen en paralelo, se encuentra que $R_{TH}=2.4\Omega$.

Ohm. El circuito equivalente de Thévenin resultante entre los puntos X e Y es, entonces, como se muestra en la figura 6D. Así resulta mucho más fácil analizar el efecto de cualquier carga. Por ejemplo, si $R_3=6\Omega$, la corriente de carga sería de 3 A, como era de esperar.

El Teorema de Norton

Una última herramienta de análisis de circuitos es el Teorema de Norton, formulado por E. L. Norton, un científico de los Bell Telephone Laboratories. El mismo establece que cualquier combinación de resistencias y fuentes de voltaje, entre cualquier par de puntos de una red, puede ser sustituída por un circuito equivalente constituido solamente por una fuente de corriente y una resistencia en paralelo. En la figura 7 se ilustra este concepto, aplicado al mismo circuito utilizado al exemplificar el Teorema de Thévenin.

Nuevamente, para convertir una red a su equivalente de Norton, se requieren dos pasos. El primero es determinar la corriente que atraviesa los puntos bajo consideración en condiciones de cortocircuito, es decir sin carga. Esta corriente se denomina corriente de Thévenin (IN), figura 7A.

En nuestro caso, $IN=10.5\text{ A}$. El segundo paso es calcular la resistencia equivalente entre dichos puntos con todas las fuentes suprimidas, es decir las fuentes de voltaje reemplazadas por cortocircuitos y las fuentes de corriente reemplazadas por circuitos abiertos. Esta resistencia se denomina Resistencia de Norton (R_N), figura 7B. En nuestro caso, $R_N=2.4\Omega$.

Con este modelo simplificado, resulta muy fácil determinar la corriente a través de cualquier carga conectada entre los puntos X e Y, puesto que todo se reduce a un simple problema de división de corriente. Por ejemplo, si la carga R_3 es de 6 ohm, la corriente a través de la misma es simplemente $10.5\text{ A} \times (2.4/8.4) = 3\text{ A}$. Esta corriente, a su vez, produce una caída de voltaje de $3\text{ A} \times 6\Omega = 18\text{ V}$.

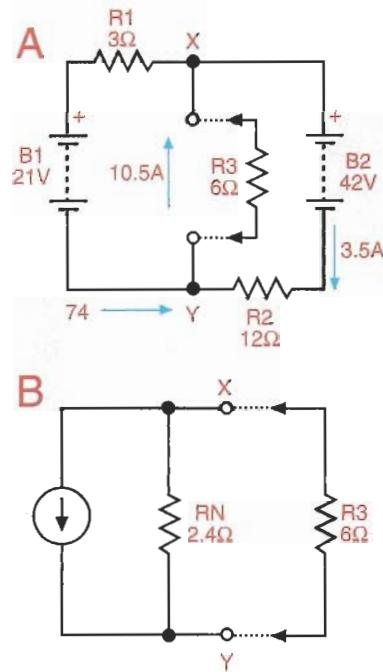


Figura 7. Ilustración del Teorema de Norton

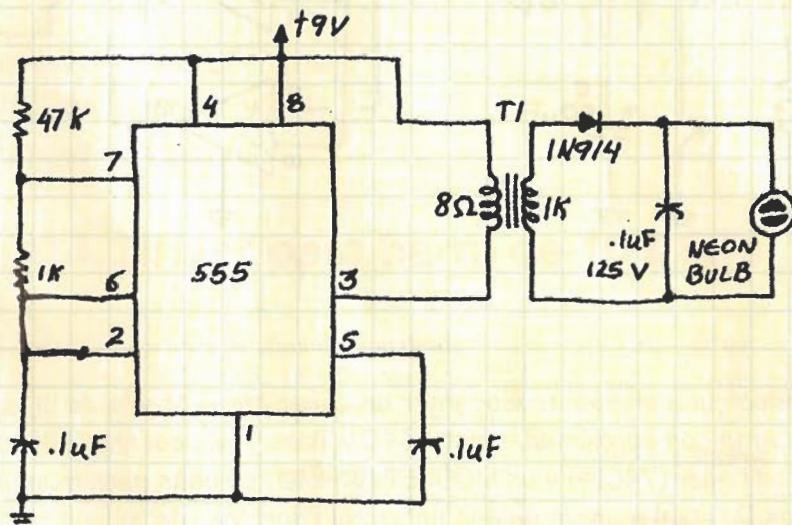
De los razonamientos anteriores resulta claro que una de las principales aplicaciones de los Teoremas de Thévenin y Norton es la sustitución de una gran parte de una red, a menudo complicada y sin interés, por un circuito equivalente muy simple. Este nuevo circuito, más sencillo, nos permite calcular rápidamente la corriente, la tensión y la potencia que el circuito original es capaz de entregar a la carga. También sirve de ayuda para elegir la resistencia o impedancia de carga.

Por ejemplo, en un amplificador de potencia con transistores, los equivalentes de Thévenin y Norton nos permiten determinar la potencia máxima que puede extraerse del amplificador y el tipo de carga necesario para obtener la máxima transferencia de potencia o para obtener la amplificación práctica máxima de corriente o de tensión. De este modo, si el objetivo es conseguir la máxima transferencia de potencia, el valor de la resistencia de carga deberá elegirse igual al valor de la resistencia de Thévenin o de Norton.

Cuaderno de experimentos

En esta sección se presentan circuitos prácticos, útiles y sencillos, con componentes fáciles de conseguir, que usted mismo puede ensamblar utilizando un protoboard, un circuito impreso universal o cualquier otro método de construcción. Todos los montajes han sido suficientemente comprobados, pero usted puede experimentar con ellos y adecuarlos a sus necesidades particulares. Una vez haya conseguido el grado de funcionamiento deseado, fabrique un circuito impreso para su proyecto e instálelo en forma definitiva en un chasis metálico o plástico.

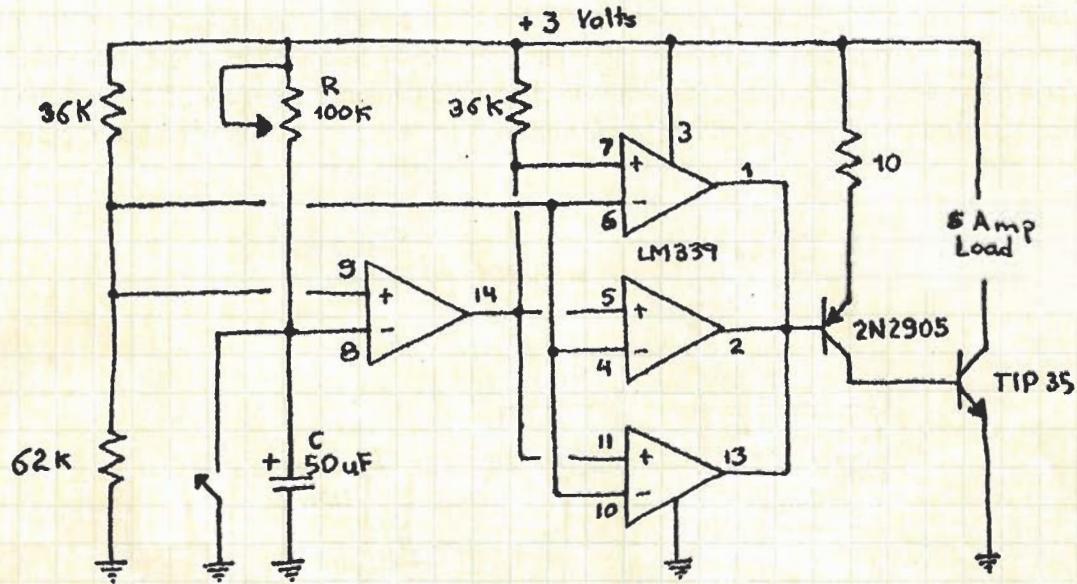
Fuente para tubos de neón



Las lámparas de neón tienen requisitos de corriente muy modestos, así que es posible excitarlas con una fuente AC o DC de bajo voltaje. Sin embargo, no todas trabajan bien bajo estas condiciones. Una mejor alternativa es utilizar un circuito como el de la figura, el cual utiliza un 555 como oscilador a una tasa de 300 Hz. El voltaje obtenido en el secundario del transformador es del orden de 90V. Este último puede ser, por ejemplo, un transformador del mismo tipo de los utilizados en la etapa de salida de muchos amplificadores de audio.

Tenga en cuenta que el circuito, aunque se alimenta a partir de 9V, produce un voltaje de salida relativamente alto, capaz de ocasionarle una descarga. Por tanto, trátelo con el mismo respeto con que lo haría ante la red pública de AC. Si el tubo de neón parpadea o no ilumina, reduzca la frecuencia de oscilación del 555 o utilice un transformador con una mayor relación de transformación.

Circuito de ignición basado en capacitor de descarga

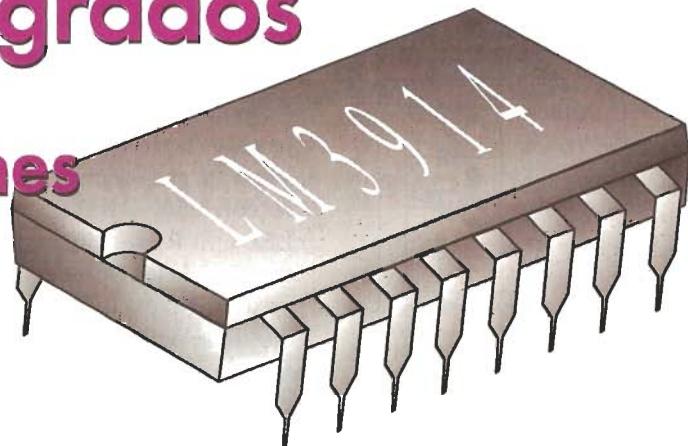


Este circuito produce una chispa al descargar un capacitor a través de la bobina primaria. Un capacitor de $2\mu F$ se carga con aproximadamente 340 voltios y se descarga bajo el control de un SCR. Un oscilador Schmitt trigger (74C14) y un MOSFET (IRF510) se usan para manejar la lado de bajo voltaje del transformador. De igual manera se usa un arreglo doblador de voltaje en el lado de alto voltaje del transformador para incrementar el voltaje del capacitor hasta 340 V.

Un oscilador Schmitt trigger se usa para disparar el SCR cerca de 4 veces por segundo. La fuente de voltaje es cerrada durante el tiempo de descarga, de tal manera que el SCR detendrá su conducción y retornará al estado de bloqueo.

El diodo conectado desde el 3904 al pin 9 del 74C14 causando que el oscilador de la fuente se detenga durante el tiempo de descarga. Tenga mucha precaución al armar este circuito. Un método alterno de conectar la bobina es llevar a tierra el terminal negativo y reubicar el capacitor entre el cátodo del diodo rectificador y el terminal positivo de la bobina. El SCR se ubica entre la tierra y el lado de 340 V del capacitor. De esta manera se reduce el riesgo para el usuario y es la configuración usual en aplicaciones automotrices.

Circuitos integrados & Aplicaciones



En esta sección mostramos diversos circuitos integrados de uso común con algunos esquemas de sus principales aplicaciones.

LM3914. Driver para barra de 10 LEDs

Este circuito integrado controla hasta 10 LEDs de acuerdo al nivel de la señal que le sea aplicada. Es muy comercial y puede ser utilizado para monitorear la amplitud de una señal de audio, del voltaje de una fuente, de la corriente en un circuito, etc.

Posee pines para indicarle la referencia del voltaje con la cual toma los valores de comparación para encender o no los LEDs de sus salidas. Por ejemplo, si el valor de la referencia de voltaje es de 5 voltios y se aplican 2.5 voltios a la entrada de señal, el circuito encenderá el LED número 5 ya que coincide con la mitad del rango con el cual se ha diseñado.

Posee dos formas básicas de funcionamiento: en forma de *Barra* y en forma de *Punto*. Para el ejemplo de los 2.5 voltios, en modo de barra, se encienden los 5 primeros LEDs formando precisamente una pequeña barra; pero en modo de punto solamente se encenderá el LED número 5 y los demás permanecerán apagados. Una de sus grandes ventajas es que funciona con

voltajes de alimentación desde 3 VDC hasta 18 VDC, un rango amplio de operación.

Observe en la figura de la derecha la distribución de pines del circuito integrado LM3914 y más abajo el esquema interno del mismo. Como vemos, está compuesto de una serie de comparadores, cada uno con un voltaje de referencia diferente y consecutivo.

La siguiente es la función de algunos pines:

VLO (VLOW). Voltaje de referencia bajo. Generalmente es llevado a tierra.

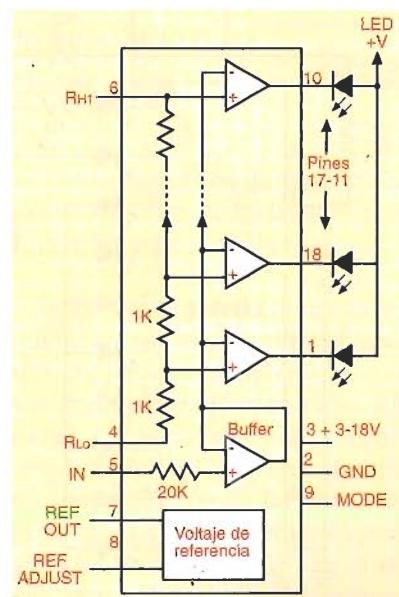
VHI (VHIGH). Voltaje de referencia alto. Es el voltaje máximo que tendría la entrada de señal.

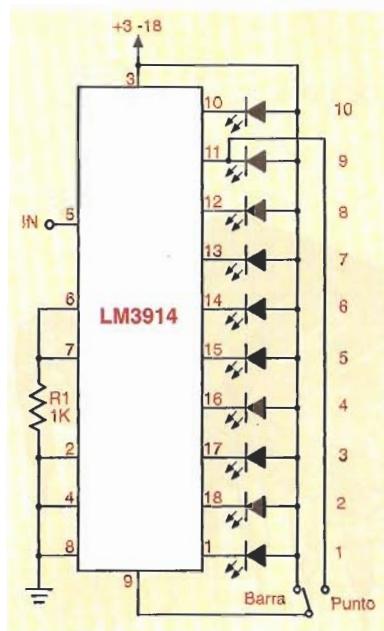
IN. Es la entrada de señal. Su voltaje debe estar entre VLO y VHI.

REF OUT. Salida de voltaje de referencia. Es un voltaje de referencia que se genera dentro del circuito integrado. Puede ser utilizado para VHI.

REF ADJ. Ajuste del voltaje de referencia.

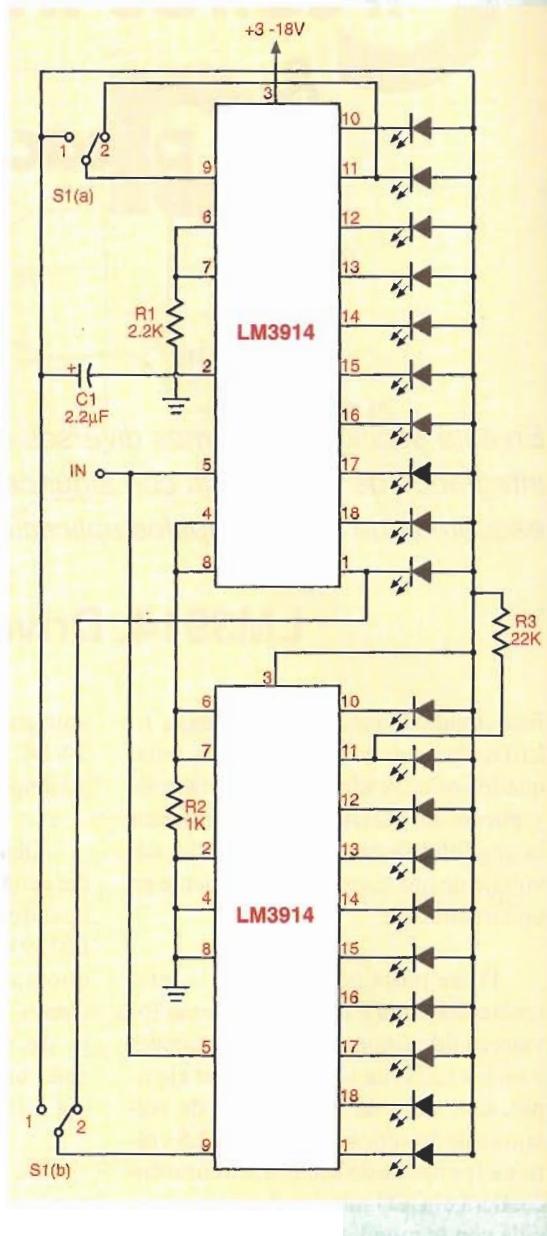
MODE. Sirve para indicar si el circuito funcionará en modo de *Barra* o en modo de *Punto*.





Display de 10 LEDs

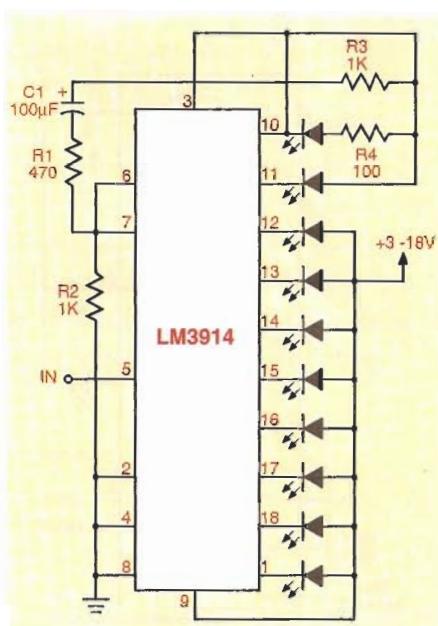
Este es el circuito básico de conexiones. El rango de la señal de entrada es de 0 a 1.3 voltios, es decir que cada LED encendido corresponde a 0.13 voltios de dicha señal de entrada. La intensidad de los LEDs depende de la resistencia R1, cuya corriente es la décima parte de la corriente a través de los LEDs. Si R1 se duplica, la intensidad de los LEDs disminuye a la mitad. Con el selector que va al pin 9, podemos configurar el circuito en modo de barra o en modo de punto.



Barra de 20 LEDs

En el circuito normal, la resolución alcanzada es de 10 puntos.

Si utilizamos dos circuitos integrados conectados en cascada, podemos duplicar la resolución mostrando 20 puntos en total. La señal de entrada es llevada simultáneamente a los integrados pero los voltajes de referencia son conectados en serie y con el VHI del integrado inferior se controla el VLO del superior. El rango del voltaje de entrada se duplica, en este caso a 2.6 voltios. De igual manera se puede configurar en los dos modos de operación por medio del selector S1. Es posible conectar más integrados en cascada.

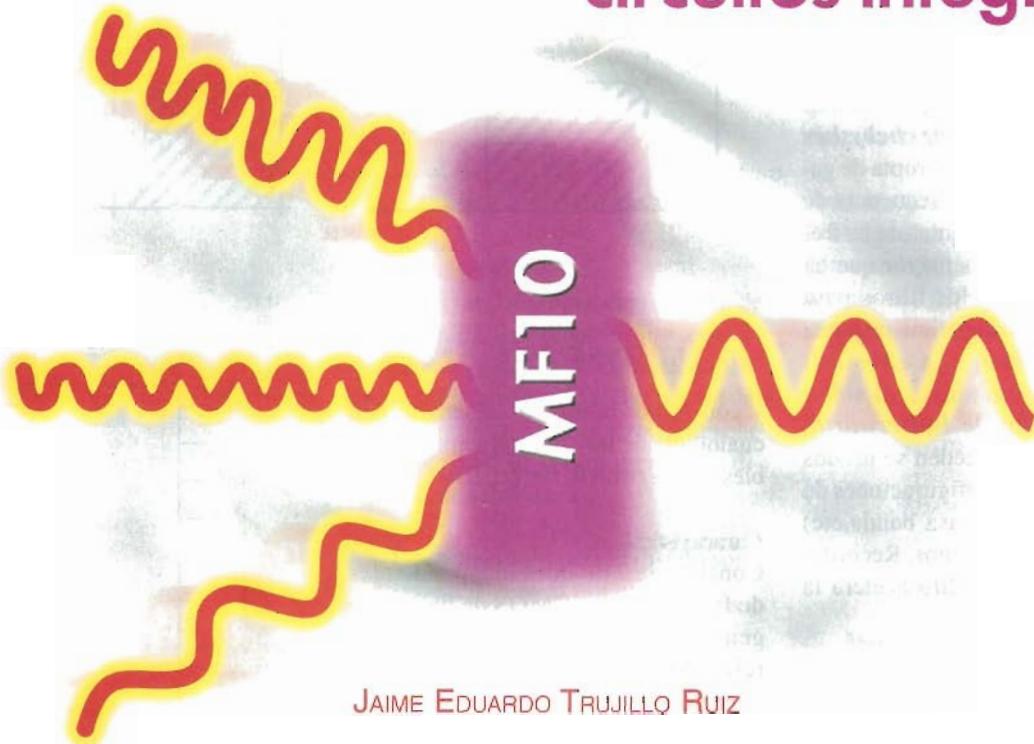


Barra intermitente

Con este circuito, cuando el nivel de la señal de entrada corresponde al último LED, inmediatamente los demás LEDs cambian su funcionamiento en forma intermitente. Esto indicaría que la señal de prueba a superado el nivel normal. Si se incrementa el valor de C1 disminuye la frecuencia del centelleo.

Diseño

de filtros activos con circuitos integrados



JAIME EDUARDO TRUJILLO RUIZ

Para el filtrado de señales, en la mayoría de los casos se necesita versatilidad, confiabilidad y facilidad en la implementación de los circuitos, por tal razón se hace casi necesario recurrir a un circuito integrado para filtrado como los de la familia MF de National Semiconductor.

Un filtro es un circuito diseñado para dejar pasar una banda de frecuencias específica; es decir, el circuito debe ser capaz de atenuar todas las señales que no estén en el rango de frecuencias especificado en el diseño del mismo. Los circuitos de filtrado pueden ser activos o pasivos, los filtros pasivos solo están configurados con elementos pasivos (resistencias, condensadores, inductancias) y los filtros activos son combinaciones de transistores, amplificadores operacionales o circuitos integrados y elementos pasivos.

Existen cuatro tipos básicos de filtros : pasa bajo, pasa alto, pasa banda y rechaza banda.

Un filtro pasa bajo es un circuito que tiene un voltaje de salida constante desde cd hasta una **frecuencia de corte (F_c)**, un pasa alto atenúa el voltaje de salida para todas las frecuencias abajo de F_c , un filtro pasa banda permite pasar sólo una banda de frecuencias mientras atenúa todas las demás y un rechaza banda se comporta exactamente de forma opuesta, es decir, rechaza

una banda específica de frecuencias mientras deja pasar todas las que se encuentren fuera de ella, figura 1.

En la función de transferencia general de los filtros se pueden obtener polos conjugados complejos que no pueden ser realizados con redes RC pasivas; por lo tanto se debe recurrir al uso de redes RLC. Para frecuencias altas, la realización de las inductancias no causa muchos problemas pero a frecuencias bajas sí, debido al tamaño requerido. Por esto lo mejor es

hacer uso de amplificadores operacionales y redes RC para la configuración del filtro deseado o encontrando un chip que reúna tales características en su interior.

Para la obtención de los coeficientes de la función de transferencia general existen varios métodos :

Los filtros paso bajo de **Butterworth** tienen una respuesta horizontal o plana y descenden bruscamente antes de la frecuencia de corte; su respuesta muestra un considerable sobreimpulso que aumenta en los filtros de orden más alto. Los filtros pasa bajo de **chebyshev** tienen una caída aún más abrupta de ganancia por encima de la frecuencia de corte y el sobreimpulso antes de la frecuencia de corte que es mayor que en los de **Butterworth** y los filtros pasa bajos de **Bessel** dan la óptima respuesta de onda cuadrada.

Los polinomios de **Butterworth**, **Bessel** y **Chebyshev** pueden ser usados en cualquiera de las configuraciones de los filtros (pasa alto, pasa banda,etc) para el diseño de los mismos. Recorremos que el orden del filtro acelera la respuesta del mismo.

Teniendo en cuenta lo anterior, veamos qué posibilidades tenemos con los circuitos integrados de la familia MF para el diseño de circuitos de filtrado.

Con esta familia podemos tener posibilidades con un chip para generar cualquier tipo de filtro y cualquier tipo de configuración como el MF10 (dual) o el MF5 (sencillo); el MF10 equivale a dos MF5. Por ejemplo, podemos tener un filtro pasa bajo de Butterworth para generar funciones de orden elevado como de orden 12 con únicamente 2 chips y unas resistencias; con el MF6 y un filtro pasa banda como el MF8.

Conozcamos los principales circuitos integrados de la familia MF:

MF10

El MF10 es un filtro activo programable con cualquiera de los tipos de filtros existentes (pasa bajo, pasa alto, pasa banda, pasa todo y rechaza banda)

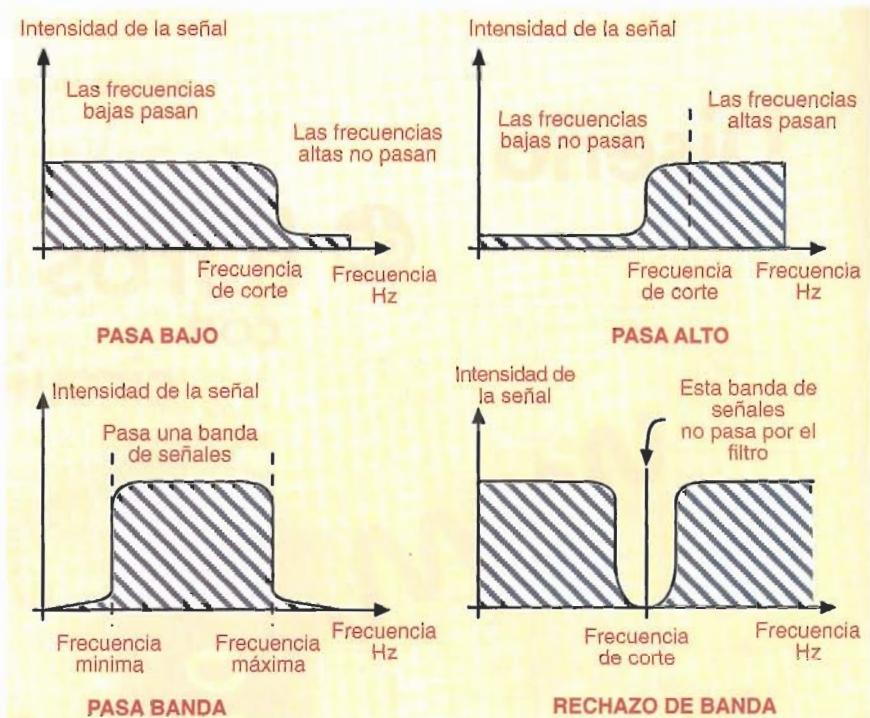


Figura 1. Tipos de filtros

y con la posibilidad de diseñarlo con cualquiera de las configuraciones posibles (**Butterworth**, **Bessel**, **Chebyshev**).

Características

Consiste de 2 bloques independientes de filtros activos CMOS de propósito general; cada bloque, a través de un reloj externo, y de 3 a 4 resistores, puede producir varias funciones de segundo orden. Cada bloque tiene 3 pines de salida; una de las salidas puede ser configurada como pasa todo, pasa alto o rechaza banda y las dos salidas restantes configurables como pasa bajo y pasa banda.

Cada bloque del MF10 es una función de segundo orden; por lo tanto, si utilizo los dos bloques en cascada puedo obtener funciones de cuarto orden. Para obtener funciones de orden mayor a 4 puedo conectar varios MF10 en cascada y así subir el orden del filtro en múltiplos de 2; la frecuencia de corte para el filtrado depende directamente de la frecuencia del reloj externo y ésta puede ser de hasta 200 kHz.

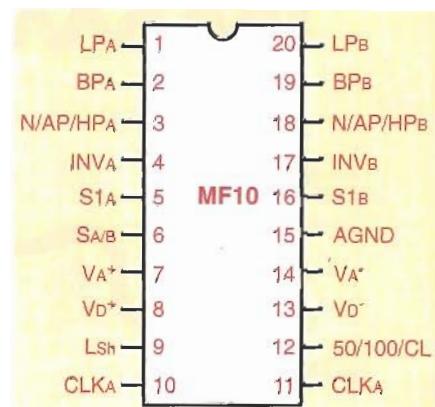


Figura 2. Diagrama de pines del MF10

Descripción de los pines

Ver pines en la figura 2:

- LP (1,20)** : Salida de función pasa bajo de segundo orden.
- BP (2,19)** : Salida de función pasa banda de segundo orden.
- N/AP/HP (3,18)** : Salida de función rechaza banda / pasa todo/ pasa alto de segundo orden.
- INV (4,17)** : Entrada inversora de cada filtro.
- S1 (5,16)** : Pin para señales de entrada usado solo para la configuración pasa todo.

Orden	Etapa	a ₀	b ₁	f _{c1} / f _{c0}	Q _i
4	1	1.8478	1	0.719	0.54
4	2	0.7654	1	1.390	1.31

Figura 3. Coeficientes para los polinomios de Butterworth

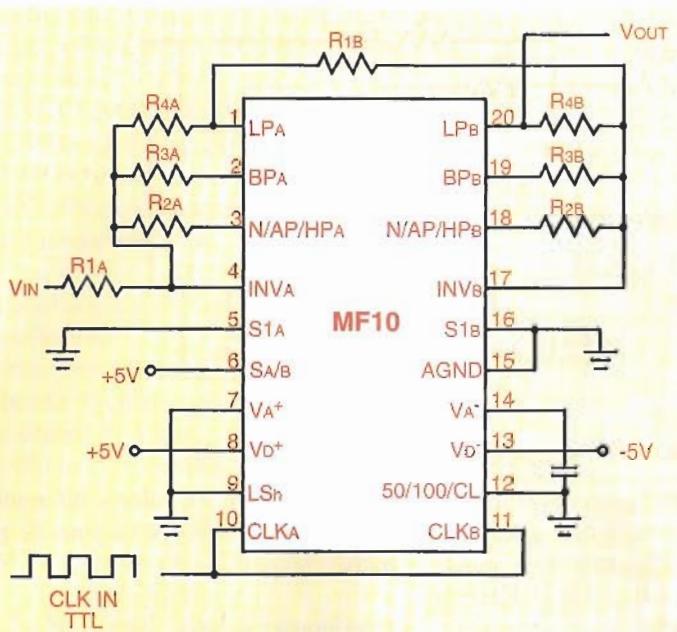


Figura 4.
Diseño de filtro
pasa bajo con
MF10

- S (a/b) (6)**: Para niveles lógicos CMOS se conecta este pin a -5V y para niveles TTL a tierra.
- V_{A+} (7)**: Alimentación análoga positiva.
- V_{D+} (8)**: Alimentación digital positiva.
- V_{A-} (14)**: Alimentación análoga negativa.
- V_{D-} (13)**: Alimentación digital negativa.
- CLKA (10), CLKB (11)**: Entradas de reloj.
- 50/100/CL(12)**: Configura la frecuencia de la señal del reloj 50 o 100 veces mayor que la frecuencia de corte.
- AGND(15)**: Tierra análoga.

Diseño con el MF10

El siguiente es un procedimiento de di-

seño para un filtro pasa bajo de 4Hz de cuarto orden de butterworth. Como la frecuencia central o de corte es dependiente directamente de la frecuencia del reloj, para el procedimiento de diseño se escoge 50 o 100 veces menor que dicha frecuencia de reloj.

Las ecuaciones para el diseño son :

- Frecuencia central por etapa :

$$f_q = \frac{f_{CLK}}{100} * \sqrt{\frac{R_2}{R_4}}$$

- Frecuencia del reloj :

$$f_{CLK} = 1100 * f_c$$

- Frecuencia central o frecuencia de corte :

$$f_0$$

- Factor de calidad del filtro :

$$Q = \sqrt{\frac{R_2}{R_4}} * \frac{R_3}{R_2}$$

- Ganancia del filtro paso bajo:

$$H_{BYP} = \frac{-R_4}{R_1}$$

Para el procedimiento de diseño se escogió el método de los polinomios de Butterworth, por esto se hace referencia a las tablas en el orden y etapas de interés.

Para la configuración de cuarto orden, ya tenemos la frecuencia del reloj que sería de 400 Hz debido a que la frecuencia de corte es de 4 Hz.

Primera Etapa: Se escogen R1A y R4A iguales para que la ganancia del filtro sea unitaria. Debido a que la ganancia del filtro pasa bajo invierte la señal, la configuración de cuarto orden evita este problema debido a la doble inversión de señal.

$$R_{1A} R_{4A} = 10 KW$$

$$R_{2A} = \frac{(f_{cl})^2 * 100^2}{(f_{CLK})^2}$$

ABCDE	50 : 1 mode		100 : 1 mode	
	F _{CLK} /F _o	Q	F _{CLK} /F _o	Q
10000	43.7	0.45	94.0	0.47
11000	45.8	0.71	95.8	0.73
01000	46.8	0.96	96.8	0.98
10100	48.4	2.0	98.4	2.0
00100	48.7	2.5	98.7	2.5
01100	48.9	3.0	98.9	3.0
11100	49.2	4.0	99.2	4.0
01010	49.3	5.0	99.3	5.0
10010	49.4	5.7	99.4	5.7
00010	49.5	7.6	99.5	7.6
11110	49.6	8.5	99.6	8.5
00110	49.6	10.6	99.6	10.6
11001	49.6	11.7	99.6	11.7
11010	49.7	12.5	99.7	12.5
11101	49.7	13.6	99.7	13.6
01001	49.7	14.7	99.7	14.7
10011	49.7	15.8	99.7	15.8
10101	49.7	16.5	99.7	16.5
01110	49.7	17	99.7	17
10001	49.8	19	99.8	19
10111	49.8	22	99.8	22
11011	49.8	27	99.8	27
11111	49.8	30	99.8	30
00101	49.8	33	99.8	33
01011	49.8	40	99.8	40
00111	49.8	44	99.8	44
00001	49.9	57	99.9	57
01101	49.9	68	99.9	68
00011	49.9	79	99.9	79
01111	49.9	90	99.9	90

Figura 6. Tabla para conversión lógica Q

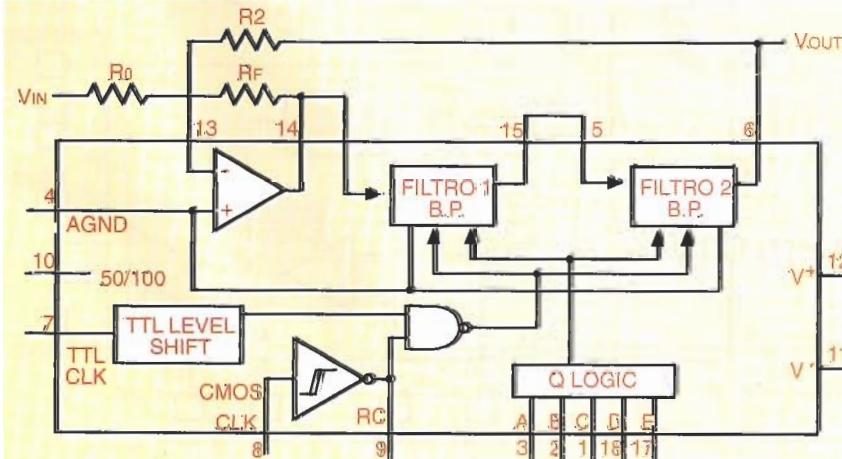


Figura 5. Diagrama de bloques del MF8

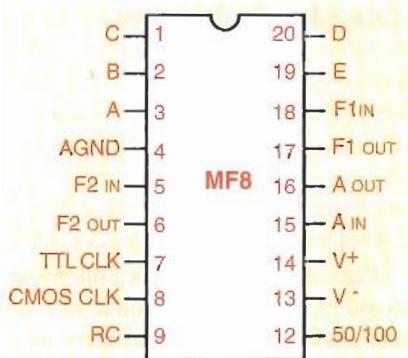


Figura 7. Diagrama de pines del MF8

Para hallar R2A necesitamos fcl que se halla de la razón de fci/fo de la tabla para la etapa 1:

$$f_{c1} = 0.719 * f_c$$

$$R_{2A} = \frac{288^2 * 100^2 * 10\text{K}\Omega}{400^2}$$

De la ecuación del factor de calidad despejamos R3A:

$$R_{3A} = Q_1 \sqrt{R_{4A} * R_{2A}}$$

De la tabla 1 se escoge Q para la etapa 1.

$$R_{3A} = 3.9\text{K}\Omega$$

Segunda Etapa: Se escoge R1B y R4B iguales a $15\text{K}\Omega$ y se hace el procedimiento anterior teniendo en cuenta los valores de la tabla para la segunda etapa. El circuito diseñado se muestra en la figura 4.

El reloj se puede diseñar con un 555 para un tren de pulsos de 400 Hz (100 veces mayor que la frecuencia de corte).

Estos circuitos son programados con una señal de reloj, dicha señal puede ser generada por niveles lógicos TTL o CMOS.

MF8

El MF8 es un filtro pasa banda programable que depende de un reloj que controla la frecuencia central del filtro y de una lógica Q que controla el coeficiente del polinomio definido y el ancho de banda de funcionamiento.

Características

El MF8 consta de dos bloques de filtros pasa banda de segundo orden que funcionan tanto en cascada como sepa-

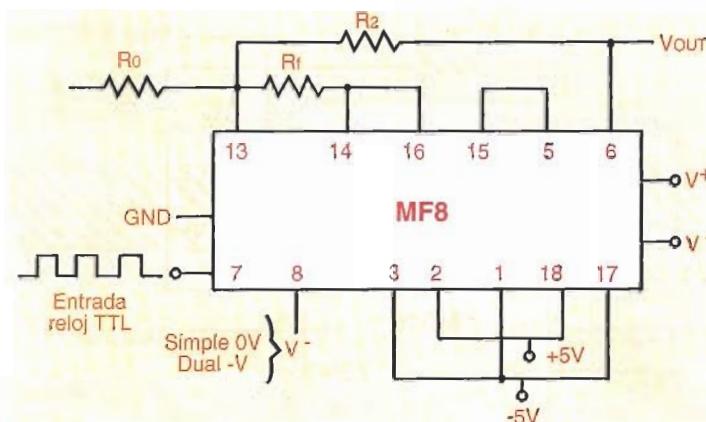


Figura 8. Diseño de un filtro pasabanda

rados y un amplificador operacional inverso. La frecuencia central es controlada por un reloj externo y puede estar entre los 0.1 Hz y los 20 KHz; la frecuencia del reloj puede ser programada 50 o 100 veces mayor que la frecuencia central; además se debe definir el tipo de señal lógica si es TTL o CMOS. Lo anterior se debe tener muy en cuenta para el diseño y configuración del circuito, figura 5.

Utilizando realimentación negativa a través de tres resistencias se puede hacer una función de filtrado de cuarto orden con cualquiera de los coeficientes de los polinomios mencionados.

La lógica Q controla el ancho de banda del circuito pues es dependiente

de ésta y posee 31 valores diferentes para la programación en los pines de entrada, figura 6.

Descripción de los pines

Ver diagrama de pines en la figura 7:

- **Entradas lógicas Q** : Entradas de programación del filtro pasa banda. **A,B,C,D,E**
(3,2,1,18,17)
 - **V+ (12) y V-** :Alimentación del circuito.
 - **F1 IN (16), F2 IN (5)** : Entradas respectivas de los filtros pasa banda.
 - **F1 OUT(15),F2 OUT (6)** : Salidas respectivas de los filtros pasa banda.
 - **A IN (13)** : Entrada inversora al amplificador operacional.
 - **A OUT (14)** : Salida amplificador.
 - **50/100/ (10)** : Para 100 veces mayor

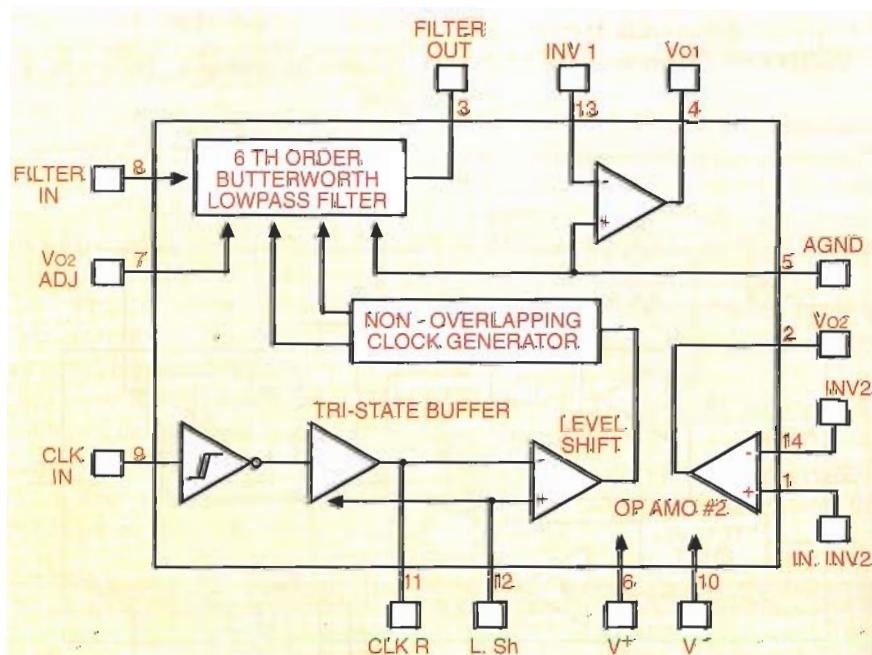


Figura 9: Diagrama de bloques y pines del MF6

la frecuencia del reloj que la frecuencia central este pin se conecta a V+ y para 50 veces a V-.

- **TTL CLK (7)** : Entrada para reloj TTL.
- **CMOS CLK (8)** : Señal de reloj tipo CMOS.
- **RC** : Generación automática de reloj mediante una red RC externa.

Procedimiento de diseño de un filtro pasa banda

Observemos el diseño de un filtro pasabanda de cuarto orden, con frecuencia central de 1 khz y ancho de banda de 400 hz con un MF8:

f_0	Frecuencia central del filtro.
BW	Ancho de banda (Banda de trabajo para el filtro)
k_2, k_q, k_0	Coeficientes de los polinomios de Chebyshev .
R_f, R_t, R_s	Resistencias a calcular para el diseño.
Q	Lógica con la que se programa el filtro según las características escogidas para el diseño.

Para el diseño de los filtros pasa banda con el MF8, se debe tener en cuenta :

- Calcular las tres resistencias.
- Definir la frecuencia central y el ancho de banda de trabajo.
- Frecuencia y tipo del reloj.
- Encontrar los coeficientes de los polinomios.
- Calcular el factor Q y buscar en la tabla su correspondiente valor lógico.

Cálculos:

$f_0 = 1 \text{ KHz}$. BW = 400 Hz
 $k_2, k_q, k_0 = 1,504 ; 1,822 ; 1,093$ Coeficientes de los polinomios de **Chebyshev** con 1.0 decibeles de "Ripple". (ver

hojas de datos)

Frecuencia del reloj: 100 KHz.

Fórmulas :

$$K_0 = \frac{R_0}{R_1}, K_2 = \frac{R_2}{R_1}, Q = K_0 * \frac{R_0}{BW}$$

Escogemos

$$R_f = 100\text{K}\Omega$$

Calculamos :

$$R_2 = K_2 * R_f = 1,504 * 100\text{K}\Omega = 15,04\text{K}\Omega$$

$$R_0 = K_0 * R_0 = 1,093 * 100\text{K}\Omega = 10,93\text{K}\Omega$$

$$Q = K_0 * \frac{f_0}{BW} = 1,822 * \frac{1000}{400} = 4,56$$

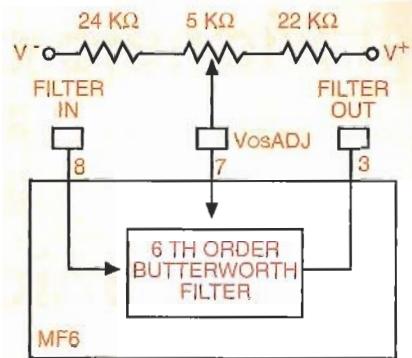


Figura 10. Corrección del voltaje de Offset

Este Q se compara en la tabla y se encuentra un correspondiente lógico que para éste caso es (01010) lo que significa poner los pines 3,1,17 a -5V y los pines 2,18 a +5V, figura 8.

Como se puede notar, el diseño de filtros con este tipo de circuitos integrados es muy sencillo, y como en algunos chips se presentan problemas de voltaje de offset como en el MF10 y MF5; en las hojas de datos se dan algunas técnicas para corregirlo. En esta familia existen circuitos integrados que brindan la posibilidad de corregirlo directamente configurando un pin del integrado como en el MF6.

MF6

El MF6 es un filtro pasabajo de 6° orden de Butterworth, con frecuencia central dependiente del reloj (TTL o CMOS). Su diagrama de bloques y de pines se muestran en la figura 9.

Descripción de los pines

- **FILTER OUT (3)** : Salida del filtro pasabajo.
- **FILTER IN (8)** : Entrada filtro pasabajo.
- **Vos ADJ (7)** : Pin para ajustar niveles DC indeseables (Voltaje Offset).

- **AGND** : Tierra analógica.
- **Vo1 (4), INV1(13)** : Salida y entrada inversora del Amp. Operacional 1.
- **Vo2 (4), INV2 (14), NINV(1)** : Salida, entrada inversora y entrada no inversora del amplificador operacional 2.
- **V+ (6), V- (10)** : Voltajes de alimentación.
- **CLK IN (9)** : Entrada para reloj CMOS.
- **CLK R (11)** : Entrada para reloj tipo TTL.
- **L.Sh** : Se configura según el tipo de reloj escogido.

La ventaja de este circuito es el ajuste de Offset en el pin 7 que se puede hacer con un potenciómetro, tal como se muestra en la figura 10. Además, se pueden obtener funciones de hasta orden 12 que brindan una respuesta excelente; para esto se conectan dos MF6 en cascada como lo muestra la figura 11.

Para más información, remitirse a las hojas de datos de los circuitos en los manuales de la firma National Semiconductor o en Internet bajar el archivo de tipo pdf de interés. \square

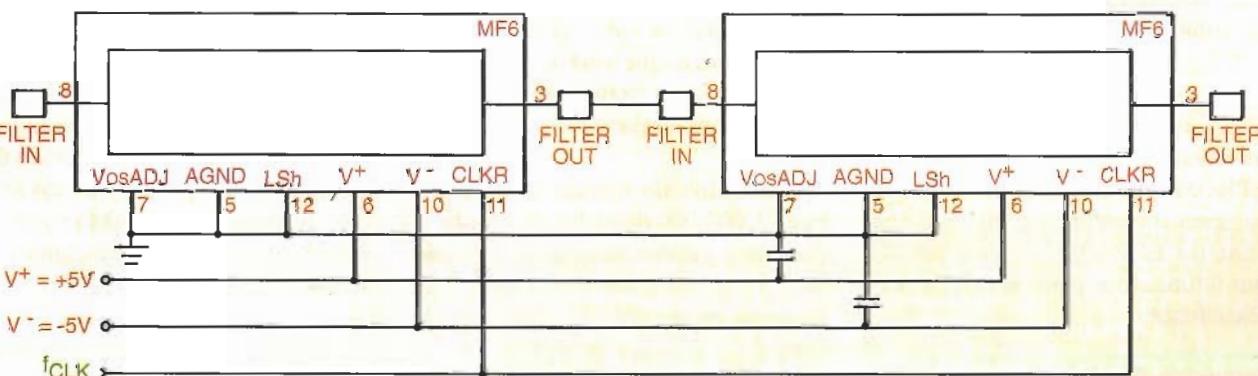


Figura 11. Conexión en cascada para obtener un filtro pasabajo de orden 12

Manejo de la memoria de datos EEPROM en microcontroladores PIC



En esta ocasión vamos a analizar el funcionamiento de las memorias de datos EEPROM.

La claridad en el manejo de estos conceptos es vital para el desarrollo de muchos proyectos. Si tiene alguna duda sobre algún aspecto del manejo de microcontroladores de Microchip, Intel, Motorola o Atmel, puede remitirla por e-mail.

JUAN ANDRÉS CASTAÑO WELGOS

welgos@telesat.com.co



xisten muchos dispositivos EEPROM disponibles en el mercado. Estos se pueden clasificar en tres tipos generales: EEPROM seriales, EEPROM paralelas y microcontroladores con EEPROM de datos internas, como por ejemplo los PIC16F84, 16F870 y 16F87X de Microchip.

En este artículo analizaremos el funcionamiento y uso de las memorias EEPROM que se encuentran en algunos microcontroladores PIC y de las memorias EEPROM seriales, por ser estas últimas muy comunes en muchas aplicaciones.

Memorias EEPROM internas

La memoria EEPROM de datos pue-

de leerse y escribirse durante la operación normal de un sistema. En esta memoria se pueden almacenar datos y variables para que no se pierdan cuando se desconecta la alimentación del sistema. Analizando los requerimientos que su aplicación requiere, puede ocurrir que esta memoria sea suficiente y no tenga que recurrir a memorias externas.

Este tipo de memorias soportan hasta 1.000.000 de ciclos de escritura/borrado y pueden guardar la información sin alterarla más de 40 años. Esta memoria no se encuentra en el espacio normal de archivos de registro; en lugar de eso, debe ser direccionada indirectamente a través de registros espe-

ciales. En la actualidad los microcontroladores PIC sólo ofrecen 64 bytes de memoria EEPROM, ubicados en el rango de direcciones de 0h-3fh.

Algunos dispositivos programadores pueden programar la memoria EEPROM de datos. Esta memoria puede ser leída o escrita durante la ejecución del programa, siendo un bloque completamente separado de la memoria de programa y de los archivos de registros (RAM), y puede usarse para almacenar cualquier tipo de datos.

El rango de direcciones de esta memoria es accesado colocando la dirección en un registro especial (EEA-

DR) y comunicando los datos por medio de otro registro especial llamado EEDATA. Los datos son leídos o escritos en un byte al tiempo. Un sólo bit no puede ser leído o escrito. Se usan cuatro registros de funciones especiales para leer y escribir en la EEPROM de datos, los cuales se muestran en la tabla 1.

El tiempo de escritura es de aproximadamente 10 milisegundos y es controlado por el temporizador que hay dentro del chip. En realidad ese tiempo resulta muy largo para la velocidad del procesador, por ese motivo existen varios bits en el registro EECON1 para supervisar la completa y correcta terminación del ciclo.

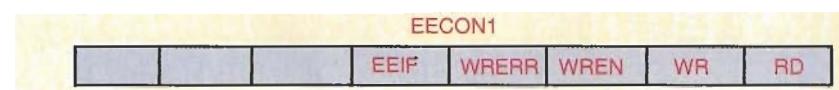
EEADR. El registro de 8 bits EEADR, es capaz de direccionar un máximo de 256 posiciones, pero sólo las primeras 64 son usadas. El rango de direcciones utilizable es de 0x00 a 0x3F.

EEDATA. Es un registro de 8 bits; se usa para pasar datos desde y hacia la memoria EEPROM de datos.

EECON2 Este no es un registro físico; sin embargo la dirección 0x89 es asignada a él. Al leerlo, todos sus bits son cero. Sólo se emplea como un dispositivo de seguridad durante el proceso de escritura de la EEPROM, para evitar las interferencias en el largo intervalo de tiempo que precisa su desarrollo. En él se escriben dos palabras de control y forman parte del ciclo de escritura.

EECON1. Este es el registro de control y tiene la función de controlar las operaciones en la EEPROM; su distribución de bits se muestra en la figura 1, y su descripción es la que se explica a continuación.

Los bits de control RD y WR inicián la lectura y la escritura. Se debe tener en cuenta que estos bits sólo pueden ser colocados en uno (no se pueden colocar en cero) por software para que inician el ciclo de escritura/lectura. Estos se limpian automáticamente al completarse un ciclo de lectura/escritura.



RD: Lectura

1: Se pone en uno cuando se va a realizar un ciclo de lectura de la EEPROM. Luego pasa a cero automáticamente.

WR: Escritura

1: Se pone en uno cuando se inicia un ciclo de escritura en la EEPROM. Cuando se completa el ciclo pasa a cero automáticamente.

WREN: Permiso de escritura

1: Permite la escritura en la EEPROM
0: Prohibe la escritura

WRERR: Señalizador de error de escritura

1: Se pone en uno cuando una operación de escritura ha terminado prematuramente
0: La operación de escritura se ha completado correctamente.

EEIF: Señalizador de final de operación de escritura

1: Cuando este señalizador se pone a 1 indica que la operación de escritura se ha completado con éxito. Se pone a 0 por programa.
0: La operación de escritura no se ha completado.

Figura 1. Registro de control de la EEPROM de datos

Una operación de escritura es válida cuando el bit WREN se pone en uno. WREN se limpia cada vez que se enciende el sistema. El bit WRERR se pone en uno cuando un ciclo de escritura se interrumpe por un reset debido al pin MCLR o al watchdog. Después de un reset, por medio del software, se puede chequear el bit WRERR y reiniciar el ciclo interrumpido de escritura. Las direcciones y los datos se mantendrán sin cambios en los registros EEADR y EEDATA.

Ciclo de lectura

Se inicia un ciclo de lectura colocando la dirección a acceder en el registro EEADR y poniendo el bit RD = 1 en el registro EECON1. En el siguiente ciclo de instrucción, los datos estarán disponibles en el registro EEDATA y permanecerán allí hasta que se realice una nueva lectura o escritura en la EPROM.

EEADR se localiza en el banco 0 (RAM) y EECON1 en el banco 1, de manera que se requiere seleccionar cada banco dentro del código, figura 2.

Ejemplo de lectura de la EEPROM:

```

movlw  dirección ;definir dirección a leer
movwf eeadr    ;informarla dirección al
                ;registro
bsf   status, rp0 ;banco 1
bsf   eecon1, rd ;bit de lectura
bcf   status, rp0 ;retorna al banco 0
movf  eedata, w  ;datos hacia W

```

Ciclo de escritura

El ciclo de escritura comienza cargando en EEADR la dirección de la posición a escribir y con el registro EEDA-TA el valor a grabar. Después se debe añadir una secuencia de instrucciones que inician la escritura, en la cual participa el registro EECON2. Este registro, que en realidad no se halla implementado físicamente, sólo asume funciones de seguridad en el proceso, car-

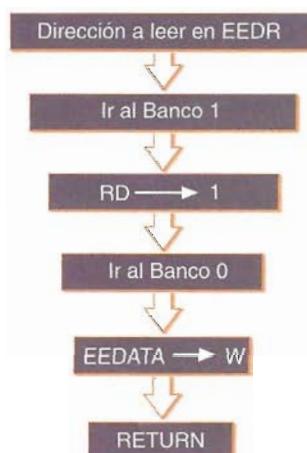


Figura 2. Diagrama de flujo del ciclo de lectura en la EEPROM de datos

Registro	Dirección
EEADR	0X09 -> dirección 8 del banco 0
EEDATA	0X08
EECON1	0X88 -> dirección 8 del banco 1
EECON2	0X89

Tabla 1. Registros especiales para el manejo de la EEPROM

```

;--MANEJO DE LA EEPROM--
list p=16c84
radix hex

;-----


; MAPEO DE MEMORIA
status equ 0x03
portb equ 0x06
eedata equ 0x08
eaddr equ 0x09
direccion equ 0x0c
byte equ 0xd0
trisb equ 0x86
eecon1 equ 0x88
eecon2 equ 0x89
;

;-----


;MAPEO DE BITS
rd equ 0
wr equ 1
wren equ 2
rp0 equ 3
;

;

inicio banco 1 org 0x000
bsf status, rp0 ;
movlw b'00000000' ;
salidas movwf trisb
bcf status, rp0 ;
banco 0 clrf portb
movlw b'00000001'
movwf byte ;byte de
;prueba a grabar
movlw 0x00
movwf direccion
call escritura
bsf status, rp0
eecon1, wr; byte
;escrito?
btfsr goto escrcompleta
bcf status, rp0
call lectura
movwf portb ;mostrar
;dato en el puerto b
goto ciclo

ciclo movf direccion,w ;
;tomar dirección
movwf eaddr ;
;dirección hacia eaddr
bsf status, rp0;banco1
bsf eecon1, rd ; leer
;dato de la EEPROM
bcf status, rp0
movf eedata,w;data-> w
return ; retorna dato
;en w

;-----

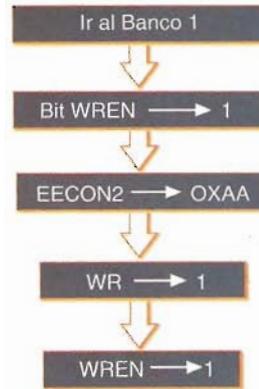

;escritura movf direccion,w ;toma
;dirección
movwf eaddr
movf byte,w
movwf eedata
bsf status, rp0
bsf eecon1, wren
; habilita escritura
movlw 0x55
movwf eecon2
movlw 0xaa
movwf eecon2
bsf eecon1, wr;WR -> 1
bcf eecon1, wren
; deshabilita escritura
bcf status, rp0;banco0
return

;

end

;FUSIBLES:
;MEMORIA DESPROTEGIDA
;WATCHDOG DESHABILITADO
;CRISTAL 4 MHZ
;POWER UP TIMER ON
;

```

Figura 4. Código del ejercicio de lectura y escritura en la EEPROM**Figura 3.** diagrama de flujo del ciclo de escritura en la EEPROM

gándose en él dos valores concretos: 55H y AAH. El tiempo de estos pasos es un aspecto crítico, de manera que recomiendo que sigan esta secuencia. Las interrupciones deben deshabilitarse mientras se ejecuta este procedimiento. El bit WREN en EECON1 debe ponerse en uno para habilitar el WRITE. Esto previene escrituras accidentales en la memoria EEPROM de datos. El bit WREN debe ponerse en cero en todas las ocasiones; excepto cuando se ejecuta WRITE, ver figura 3.

Al escribir un dato en esta memoria, se borra automáticamente el dato anterior y se graba el nuevo. El tiempo de escritura es controlado por el timer del chip. Este tiempo puede variar según las condiciones de voltaje y temperatura, al igual que entre diversos tipos de chips.

Al completar el ciclo de WRITE, el bit WR se pone en cero automáticamente y la bandera de escritura completa en EE es activada. El programa puede detectar si el ciclo de WRITE se ha cum-

plido habilitando esta interrupción, mirando la bandera EEIF o el bit WR. Los bits EEIF y WREN deben ser limpiados por software. En la figura 4 vemos un ejemplo de manejo de lectura y escritura en la EEPROM, cuyo diagrama esquemático se ve en la figura 5.

Una buena precaución es verificar si la escritura de la EEPROM ha sido correcta, para lo cual se suele restar el dato escrito con el que existe en el registro EEDATA. Si no se ha producido error, el señalizador Z pasa a valer 1.

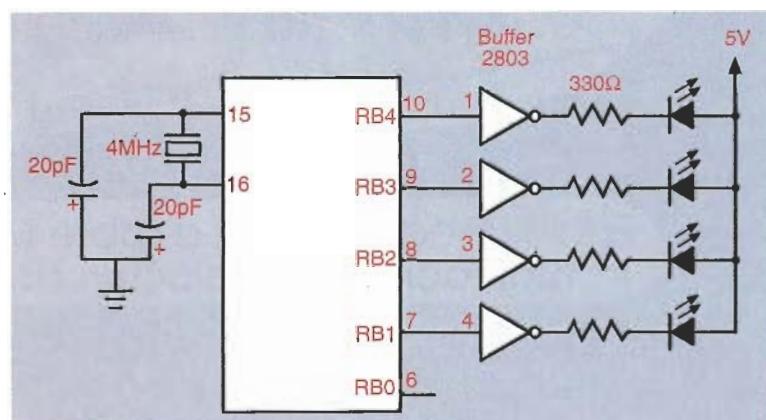
Memorias EEPROM seriales

La tecnología de las memorias seriales ha surgido como solución para sistemas que requieren bajo tamaño, poca potencia y efectividad. Entre sus principales ventajas se cuentan:

- Se pueden conectar fácilmente con microprocesadores o microcontroladores; incluso algunos de ellos tienen pines dedicados a esa labor.
- Transferencia de datos de manera serial, lo que permite ahorrar pines del micro, al igual que ocupan poco espacio y consumen poca potencia.

Cuando se ha decidido por el uso de este tipo de dispositivos, el paso siguiente es seleccionar uno de los dos protocolos posibles de comunicación con la EEPROM serial (2 o 3 hilos).

La elección de los dos hilos se debe hacer en aplicaciones que requieran bus I²C, inmunidad al ruido y disponibilidad limitada de pines. La elección de los 3 hilos es conveniente en casos de

**Figura 5.** Diagrama esquemático del ejercicio de lectura y escritura en la EEPROM

requerimientos limitados de protocolos, un protocolo SPI, alta frecuencia de reloj o en aplicaciones de 16 bits.

BUS DE 3 HILOS En este grupo se encuentran los dispositivos con la nomenclatura 93XXXX. En ellos se requieren de cuatro pines.

BUS DE 2 HILOS La nomenclatura común es 24XXXX y 85XXXX, ver tabla 2. El protocolo I²C utiliza comunicación bi-direccional maestro/esclavo. Tanto el maestro como el esclavo pueden operar como transmisor o como receptor. La operación del bus debe ser manejada por el maestro, quien genera la señal de reloj serial y las condiciones de inicio de detención.

Estas memorias tienen un pin llamado SCL que recibe los pulsos generados por el maestro, y otro llamado SDA que maneja el flujo de datos bidireccionalmente. Este dispositivo no requiere de un pin habilitador, ya que en este esquema la transferencia de información sólo puede de iniciar cuando el bus esté libre. Cada memoria tiene su dirección, determinada mediante los pines A0, A1 y A2; en una red sólo responderá la memoria cuya dirección coincida con la dirección que va encabezando la trama de información.

Cuando un microcontrolador desea establecer comunicación con la memoria, debe enviarle una serie de bits que llevan la siguiente información:

1. Se envía el bit de arranque
2. El código 1010 (propio de estas memorias)
3. La dirección del dispositivo (A2, A1, A0)
4. Un bit que indica que se desea escribir ('0') en la memoria

Después de esto, la memoria debe enviar un reconocimiento para informarle al microcontrolador que recibió la información. Dicha señal, llamada ACK, consiste en poner el bus en un nivel bajo (lo hace la memoria). Después el microcontrolador debe enviar los bits que corresponden a la posición de memoria que se quiere leer o escribir; nuevamente la memoria envía un reconocimiento. El paso siguiente depende de la operación que se vaya a ejecutar.

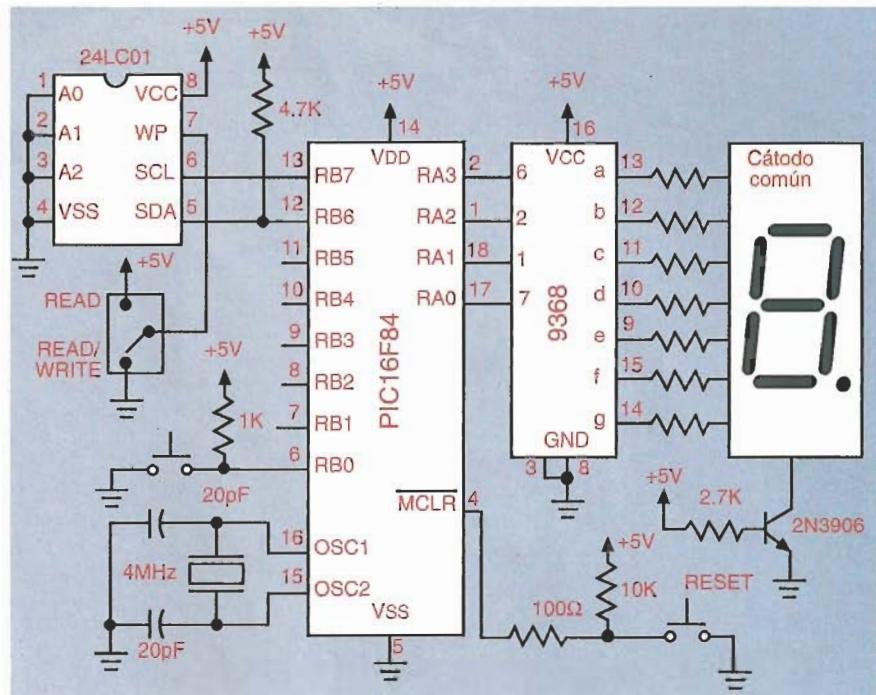


Figura 6. Ejemplo de manejo de una memoria serial



Figura 7. Montaje del ejercicio de manejo de memoria serial

Referencia	Capacidad en K bits	Bloques internos	A0	A1	A2	Dispositivos en el bus
24LC01B, 24C01	1	1	1 0 0	1 0 0	1 0 0	8
24LC02B, 24C02	2	1	1 0 0	1 0 0	1 0 0	8
24LC04B, 24C04	4	2	X	1 0 0	1 0 0	4
24LC08B	8	4	X	X	1 0 0	2
24LC16B	16	8	X	X	X	1

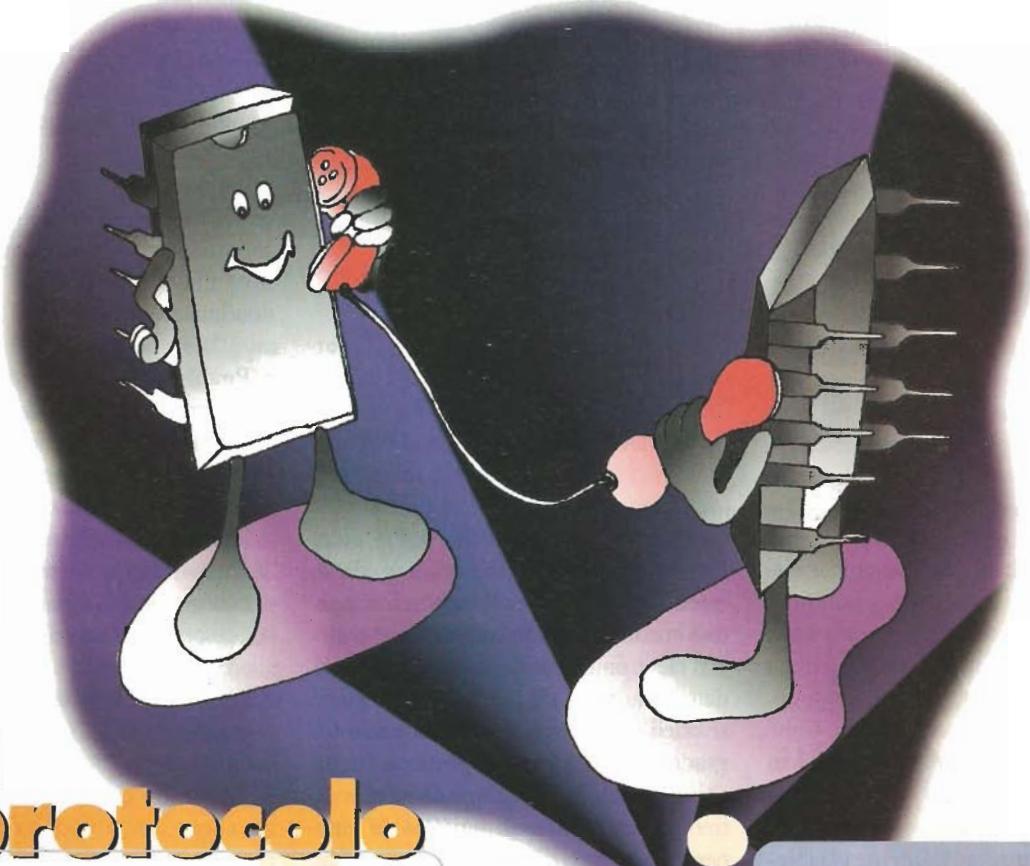
Tabla 2. Nomenclatura de las memorias seriales con bus de 2 hilos

Ejemplo de aplicación

En este ejemplo se realizará un contador de 0 a 9 con un interruptor pulsador y un display de siete segmentos, figura 6. El número que se muestra en el display se va a almacenar simultáneamente en una memoria 24LC01. En este caso, los pines de la dirección se conectaron a tierra, al igual que el pin WP. La resistencia de 4.7 Kohm conectada al pin SDA es necesaria dado que dicho pin tiene colector abierto. El display se conecta al

puerto A y el pulsador al pin RB0, figuras 6 y 7.

El programa lleva el control del conteo decimal y almacena en la memoria el mismo dato que se envía al display. Si necesita el código de este ejercicio o tiene alguna duda del manejo de memorias EEPROM o sobre otro tema que le gustaría que desarrolláremos en esta sección, puede enviarme un e-mail cekit@cekit.com.



El protocolo 1-wire

RICARDO RESTREPO

Profesor Universidad Tecnológica de Pereira

EDISON DUQUE

Profesor Universidad Tecnológica de Pereira

de Dallas Semiconductor

En este artículo veremos la forma más sencilla de implementar una comunicación serial entre un microcontrolador PIC y los circuitos integrados construidos por Dallas Semiconductor, cuya comunicación con el mundo exterior se hace mediante una interface específica diseñada por ellos, conocida como interface 1-wire.

La transferencia de información entre diferentes dispositivos o equipos ha permitido que se construyan aparatos de alto desempeño, con mayores prestaciones y obviamente, con mayor eficiencia. A nivel de dispositivos electrónicos, la comunicación se ha desarrollado fundamentalmente de dos formas: Comunicación en paralelo y comunicación serial. La comunicación en paralelo, por ejemplo entre una computadora y una impresora o entre un microprocesador y una memoria EPROM, tiene como ventaja que el intercambio de información es muy rápido ya que se pueden enviar y recibir datos de 8, 16, 32 o más bits al mismo tiempo. Su desventaja radica en que se requieren muchas líneas de conexión entre los equipos o dispositivos a unir.

La comunicación serial por su parte presenta diferentes modalidades. Por ejemplo, la interface RS-232, muy popular para comunicar computadoras, PLC's (controladores lógicos programables) y microcontroladores, entre otros, es un tipo de comunicación serial asíncrona ya que no existe ninguna línea con señal de reloj entre los dispositivos a unir; en este caso la validez de los datos en la línea de comunicación depende de tiempos previamente establecidos o configurados en los equipos.

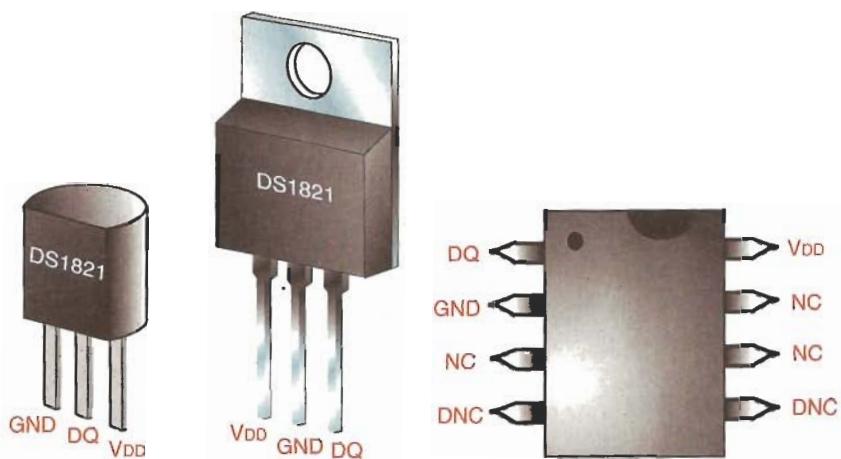


Figura 1. Aspecto físico del sensor de temperatura DS1821

Cuando se habla de comunicar circuitos integrados entre sí, las posibilidades aumentan. Esto se debe a que cada fabricante de dispositivos electrónicos ha creado diferentes interfaces o protocolos de acuerdo a sus necesidades. Cabe aclarar que muchos otros fabricantes han adoptado también dichas interfaces para que sus dispositivos sean compatibles con los estándares previamente establecidos. Por ejemplo, la empresa *Philips* creó la interface serial I^C (Inter-Integrated-Circuit), la cual se puede encontrar en las memorias seriales de la serie 24LC01, 24LC02, etc.; la empresa *National Semiconductor* creó la interface *Microwire*, que se puede encontrar en las memorias seriales de la serie 93LC46, 93LC56, etc.; la empresa *Motorola* creó la interface SPI (Serial-Peripheral-Interface), la cual se puede encontrar en diferentes convertidores A/D, convertidores D/A, etc.; por su parte, la empresa Da-

llas Semiconductor ha popularizado la interface 1-wire, que se caracteriza porque la comunicación entre los dispositivos electrónicos utiliza una sola línea de datos y la señal de tierra; esta interface se puede encontrar en los circuitos integrados especiales que produce dicha empresa, tales como chips de relojes de tiempo real, sensores digitales de temperatura, convertidores A/D, etc..

Procedimiento

La idea con este proyecto es construir un equipo que nos permita conocer y aplicar la interface 1-wire. En vista de ello se decidió buscar la opción más sencilla y fácil de explicar, lo que nos condujo al diseño y construcción de un termómetro digital que utiliza como sensor un dispositivo de medición y control de temperatura construido por *Dallas Semiconductor* (obviamente con interface 1-wire) referencia DS1821, figura 1.

También se escogió un microcontrolador PIC para que controlara el sistema gracias a su popularidad y facilidad de uso, y como elemento de visualización, se optó por un módulo LCD.

En la figura 2 se muestran los bloques internos que conforman el DS1821. Este dispositivo posee varias características muy especiales que nombramos a continuación:

- Realiza la medición de temperatura y entrega el valor obtenido en forma de una palabra digital de 8 bits.
- Posee únicamente tres pines: tierra, fuente y la línea de datos bidireccional.
- Puede ser configurado para que actúe como un controlador de temperatura autónomo o para que se comunique con un dispositivo maestro (microcontrolador).
- Posee un rango de medición de temperatura de -55° a +125°C.

Cuando este dispositivo se utiliza como controlador de temperatura, el microcontrolador debe enviarle los valores máximo y mínimo que son aceptados por el sistema (para definir la histéresis); una vez se han programado estos datos, simplemente se envían los parámetros de configuración requeridos para que el DS1821 entre en modo de funcionamiento como controlador, en cuyo caso la línea de comunicación de datos (pin DQ) se convierte en la salida que manejará el actuador del control de temperatura (ya sea un ventilador o un calentador).

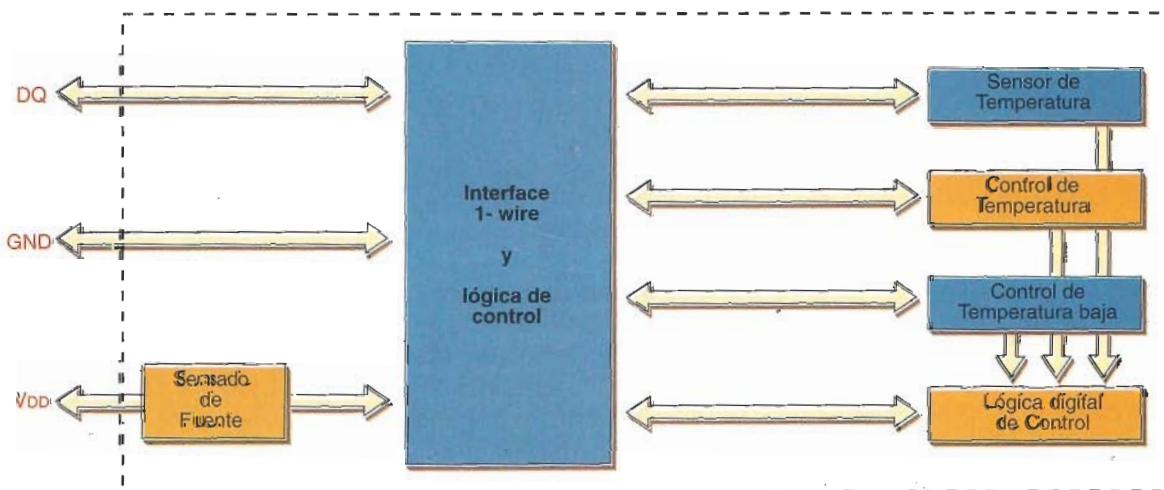


Figura 2. Arquitectura interna del DS1821

Temperatura	Dato digital del DS1821	Equivalente Hexadecimal
+125°C	01111101	7Dh
+25°C	00011001	19h
0°C	00000000	00h
-1°C	11111111	FFh
-25°C	11100111	E7h
-55°C	11001001	C9h

Figura 3. Formato en que entrega el valor de temperatura el DS1821

Cuando el DS1821 se configura para que trabaje como un esclavo del sistema maestro (o sea del microcontrolador), la línea de datos (pin DQ) permanece siempre dedicada a la comunicación. Este último caso es el que nos interesa ya que la idea en este proyecto es que el microcontrolador PIC le indique al DS1821 en qué momento debe empezar la medición y este último debe entregarle como respuesta una palabra digital de 8 bits que equivale a la temperatura medida. En la figura 3 se muestran los valores digitales que entrega el sensor.

Termómetro digital

En la figura 4 se muestra el diagrama esquemático del circuito; note que el módulo LCD se ha configurado para

ser controlado con interfaz de 4 bits. El DS1821, por su parte, emplea una sola línea de datos bidireccional, razón por la que el microcontrolador PIC debe estar permanentemente cambiando el estado del pin RA0 (configurándolo como entrada o salida). En esta línea de datos, debe colocarse una resistencia de *pull-up* para que el sistema funcione correctamente (en nuestro caso es de 10k).

Dado que la especificación de la interfaz 1-wire es bastante larga y que los datos de configuración y operación del DS1821 son muy extensos, se deja al lector la tarea de buscar en Internet la hoja de datos de dicho dispositivo. De esta forma, el lector puede encontrar la información necesaria para con-

figurar el dispositivo en el modo de funcionamiento deseado y puede ir comparando dichos valores con los que se utilizan en el programa del microcontrolador. Dicho software se encarga de manejar el módulo LCD y de configurar el DS1821 para que realice una medición de temperatura cada vez que este le da la orden. La secuencia requerida para lograr dicha configuración y los pasos utilizados en la comunicación serial, están escritos en forma de rutinas, lo cual puede ser de gran ayuda para que los lectores las utilicen en otras aplicaciones con dispositivos 1-wire. Si desea recibir el archivo puede escribir a la siguiente dirección de correo electrónico: eduque@utp.edu.co Ω

Referencias

1. La hoja de datos del DS1821 se puede buscar en la página web de Dallas Semiconductor: www.dalsemi.com
2. La hoja de datos del microcontrolador PIC16F84 se puede buscar en la página en Internet de Microchip Technology: www.microchip.com
3. El funcionamiento del módulo LCD se puede encontrar en la revista Electrónica & Computadores N° 36. Editorial CEKIT.

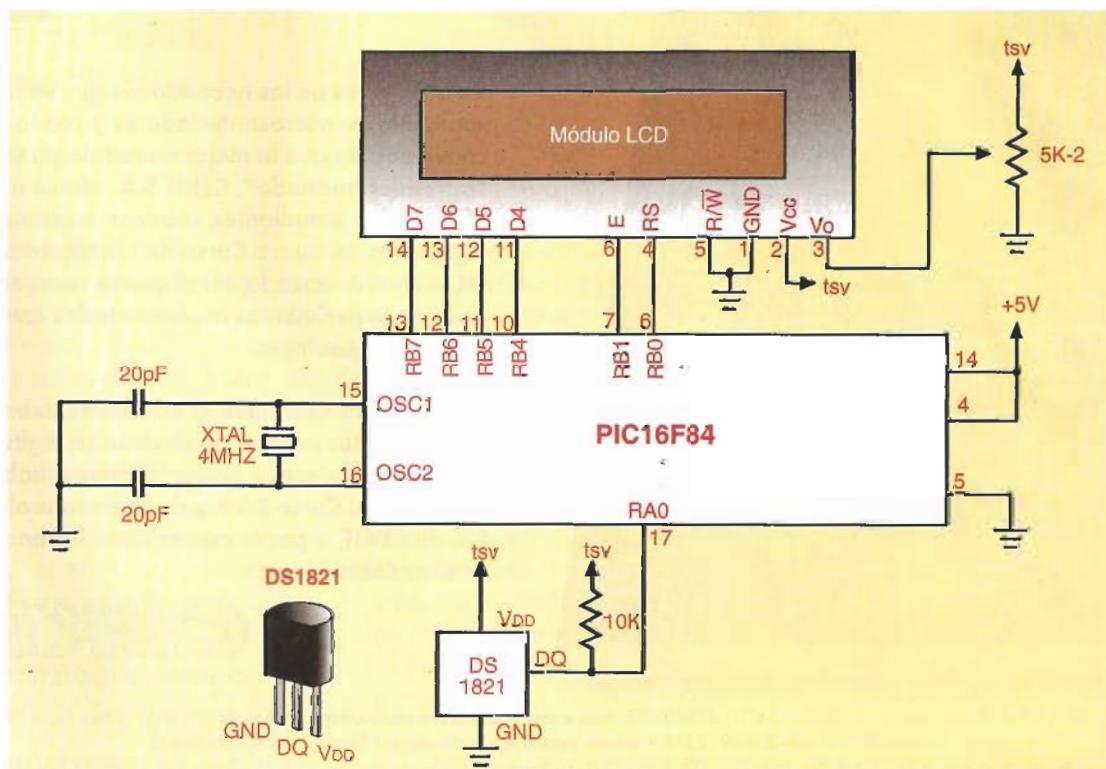
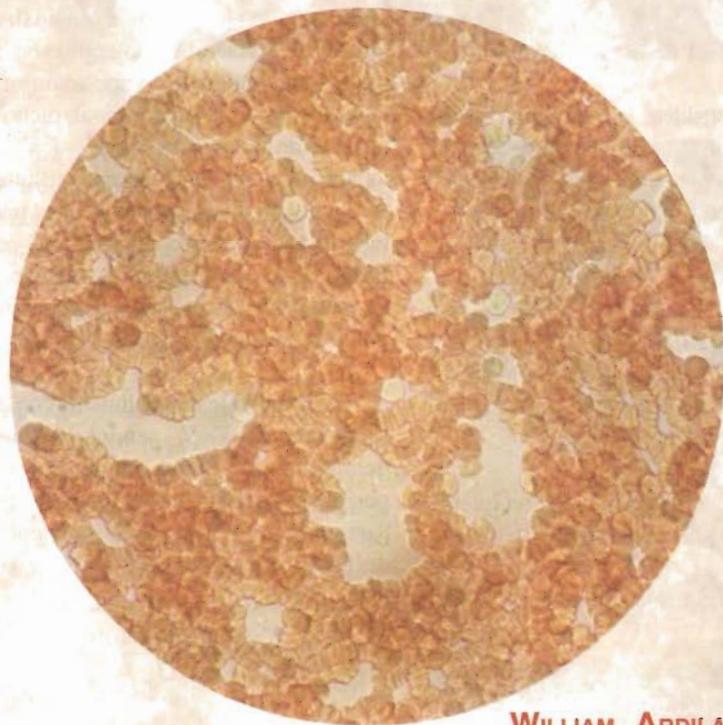


Figura 4. Circuito del termómetro digital construido para estudiar la interface 1-wire

Propiedades Eléctricas de las Células Sanguíneas

En este artículo se pretende demostrar experimentalmente las propiedades dieléctricas de la célula, en particular de los glóbulos rojos, cuya capacidad por unidad de superficie es aproximadamente de 0.9 $\mu\text{f}/\text{cm}^2$. Se describe detalladamente el protocolo experimental y se utiliza el método de desfasamiento para obtener dicho valor.



WILLIAM ARDILA U.
CARLOS A. LÓPEZ A.
ALVARO A. OROZCO G.

F

Fundamento teórico

El modelo de circuito equivalente de la membrana celular se reduce en primera aproximación a un circuito paralelo de una resistencia y un condensador, a partir del cual se desarrolla el potencial de acción, de ahí la importancia de conocer experimentalmente el valor de la capacidad para este condensador que dada la composición, fundamentalmente grasa de la membrana celular, es relativamente alta y comparable a los condensadores industriales como: aceite, mica, etc. Es importante anotar que los primeros trabajos en este campo de medición se deben a Fricke y Cole (1910), quien utilizando métodos electrónicos rudimentarios pero muy ingeniosos para su época, hallaron valores sorprendentes de $0.9 \mu\text{f}/\text{cm}^2$.

Consideremos una célula en forma esférica inmersa en un campo eléctrico, figura 1. Existen tres condiciones de contorno que se deben tener en cuenta para este tipo de medición:

- En zonas muy alejadas de la célula, o sea para grandes valores de r , el campo eléctrico es igual a E_0 y las líneas de campo son paralelas.
- Considerando el medio intracelular como un conductor perfecto, el campo en el interior de la célula es nulo.
- El potencial cumple la ecuación de Laplace.

Teniendo en cuenta estas consideraciones, se pueden hacer los desarrollos matemáticos que no son objetivo de esta

publicación ya que pueden ser fácilmente consultados en cualquier libro tradicional de electromagnetismo. La diferencia de potencial entre los medios intracelular y extracelular está dada por;

$$U(r_0) = \frac{3E_0r_0 \cos\theta}{2} \quad (1)$$

donde:

E_0 : Campo eléctrico constante.
 r_0 : radio de la célula.

El campo eléctrico en el interior de la membrana celular viene dado por consiguiente;

$$E_m = \frac{3E_0r_0 \cos\theta}{2\delta} r_0 \quad (2)$$

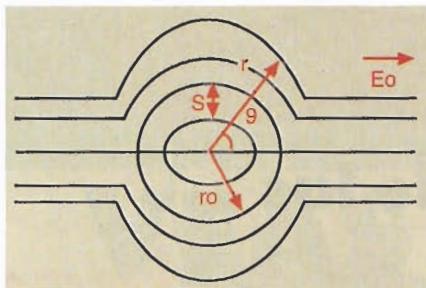


Figura 1. Distribución de campo y potencial en el entorno de una célula esférica cuando se aplica un campo eléctrico externo E_o .

donde:

d: Espesor de la membrana.

Si se considera la membrana en pequeñas coronas esféricas, se puede por integración encontrar la energía capacitativa almacenada en una célula, figura 2.

$$E = \frac{3\pi}{2} C_m r_o^4 E_o^2 \quad (3)$$

Si se confinan N células de concentración n en un volumen, figura 3, $V=sd$, donde :

d: distancia entre paredes del recipiente
S: área lateral del recipiente.

La energía total almacenada en el depósito es:

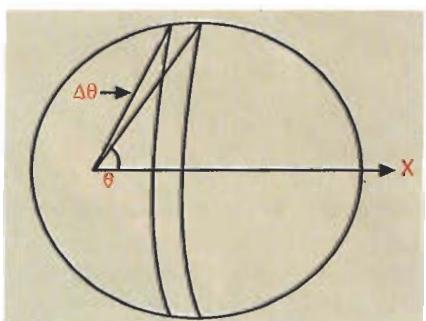


Figura 2. Corona esférica de una célula.

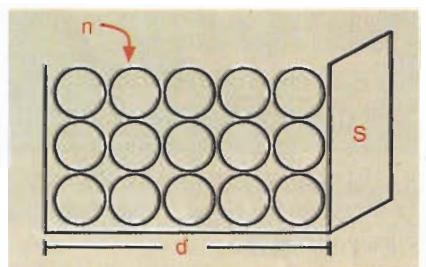


Figura 3. Células confinadas en un volumen

$$E_T = \frac{3\pi}{2} C_m E_o r_o^4 nsd \quad (4)$$

donde:

C_m : capacidad por unidad de superficie.

Lo que hasta aquí tenemos es un condensador formado por células que almacenan una energía capacitativa total dada por la expresión 4.

Podemos entonces estudiar el comportamiento de dicho condensador si se aplica una fuente de voltaje senoidal, sin embargo es necesario sumergir estas células en un líquido de composición similar al medio biológico original, dicho líquido es de baja resistencia eléctrica y por lo tanto el equivalente eléctrico resultante. Dicho circuito equivalente es el paralelo formado por un condensador C_{eq} por efecto de la capacidad de las células en suspensión y una resistencia en paralelo por la presencia del líquido, figura 4.

Si el equivalente se alimenta con una señal senoidal $V=V_m \sin \omega t$, se puede demostrar que dicho condensador es posible conocerlo en función de los valores máximos de voltaje, corriente y desfase, un sencillo cálculo nos lleva:

$$C_{eq} = \frac{I_m \sin \varphi}{\omega U_m} \quad (5)$$

Y que almacena una energía total:

$$\text{Energía} = \frac{I_m \sin \varphi}{\omega U_m} \frac{d}{3\pi r_o^4 ns} \quad (6)$$

Ahora bien, manipulando adecuadamente las expresiones, se obtiene la capacidad por unidad de superficie de una célula como:

$$C_m = \frac{I_m \sin \varphi d}{\omega U_m 3\pi r_o^4 ns} \quad (7)$$

Debemos diseñar un experimento que nos permita conocer el desfase y los valores máximos de corriente y voltaje en un sistema R-C paralelo, y mediante un delicado protocolo los valores de:

r_o :radio de la célula.
 n :concentración.

Protocolo

Como células experimentales utilizamos eritrocitos que junto con los leucocitos y las plaquetas son compuestos fundamentales en la sangre. Los eritrocitos son las células más numerosas en la sangre, contienen hemoglobina que constituye el principal transportador de oxígeno.

El plasma

A la porción líquida de la sangre que circula se le denomina plasma. Es un fluido color pajizo, químicamente muy complejo, el cual contiene una amplia variedad de sustancias. Los glóbulos rojos, los glóbulos blancos y las plaquetas flotan en este medio líquido. Desde este punto de vista la sangre se puede mirar como un tejido líquido; contiene glóbulos, pero la sustancia intercelular es un líquido cambiante mas que un material de soporte algo más sustancial.

Glóbulos rojos o eritrocitos

Los glóbulos rojos o eritrocitos, son discos bicóncavos de unos 7 a 8 micrometros de espesor. A diferencia de la mayoría de las células, no poseen núcleo. Una armazón elástica mantiene la forma discoidal y hace que las células puedan encorvarse y deformarse para pasar por los vasos sanguíneos de menor diámetro que el de ellas. Los eritrocitos no pueden desplazarse en forma activa, sino que flotan simplemente en la corriente sanguínea, y son movilizados por el efecto impenetrante del corazón. El hombre adulto tiene un promedio de 5.400.000 glóbulos rojos por milímetro cúbico de sangre y la mujer adulta aproximadamente 5.000.000 por milímetro cúbico. Los recién nacidos tienen un número mayor de eritrocitos, que llega a 6 o 7.000.000 por milímetro cúbico; esta cantidad decrece después del nacimiento y aproximadamente a los tres meses alcanza el valor normal para

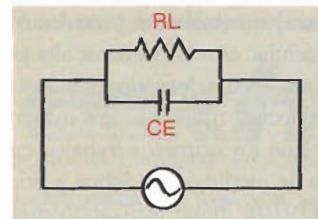


Figura 4. Condensador formado por efecto de las células de suspensión

el adulto. El cuerpo humano contiene alrededor de 30 billones de glóbulos rojos. Puesto que el número de glóbulos rojos por milímetro cúbico es necesario para determinar el estado de salud general de una persona, casi todos los exámenes clínicos incluyen un recuento de glóbulos rojos.

Glóbulos blancos o leucocitos

Existen cinco tipos de glóbulos blancos o leucocitos, cada uno de los cuales difiere considerablemente de los hematíes; tienen núcleo, no contienen hemoglobina (y por lo tanto son incoloros) y se desplazan activamente por medio de movimientos amiboidales. Los leucocitos pueden desplazarse en contra de la corriente sanguínea, y aún deslizarse de las paredes de los vasos sanguíneos y penetrar en los tejidos.

Los glóbulos blancos son mucho menos numerosos que los hematíes, hay un promedio de cerca de 7,000 por mm cúbico, pero fluctúan entre 5 y 9 ó 10 mil en las diferentes personas y aún en la misma persona a diferentes horas del día. La cantidad de leucocitos es mínima en las primeras horas de la mañana, y máxima a la tarde. Las personas deficientemente nutridas tienen un número de leucocitos inferior al normal, y una menor resistencia a las enfermedades; la disminución del número de leucocitos a 500 por mm. cúbico, o menos es fatal. Existen cinco tipos de glóbulos blancos que pueden distinguirse empleando diferentes técnicas con colorantes.

Las plaquetas

Las plaquetas sanguíneas, un tercer tipo de elementos figurados de la sangre, son importantes en la iniciación del proceso de coagulación de la sangre. Son cuerpos esféricos incoloros, desprovistos de núcleos con un diámetro aproximadamente igual a la tercera parte del de los hematíes. Se cree que la mayoría de ellos se origina por la fragmentación de las células gigantes en la médula ósea roja, pero experimentos recientes indican que algunos se forman a partir de células fagocitarias pulmonares. Su ciclo de vida se calcula en cuatro días. Se estima que hay entre 250,000 y 400,000 por milímetro cúbico. Las pla-

Figura 5. Toma de muestra sanguínea



quetas sanguíneas contienen cierto número de sustancias químicas, probablemente las más importantes son el fosfolípido cefalina, un activador de la protrombina similar a la tromboplastina de los tejidos pero más débil, y la serotoninina, una sustancia vasos-constrictora.

Para medir la capacidad dieléctrica de los glóbulos rojos es necesario aislarlos de la sangre de un voluntario, figura 5, para ello la muestra de sangre se lleva a una centrífuga, para separar los glóbulos rojos del plasma y los leucocitos, los cuales se extraen, dejando así solo los eritrocitos, figura 6.

Es necesario utilizar heparina para evitar la coagulación de la sangre. La medición del parámetro (concentración), se hace con un microscopio electrónico obteniéndose valores aproximados de 6×10^6 , número de glóbulos por milímetro cúbico, figura 7.

Estos eritrocitos ya cuantificados son inmersos en cloruro de sodio, un líquido que tiene características similares al plasma sanguíneo. Este es un líquido isotónico, lo cual significa que no hay forma que el glóbulo se hemolize o se reviente por presión osmótica. Para que no suceda esto se debe resolver el siguiente problema:

Un glóbulo rojo típico de 5000 A° de radio tienen una membrana de tensión superficial máxima 5×10^{-2} N m⁻¹. Lo introducimos en una disolución 1 M de NaCl. Suponemos que el NaCl no puede atravesar la membrana de la



Figura 6. Aspecto de la sangre después del proceso de centrifugado

célula. La concentración interior de sustancias que no pueden atravesar la membrana es 3 M. a) Reventará el glóbulo rojo? b) Cuál es la concentración mínima de de NaCl necesaria para que el glóbulo rojo no reviente? ($T=300\text{ K}$; $R=0,082\text{ atm l K}^{-1}\text{ mol}^{-1}$)

Al introducir un glóbulo rojo en una disolución de NaCl, dependiendo de la concentración de la misma, habrá una tendencia del agua a entrar o salir del glóbulo rojo. Esta tendencia vendrá reflejada por la presión osmótica π , que es función de la diferencia de concentraciones y de la temperatura.

Si la disolución es 1M, quiere decir que hay 1 mol l⁻¹ de NaCl y, por



Figura 7. Conteo electrónico de erictrocitos

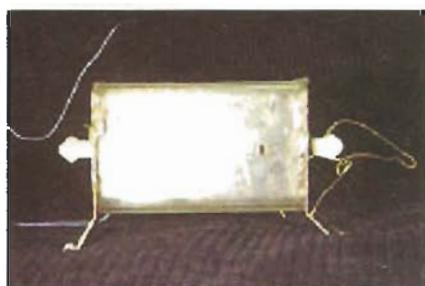


Figura 8. Recipiente donde se deposita la solución del cloruro de sodio en los glóbulos rojos

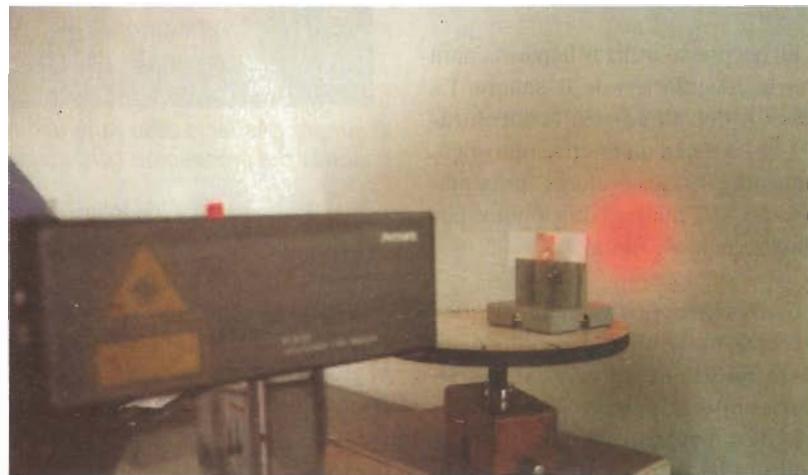


Figura 9. Fenómeno de difracción del glóbulo rojo

L(cm)	1	1	1	5	10	15	2
A(cm ²)	20	15	19	30	20	30	80
C _b (nM/cm ²)	2.63	2.42	2.01	0.85	0.29	0.28	5.75
C _m (mM/cm ²)	0.73	0.71	0.75	0.69	0.76	0.81	0.73
X(grados)	6	7	5	6	7	5	6

Tabla 1. Datos experimentales en la medición de la capacidad por unidad de área en un globulo rojo.

Cuando el agua que haya entrado en el glóbulo rojo haya producido un aumento de presión igual a la presión osmótica, el flujo de agua cesará. El valor de C se supone constante por el hecho de que aunque el flujo de agua sea suficiente como para variar la presión, es muy pequeño para variar la concentración apreciablemente. Aplicando la ley de Laplace.

$$T = \frac{r\Delta p}{2} = \frac{5x10^7 mx,49x10^6 Nm^{-2}}{2} \\ = 62,3x10^2 Nm^{-2} \quad (9)$$

Como esta tensión es mucho mayor que la tensión superficial máxima, el glóbulo rojo se rompe. Podemos calcular la diferencia de presiones máxima que puede haber entre el exterior y el interior sin que la membrana se rompa.

$$\Delta p = \frac{2T}{r} = \frac{2x5x10^2 Nm^{-2}}{5x10^7 m} \\ = 2x10^{-5} Nm^{-2} = 1,97 atm \quad (10)$$

Y, por tanto, la máxima diferencia de concentraciones viene dada por

$$\Delta C = \frac{\pi}{RT} = \frac{1,97 atm}{0,082 atm K^{-1} mol^{-1} x 300 K} \\ = 0,08 mol l^{-1} \quad (11)$$

Por lo que la concentración de iones Na y Cl conjunta debe ser como mínimo de 2,92 M, es decir una concentración de NaCl de 1,46 mol l⁻¹. Luego si el glóbulo rojo se introduce en una disolución de concentración acuosa 1,46 M o menor se romperá.

Este líquido se deposita en un recipiente de acrílico con dos electrodos de plata, figura 8, previa medición de la longitud del recipiente y área de los electrodos. Se han utilizado recipientes de diferentes longitudes y áreas, también se recomienda utilizar electrodos de electrocardiografía.

La medición del radio medio del glóbulo rojo se hace mediante la técnica de la interferencia haciendo uso de un láser como fuente de luz, ya que se produce el fenómeno de la difracción, pues el glóbulo rojo hace el papel de rendija circu-

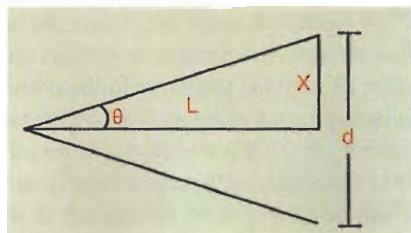


Figura 10. Proyección de un glóbulo rojo debido al laser

lar y tiene dimensiones comparables con la longitud de onda del láser, figura 9.

Se ubica un porta objetos con una gota de sangre a una distancia tal que se pueda identificar el mínimo círculo de color tenue sobre la pared para obtener su diámetro.

L: distancia del porta objetos a la pared.
d: diámetro del primer mínimo.
a: diámetro del glóbulo rojo, figura 10.

$$\tan\theta = \frac{x}{L} \quad a \sin\theta = 1.22\lambda \quad (12)$$

$\lambda = 6328 \times 10^{-10}$ mt, longitud de la onda del láser a $^{\circ} 32.200$ A°

Haciendo uso de un generador y trabajando a frecuencias entre 1 kHz – 10 KHz se implementa el montaje experimental de la figura 11. La señal de voltaje $V(t)$ se adquiere a través del canal 0 y la corriente a través del canal 1, previa conversión de corriente a voltaje y amplificación. Estas señales son adquiridas haciendo uso de una tarjeta Lab PC 1200 de la National Instruments, se calcula el estado de dos señales senoidales, la subrutina puede ser vista en la dirección web:

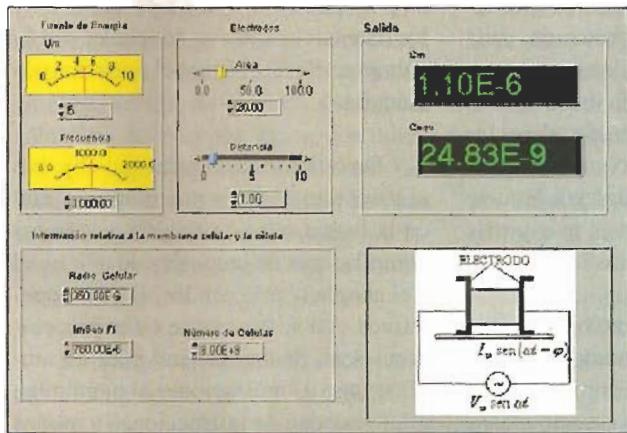


Figura 12. Simulador realizado en LabVIEW

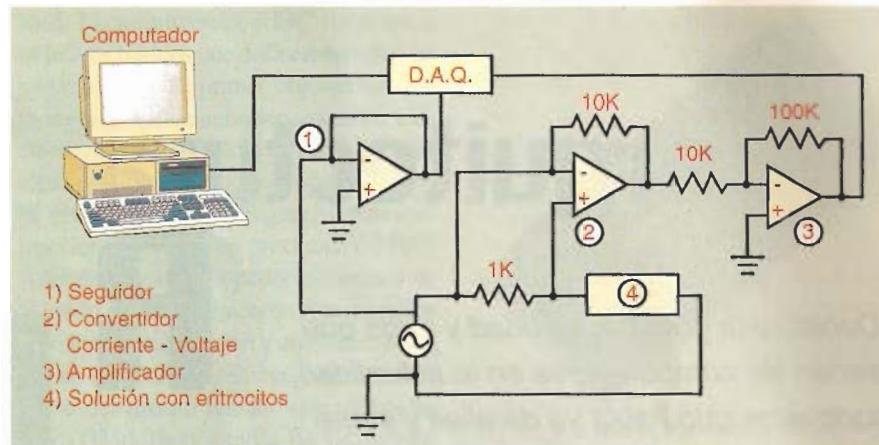


Figura 11. descripción del circuito conectado a la tarjeta de adquisición de datos

<http://ohm.utp.edu.co/fiepages/labs/fp.html>

Es de anotar que el modo en que se adquieren las señales fue el diferencial, el número de muestras por segundo fue de 1,000, y la velocidad de muestreo por canal de 10,000 muestras por segundo. Número de muestras y velocidad suficientes para hacer el análisis correspondiente.

Para la realización del proyecto se contó también con un simulador realizado en LabVIEW para el cálculo de la capacidad de los glóbulos rojos, figuras 12 y 13.

Capacitancia parásita

La interfase entre el electrodo y el líquido forma capacidad parásita del orden de 3 microfaradios, para eliminar esta capacidad parásita se intercalan entre el líquido y el electrodo dieléctricos delgados de espesor de 2 milímetros, logrando resultados bastante aceptables, tabla 1.

Conclusiones

Es posible mediante técnicas experimentales sencillas medir la capacitancia dieléctrica de la membrana celular, la cual es del orden de 0.8 microfaradios por centímetro cuadrado. Podemos pensar que las células de la sangre son un buen dieléctrico, comparable con los dieléctricos industriales como mica, aceite, etc. Sin embargo la experiencia recogida por los autores pone de manifiesto la dificultad en la manipulación de células vivas dada su fragilidad. La capacitancia de la membrana es independiente de la frecuencia de excitación de la fuente de alimentación. Q

Bibliografía

- [1] José María Ferrero. Bioelectrónica. Señales Bioeléctricas. José M. Ferrero Corral. Universidad Politécnica de Valencia.
- [2] Guyson, Fisiología Médica. Mc Graw Hill. 8^a Edición, Capítulo 2, pp 12, 13.
- [3] Vernon B. Mountcastle. Fisiología Médica. The CV. Mosby Company. Saint Louis. Mo. USA, 2^a Edición Española., pp 17,21; pp 4 – 8; pp 17 – 21.

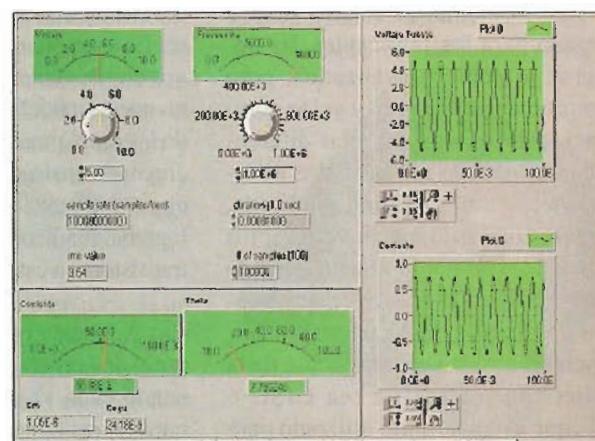


Figura 13. Simulador realizado en LabVIEW

La arquitectura

Debido a la gran popularidad y auge que tienen las computadoras en la actualidad, todo el mundo habla ya de ellas y sobre sus especificaciones técnicas, cuya parte más significativa es el microprocesador y su velocidad de proceso en MHz. Otra de las características más importantes, pero poco conocida por los usuarios comunes, es su arquitectura de la cual hay dos tipos principales: la CISC y la RISC, tema sobre el cual se ocupa este artículo.

Los microprocesadores son los dispositivos semiconductores o circuitos integrados más complejos y existen en varias marcas, modelos y clases, siendo unos más populares que otros. Todos hemos escuchado al menos nombrar los procesadores Intel 486 y Pentium o los Motorola M68000 y PowerPC. Sobre un microprocesador se pueden decir muchísimas cosas, ya que es un dispositivo tan complicado y extenso como se quiera a la hora de analizarlo o conocerlo en su totalidad.

Anteriormente, este tema estaba reservado para los fabricantes, los ingenieros y técnicos avanzados; pero como producto de la gran y veloz competencia permanente entre las diferentes marcas, con la publicidad e información técnica ampliamente publicada, se han popularizado muchos conceptos de los cuales se habla mucho pero de algunos de ellos no se conoce su verdadero significado e implicaciones reales. Un ejemplo de esto es la mención de la arquitectura interna ya sea CISC o RISC, que es un término utilizado para tratar de mostrar que un procesador puede ser mejor que otro de cierta for-

ma. Cuando se habla del término CISC o RISC, se refiere a la arquitectura o estructura general interna sobre la cual está diseñado un microprocesador, es decir, su forma de construcción y operación internas y cómo afecta esto su desempeño u operación.

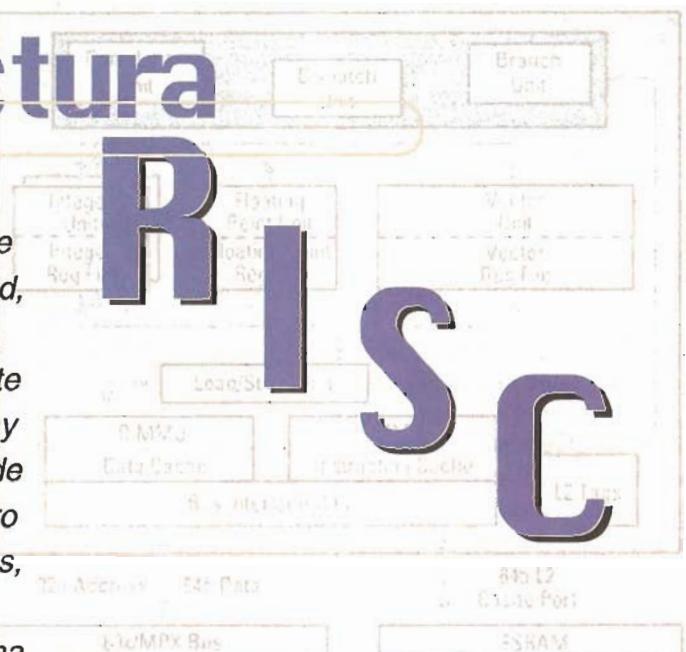
Cómo surge la tecnología RISC

En un principio, no se hablaba de RISC ni de CISC (Complex Instruction Set Computer) o Conjunto Complejo de Instrucciones para Computador, porque simplemente todo se hacía de una misma forma en cuanto a los procesadores se refiere. En este punto de la evolución, cada vez que se aumentaba el desempeño con un nuevo modelo, se hacía incrementando la complejidad de los procesadores incluyendo instrucciones complejas para ejecutar operaciones complejas. Esto se lograba añadiendo cada vez más y más transistores a estos circuitos integrados.

A medida que la tecnología lo permitió, los procesadores fueron evolucionando cada vez más permitiendo más transistores en un mismo espacio creando así varias generaciones de procesadores basados en una familia básica conser-

vando la compatibilidad y trayendo mejoras y nuevas instrucciones que mejoraron el desempeño. Por ejemplo, en la familia Intel, los modelos 8080, 8085, 8086, 80186, 80286, 80386, 80486, Pentium III, etc., figura 1, y en la marca Motorola el 68000, 68020, 68030, 68040 y 68060. Cada vez que salía al mercado una nueva generación, surgían beneficios en el desempeño de tal forma que el set o conjunto de instrucciones de estos procesadores, se acercaba cada vez más a los lenguajes de alto nivel, reduciendo así la brecha existente entre el lenguaje de máquina o assembler, el código de alto nivel y los sistemas operativos, los cuales son finalmente el enlace entre el operador y la máquina.

De esta forma, el principal objetivo al tener uno de estos procesadores, está en la buena relación que tenga con los compiladores de lenguajes de alto nivel y el acoplamiento con los sistemas operativos. Así se busca que estos procesadores sean de uso general para un amplio rango de aplicaciones al incluir una gran variedad de instrucciones y modos de direccionamiento. En la figura 2 podemos ver un procesador UltraSparc, que



FELIPE GONZÁLEZ SÁNCHEZ

felipecgonzalez@mail.com

se utiliza en estaciones de trabajo SUN, uno de los microprocesadores de más alto rendimiento hasta la fecha. El análisis del código generado por un compilador en un procesador CISC, arroja resultados impresionantes, revelando que el 80 % de las instrucciones generadas por el compilador corresponden a un 20 % de las instrucciones que posee el procesador; esto se conoce como la regla 80/20. Es bastante obvio que si este 20 % de las instrucciones del procesador se optimizan y aceleran, los beneficios obtenidos serían bastantes. El análisis de estas instrucciones también reveló que eran instrucciones simples y que se utilizaban los modos de direccionamiento más sencillos; esto puso en duda entonces todo el trabajo y esfuerzo realizado durante años para desarrollar procesadores que redujeran la carga de los compiladores.

Otra implicación de esta situación es que si sólo las instrucciones más simples eran las que se necesitaban, entonces el hardware interno del procesador que se usaba podría ser reducido en cuanto a complejidad; de esta forma se podría diseñar un microprocesador superior con menos transistores y por lo tanto a menor costo y un menor consumo de energía, uno de los principales problemas de los microprocesadores. En base a esta premisa, se diseñó una nueva arquitectura llamada RISC (Reduced Instruction Set Computer), o Conjunto Reducido de Instrucciones para Computador. Un procesador de este tipo, con un conjunto de instrucciones más simple, podría entonces ejecutar instrucciones en un sólo ciclo de reloj y realizar operaciones complejas usando varias de esas instrucciones simples. Siendo así, se ganaría mayor velocidad con respecto a un procesador CISC, ya que aunque fueran más instrucciones, en el nuevo tipo se tardaría máximo un sólo ciclo de reloj por instrucción mientras que un procesador CISC puede tardar hasta 10 o más ciclos de reloj en una sola instrucción. Esta es la principal diferencia entre las dos arquitecturas con las cuales se fabrican los procesadores.

El crédito para la primera compañía en definir y usar la arquitectura RISC se lo lleva IBM, que lo hizo en la década de 1970 y la investigación continuó luego en las universidades de Berkeley y Stan-

ford. La arquitectura RISC tiene unos objetivos básicos que definen este tipo de procesadores. El primer objetivo es que todas las instrucciones tienen que ser ejecutadas en un sólo ciclo de reloj, para así alcanzar un máximo desempeño ya que se van a ejecutar por lo general más instrucciones que en un procesador CISC. Además de esto, el opcode o formato de la instrucción, que es como el procesador reconoce la instrucción y que tiene la forma de una palabra binaria, debe ser menor o del mismo tamaño que el bus de datos (Data Bus) y todas las instrucciones deben tener la misma longitud de palabra, es decir, el mismo número de bits; de esta forma no se aceptan modificadores para la instrucción ni operadores adicionales lo que causaría un aumento de ciclos de acceso a la memoria, aumentando el tiempo de ejecución.

En un procesador CISC, los códigos de operación de las instrucciones pueden ser de diferente tamaño, lo cual complica el ciclo de decodificación de la instrucción aumentando el tiempo, al igual que cuando la instrucción tiene operandos adicionales; por ejemplo cuando se utiliza una misma instrucción con diferentes modos de direccionamiento. De esta forma el procesador lee primero el código de la instrucción, luego el modificador y finalmente la información adicional para usar el modificador, tomándose un buen tiempo en la sola etapa de decodificación de la instrucción, aumentando el tiempo total de ejecución de cualquier proceso. Recordemos que todos los microprocesadores tienen un ciclo básico general de funcionamiento que puede observarse como un ciclo de tres pasos básicos: la primera o FETCH, que es cuando se busca la instrucción y entra al procesador, la segunda o DECODE cuando el microprocesador analiza el código y reconoce qué tipo de instrucción es y por último, la tercera o EXECUTE que se encarga de ejecutar la instrucción.

Otro objetivo y característica de un procesador RISC, es que la memoria sólo se accede con dos instrucciones, LOAD y STORE. Esta condición es necesaria para poder mantener la anterior característica de ejecutar en un ciclo de reloj la instrucción. En un procesador RISC, los



Figura 1. Microprocesador Pentium III

datos de memoria que se necesitan están almacenados previamente en un registro interno que luego envía el resultado a la memoria principal fuera del procesador. Esto toma como mínimo tres instrucciones; de esta forma se mantiene la regla de un ciclo de reloj por instrucción así sean más, pero trae como consecuencia que los procesadores RISC requieren muchos registros internos de uso general para poder mantener los niveles de desempeño apropiados. El tercer y último objetivo consiste en que las diferentes unidades de ejecución deben tener conexiones por hardware y no usar microcódigo, ya que este requiere múltiples ciclos de reloj para cargar las secuencias que envían la instrucción a las unidades de ejecución y por lo tanto no se podrían implementar unidades de ejecución de un sólo ciclo de reloj.

Popularización de las máquinas RISC

A pesar de que los conceptos de la tecnología RISC surgieron y se empezaron a aplicar durante la década de los 70, fue la reserva o cautela del mercado la que hizo que sólo estaciones de trabajo y máquinas



Figura 2. Procesador UltraSparc III

especializadas utilizaran microprocesadores basados en la arquitectura RISC. El final de la década de los 80 y principios de los 90, fue una época bastante difícil para los fabricantes y usuarios de procesadores CISC y RISC de alto desempeño, ya que aunque RISC prometía mejor desempeño a un precio menor y con consumos de potencia más bajos, el mercado masivo ya estaba inundado por familias como la Intel x86 y la Motorola 680x0. En ese entonces era muy difícil predecir cual iba a ser la arquitectura dominante en el futuro o si realmente iba a ocurrir una migración del mercado masivo de CISC a RISC, ya que inclusive en los planes de Motorola e Intel se contemplaba la evolución de estas familias a niveles muy sofisticados donde no había un límite en cuanto al aumento de desempeño con los procesadores que se tenían planeados.

El estancamiento del RISC era muy visible, ya que no existía ninguna compañía con altos volúmenes de ventas que apoyara esta tecnología para asegurar una difusión suficiente y hacer rentable su desarrollo y continuidad. El mercado estaba demasiado desbalanceado; mientras que los procesadores RISC se vendían en una cifra alrededor de cientos de miles de procesadores, los procesadores CISC se vendían en cifras de millones, durante el mismo período.

La única solución a este inconveniente era que alguna compañía con

participación suficiente en el mercado de las PCs usara este tipo de procesadores. De todas maneras estos procesadores tienen un mercado significativo en aplicaciones de sistemas de control embebidos, ya que son muy populares en este campo. En cuanto al mercado de las PCs, existía un gran problema, ya que en esta época están bastante definidas por lo menos las tendencias en cuanto al mercado masivo se refiere. La migración de máquinas basadas tradicionalmente en CISC hacia una arquitectura RISC tendría muchas implicaciones, ya que el software anterior no es compatible, siendo esto un problema muy grave y difícil de resolver ya que tendría que emularse por software para mantener la compatibilidad con aplicaciones anteriores, solución que no daría la misma velocidad de ejecución para aplicaciones antigua.

Algunos procesadores RISC

Comercialmente, los fabricantes con mayores ventas, es decir Intel y Motorola, fueron desarrollando procesadores RISC. Intel sacó al mercado el primer procesador que tenía algo de operación superescalar. Este procesador, el i860 permitía la ejecución simultánea, bajo ciertas circunstancias, de una instrucción de enteros y otra de punto flotante. Esto hacía que se pudieran ejecutar hasta cien millones de instrucciones en un segundo con un reloj de 50Mhz, pero todo con una dependencia que limitaba

el uso de esta característica. El problema estaba en que la doble ejecución no era controlada por el hardware, sino por el compilador, de forma que para ejecutar dos instrucciones simultáneamente, se necesitaban conectar una instrucción entera y otra de punto flotante de forma que el par entrara al procesador donde se despachaba cada una a la unidad de ejecución apropiada.

El punto está en la habilidad del compilador para predecir cuando es seguro ejecutar pares de instrucciones de este tipo, y además se necesitaba reorganizar el flujo de instrucciones para crear el par apropiado, de forma que si no se podían organizar, habría que esperar usando instrucciones nulas (NO-OP) para prevenir la doble ejecución. Esto hacía bastante difícil el desarrollo del compilador ya que no siempre el funcionamiento era óptimo en todos los casos así que se obtenían grandes diferencias entre el desempeño máximo y promedio del procesador. Por el lado de Motorola, el primer intento de una familia RISC, fue la serie 88000, que sólo hasta su segunda generación, el 88110, logró operación superescalar incondicional sin el problema presentado por el Intel i860, ya que podía ejecutar varias instrucciones por ciclo de reloj, sin la necesidad de hacer una reorganización en el flujo de llegada de estas al procesador. Este procesador tenía 1.3 millones de transistores, fabricado en tecnología CMOS de 0.8 micrones, 10 unidades de ejecución repartidas en múltiples unidades de enteros, punto flotante y de manejo gráfico, un bus de datos de 64 bits. En la figura 3 podemos ver su diagrama simplificado.

Desde un punto de vista tecnológico, la familia 88000 se consideraba como la mejor arquitectura RISC desarrollada hasta el momento, pero no tuvo gran éxito en cuanto a la acogida que tuvo en la industria de los computadoras de consumo masivo, todas sus esperanzas estaban puestas en que podría ser adoptado por Apple Computer, para ser el procesador de la siguiente generación de los computadoras Macintosh. Estas esperanzas se enterraron completamente en 1991 cuando se anunció una alianza entre Apple, IBM y Motorola para desarrollar una nueva familia de procesadores Ila-

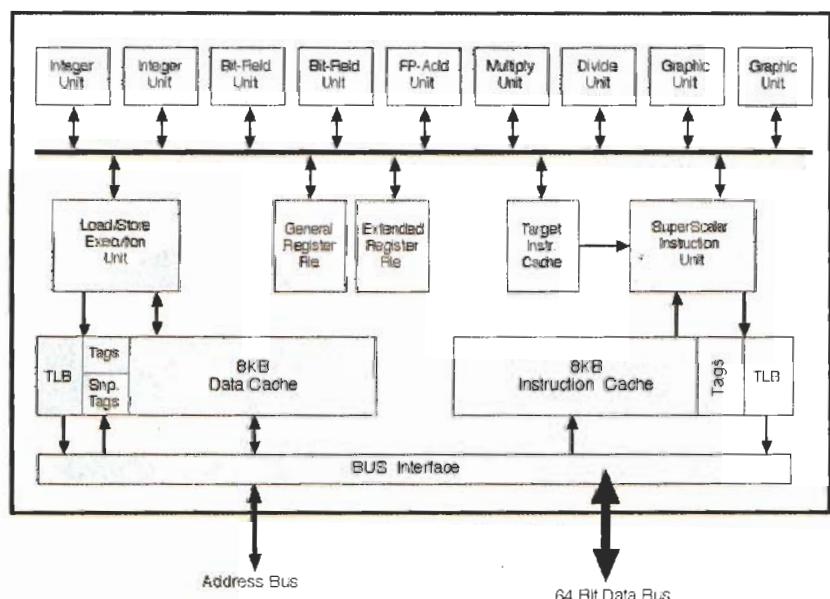


Figura 3. Diagrama simplificado de un microprocesador RISC

mada PowerPC, que se basaría en un principio en la arquitectura POWER de IBM, que era utilizada en las estaciones de trabajo como la IBM RS/6000.

Para Motorola esta alianza era una segunda oportunidad para dominar el mercado RISC, ahora con la garantía de dos fabricantes de PCs con suficiente participación en el mercado lo que aseguraría una amplia difusión. El desarrollo de la familia M88000 fue cancelado rápidamente después del anuncio de la alianza dejando al M88110 como el segundo y último miembro fabricado de esta familia. Esta alianza traía también beneficios para Apple, que venía basando sus computadoras Macintosh en la familia Motorola 680x0, donde aparentemente ya empezaba a haber problemas con respecto a aumentos de desempeño de forma significativa dejando así al 68040 como el último miembro de esta familia en ser utilizado como procesador de los Macintosh.

Hubo estudios en ese entonces para incorporar al 68060 para crear una nueva generación de equipos Macintosh pero se llegó a la conclusión de que no se debía continuar con la arquitectura M68000 y migrar de una vez a una nueva tecnología diferente como lo era la RISC, con el PowerPC, a pesar de que iba a ser un problema mantener la compatibilidad con todo el software ya existente para la familia M68000 con más de una década de desarrollo. Inclusive las primeras versiones de procesadores PowerPC que fueron utilizadas en Macintosh, presentaban facilidades y opciones por hardware que permitían la emulación de un 68040 para poder brindar compatibilidad con el software anterior.

Esta fue una transición bastante difícil, pero se llevó a cabo de la manera menos impactante posible, de tal forma que si se quería aprovechar al máximo el nuevo procesador PowerPC, el software debería ser recompilado sobre este procesador para así notar la superioridad del desempeño con respecto a la familia M680x0.

La arquitectura PowerPC incluye muchas instrucciones heredadas de la arquitectura POWER de IBM, que pueden ser ejecutadas sin modificación alguna, haciendo al PowerPC parcialmente com-

patible con la plataforma POWER, de forma que el resto de instrucciones no soportadas por el PowerPC se emulan en software de manera similar al caso del 68040. En la figura 4 podemos ver un diagrama que muestra la evolución de la arquitectura RISC de IBM, desde que nació el primer procesador RISC, el IBM 801, cuya evolución dio para dos procesadores de familias diferentes, el RT, que tuvo una duración muy corta y por otro lado el RS/6000 que tuvo mayor éxito.

El procesador PowerPC fue producto de aportes tecnológicos de Apple, IBM y Motorola y gran parte de la implementación de hardware fue heredada de las ideas del M88000 como por ejemplo las interfaces de los buses de los PowerPC 601 y 603. El desarrollo del PowerPC tiene un modelo diferente comparado con otras familias de procesadores, ya que ha sido ideado de tal forma que con un modelo simple, se asegure la compatibilidad de los futuros miembros de la familia. Así los desarrolladores de software no tienen que preocuparse por esto, lo que sucedía con la serie M68000 donde se daban pautas que servían como indicaciones de lo que debería ser implementado el software para mantener la compatibilidad con generaciones futuras. En la figura 5 se puede observar el diagrama de bloques internos del PowerPC 7400 que es el procesador más reciente de la familia PowerPC.

Actualmente la familia PowerPC es tal vez la familia RISC más popular y con mayor difusión y ha evolucionado bastante pasando por los procesadores 601, 603, 604, 750 y el más reciente, el 7400 que es el que presenta mayor velocidad y es la CPU de las últimas computadoras PowerMacintosh de Apple. Otro campo muy extenso de aplicación de los procesadores PowerPC, está en el campo de control y automatización donde también es ampliamente utilizado.

Con este artículo se pretendió dar una idea real de lo que es realmente un

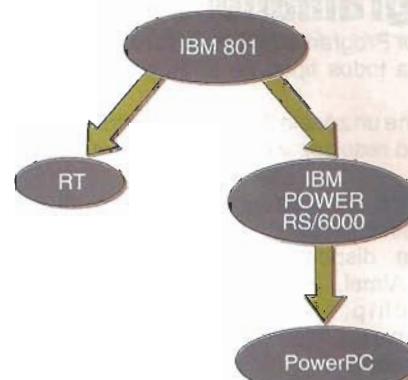


Figura 4. Historia de la arquitectura RISC de IBM.

procesador RISC y sus principales diferencias con los procesadores CISC, así como también hacer una breve descripción de los procesadores más populares desde un punto de vista muy general. Como conclusión, se puede decir que un procesador RISC hace que se realce más la relación entre el software y el hardware, de forma que los compiladores son una pieza fundamental para el desarrollo de los sistemas operativos y las aplicaciones. Además, como en un procesador RISC, el código tiende a ser más extenso, a los programadores se les dificulta escribir directamente en ensamblador, en vez de un lenguaje de alto nivel. Esto se aplica también al proceso de depuración que se hace muy complicado en assembler con un código tan largo ya que un objeto compilado en un procesador RISC es entre el 20% y 40% más largo que el código para un CISC. \square

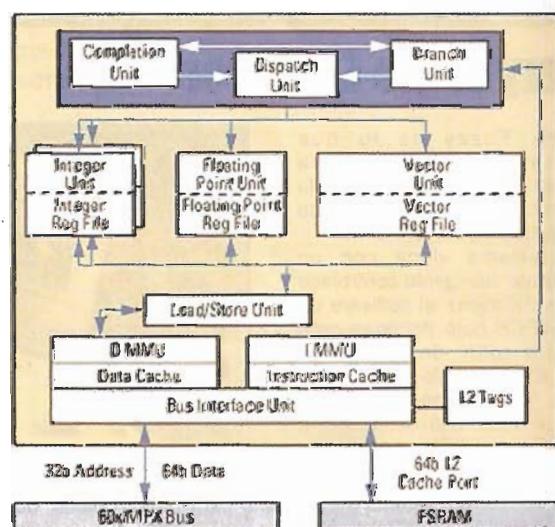
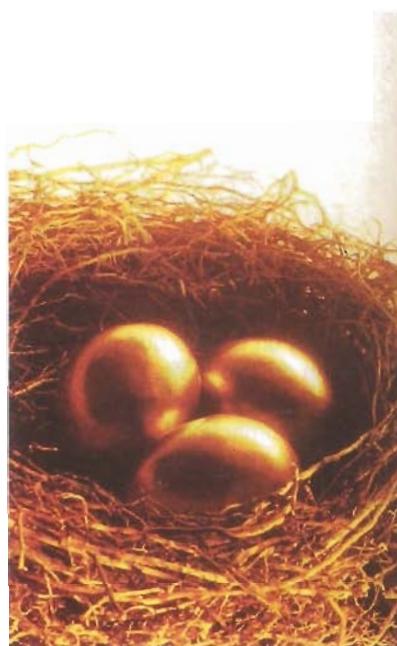


Figura 5. Diagrama de bloques del microprocesador PowerPC 7400

Qué hay de nuevo en...



En esta sección hacemos referencia a aquellos dispositivos relacionados con la electrónica y las computadoras que, además de ser novedosos, llaman la atención por sus características y servicios prestados

Robot - Aspirador Cye



Cye-SR es un pequeño dispositivo que se desplaza en forma autónoma por medio de un radio control conectado a una PC. Tiene una fuerza de 7lbs y si se le adiciona una pequeña aspiradora, podrá ejecutar labores domésticas previamente programadas en una PC distante. Otra de sus aplicaciones consiste en el transporte de pequeños objetos a través de la oficina u hogar, simplemente se le programa la ruta desde una PC y listo.

Más información en :
www.personalrobots.com/home.html

Microporcesador Celeron Mobile de 500 MHz para portátiles

Este nuevo microporcesador, además de ganar en velocidad de proceso, permite el ahorro de dinero gracias a su bajo costo. Recordemos que la línea de Intel Celeron ha sido diseñada para competir en costo con los microporcesadores de AMD. Así mismo, la línea Mobile, que se ha caracterizado por ser para equipos portátiles, ha querido utilizar las propiedades del Intel Celeron.

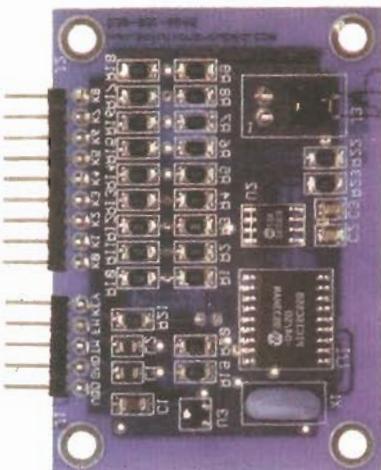
Más información en:
<http://developer.intel.com/design/mobile/celeron/index.htm>



Módulo programable para teclado

Este es un módulo programable para la conexión de teclados en forma de matriz. Cuando se le conecta la matriz de teclas, el dispositivo detecta la que se ha oprimido y envía a través de sus pines de salida un código que puede ser programado de acuerdo a la exigencia del proyecto. Su modo de comunicación puede ser serial a 2400 bps. Adicionalmente posee 64 bytes de memoria EEPROM para almacenamiento de datos. Su forma física permite una fácil conexión a circuitos impresos o a tableros experimentales.

Más información en:
www.solutions-cubed.com

**Computadora compacta eOne de tipo PC**

Esta es una computadora con apariencia similar a la famosa iMAC de Apple; la diferencia es que pertenece a la plataforma PC. Funciona con microprocesador Celeron de 433 MHz y posee 64 MB de memoria RAM. Su capacidad de almacenamiento en el disco duro es de 6.4 GB. A pesar de su tamaño reducido, la pantalla es de 15 pulgadas e incluye sistema de audio con altavoces estereofónicos, CD-ROM de 24X, unidad de disquete de 1.44 MB, fax módem y puertos USB.

Más información en:
www.e4me.com/infocentral/product_eone500.html

Impresoras para etiquetas

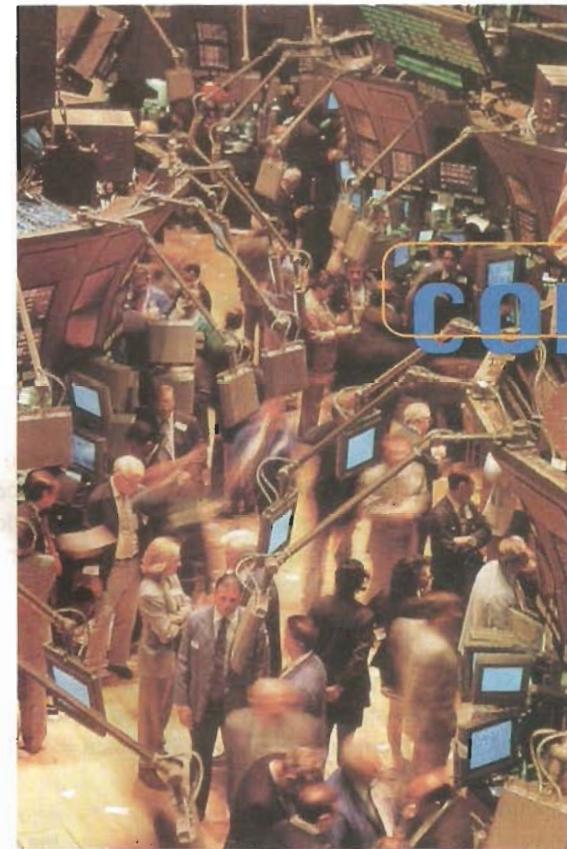
Este tipo de impresoras ofrece como sus mejores características su tamaño y apariencia externa. Su forma aerodinámica la convierte en un estilo futurista y agradable en el entorno donde se ubique. Su funcionamiento se basa en los principios térmicos, lo que hace que sea completamente silenciosa. El software que se incluye con la impresora posee programas con herramientas de diseño para la impresión de etiquetas; además permite el almacenamiento de los estilos más usados con su respectivo nombre.

Más información en:
<http://www.dymo.com/>



Fundamentos de redes de computadoras

Parte 1



FELIPE GONZÁLEZ SÁNCHEZ
felipecgonzalez@mail.com

Prácticamente desde el principio de la historia de las computadoras, se tenía en mente la idea de comunicarlas entre sí para el intercambio de información y para compartir procesamiento y recursos externos. Hoy, las redes ya se utilizan en todo tipo de organizaciones, desde pequeñas empresas hasta el hogar. En este artículo veremos algunos conceptos de redes con la evolución del hardware, el software y los nuevos medios de comunicación.

Los cambios y avances en el campo de las comunicaciones han permitido una fusión con el mundo de las computadoras, donde el modelo de la sala de cómputo central que existía antes ha ido desapareciendo, y ahora en cambio de una gran computadora que haga todo, se tiene un número apreciable de máquinas separadas, que conectadas entre sí, realizan el mismo o un mayor trabajo. En este tipo de redes, cada computadora es autónoma en sus tareas, es decir no está controlada por otra computadora, como en el modelo inicial maestro/esclavo con terminales brutas que son simplemente módulos o periféricos de un computadora central, figura 1.

Así, las computadoras de una red tienen la habilidad de intercambiar información con otras partiendo de una interconexión física, que puede ser a través de diferentes medios como por ejemplo, alambre de cobre, fibra óptica, enlaces de microondas o satelitales o una mezcla de todos estos medios con diferentes jerarquías como ocurre con las redes de área extensa o WANs que mencionaremos más adelante. Existe otro concepto clave que puede prestar-se para confusiones que tiene similitud con la definición de redes de computadoras y es lo que se denomina un sistema distribuido o DCS (*Distributed Computer System*), en el cual hay una

configuración donde la existencia de un grupo de computadoras autónomas, es transparente al usuario.

Por ejemplo, puede haber un sistema de multiprocesamiento, en donde un usuario corre un programa y el sistema operativo se encarga de ejecutarlo en alguna parte del hardware dependiendo de la disponibilidad de recursos. La tarea es transparente al usuario, tal como si fuera una sola computadora. Este esquema se utiliza mucho hoy en día para sistemas de control, de comunicaciones y para servidores con funciones trascendentales; de tal forma que se tenga redundancia en un sistema y si

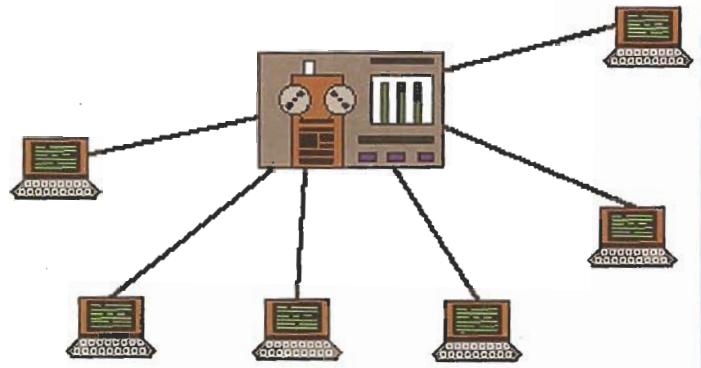


Figura 1. Modelo antiguo donde muchas terminales brutas se conectan a una máquina central.

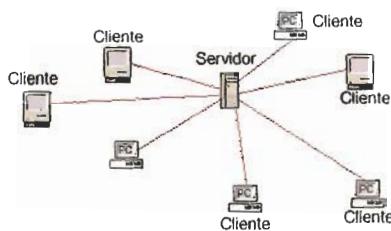


Figura 2. Modelo Cliente-Servidor

algo falla, se ejecuten automáticamente las operaciones en otro lugar, tarea que como ya dijimos, corresponde a un sistema operativo, que se encarga de gestionar el transporte de recursos entre los diferentes módulos de un sistema distribuido. Otro objetivo de un sistema distribuido es tener mayor desempeño, claro está dependiendo de la aplicación. Por ejemplo, muchos servidores Web de altísimo tráfico, utilizan este sistema para mantener al máximo posible la seguridad y el rendimiento.

En una red común, los usuarios tienen que escoger los recursos que desean compartir, todo bajo una jerarquía de permisos ya establecida, y también deben ordenar manualmente las tareas que se deseen ejecutar en diferentes puntos de la red, donde cada punto se identifica de alguna forma, como por ejemplo con una dirección. La gran diferencia entre un sistema distribuido y una red de computadoras, se ve reflejada en el software y no en el hardware, ya que se puede tener un conjunto de computadoras interconectadas entre sí actuando como un DCS, donde el sistema operativo se encargaría de todo haciendo que todos los procesos sean transparente-

tes al usuario; mientras que en el otro extremo se puede tener el mismo hardware y el funcionamiento es totalmente diferente.

Gracias a la incorporación masiva de las computadoras en las empresas y organizaciones y a la proliferación de medios de comunicación a bajo costo, se ha incrementado notablemente el crecimiento de las redes, ya que es una necesidad que surge al querer interconectar, por ejemplo diferentes sucursales o recibir información de una oficina central ya sea para cualquier aplicación que involucre el procesamiento de volúmenes de información significativos, todo haciéndose de igual forma si dos computadoras están a tan sólo unos pasos o a miles de kilómetros. Otra gran ventaja de las redes es que la comunicación entre las personas se puede simplificar considerablemente a un costo muy bajo, ya que se utilizan aplicaciones como el correo

electrónico, donde la comunicación es casi instantánea sin importar la distancia.

Modelo Cliente-Servidor

En los tipos de redes las tendencias han cambiado drásticamente. Antes se utilizaban Mainframes o computadoras muy grandes, poderosas y costosas, mientras que ahora se utilizan muchas computadoras «pequeñas» para repartir el trabajo, principalmente debido a que estas son más económicas. De cierta forma se mantiene el modelo de un computadora central o servidor cuya carga es muy baja comparada con las

tendencias antiguas utilizando por ejemplo para compartir o difundir información común a cierta cantidad de computadoras de la red a las que se les llama clientes, cuyos intereses son similares en cuanto a la información se refiere. Este tipo de aplicación se conoce como el modelo cliente-servidor y se puede observar en la figura 2, donde el servidor, además de repartir información, también se puede usar como ejecutor de tareas, y luego de que se tenga el resultado, este es enviado a las computadoras que hicieron la solicitud de ejecución.

Con el modelo cliente-servidor, se tiene otra ventaja sobre la estructura antigua del *Mainframe* o centro de procesamiento. La gran ventaja se conoce con el término de escalabilidad, donde a medida que crecen los requerimientos de procesamiento o demanda de mayor manejo de información, se necesitaría cambiar el *Mainframe* por uno de mayor capacidad lo cual implica un costo altísimo y en muchos casos problemas de compatibilidad y necesidad de nuevo software.

En el modelo cliente-servidor, la escalabilidad se da simplemente añadiendo más microcomputadoras, como más clientes y más servidores de forma transparente, ya que simplemente consiste en añadir elementos del mismo tipo al sistema conjunto ya existente y sin límite alguno permitiendo un crecimiento que satisfaga las necesidades que vayan surgiendo en cada organización.

Difusión masiva de las redes

Durante los años 70s y 80s este avance tecnológico era exclusivo de empresas muy grandes o de comunidades con un alto poder adquisitivo; fue entonces hasta que las computadoras personales lograron la integración suficiente con el mundo de las comunicaciones y se logró una utilización masiva del concepto de red, gracias al acceso de información remota que se podía tener desde el hogar o desde una pequeña empresa por ejemplo, con sólo usar un módem, o las comunicaciones entre personas que estuviesen en sitios muy lejanos, y también viéndolo como una

forma de entretenimiento interactivo que ha logrado que el interés de las masas cree el impulso tecnológico que hoy hace a Internet, por ejemplo, la más grande red de todos los tiempos y el fenómeno tecnológico con el que se terminó el milenio.

Tipos de redes

Con todo el boom que existe sobre este tema, sería muy extenso enumerar todos los tipos de topologías de redes que existen actualmente, pero sí se pueden clasificar los principales bajo ciertos parámetros; básicamente el tamaño o área de cubrimiento y la tecnología que utilizan. En cuanto a tipos de transmisión, se puede hablar de dos sistemas básicos: las redes de difusión masiva (*Broadcasting Networks*) y las redes de comunicación entre dos puntos fijos (*Point to Point Networks*).

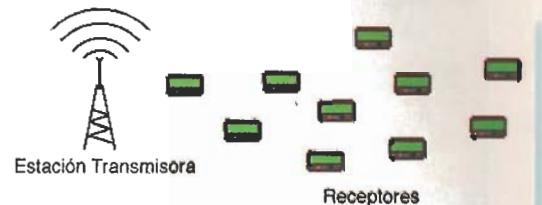
Redes de difusión masiva (*Broadcasting Networks*)

Este tipo de redes se basa en un concepto simple, pero eficaz, ya que se compone básicamente de dos elementos: un transmisor y muchos receptores que comparten el mismo medio de transmisión. Partamos de un ejemplo simple como la radio o la televisión donde todos reciben la misma información.

En un ámbito más aplicado a nuestro tema, una computadora envía varias tramas (conjuntos de bits) que representan cierta información que podremos llamar «paquete». En este modelo, se envía un paquete y todas las computadoras lo ven pasar; cada paquete tiene un trozo de información que indica una dirección destino, de forma que cada computadora tiene su dirección asignada que la identifica como única en la red.

Con más detalle, el paquete contiene un campo que indica la dirección y todos los clientes comparan este campo de la trama o paquete con su dirección hasta que el dueño de ésta lo identifica y procesa el resto del paquete de cierta forma previamente establecida por lo que se quiera hacer. El caso de un *beeper* o buscapersonas es exactamente el mismo, donde el medio que

Figura 3. Modelo de comunicación de una red de buscapersonas o beepers.



comparten todos los receptores es el aire y dependiendo del código que va en el campo de direcciones, se recibe el mensaje en el receptor apropiado como se puede ver en la figura 3.

En aplicaciones de *broadcasting* también se pueden hacer otras cosas como por ejemplo usar un código de dirección para escoger subgrupos como destino, dependiendo de lo que se deseé, permitiendo así comunicación a uno o a múltiples puntos de una red; esto es conocido como *Multicasting*.

Redes Punto a Punto (*Point-to-Point Networks*)

Este sistema se basa en múltiples enlaces entre pares de puntos de una red. De esta manera, cuando la información viaja de un punto a otro es posible que tenga que pasar por puntos intermedios para llegar a su destino final; también puede viajar por diferentes rutas para alcanzar ese destino, así que desde el punto de vista del software, se debe tener en cuenta la importancia de los mé-

todos de enrutamiento de la información dependiendo de las condiciones dadas en la red en ese momento para asegurar la mejor manera de transportar la información; este es el caso por ejemplo de Internet. Podemos ver en la figura 4 un esquema de conexión entre dos puntos remotos pasando por rutas que podrían ser diferentes dependiendo del caso.

Como mencionamos anteriormente, las redes también se pueden clasificar por su área de cubrimiento y tecnología utilizadas. Básicamente, existen tres categorías donde se pueden clasificar todas las redes como lo muestra la figura 5.

LANs (Local Area Networks)

Las redes de área local, tal vez las más populares, consisten básicamente en redes privadas pequeñas con un cubrimiento que abarca una casa, una oficina, un edificio o como máximo una universidad. Se usan para conectar grupos de computadoras que comparten

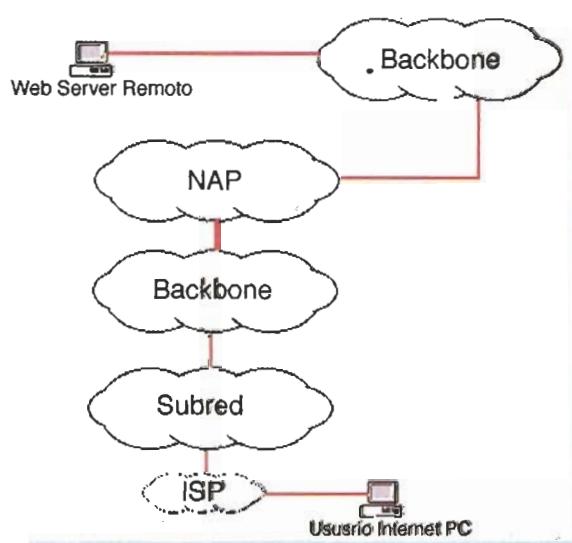
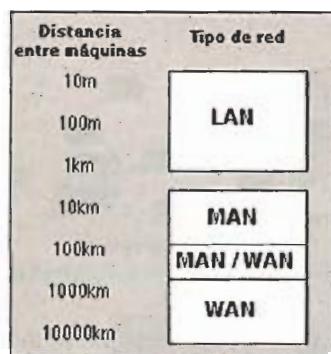


Figura 4. Esquema simplificado de la conexión de dos puntos en Internet

**Figura 5.** LANs, MANs y WANs

recursos como por ejemplo impresoras, dispositivos de almacenamiento masivo o acceso a bases de datos. Las redes de área local tienen un límite en cuanto a cubrimiento; de esta manera se sabe con certeza hasta donde se extienden, no como sucede con redes de área extensa o WANs, en las cuales no se puede conocer bien su cubrimiento. Este conocimiento permite el uso de ciertos tipos de conexión o topologías, que hacen por ejemplo que se conozcan las características de la información en los peores casos, cuando esta viaja entre los puntos más distantes. Otra ventaja de las LANs es que, debido a su tamaño restringido, se pueden administrar con relativa facilidad.

Las redes LAN más comunes tienen velocidades de transmisión de 10Mbps a 100 Mbps, tienen bajos tiempos de retardo y la tasa de error es bastante baja. Hoy en día el tipo de red

LAN más común, es la red Ethernet. También son populares las redes Token-Ring, tecnología de IBM con velocidades de 4 y 16 Mbps, y en la plataforma Macintosh, las redes AppleTalk sobre Ethernet o la versión de bajo costo y baja velocidad conocida como LocalTalk.

Existen varios modelos para la operación de redes LAN, por ejemplo en el caso donde los diferentes puntos de red comparten un mismo cable o medio de transmisión, como en la figura 6. En este caso, se asignan puntos de origen y destino, permitiendo que una máquina transmita y otra reciba mientras que todas las otras máquinas de la red deben esperar y abstenerse de involucrarse en la comunicación dejando el canal libre.

En una red Ethernet existen varias configuraciones posibles con respecto al tipo de cableado que se vaya a utilizar, ya sea 10Base5, 10Base2 o 10BaseT. Todas estas configuraciones tienen en común el mecanismo de coordinación para transmitir datos; cuando una máquina quiere transmitir, evalúa o "escucha" el medio compartido para detectar si hay tráfico; si lo hay, espera hasta que el canal esté libre; de lo contrario, inicia la transmisión instantáneamente. Puede darse el caso en el cual dos o más máquinas inicien la transmisión simultáneamente; en esta ocasión los paquetes transmitidos chocarán, por lo cual se suspenderá la transmisión inmediatamente desde ambas partes; entonces cada máquina espera un tiempo aleatorio y reinicia la transmisión para evitar que se presente la colisión otra vez y así se inicia el ciclo de solicitud del medio o canal nuevamente.

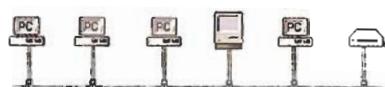
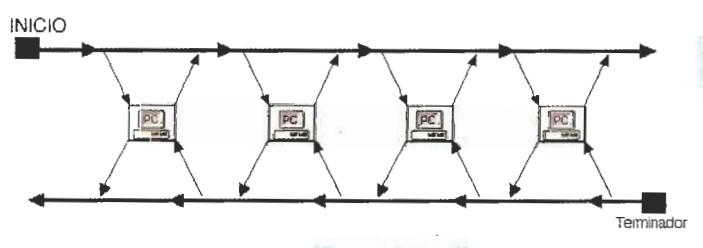
En otro tipo de red LAN, como por ejemplo la del tipo anillo, cada paquete

tiene que pasar en serie por todos los puntos de la red. Este tipo de red Broadcast, requiere también un método para asignar el uso del canal a cada uno de los miembros del anillo. Existen otros tipos de redes que utilizan técnicas que no requieren un método de asignación de canal a cada una de las diferentes estaciones de la red. Una técnica que se utiliza es usando multiplexación en tiempo, de forma que se asignan intervalos a cada estación, tiempo durante el cual puede trasmisir una máquina y se continúa la transmisión en el siguiente intervalo correspondiente, después de haber dado toda la vuelta. Este esquema es eficiente sólo en el caso donde todas las máquinas estén transmitiendo simultáneamente; de lo contrario, se desperdicia el uso del canal ya que va a existir un porcentaje de tiempo considerable donde no va a haber flujo de información puesto que los "time slots" o intervalos correspondientes a otras máquinas, no van a tener ningún tráfico.

MANs (Metropolitan Area Networks)

En el siguiente nivel de la clasificación con respecto al tamaño y área de cubrimiento, se encuentran las MANs o redes de área metropolitana. Estas redes son básicamente versiones extendidas de LANs y utilizan tecnología muy similar y dan cubrimiento a zonas extensas como por ejemplo una ciudad o conexiones entre LANs, que no estén muy distantes geográficamente. Una MAN consiste básicamente en dos líneas de transmisión que hace que se puedan extender fácil y ampliamente sobre la zona que se desea cubrir. La IEEE, ya ha estandarizado este tipo de redes, conocidas con el código 802.6 o con las siglas DQDB (*Distributed Dual Queue Bus*). La topología se conforma por dos cables unidireccionales, de los cuales se derivan las conexiones a las máquinas como se puede observar en la figura 7.

Cada cable tiene un iniciador y un terminador que vela por el flujo de datos iniciando la transmisión de estos al principio del cable. El tráfico se enruta de acuerdo a la ubicación del destino. Observando la figura, si se quiere trans-

**Figura 6.** Un sólo cable permite la conexión de varios computadoras.**Figura 7.** Modelo de conexión de una red MAN.

mitir hacia una máquina que está a la derecha, se usa la línea superior, y si es a la izquierda, la inferior; de esta manera se justifica el uso de líneas unidireccionales.

WANs (Wide Area Networks)

Las redes de área extensa son el ejemplo de qué tan lejos ha llegado el mundo de las comunicaciones hoy en día; estas tienen cubrimientos que van desde ciudades aledañas hasta países, continentes y atraviesan en algunos casos todo el planeta. Las aplicaciones de estas redes consisten principalmente en compartir datos, ejecución de aplicaciones y circuitos o líneas privadas de voz y video-conferencia, y pueden conectar un número ilimitado de máquinas. En estas redes se habla de dos tipos de comunicación: la comunicación entre las máquinas o *hosts* que es la información en sí que se desea intercambiar, y la comunicación de los aspectos de la red como tal, que se denomina subnet o subred, que cubre los aspectos operativos que hacen que la comunicación sea eficaz y posible.

Estas subredes o subnets, comprenden la parte física del enlace y se compone de dos elementos esenciales: las líneas de transmisión y los elementos o dispositivos de conmutación. Estos últimos son módulos especializados, generalmente computadoras dedicadas más conocidos como enrutadores o *routers*, que se utilizan para conectar las líneas de transmisión de forma que cuando llegan datos, el elemento de conmutación debe escoger un camino usando otra línea de transmisión para continuar el trayecto de lo que se transmitió. En la figura 8 se puede observar un diagrama que ilustra la posición de los enrutadores en una red WAN y la demarcación de la subred o subnet.

Este tipo de conexión ofrece una flexibilidad importante, ya que si por ejemplo dos enrutadores no poseen una conexión directa o se pierde el enlace por problemas del medio de comunicación, estos pueden llegar a su destino escogiendo caminos o rutas alternas, así se recorra una distancia mucho mayor introduciendo, claro está, un mayor re-

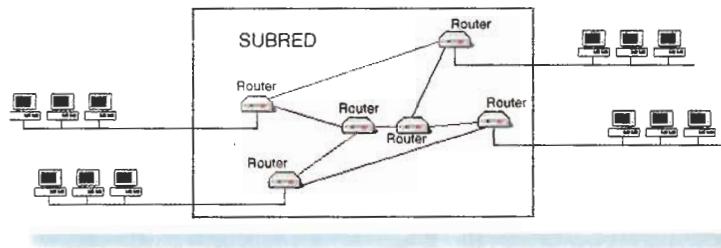


Figura 8. Red WAN. Esquema de enrutadores y conexión entre líneas de transmisión.

tardo en el flujo de la información. Otra causa que amerita el desvío de la ruta más corta es por ejemplo también el tráfico excesivo que se pueda dar en ciertas zonas de la red.

El tráfico juega un papel bastante influyente, ya que se va acumulando información de forma que queda en espera hasta que el enrutador considere que la línea de transmisión de salida está libre para enviar la información que estaba en espera. Una subred o subnet que utiliza esta técnica se conoce como red punto a punto o PSS (*Packet Switched Subnet*).

Las redes WAN poseen una característica bastante peculiar que consiste en que su crecimiento es prácticamente ilimitado y más importante aún, su crecimiento puede salirse de control con respecto a las topologías e interconexiones posibles entre enrutadores de forma que se obtienen conexiones amorfas que pueden ser de muchas formas diferentes.

En una WAN se puede encontrar una mezcla muy diversa de medios de transmisión, debido a la característica de difusión amplia y adaptabilidad que se requiere; esto hace que puedan haber conexiones que usen desde pares de cobre, pasando por fibra óptica, enlaces de microondas y satelitales, lo cual causa que ciertas partes de la red sean más rápidas que otras haciendo aparecer el fenómeno conocido como cuello de botella (*bottleneck*) donde se represa la información esperando disminución en el tráfico para poder continuar el trayecto.

Internetworks

Como vimos en la definición de las WANs, existen muchos tipos de redes,

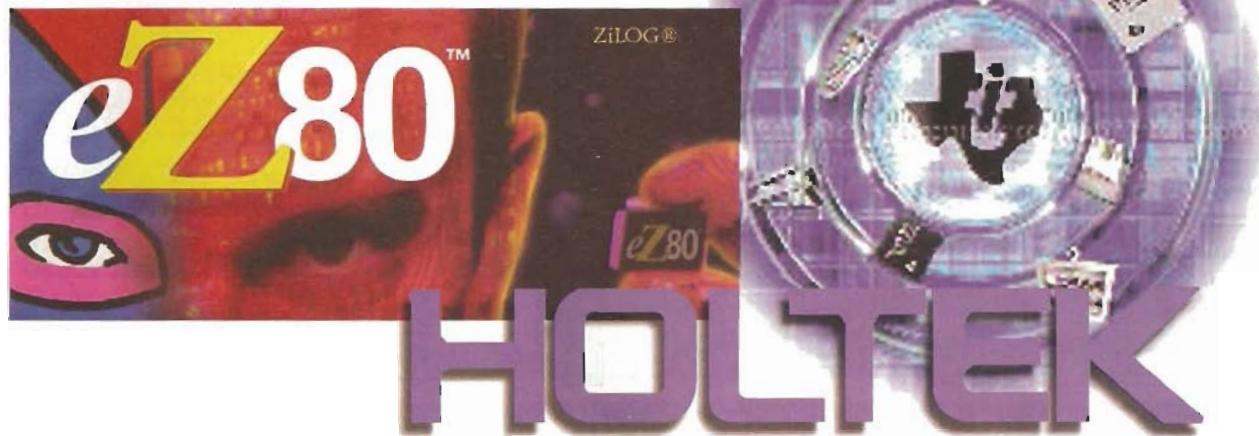
y más si lo vemos desde un punto de vista global examinando todas las posibilidades. Así entonces, hoy en día se puede encontrar una diversidad muy amplia de tipos de redes tanto como en hardware como en software, y es muy usual que estas redes entre sí sean incompatibles. Cuando se desean comunicar entre sí, como sucede ahora que todos quieren estar conectados con todos, es necesario usar un elemento bidireccional que permita hacer la traslación o traducción entre los formatos de estas diferentes redes para así permitir su comunicación.

Las máquinas que hacen esta tarea se conocen como *gateways* y se encargan de hacer la traslación a todo nivel. Un conjunto de redes interconectadas es lo que se conoce como *internetwork* y es precisamente de donde sale el término **Internet** que no es precisamente la red que todos conocemos, sino el nombre de una red con las características mencionadas previamente.

Es importante notar bien las diferencias entre una red, una WAN, Subnet, LAN e Internetwork, ya que algunas veces se presta para confusiones; por ejemplo una interconexión de redes LAN a través de una red WAN, se puede considerar como una Internetwork.

En este artículo vimos una introducción a los conceptos básicos de lo que concierne a las redes de datos, explicando los principales conceptos que sirven como base para entrar en un detalle más técnico del cual hablaremos en el próximo artículo cuando se defina la parte de software que está detrás de todas las redes, iniciando por el modelo de múltiples capas u OSI (Open Systems Interconnect) y el punto de vista de TCP/IP. ☐

Nuevos Circuitos Integrados



En esta sección destacamos las últimas novedades de las principales empresas de desarrollo en circuitos integrados que surgen en el mercado y que nos pueden ser de gran utilidad.

JUAN ANDRÉS CASTAÑO WELGOS
welgos@telesat.com.co

Nuevos procesadores DSP

De alta velocidad y baja potencia

La firma Texas Instruments, líder mundial en Procesadores Digitales de Señales ha anunciado dos nuevos circuitos DSP, El **TMS320C64x™** y **TMS320C55x™**. Ellos ofrecen lo último en cuanto eficiencia y bajo consumo de potencia. Muchos aseguran que ellos configuran la siguiente generación en aplicaciones digitales, en voz, audio, datos y video.

En realidad los nuevos DSP *C64x* y *C55x* ofrecen las más altas cualidades y más bajo consumo de potencia ofrecidos hasta la fecha en la industria, a la vez que se mantiene la compatibilidad de código con toda la actual familia de productos DSP. El *C64x* es el DSP más completo del mundo y el *C55x* es el que menos potencia consume del mercado, su diagrama de bloques se muestra en la figura 1.

El nuevo **DSPTMS320C64x™** tiene entre sus características:

- Velocidad de reloj por encima de 1.1 GigaHertz (GHz)
- La estructura de las instrucciones permiten casi 9.000 millones de operaciones por segundo (MIPS).
- Instrucciones especiales que permiten reducir el tamaño del código y aceleran el desarrollo de las aplicaciones debido a factores como corrección de errores y manipulación de bits.
- Su campo de acción incluye comunicaciones y tratamiento preciso de imágenes, por ejemplo en aplicaciones médicas, de seguridad o en TV de alta calidad.
- Soporta operaciones duales de 16-bits.

La combinación de ratas super rápidas de reloj con las nuevas características de arquitectura, hace posible que sea 10 veces más rápido que sus antecesores.

El nuevo **DSPTMS320C55x™** tiene entre sus características:

- 800 MIPS a un reloj de 400 MHz.
- Consumo de 0.05 mW por MIPS.
- El tamaño de las instrucciones permite reducir el tamaño del código en un 30 por ciento.
- Manejo automático del consumo de potencia, mediante la utilización de técnicas que desactivan los periféricos, memoria y unidades funcionales inactivas.
- Su arquitectura *Dual-MAC* permite duplicar el número de instrucciones por ciclo, al tiempo que reduce el con-

TECNOLOGIA

- sumo de potencia e incrementa su paralelismo.
- El campo de mercado incluyen aplicaciones que se alimenten con batería, como teléfonos celulares, música, fotografía y aplicaciones médicas.
 - Posee memoria caché de instrucciones lo que reduce la necesidad de accesar a memorias externas.
 - Posee una interface para memoria externa que mejora su potencialidad y flexibilidad del sistema.
 - Buses adicionales y capacidad expandida de direccionamiento.
 - El tamaño de las instrucciones de las variables incrementa la densidad del código.

El DSP C55x es una mejora al ya exitoso C54x. Soporta nuevas capacidades de programación manteniendo la compatibilidad en software con cualquier código del C54x, y reduciendo el consumo de potencia en una sexta parte. El C55x permite al

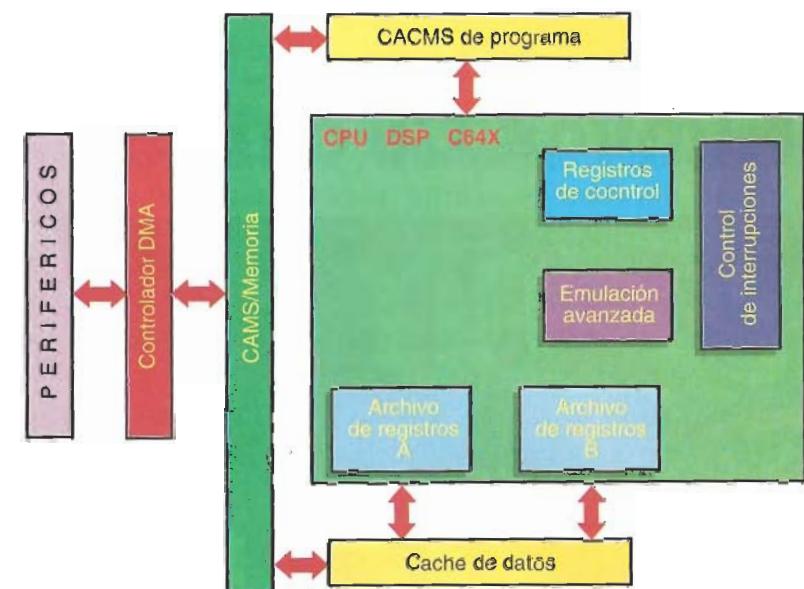


Figura 1. Diagrama de bloques del DSP C64x

diseñador todo el entorno de desarrollo de Texas Instruments. El expressDSP™ Real-Time Software Technology, soporta estos nuevos

DSP C64x y C55x proporcionando todas las herramientas y garantías que permiten iniciar rápidamente con el desarrollo de aplicaciones.

HT3670C Tambor electrónico

Este es un integrado CMOS diseñado para aplicaciones específicas que necesiten el sonido electrónico de un instrumento musical. Tiene un rango de 18 instrumentos y 36 efectos rítmicos

diferentes. Cuenta con funciones que permiten ajustar el tiempo y el volumen, al igual que 13 teclas con las cuales se puede seleccionar entre el rango de los 18 instrumentos y seis te-

clas para seleccionar los 36 sonidos rítmicos. La salida del sonido se da por cuatro canales ETS (Electronic Tone Synthesizer) además de un circuito de procesamiento de sonido de dos canales.

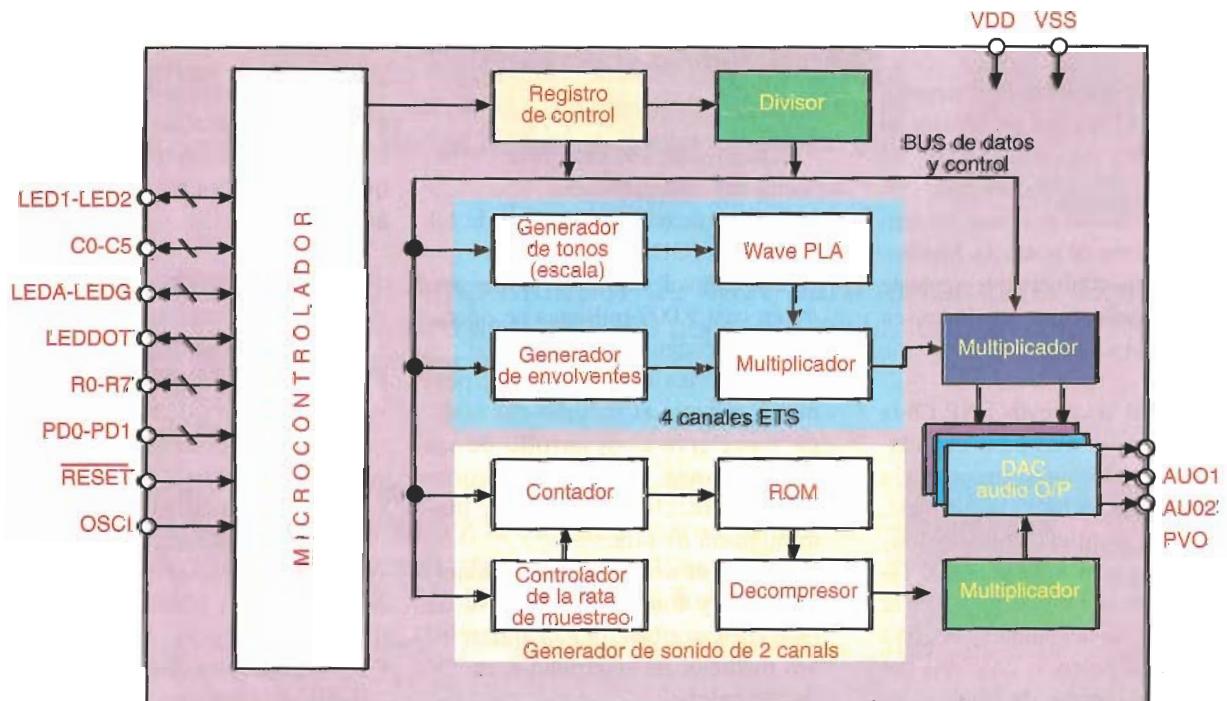


Figura 2. Diagrama de bloques del HT3670C

Con la habilidad de controlar un teclado externo y con el control total sobre los circuitos internos que procesan el sonido, la firma Holtek empaquetó su microcontrolador de 8 bits que permite el desarrollo de aplicaciones multifuncionales. Con esa característica, el dispositivo puede controlar el teclado de entrada y el sintetizador de tonos interno (ETS).

Después de recibir la acción requerida desde el teclado, el microcontrolador genera y envía el código correcto de control hacia el ETS para producir la señal de sonido requerida. Las sali-

das de sonido son dos salidas de melodías y una salida de sonido, todas manejadas por un conversor tipo D/A. La circuitería interna del integrado permite manejar dos displays de siete segmentos al igual que una conexión para

un amplificador apropiado de sonido externo. Con todas estas características podemos decir que podemos crear un sistema completo de sonido electrónico. En la figura 2 podemos ver el diagrama de bloques.

Características

- Rango de voltaje de operación: 3.3V~5.2V
- Frecuencia del sistema: 3.58MHz
- Oscilador tipo RC
- Tres salidas D/A de audio de 8 bits
- Aplicación embebida con el microcontrolador Holtek
- Procesamiento de melodías de cuatro canales
- Dos canales de procesamiento de sonido
- Control principal de volumen de ocho niveles
- 36 efectos ritmicos
- Capacidad de manejar dos displays de siete segmentos
- Encapsulado de 40 pines

Micropocesador ez-80 de ZILOG

El ez80 es la siguiente generación del conocido procesador z80. Sus características de funcionamiento lo hacen comparable con los procesadores de 16 bits, pero con el ahorro de energía de un procesador de 8 bits. Se trata de un micropocesador de 8 bits capaz de realizar operaciones de 16 o 24 bits, para ambos tamaños de datos, el procesador incluye un acumulador. Una de sus características más llamativas es que viene listo para Internet; este procesador ofrece el paquete de TCP/IP para aquellos que quieran diseñar productos para conectarse a la red. El ez80 ofrece una de las CPU más rápidas que hay en la actualidad, efectúa operaciones 4 veces más rápido que el z80 bajo el mismo reloj; este incremento en la velocidad de procesamiento puede usarse para mejorar el ancho de banda o para disminuir el consumo de potencia.

El ez80 es el primer micropocesador de 8 bits, capaz de soportar 16 MB de direccionamiento lineal. Esta característica permite trabajar con memorias grandes que pueden soportar aplicaciones complejas de software. Un módulo de multiplicación y acumulación (MAC) opera en paralelo con el procesador z80 para calcular sumas de productos, muy útil en aplicaciones de procesamiento digital de señales. Este módulo permite multiplicación de 16x16 y acumulación en 40 bits.

Arquitectura

El ez80 es la cuarta generación de procesadores z80. Es la base de una nueva familia de micropocesadores, e incluye las siguientes características:

- Código compatible con z80 y z180
- Varios modos de generación de direcciones incluyendo direccionamiento lineal de 24 bits.
- Registros de 24 bits y ALU
- Módulo MAC opcional para aplicaciones DSP.

Registros de programa

El registro A es el acumulador para operaciones de 8 bits, y el par de registros HL es el acumulador para operaciones de 16 y 24 bits. Además del registro A, hay seis registros más llamados B, C, D, E, H y L, que pueden operar también como registros pares BC, DE y HL. El registro bandera F completa el banco de registros.

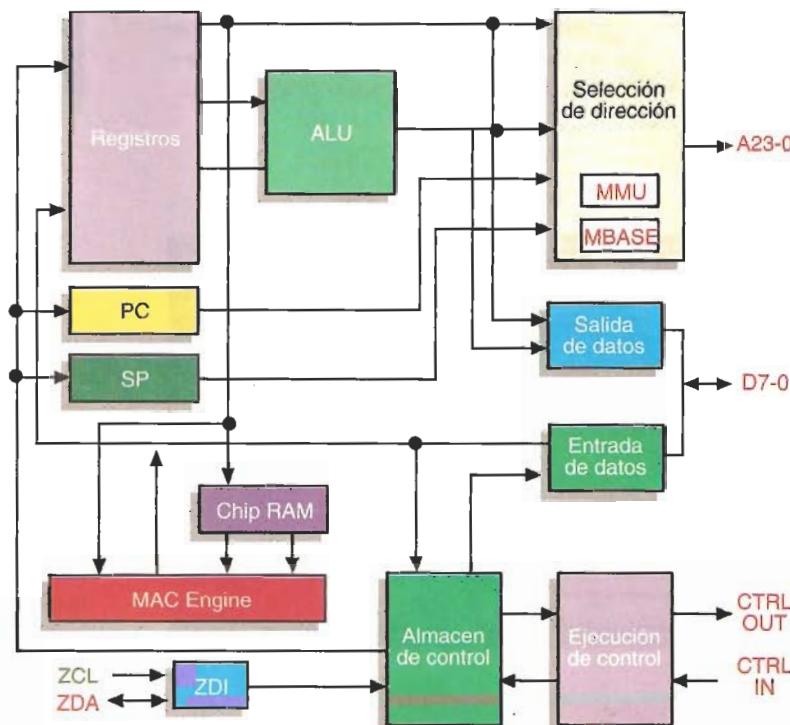
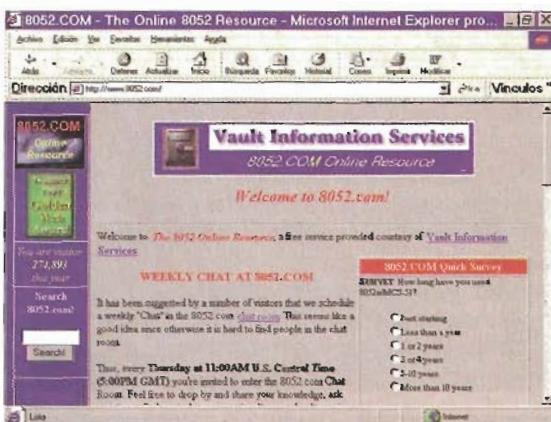


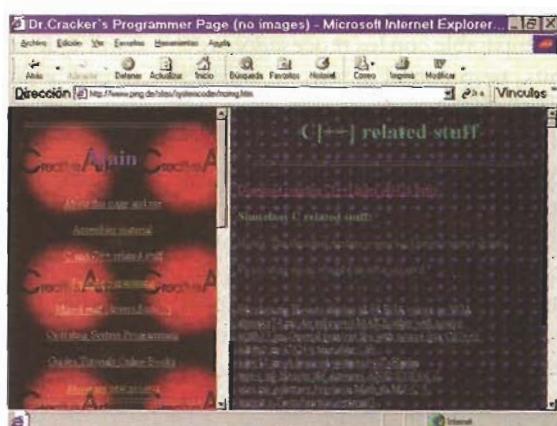
Figura 3. Diagrama de bloques del ez80



La página del 8052

Esta página se dedica a la familia de microcontroladores MCs51 y derivados. Tiene disponible un simulador shareware para el 8052. Incluye tutoriales, enlaces a otras páginas. Los jueves se hacen reuniones en el chat de la página para hablar sobre el tema.

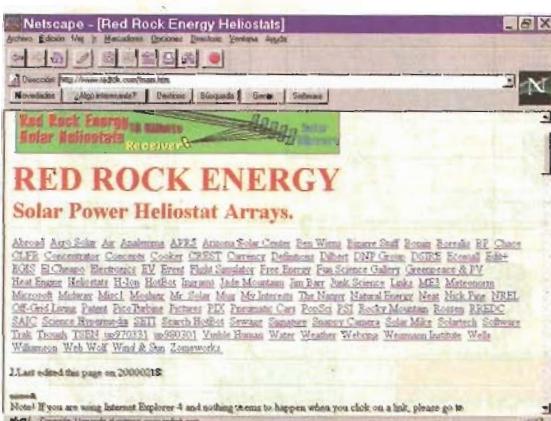
<http://www.8052.com>



La página de programación del Dr. Cracker

En esta página se encuentra excelente información, utilidades, bibliotecas, programas y drives hechos en C, Pascal y Assembler que pueden ser de gran utilidad. Manejo de GUI, puertos, sonido, imágenes etc., están disponibles sobre todo si su entorno es el Borland C++ o el Turbo C++.

<http://www.ping.de/sites/systemcoder/noimg.htm>



Energía de la piedra roja

La energía solar será en un futuro no muy lejano, una de las mejores fuentes a la que nos veremos obligados a acudir para subsistir. En esta página se encuentra una muy buena información introductoria sobre el manejo de esta clase de energía, y se muestran algunos ejemplos concretos de aplicaciones reales ya implementadas y que funcionan con éxito.

<http://www.redrok.com/main.htm>

Fuente para desarrolladores en Java

Aquí se ofrece una muy buena información, noticias que le serán muy útiles a los que trabajan con Java.

Se enseñan nuevas técnicas y se muestran documentos y ejemplos de manejos de objetos y clases.

<http://www.artima.com/index.html>

Foros de tecnología avanzada

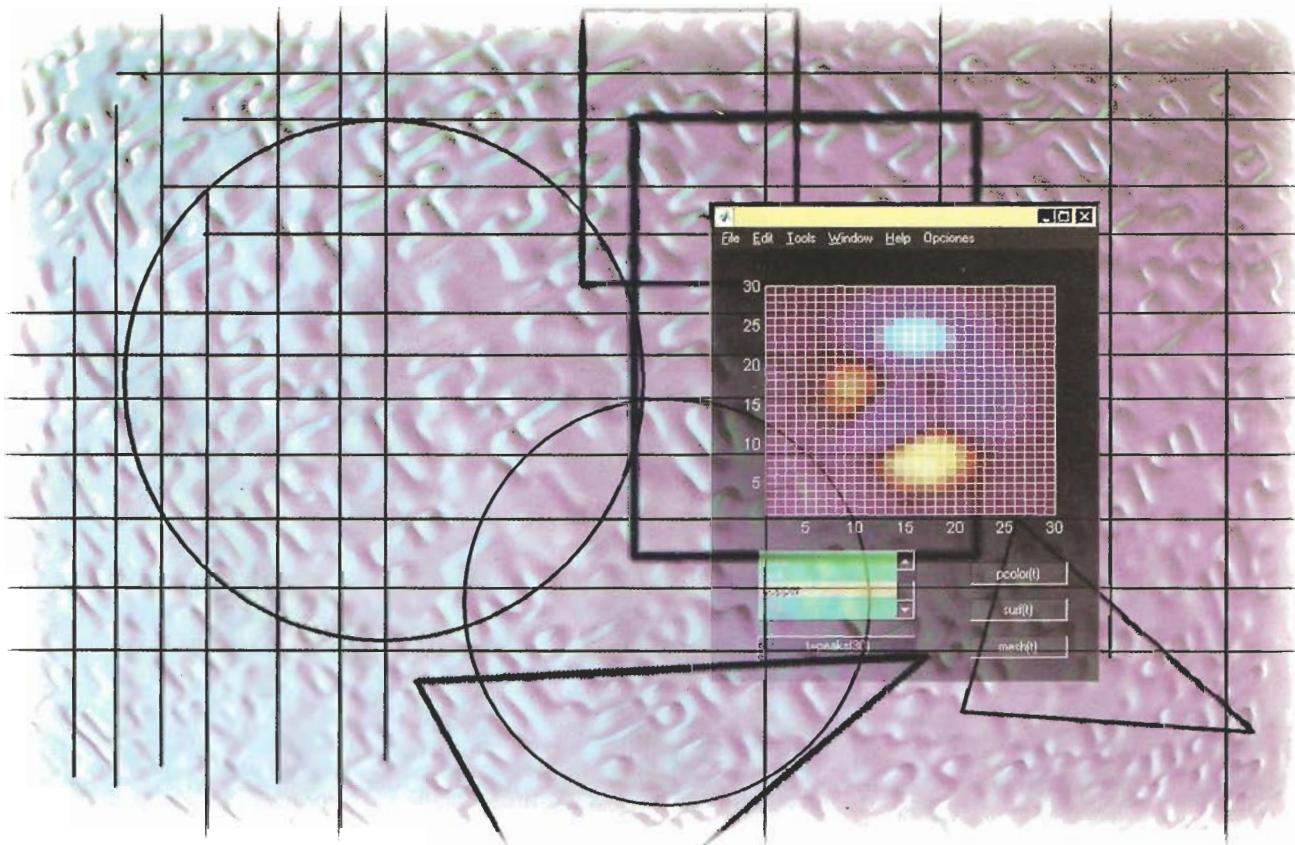
En esta página se consigna los últimos avances tecnológicos de todo el mundo en diversos campos. Es una buena página para todos los que les guste mantener actualizados en el mundo de las ciencias. Incluyen áreas como arquitectura de PC, software, circuitos de audio, aviación militar, fotografía digital, electrónica, mecánica, ingeniería naval, etc.

<http://www.deja.com/~tech/>

La página de educación en virus
Los virus de computadora son uno de los problemas con los que tenemos que lidiar frecuentemente. Esta página contiene documentos muy serios, recopilados de distintos investigadores y empresas, que nos puede servir para conocer más acerca de los virus y cómo defendernos de ellos. Incluye información que va desde la bases hasta los virus más temidos.

<http://members.xoom.com/kbechtel/edu.htm>

Construya una interfaz gráfica con Matlab



CARLOS ALBERTO LÓPEZ A. WILLIAM ARDILA U.

LUIS E. LLAMOSA

Grupo de Electrofisiología
Universidad Tecnológica de Pereira

La interacción de los usuarios con las computadoras personales (PC) se facilita cada vez más gracias a las interfaces gráficas, que orientan a la persona mediante menús donde están categorizadas las diferentes operaciones que se pueden realizar con un programa (software) para PC. El propio Matlab está controlado por menús.

La interfaz gráfica con el usuario (GUI, por sus iniciales en inglés, *Graphical User Interface*) hace que un programa sea fácil de usar. El objetivo de este artículo consiste en presentar una introducción sencilla a las interfaces gráficas que pueden programarse con Matlab.

El flujo de las operaciones que debe realizar la computadora para llevar a cabo una tarea en particular puede estar controlado por un *script* (conjunto de instrucciones que conforman un programa específico) o por una GUI. En el caso del *script*, el algoritmo implementado establece un orden de ejecución de las instrucciones, mientras que esto no ocurre con una GUI. En efecto, los comandos para crear una interfaz con el usuario se escriben de la misma forma que en un *script*; pero una vez que este guión se ejecuta, permanece en la pantalla hasta tanto el usuario decida cerrarlo o hasta que elija otra opción del menú.

Para facilitar el trabajo de diseño y construcción de una GUI, Matlab ofrece una herramienta conocida como GUIDE (*Graphical User Interface Development Environment*) con la cual se simplifica enormemente el procedimiento que se debe seguir para su creación.

Las herramientas que conforman el entorno de desarrollo de Interfaces gráficas de usuario –GUIDE–, son las siguientes:

- El editor de propiedades (The property editor)
- El panel de control (The Guide Control Panel)
- El editor de guiones (The Callback Editor)

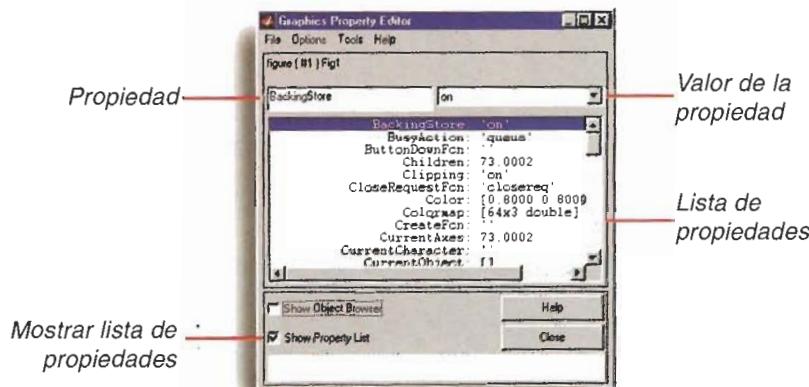


Figura 1. Ventana del editor de propiedades

- La herramienta de alineación (The Alignment Tool).
- El editor de menús (The Menu Editor)

Una síntesis de las características de cada una de ellas se presenta a continuación:

El editor de propiedades (Property editor)

Los objetos gráficos que se manejan en Matlab están caracterizados por un conjunto de propiedades, tales como el color, los ejes, las retículas, etc. Para ver y cambiar estas propiedades desde la ventana de comandos, se emplea la orden **propedit**, cuyo uso se ilustra con el siguiente ejemplo que consiste en una serie de instrucciones para crear una gráfica de la función $\sin(t)/t$:

```
t = 0:0.1:20;
plot(t,(1/t).*sin(t))
get(gcf, 'color')
```

En las instrucciones anteriores, el comando **get** se usa para obtener información sobre el color de la figura ge-

nerada con la instrucción **plot**; **gfc** (get current figure) da información sobre la ventana de gráficos.

La respuesta que se observa en la ventana de comandos es la siguiente:

```
Ans =
0.8000 0.8000 0.8000
```

Los anteriores valores representan un color entre el blanco [1 1 1] y el negro [0 0 0]. Se puede recurrir al editor de propiedades para ubicar y modificar la anterior información o cualquier otra propiedad de la figura actual. Para ello, debe escribir en la ventana de comandos la siguiente instrucción:

```
propedit (gcf)
```

De esta manera aparece una ventana como la que se muestra en la figura 1, en cuya parte central se despliega una lista de las propiedades de la gráfica. Ubique la palabra **color** y haga clic sobre ella para modificar los

valores iniciales en el campo de edición situado en la parte superior de la lista. Pruebe varios conjuntos de valores teniendo en cuenta que los números de la tripleta indican la proporción de rojo, verde y azul que se combinan para generar el color que usted ve en la ventana de la figura.

Más adelante se indica otra forma de acceder al editor de propiedades.

El panel de control (Guide Control Panel)

La ventana del panel de control del GUIDE, figura 2, aparece en la pantalla al escribir en la ventana de comandos la instrucción **GUIDE**.

El Panel de Control simplifica la creación y manipulación de objetos gráficos, puesto que facilita el acceso a sus propiedades; además permite colocar y organizar los diferentes elementos de una GUI seleccionándolos con el ratón en una paleta que está ubicada en la parte inferior de la ventana, y luego arrastrándolos hasta la posición deseada en la ventana donde se diseña la GUI.

El Panel de Control está dividido en tres partes principales:

- Las herramientas del GUIDE (Guide Tools): Editor de propiedades (Property Editor), Editor de guiones (Callback Editor), Herramienta de alineación (Alignment Tool), Editor de menús (Menu Editor).
- Lista de Control de Figuras: Para controlar y activar las figuras.
- Paleta de objetos nuevos (New object palette): permite añadir objetos a las figuras. Existen nueve tipos diferentes de controles que se pueden integrar en una figura:
 - Botones de pulsar con el ratón (push buttons).
 - Botones de radio (radio buttons).
 - Cajas de opciones (checkboxes).
 - Cuadros de texto editable (editable text).
 - Cuadros de texto estático (static text).
 - Deslizadores (sliders).
 - Marcos (frames).
 - Cajas con listas de opciones (listboxes).
 - Menús desplegables (pop-up menus).



Figura 2. Ventana del Panel de Control

El editor de guiones (Callback Editor)

En este editor se escriben las instrucciones que hacen funcionar los diferentes objetos (botones, cajas, cuadros, etc.), cuando reciben la acción del ratón. Una vez activada la figura (procedimiento que se indica más adelante), basta con hacer clic sobre uno de los objetos para que se ejecute el código asociado a él, que ha sido previamente escrito en el recuadro de edición.

La ventana correspondiente al Editor de guiones se muestra en la figura 3, junto con las partes que la componen.

El acceso al editor se puede efectuar desde la ventana de comandos mediante la orden **cbedit**, o haciendo clic en el botón identificado como callback editor en la ventana Guide Control Panel.

El editor de guiones también se puede usar para realizar cambios en cualquier figura. Comience colocando una marca de verificación en la casilla del visualizador de objetos (*show object browser*), para desplegar una lista de ellos en el visualizador (*object browser*) situado en la parte superior de la ventana. Luego seleccione la propiedad que desea modificar, haga los respectivos cambios y finalice haciendo clic sobre el botón Aplicar (*Apply*).

La herramienta de alineación (Alignment tool)

La herramienta de alineación, figura 4, permite organizar objetos seleccionados usando una colección de botones de pulsar que facilitan las operaciones de distribución y alineación de objetos en una figura.

Se puede activar la herramienta de alineación desde la ventana de comandos al ejecutar la orden **align**, o al hacer clic sobre el botón respectivo en la ventana Guide Control Panel.

Las opciones disponibles son:

- Alineamiento vertical con respecto a la parte superior de los objetos; alineamiento con respecto al punto me-

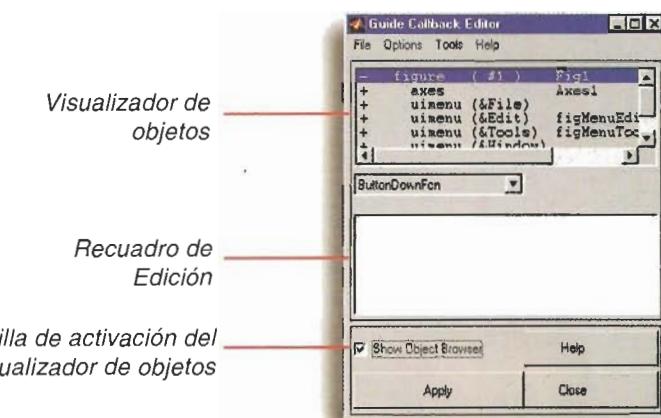


Figura 3. Elementos de la ventana del Editor de guiones

dio o a la parte inferior de ellos. Distribución de arriba abajo y fijación del espacio entre objetos.

- Alineamiento horizontal con respecto a la izquierda, al centro o a la derecha; distribución de objetos de izquierda a derecha y establecimiento de espacio entre los diferentes objetos.

El editor de menús (Menu Editor)

El editor de menús, figura 5, se usa para agregar y editar los menús desplegables creados por el usuario, que se encuentran asociados con una figura. Los menús que usted crea con el Editor de menús se agregan a los menús básicos que normalmente pertenecen a una figura: Archivo (File), Editar (Edit), Ventana (Window), Ayuda (Help). Estos cuatro menús son "invisibles" e inmodificables, pero se pueden ocultar cambiando la propiedad *MenuBar* de la figura al valor *none*. El editor no muestra los

nuevos menús hasta tanto la figura haya sido activada.

Un ejemplo completo

A continuación se va a diseñar y construir una GUI que permita modificar los colores de una gráfica, el fondo de la ventana de gráficas y el tipo de gráfica que será representada. Se usará la herramienta GUIDE a fin de simplificar al máximo el código y la sintaxis requeridos para construirla.

Se empieza escribiendo la palabra GUIDE en la ventana de comandos. Aparecen entonces una nueva ventana gráfica y el Panel de Control de GUIDE. Presione el botón "axes" en el Panel de Control y arrástrelo para que establezca un área en la cual quedará ubicada la figura en la ventana gráfica. Añada una Caja de lista (listbox) seleccionando el botón respectivo en el Panel y arrastrándolo hasta la ventana. A continuación agregue una caja de texto editable (Edit)

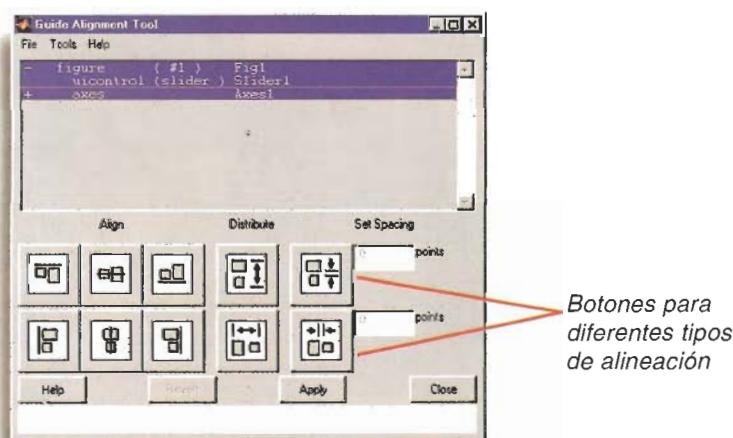


Figura 4. Ventana con las herramientas de alineación

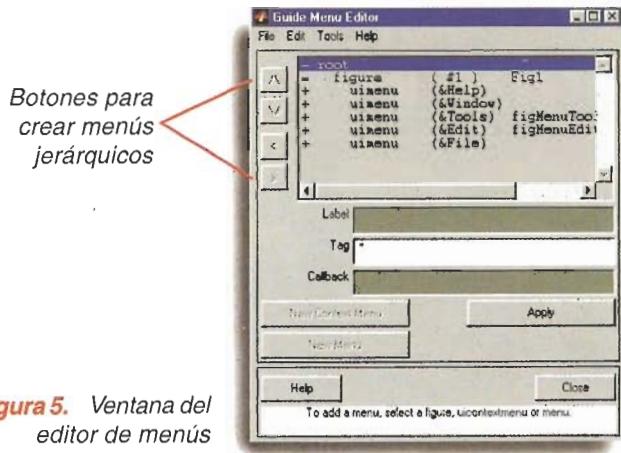


Figura 5. Ventana del editor de menús

debajo de la anterior. En este momento la ventana gráfica deberá tener el aspecto mostrado en la figura 6.

Luego alinee a la izquierda la Caja de lista, la Caja de texto editable y los ejes. Para ello se utiliza la herramienta de alineación del Panel de control; al pulsar sobre ella se abre una ventana. Seleccione con el ratón la lista que contiene cuatro elementos ubicados en la parte superior de la ventana, figura 7; luego haga clic sobre el botón de alineación horizontal con respecto a la izquierda (Horizontal align left) y finalice esta operación pulsando sobre el botón Apply (aplicar). Así obtendrá tres objetos aliñeados según su requerimiento.

Ahora se añadirá a la figura tres botones de pulsar (push buttons). Para ello seleccione del Panel de Control el botón identificado con dicho nombre y arrás-

trelo hasta la figura; luego cópielo dos veces. Seguidamente seleccione los tres botones y mire sus posiciones en el Editor de propiedades. Para conseguir que sus anchuras y alturas sean iguales, escriba en el campo de valor del Editor de propiedades las especificaciones [? ? 65 15] para la propiedad position .

Para concluir el diseño, alinee a la derecha estos tres botones empleando nuevamente la herramienta de alineación. Al efectuar lo indicado hasta este momento, se debe obtener la siguiente distribución de los elementos en la ventana gráfica, figura 8.

El paso siguiente consiste en activar la figura haciendo clic sobre la expresión "controlled #1" y pulsando luego el botón Apply del Panel de Control. Cuando una figura se encuentra en el estado controlado, se pueden mover los

botones, ejes, y otros objetos que existan dentro de ella haciendo uso del ratón. Una vez concluida la ubicación de los elementos, se activa la figura para que vuelva a su comportamiento normal; es decir, sin posibilidad de mover los objetos que la componen. Grabe la figura cuando se le ofrezca esta posibilidad.

En la siguiente etapa se van a rotular los tres botones, colocar varias opciones de color en la caja de lista y habilitar el recuadro de texto editable.

Use el Editor de propiedades para colocar las siguientes leyendas en la propiedad "string" de los botones: surf(t), pcolor(t) y mesh(t).

La manera de escribir una larga cadena de texto que contenga nombres de colores válidos en Matlab para seleccionar en la caja de lista, consiste en crear primero un arreglo de celdas en la ventana de comandos escribiendo allí algunas de las funciones que generan colores predefinidos:

```
Colores = {'Jet'; 'autumn'; 'bone';
'pink'; 'copper'; 'flag'}
```

Esta lista contiene sólo algunos de los colores disponibles. Ahora se debe seleccionar la propiedad 'string' para la caja de lista en el Editor de propiedades. Asignele como valor la palabra **color** es escrita sin comillas; así se recuperá esta variable del espacio de tra-

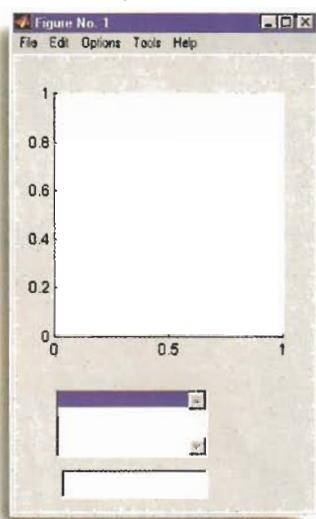


Figura 6. Primeros pasos en el diseño de la GUI

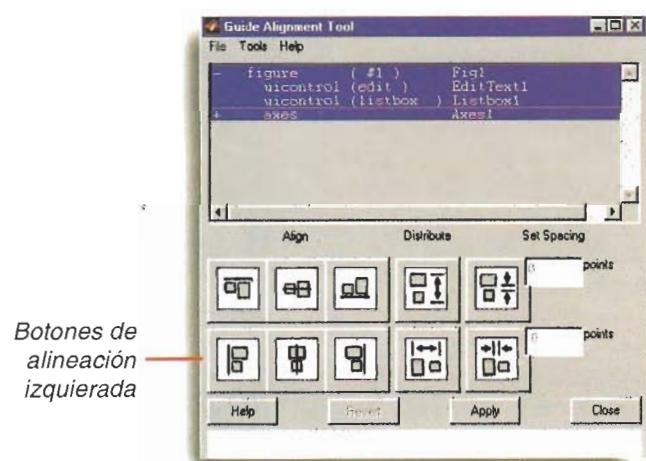


Figura 7. Ventana correspondiente a la herramienta de alineación. En ella se señala el botón de alineación con respecto a la izquierda

abajo del Matlab. Cuando se aplica este valor a la gráfica, el campo de edición exhibe ahora la palabra **towork** en lugar de la que se escribió inicialmente (colores); esta nueva variable es equivalente a la anterior. La expresión **towork** se usa siempre que el valor de una propiedad contiene varias líneas. En la ventana de la figura se observa ahora la caja con los colores seleccionados.

Para continuar el proceso, se usa el editor de guiones (Callback Editor) con el fin de asignar las órdenes que se deben ejecutar al pulsar los botones o al elegir un color de la lista.

Primero, seleccione la caja de lista (Listbox) en el visualizador de objetos del Editor y elija la propiedad **callback**. En el recuadro de edición, escriba el siguiente código:

```
Value = get(gcfbo, 'Value');
String = get(gcfbo, 'String');
Colormap = (string{Value})
```

La primera instrucción obtiene el valor para el objeto seleccionado; es decir, la caja de lista. La segunda instrucción obtiene la cadena de texto seleccionada de la lista que presenta la caja. Finalmente, la tercera instrucción asigna a la figura el color elegido.

Ahora se escribirá el código que corresponde a cada uno de los botones de pulsar. Tal como se procedió en el caso anterior, seleccione los tres botones en la lista que presenta la ventana del Editor de guiones; luego elija la propiedad **callback** y escriba el siguiente código en el cuadro de edición:

```
CommandString = get(gcfbo, 'String');
EditHandle = findobj(gcf, 'Tag', 'EditText1');
ZString= get(EditHandle, 'String');
eval(ZString);
eval(CommandString);
```

Las acciones que realiza el código anterior son las siguientes:

1. Invoca el comando de graficación para que ejecute lo indicado en el respectivo botón.
2. Ubica el manejador de la Caja de edi-

ción usando para ello el comando **findobj** y el rótulo (**Tag**) asignado por GUIDE (**EditText1**). El editor de guiones presenta la información correspondiente al rótulo al final de la lista de propiedades.

3. Obtiene la información escrita en la caja de edición, consistente en el argumento de entrada para la función que se requiere graficar.
4. Evalúa la cadena de texto que se ha escrito en la caja de edición.
5. Evalúa el comando de graficación.

En este punto ha concluido la implementación básica de la GUI. Se puede constatar su correcto funcionamiento escribiendo en la Caja de edición de la figura un argumento para la función **pcolor**; por ejemplo, **t=peaks(50)** y pulsando luego sobre el botón que tiene como rótulo la palabra **pcolor(t)**. El resultado será la gráfica solicitada por el usuario.

Antes de continuar, grabe la figura empleando el comando **Save** (grabar) del menú **File** (Archivo) en la ventana de la figura.

El último paso consistirá en agregar algunos menús a la figura para facilitar al usuario los cambios de color. Para ello se usa el Editor de menús y se procede como se indica a continuación:

- Seleccione la figura en el visualizador de objetos y haga clic en el botón **New Menu** (Nuevo menú).
- Escriba '**opciones**' como etiqueta para el menú y pulse nuevamente en el botón **New Menu** para definir un submenú dependiendo del menú '**Opciones**'.

Coloque como etiqueta del submenú la palabra '**color**' y pulse una vez más sobre el botón **New menu**. Este nuevo menú tiene como etiqueta la palabra '**color1**' y su código (Callback) es: '**set(gcf, "color", "jet")**'.

Continúe el procedimiento para los demás colores del submenú. Si lo desea, puede agregar otros menús a la figura. Grabe nuevamente y pruebe la GUI que ha construido. El resultado final debe ser similar al que se muestra en la figura 9.

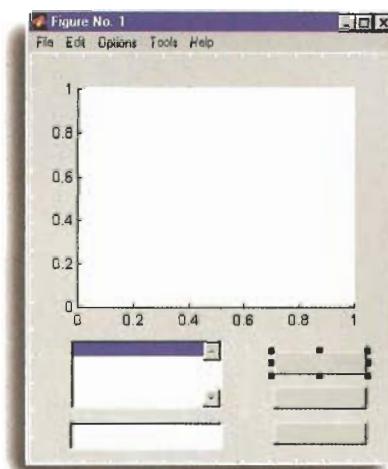


Figura 8. Disposición final de los objetos que conformarán la GUI.

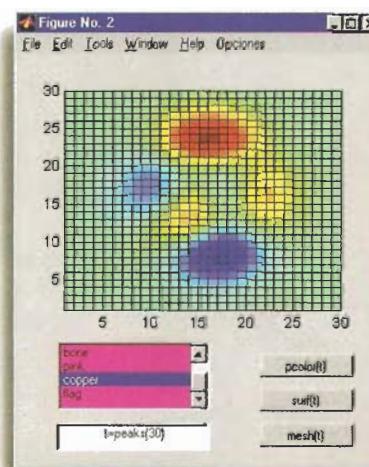


Figura 9. Presentación final de la interfaz gráfica de usuario.

Matlab e internet

Para ampliar y profundizar sus conocimientos en el tema de la construcción de GUIs, se recomienda la siguiente página:

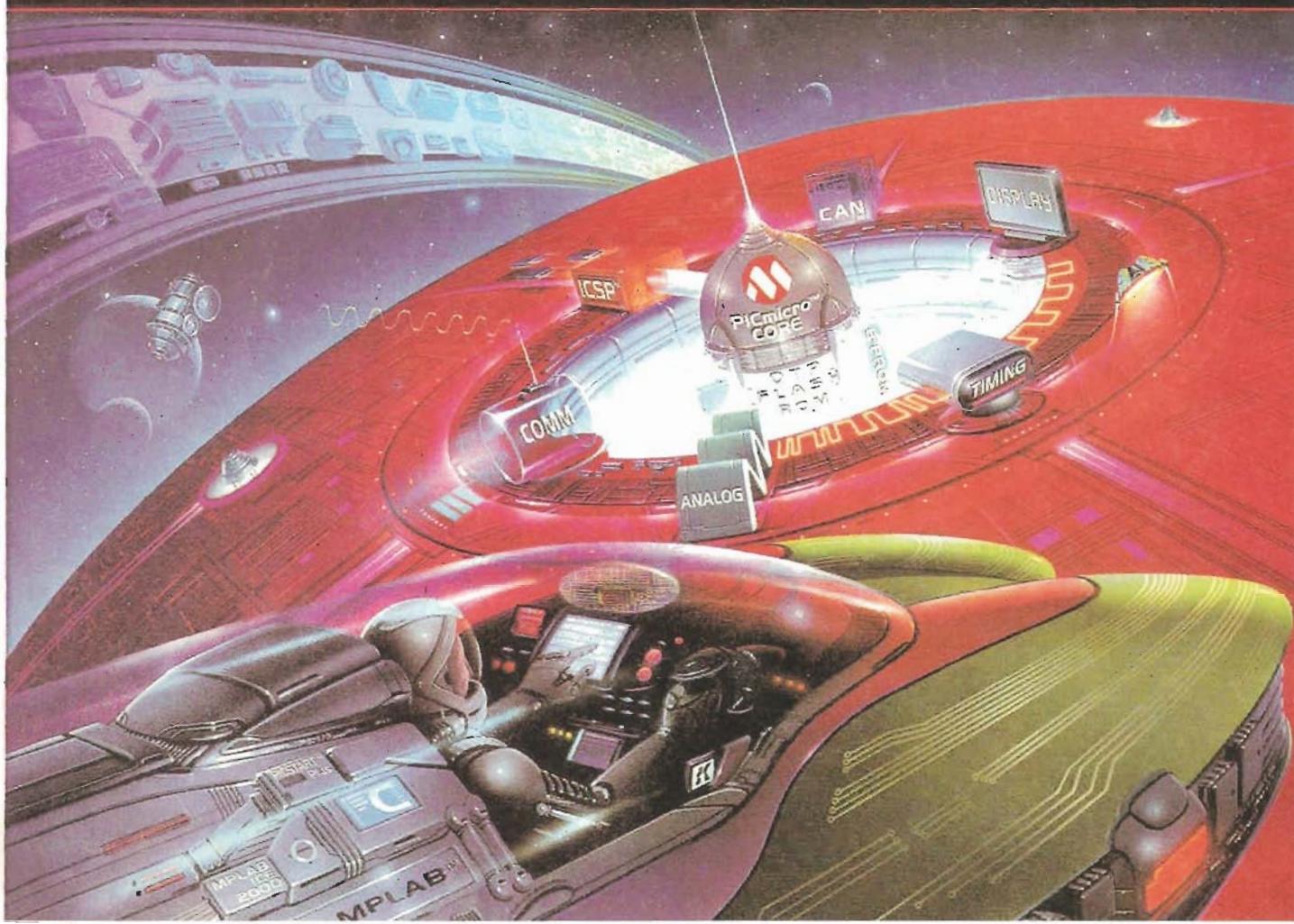
[ftp://ftp.mathworks.com/pub/mathworks/toolbox/building-guis/](http://ftp.mathworks.com/pub/mathworks/toolbox/building-guis/)

Bibliografía

HANSELMAN, Duane y LITTLEFIELD, Bruce. Mastering Matlab 5. A Comprehensive Tutorial and Reference. New Jersey : Prentice Hall. 1.998.

MATLAB. The Language of Technical Computing. Building GUIs with MATLAB. Versión 5. The Math works Inc. 1.997.

NAKAMURA, Shoichiro. Análisis numérico y visualización gráfica con Matlab. México : Prentice-Hall Hispanoamericana, S.A. 1.997



Ir Más Allá De La Imaginación Con Planeta Microchip

**Empiece A Diseñar Hasta El Límite Con Microchip. La Solución
Completa de Microcontrolador RISC de 8-bit**

Curso Intensivo de PICmicros

Dirigido a: Ingenieros, Técnicos, Informáticos, Hobbistas, Docentes y Estudiantes

Nivel: Básico & Intensivo.

Requisitos: que los participantes tengan el conocimiento básico de Electrónica Digital y PC.

Temas: Arquitecturas de PICmicros, Programación con Assembler de PIC, Uso de Herramientas como Simulador, Compilador y programador, Trucos & Precaución sobre el diseño con PIC, Implementación de Módulo LCD Display, Memorias Serial EEPROM, Teclados, Conversor A/D y demás periféricos para cualquier proyecto.

Duración: 6 clases de 18 horas totales que se dictan todos los meses incluso en el verano 1999.

Disertantes: Ingenieros de Aplicación certificados por Microchip Technology Central en Arizona USA.

Aplicación Típica: Alarma para auto y casa, Temporizador, Controlador de Voltaje y de Presión, Radio Comandos, Reemplazo a PLC, Telemetría, Instrumento de Electromedicina, Controlador de Motor, Robot, Codificador de TV, Cartel de Display para Propaganda, Control de Acceso, etc. Solo está limitado por su imaginación.

Precio de PICmicros: Desde \$2 para PIC de 8 Pin hasta \$22 para Super PICMicro de 64 Pin. (precio varia sin previo aviso)

Llamenos ya:

Capital y Gran Bs.As.: 011-4522-5466

Córdoba : 0351-4526698 Rosario: 0341-4400092

**Cika
Electrónica SRL**

Av. De Los Incas 4821 (1427) Buenos Aires Argentina

TEL. 4522-5466 FAX. 4523-6068

Email. cika@cika.com Website: www.cika.com



MICROCHIP
The Embedded Control Solutions Company