

elektor

LA REVISTA INTERNACIONAL DE ELECTRONICA Y ORDENADORES

Nº 313

4 €

TENSIÓN

¡Seguridad eléctrica primero!

¿Es seguro?

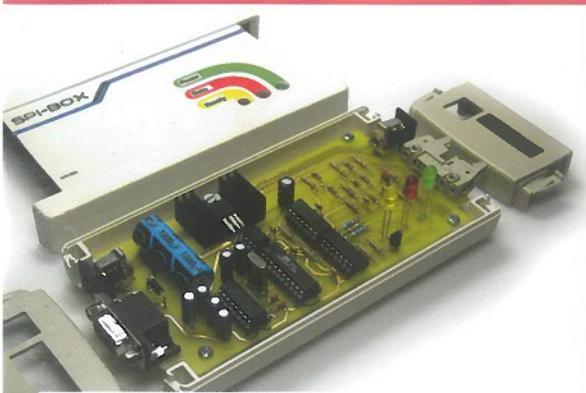
- Equipos eléctricos
- Fuentes de alimentación chinas

- **Caracterizador de Pilas AA**

- **Funciones de Control para el R8C**

Caja SPI Universal

una manera rápida de programar microcontroladores



Planera



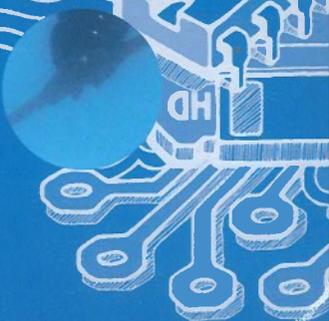
00313

Todas las revistas del año 2005 en CD

elektor

2005

Enero
Febrero
Marzo
Abril
Mayo
Junio
Julio
Agosto
Septiembre
Octubre
Noviembre
Diciembre



Humano
Planera

Redacción
VIDELEC, S.L.

Dirección
Eduardo Corral
Colaboradores
Jose M^o Villoch, Pablo de la Muñoza, Andrés Ferrer,
José Muñoz Carmona

Publicidad
jefe de grupo dpto comercial
Javier Menéndez
jefe de publicidad
Isidro Iglesias
Valportillo primera nº11
28108 Alcobendas (Madrid)
Tel.: +34 91 141 69 16 - Fax: +34 91 661 47 54
email: iglesias@grupov.es

Delegación Cataluña
AD PRESS, S.L.
Director:
Isidro A. Iglesias
Publicidad:
Annie Cabarubias
Comte d'Urgell, 165-167, B-1º-3ª
08036 Barcelona
Tel.: +34 93 451 89 07 - Fax: +34 93 451 83 23
email: acc@ad-press.com

Humano Planeta

EDITOR
Martín Gabilondo Viqueira
SUBDIRECTOR GENERAL
José Manuel Alonso Viguera
DIRECTOR COMERCIAL
Amador Moreno
DIRECTOR DE EXPANSIÓN
Rafael Morillo
DIRECTOR DE PRODUCCIÓN
Andrés Valladolid
DIRECTOR DE PUBLICACIONES
Juan Francisco Calle
DIRECTORA DE ADMINISTRACIÓN
Mar Molpeceres
REDACCIÓN, PUBLICIDAD Y SUSCRIPCIONES
C/Valportillo Primera, 11, 28108 Alcobendas, Madrid
Teléfono: 91 662 21 37 Fax: 91 662 26 54
www.grupov.es

Servicios
Redacción, traducciones y maquetación
VIDELEC, S.L.

Imprime
IBÉRGRAPHI 2000 S.L.L.

Distribución en España:
S.G.E.L. Avda. Valdeparra, 29
Tel.: 91 657 69 00

Distribución en el Exterior
Argentina
Importador
Edilogo S.A.
Av. Belgrano 225, 1º, B. C1276ADB Buenos Aires
Distribución Capital
DISTRIBED
Distribución Interior
D.G.P.

Chile
Importador
Iberoamericana de Ediciones, S.A.
C/Leonor de la Corte, 6035. Quinta Normal. Santiago de Chile

México
Importador exclusivo
Compañía Importadora de Revistas S.A. de C.V. ("CIRSA")
Negra modelo Nº 6, Col. Alce Blanco
Municipio de Naucalpan de Juárez, (53330) Estado de México
Tel.: (52-55) 5360-4167 - Fax: (52-55) 5360-7774

Distribución Estados: Citem
Distribución D.F.: Unión de Voceadores

Portugal
Importador
Ediber-Edicao e distrib. de Public. L.D.A.
Rua D. Carlos Mascarenhas, 15 - 1000, Lisboa

Venezuela
Distribuidora Continental

Colombia
Disunidas, S.A.

Depósito legal: GU.3-1980
ISSN 0211-397X
30/Junio/2.006

Reservados todos los derechos de edición.
Se prohíbe la reproducción total o parcial del contenido de este número ya sea por medio electrónico o mecánico de fotocopia, grabación u otro sistema de reproducción, sin la autorización expresa del editor.
Las opiniones expresadas a lo largo de los distintos artículos, así como el contenido de los mismos, son responsabilidad exclusiva de los autores. Así mismo, del contenido de los mensajes publicitarios son responsables únicamente los anunciantes.
Copyright = 1996 Segment BV

PVP en Canarias 4,15 € (sobretasa aérea)

conocimientos

62 Comunicaciones de Datos Serie

sobre el terreno

32 Curso FGPA (1)

44 Simple Caracterizador de Pilas AA Recargable

48 Caja SPI Universal

56 Funciones de Control para el R8C

68 Trucos de Diseño

- Conmutador DC controlado por RS-232
- Tension de referencia con una diferencia
- ¿Como cablear un núcleo toroidal?
- Soportes de batería DIY
- Limpieza del aluminio con bicarbonato de sosa

tecnología

6 Ahorro y Economía

22 Diseños para las Masas

28 Generación de Ondas en C

38 Manifiesto sobre Alimentaciones en Modo Conmutado

informática y mercado

10 Noticias

20 Próximo número

información y entretenimiento

71 Nuevos Libros

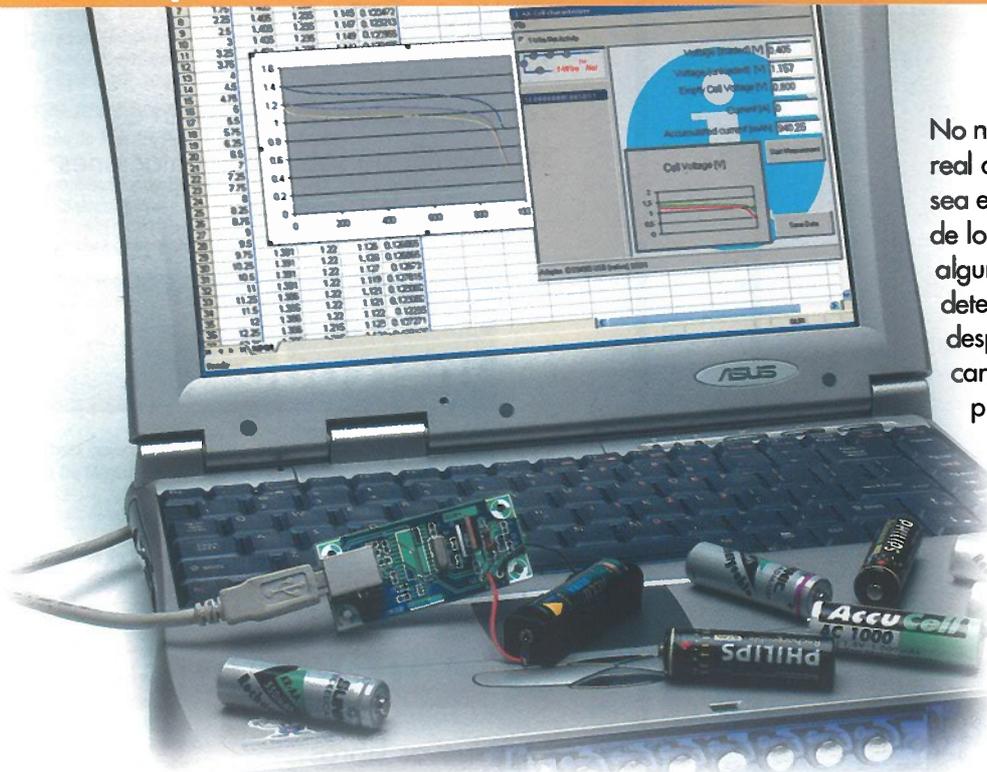
72 EPS

18 Ahorro y Economía

¿Cómo debería construirse un circuito seguro de usar? Para encontrar la respuesta vamos a echar una ojeada al interior de un par de equipos y a su circuitería interna. Es fascinante ver los trucos que los profesionales desarrollan para conseguir el compromiso y limitaciones entre el precio y la miniaturización.



44 Simple Caracterizador de Pilas AA Recargable



No nos sorprende que la capacidad real de unas pilas AA recargables sea en ocasiones bastante diferente de lo que se especifica y que, en algunos casos, la capacidad se deteriore de forma dramática justo después de un par de ciclos de carga y descarga. Este circuito nos proporciona información detallada sobre el estado de unas pilas recargables del tipo AA.

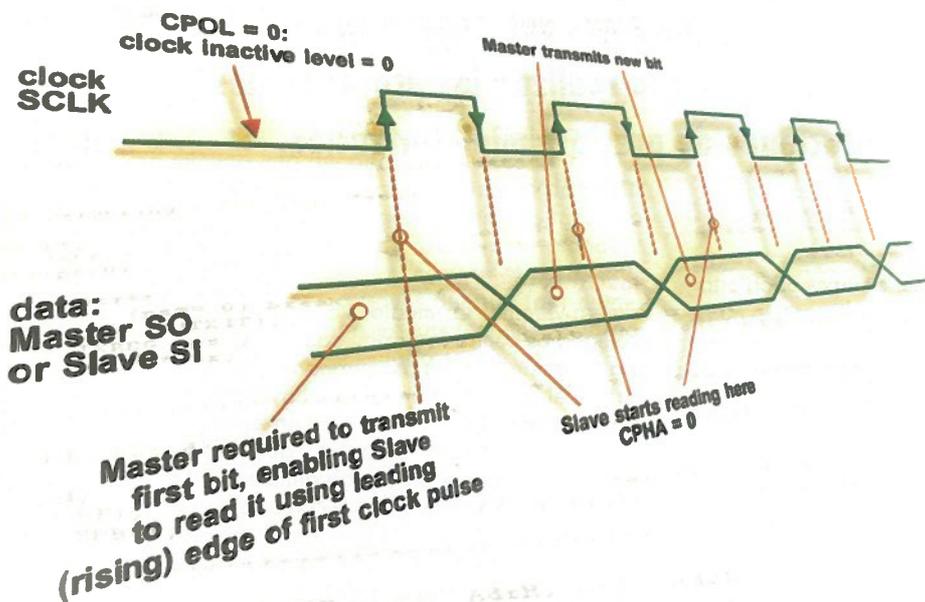
CONTENIDO

48 Caja SPI Universal



Una gran cantidad de microcontroladores pueden programarse a través del bus SPI. Trabajar con este bus desde un puerto serie de un ordenador, añadiendo sólo un par de componentes extras es una solución económica pero lenta. Una alternativa mejor es la de utilizar la Caja SPI que se describe en este artículo, la cual proporciona una velocidad aceptable incluso cuando se usa con un dispositivo conversor USB a serie.

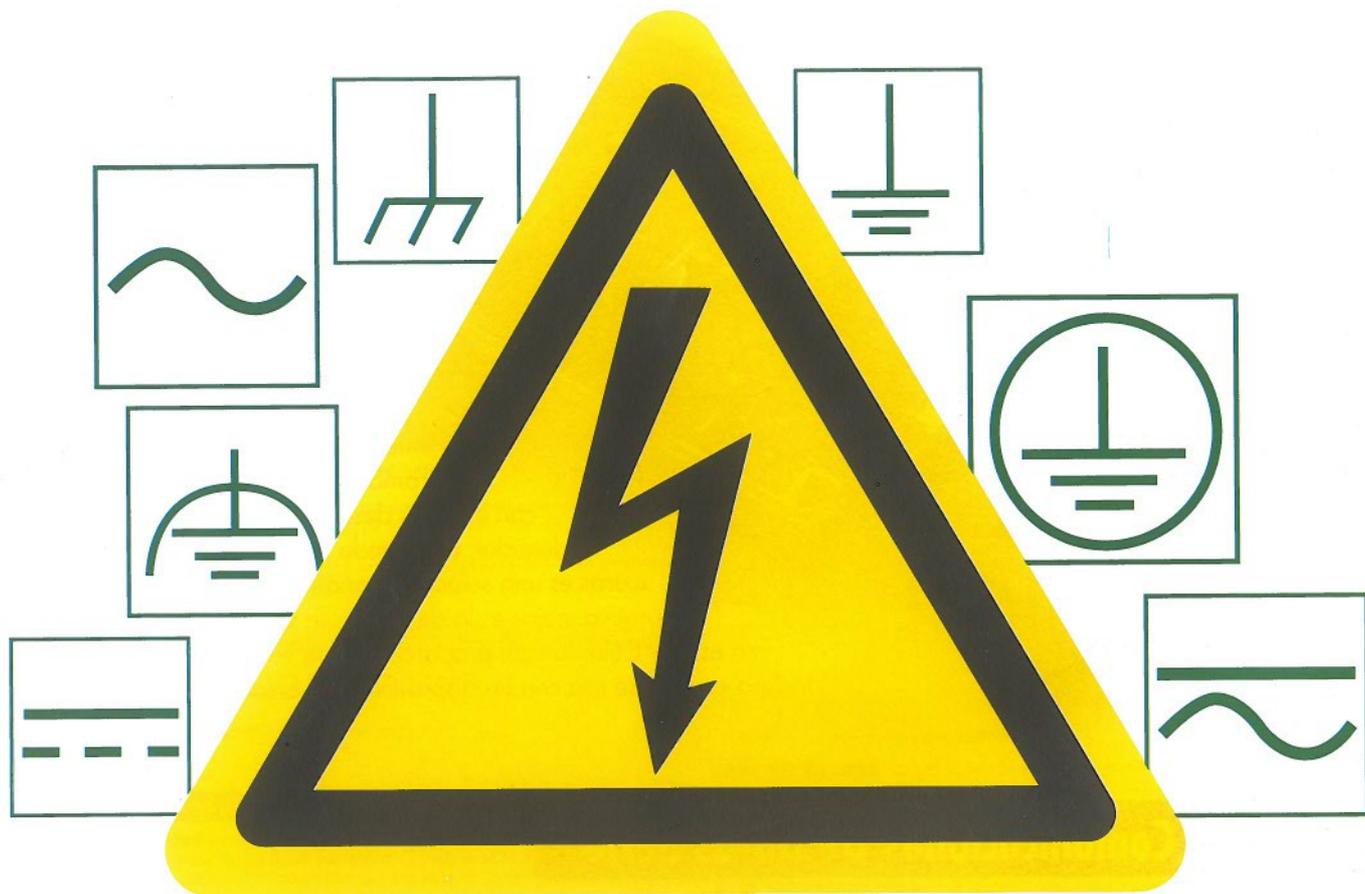
62 Comunicaciones De Datos Serie



La transferencia de información a través de una interfaz serie asíncrono entre un microcontrolador y un dispositivo periférico se ha convertido en algo cada vez más de moda en los últimos años, especialmente desde que la velocidad de datos se ha conseguido aumentar hasta más de los 10 MBit/s.

Ahorro y Economi

Martin Ossmann



**¿Cómo debemos construir un circuito para que sea segura su utilización?
Hemos echado un vistazo a los trucos que realizan los profesionales para
conseguir satisfacer las limitaciones de precio y miniaturización.**

La cuestión de la seguridad debería ser siempre prioritaria cuando diseñamos circuitos y equipos. Un dispositivo no ha de suponer ningún riesgo para las personas durante su manipulación; el dispositivo debe funcionar de forma satisfactoria en todas las situaciones posibles y, finalmente, no debe causarnos (a nosotros o a otros dispositivos) daño alguno, en el caso de que se produzca un fallo.

Teniendo en cuenta todos estos requerimientos, para una unidad realizada de forma casera puede ser un gran reto. Por supuesto, los diseñadores industriales se enfrentan al mismo problema cuando hacen productos para usuarios finales: esos productos deben ser infalibles. ¿Por qué no echamos un vistazo a un par de elementos del equipo y vemos lo que podemos aprender de los profesionales?

Ahorro de dinero

Primero veremos las medidas que debemos tomar para protegernos contra la posibilidad de un shock eléctrico. Esta situación se evita asegurándonos que las distintas partes del dispositivo en las que hay alta tensión no se pueden tocar, que normalmente corresponden con aquellas donde hay tensión de red. En general los cables y componentes ya vienen con aislamiento, a lo cual el reglamento estándar se refiere como 'aislamiento básico'. La situación en la que el aislamiento básico no es fiable no debería depender del azar. También son necesarias otras medidas, tales como aislamiento adicional y puesta a tierra del chasis, para aumentar la seguridad en situación de fallo.

Algunas observaciones en equipos eléctricos

Veamos en primer lugar un reproductor de CD (**Figura 1**). De acuerdo a la etiqueta de la unidad (**Figura 2**) éste es un dispositivo de Clase II; por lo tanto, no hay conexión a tierra. Como podemos ver, el cable de red entra directamente al transformador. El interruptor de on-off está en el lado secundario del transformador y por lo tanto es una parte de baja tensión. Los fabricantes han sido cuidadosos para dar un buen aislamiento. Todos los componentes que llevan tensión de red tienen un aislamiento adicional, y hay una separación de protección en los dos bobinados del transformador. Esto nos asegura que el fallo del esmalte en el cable de uno de los bobinados no es peligroso. En la caja no hay fusible en el bobinado primario, lo cual nos ayuda a mantener el bajo coste total. En el Reino Unido se incorpora un fusible en el enchufe de conexión del equipo. Esta disposición sólo se puede usar con un transformador adecuado y las correspondientes precauciones de seguridad en el lado secundario. Una desventaja de esta disposición es que el transformador funciona de forma continua, incrementando el consumo de energía en situación de reposo.

Análisis del transformador

El transformador de red hace un trabajo importante en el aislamiento del circuito de la red. Para que un transformador clásico de 50 Hz sea seguro debe construirse usando un núcleo especial con bobinas separadas (**Figura 1**), lo cual debe encontrar todos los requerimientos necesarios en términos de caminos de conducción. Las modernas fuentes de alimentación incrementan su seguridad usando tecnología de conmutación y pequeños transformadores, y revisando los tipos de transformadores usados.

La **Figura 3** nos muestra un transformador toroidal desde una fuente de conmutación reductor para lámparas halógenas de pequeña tensión, proporcionando aislamiento desde la red. El bobinado primario está colocado dentro de la caja de plástico. Este diseño de núcleo particular asegura que los bobinados primario y secundario se mantienen a una distancia de seguridad mínima uno del otro. Los huecos de aire y el espacio que deja el material aislante han de determinarse cuidadosamente. En general es más fácil proporcionar los requerimientos de aislamiento usando una barrera física, aunque es necesario asegurar que el camino de conducción a través del aire (por ejemplo entre los dos bobinados) es suficientemente largo. En este transformador el bobinado secundario no tiene un aislamiento extra.

Una forma relativamente breve de proporcionar un orificio adecuado es usar bobinas diseñadas especialmente. La **Figura 4** muestra los componentes de un ejemplo. Los bobinados primario y secundario están bobinados en bobinas separadas con collares y sus extremos nos dan orificios adecuados. Esta construcción hace más fácil satisfacer las regulaciones de seguridad. Sin embargo, hay algunas desventajas: las bobinas son caras, y el transformador resultante es a

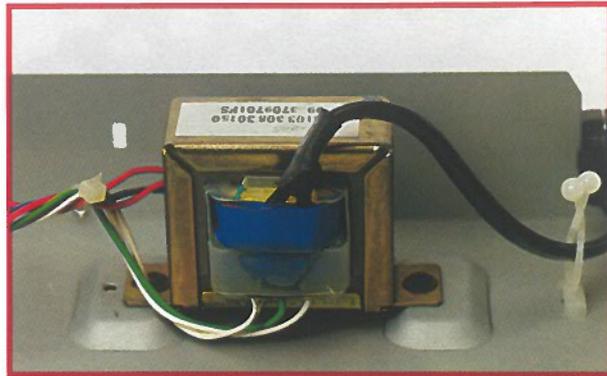


Figura 1.
Un vistazo dentro de un reproductor de CD comercial.



Figura 2.
El símbolo indica un dispositivo de Clase II.

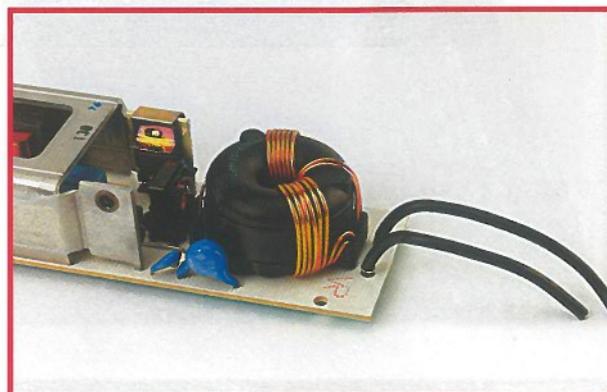


Figura 3.
Transformador toroidal en una fuente de alimentación para una lámpara halógena.

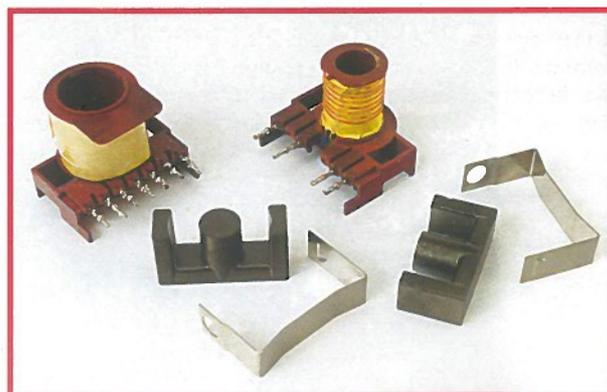


Figura 4.
Bobinas con aislamiento propio.

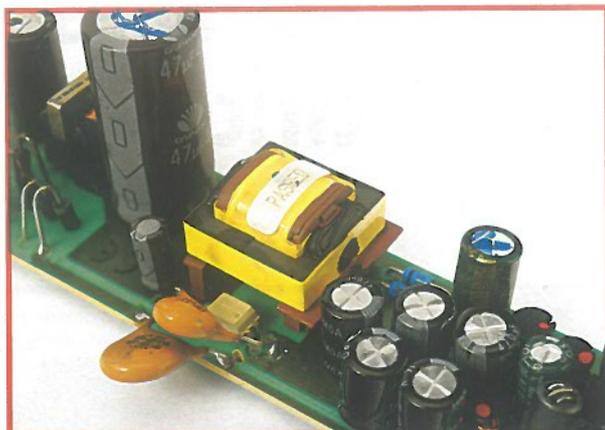


Figura 5. Transformador montado, opto-acoplador y condensadores de clase Y.



Figura 6. Malla bobinada en capas.

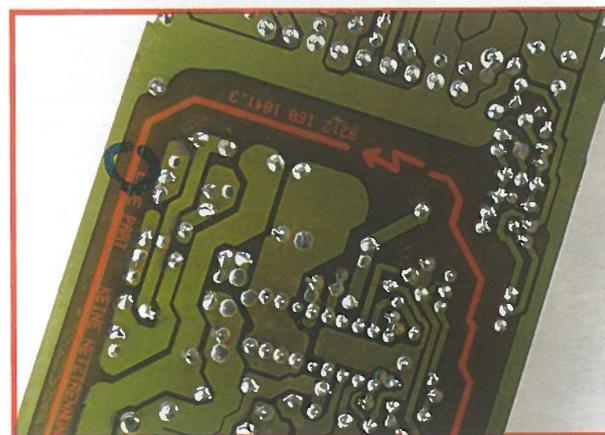


Figura 7. Barrera de aislamiento entre la red y el lado secundario de la alimentación.

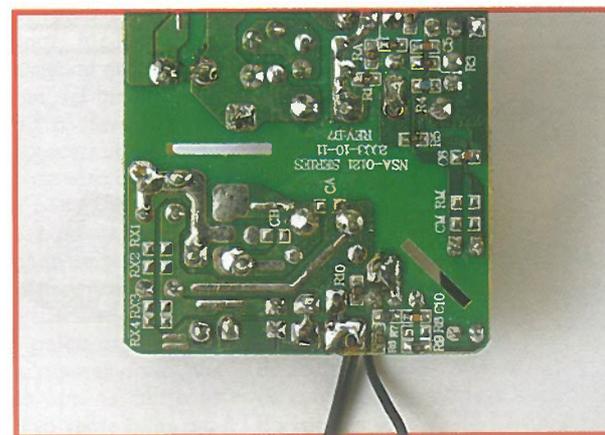


Figura 8. Slots que proporciona una barrera adicional.

menudo más grande que lo estrictamente necesario desde el punto de vista eléctrico. Los extremos primario y secundario están también bien acoplados juntos, lo cual crea problemas extra en el diseño de los convertidores de retorno de bajo coste. Por otro lado, no es necesario poner hojas aislantes entre los bobinados en el transformador completo. La **Figura 5** muestra un transformador en uso. Éste es relativamente pequeño, como puede verse comparando su tamaño con el opto-acoplador en un encapsulado DIL de 4 pines. Además del opto-acoplador también hay dos condensadores de clase Y en el primer plano. Estos componentes puentean el orificio de aislamiento; volveremos sobre este punto más adelante.

Seguridad en nuestros propios proyectos

Una forma de evitar la necesidad de las capas de aislamiento o bobinas especiales es usar cable de múltiples aislamientos. Por ejemplo, existe un tipo llamado 'cable de triple aislamiento' disponible para aislamiento de red (ver artículo de este número referente al diseño inteligente de la fuente de alimentación conmutada). La **Figura 6** muestra otra variante del tema, donde los bobinados están hechos de malla bobinada con varias hojas finas y gruesas aislantes. Con el uso de cable especial como éste es posible construir transformadores que presentan simultáneamente baja inductancia y un buen aislamiento.

Una vez que el transformador se ha diseñado para conseguir un aislamiento adecuado, podemos aplicar los mismos principios a la placa de circuito impreso y el resto de los dispositivos. En el diseño que se muestra en la **Figura 7** la parte interior de la placa es el lado primario. La fuente de alimentación se coloca en una caja de metal que está conectada al bobinado secundario. La barrera se puede ver claramente. También está etiquetada, por lo que el servicio técnico puede inmediatamente ver dónde cae.

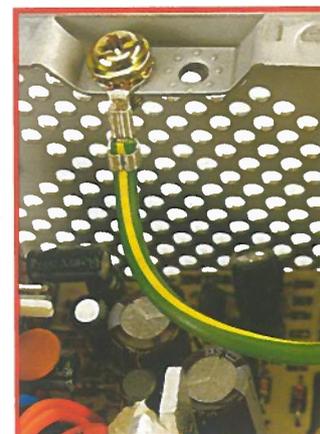


Figura 9. Conexión a tierra en

En la **Figura 8** se muestra con mayor detalle la siguiente situación: dos slots se han cortado en la placa de circuito impreso a puntos críticos particulares. Esto incrementa considerablemente la longitud del camino de la corriente de fuga sobre la superficie de la placa de circuito: cualquier corriente de fuga debe hacer su camino alrededor de los slots, lo que es particularmente importante porque la placa de circuito impreso puede llegar a estar sucia con el tiempo o atraer humedad, ya que tiene un aislamiento menor.

Dentro de un PC

Tradicionalmente los PCs están construidos como dispositivos de Clase I con una conexión a tierra. La **Figura 9** muestra la tensión de alimentación y su caja de metal. En la derecha tenemos una conexión fría. Esto es necesario para asegurar que la conexión de tierra puede hacer su trabajo de forma fiable.

Nosotros colocaremos el transformador dentro de esta fuente de alimentación (**Figura 10**). Una base estándar está montada verticalmente entre dos disipadores de aluminio. El disipador frontal enfría el transistor de conmuta-

ción en el lado primario, y el disipador posterior enfría los diodos rectificadores en el lado secundario. Toda la disposición está tan compactada que el orificio entre las placas de aluminio y el transformador no proporciona un aislamiento adecuado, por lo que para cumplir con las regulaciones se han fijado con pegamento dos hojas transparentes de material aislante (presumiblemente como una idea tardía) entre el transformador y la placa de aluminio. No es una idea elegante, pero sí efectiva.

En una construcción de este tipo, es esencial asegurar que el aislamiento entre la parte inferior de la placa de circuito impreso y la caja es adecuado. Frecuentemente las hojas aislantes se encuentran aquí como pequeños orificios de aire.

Por último vemos los condensadores llamados de clase Y, que normalmente se colocan para reducir interferencias, y en tal caso está permitido conectarlos entre partes del circuito de alta tensión (tales como el vivo de entrada) y tierra. La seguridad es particularmente crítica para esas capacidades, y deben ser de tipo apropiado. El circuito no debe exceder la corriente de fuga a tierra prescrita. La **Figura 11** muestra un ejemplo de este tipo de condensador. El valor de la capacidad (472 significa 4,7 nF) es difícil de encontrar entre los proveedores y marcas. Este condensador se puede conectar entre las partes

de alta tensión del circuito y el secundario o caja metálica en equipos de Clase II. Si un condensador de estos falla (por ejemplo si pasara una señal de 50 Hz) el lado secundario no debería estar más aislado que el primario. Dicho fallo de un simple componente podría tener consecuencias inciertas, lo que explica lo estrictas que deben ser las especificaciones que debemos encontrar. Debido a que las conexiones a un condensador de clase Y generalmente puentean la barrera de aislamiento (ver Figura 5), deben estar adecuadamente separados.

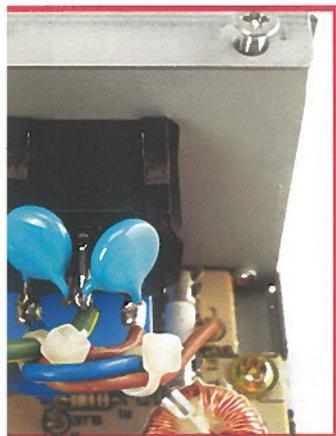


Figura 10. Hojas de aislamiento colocadas entre el transformador y los disipadores.



Figura 11. Un condensador de clase Y aprobado.

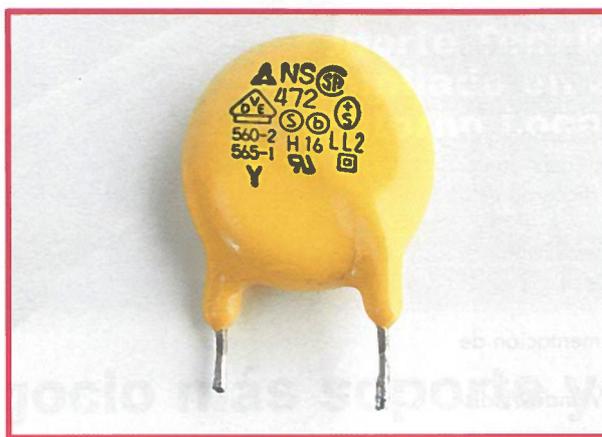


Figura 12. Fuente de alimentación para lámparas halógenas.

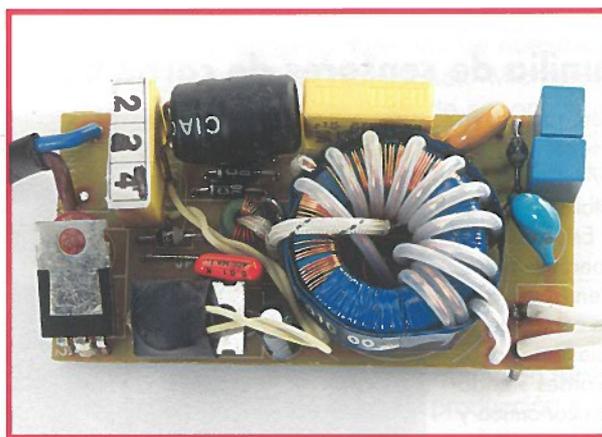
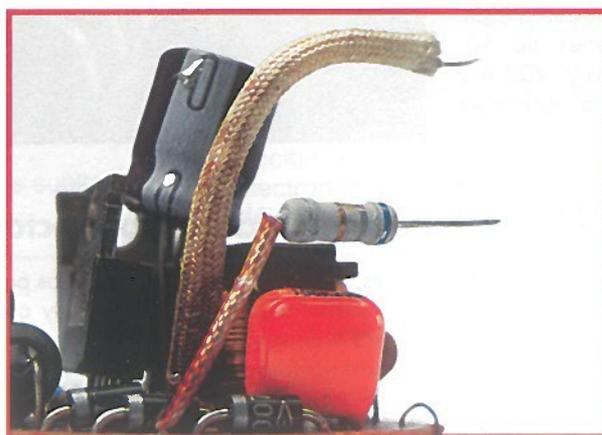


Figura 13. El 'fusistor' es un fusible y una resistencia en un dispositivo.



fuente de alimentación de un PC.

Fallo seguro

El sobrecalentamiento es una de los aspectos más importantes que tiene que considerar un diseñador cuando ocurre un fallo. En casos extremos un fallo puede provocar un incendio, por lo que muchas fuentes de alimentación conmutadas incluyen protección contra sobrecalentamiento. La **Figura 12** muestra un ejemplo de una fuente de alimentación para lámparas halógenas. Unos de los dos transistores de potencia está fijado con un sensor de temperatura que inhibe el convertidor si la temperatura llega a ser demasiado alta. Esta clase de diseño también es útil si una fuente conmutada debe resistir sobrecargas o corto-circuitos en su lado secundario. En general es necesario considerar el efecto de un fallo en cualquier componente en el diseño de la fuente; sólo entonces podemos también considerar la incorporación de un fusible.

La **Figura 13** muestra una interesante posibilidad, el 'fusistor' es una resistencia que limita la corriente en el encendido. La resistencia se quema con una corriente de sobrecarga sin riesgo de producir fuego. Ésta es una solución ideal en muchas aplicaciones.

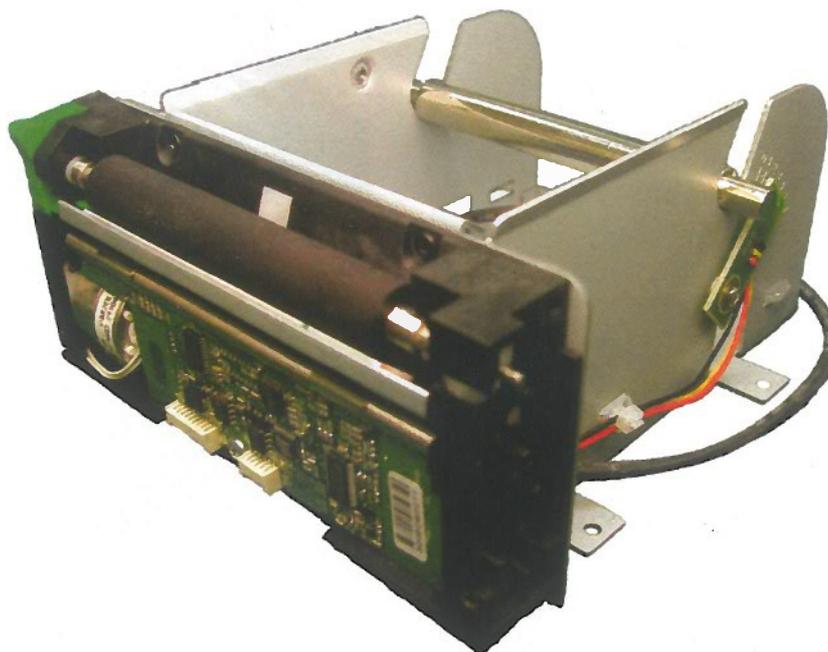
(050327-1)

Nueva impresora térmica KM324-HRS

APS, compañía representada por Monolitic acaba de lanzar al mercado la nueva KM324-HRS. Se trata de una nueva impresora térmica tipo kiosco entre cuyas principales ventajas destaca sus reducidas dimensiones de tan solo 121 x 73 x 147 mm.

Esto la hace ideal para un buen número de aplicaciones como máquinas de vending, expendedoras de billetes, etc donde la falta de espacio es un condicionante importante. Otras características de esta impresora son:

- Ancho de papel 80mm
- Comunicación RS-232 / USB 2.0
- Tensión de alimentación de 24V
- Drivers para Windows disponibles
- Opción cutter disponible

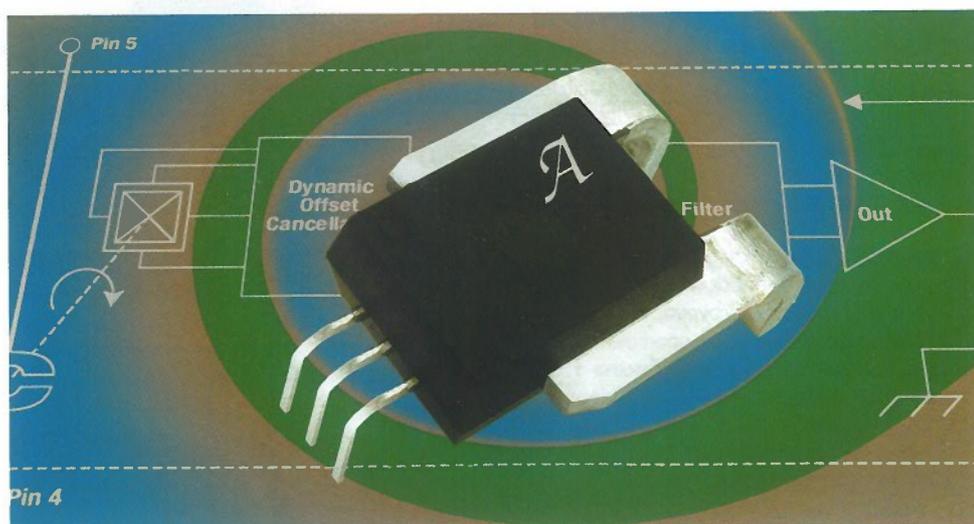


Nueva impresora KM324-HRS, para las necesidades en las que la falta de espacio es importante.

Nueva familia de sensores de corriente

Monolitic presenta la nueva familia de sensores de corriente ACS75x basados en tecnología Hall de Allegro Microsystems.. Estos dispositivos proporcionan numerosas ventajas en cuanto a tamaño y costes con respecto a las tradicionales soluciones basadas en toroides siendo una alternativa económica y precisa para innumerables aplicaciones.

Estos dispositivos se presentan en versiones de 50, 100, 130, 150 y 200 A y cumplen con la normativa RoHs.



Familia MFR, equipos de soldadura perfectos para procesos libres de plomo

Sagitrón, distribuidor para España de **OK Internacional**, fabricante líder durante más de 50 años en herramientas para la electrónica como soldadores, desolda-

dores, máquinas de reposición presenta la nueva familia **MFR** de soldadores/desoldadores preparados para la soldadura libre de plomo. Estos nuevos sistemas tienen

altas prestaciones y un precio muy competitivo. El multifuncional sistema **MFR** es una herramienta muy versátil que no solo ofrece muy buenas prestaciones en trabajos de reu-

tilización o reproceso sino que también es una herramienta para soldadura en SMD o en montaje de inserción.

Este nuevo equipo **MFR** tiene dos salidas conmutables

con la posibilidad de configurarlas con una pinza de precisión, una pinza estándar, un soldador manual, un soldador manual especial para producción con más potencia y una pistola desoldadora ajustable. Este sistema se puede adaptar a las necesidades exactas de cada cliente porque todas las unidades incluyen funciones de ajuste como "Auto Standby" y "Auto off" las cuales pueden ser programadas con diferentes temporizaciones con el fin de proteger a las puntas sobretodo

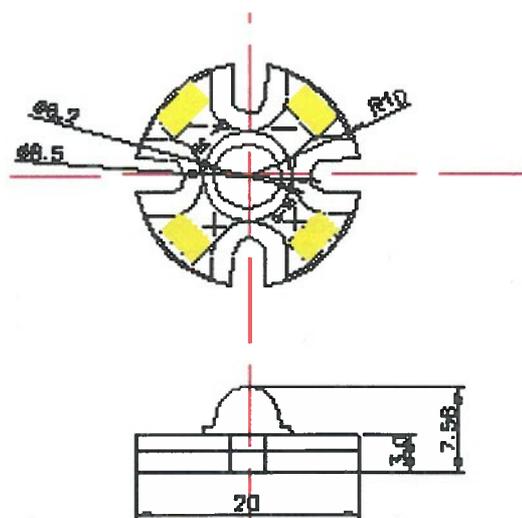
trabajando con soldadura libre de plomo. La serie **MFR** usa la tecnología **SmartHeat®** para entregar toda la potencia requerida por la soldadura libre de plomo. La tecnología **SmartHeat®** detecta la energía térmica demandada por la huella de la PCB y varía la potencia instantáneamente entregando de una forma precisa y continua la potencia necesaria; produciendo una gran calidad en la unión de la soldadura mientras protege tanto al componente como a la huella de la PCB.



Diodos LED de 1 watio de potencia para aplicaciones de iluminación

Sagitrón, distribuidor para España de **Quantum**, presenta la gama de LEDs de 1 watio de potencia con una eficiencia luminosa. Se disponen de toda la gama básica de colores, que incluyen: el rojo, amarillo, verde, azul, blanco y RGB con tres LEDs en el mismo encapsulado permitiendo así hasta 3Wattios totales con los tres LEDs al máximo de funcionamiento. Estos leds se controlan como los LEDs estándar, aplicando una corriente en directo de 350 mA para obtener una eficiencia óptima, con caídas de tensión de alrededor de 2V para el rojo y el amarillo con una eficiencia luminosa de 20 a 35 lúmenes y de 3,5V para el verde, azul y blanco, con 25, 12 y 50 lúmenes respectivamente. Esta familia de led utiliza un encapsulado basado en una base circular de aluminio con 4 muescas para fijar con unos tornillos a una superficie de

soporte y radiador, sobre la que se solapa una superficie de fibra de vidrio con la huella para la soldadura de los contactos hacia el exterior y la conexión interna al semiconductor alojado sobre un cono y recubierto todo por una resina de epoxi transparente, resistente a los Ultravioletas, que por la forma hace de lente a la vez que protege contra humedad y agentes externos. Aunque todas se basan en un mismo semiconductor para cada color, se dispone de distintas series según el ángulo de emisión que son **QT-50Kxx310** para 80°, **QT-50Kxx610** para 120° y **QT-50Kxx810** para 165°. También se pueden utilizar lentes externas para abrir o cerrar mas el ángulo de emisión. Las aplicaciones típicas de este tipo de LEDs son la de iluminación decorativa, retroiluminación de paneles informativos, indicadores lumino-



sos, semáforos, pantallas de visualización y en general la sustitución de otras fuentes de luz menos eficientes energéticamente y con menor vida y fiabilidad. En este mismo encapsulado con 4 terminales existe una versión RGB **QT-50KFC310** con 80° que permitirá realizar

efectos de iluminación con juegos dinámicos en toda la gama de colores, pantallas de imágenes o video para exterior y en general aplicaciones de iluminación, información o señalización utilizando toda la gama de colores. Se suministran en tubos con 20 unidades en cada uno.

Indicador Multifunción MICRA-M

El MICRA-M es un nuevo indicador multifunción programable para: célula de

carga, proceso y temperatura, que además ofrece un nuevo display tricolor

(Verde, Naranja y Rojo) para cada fase de medición, programación o alarma.

Con display: ± 19999 (4 dígitos), es fácilmente escalable, cuenta con funcio-

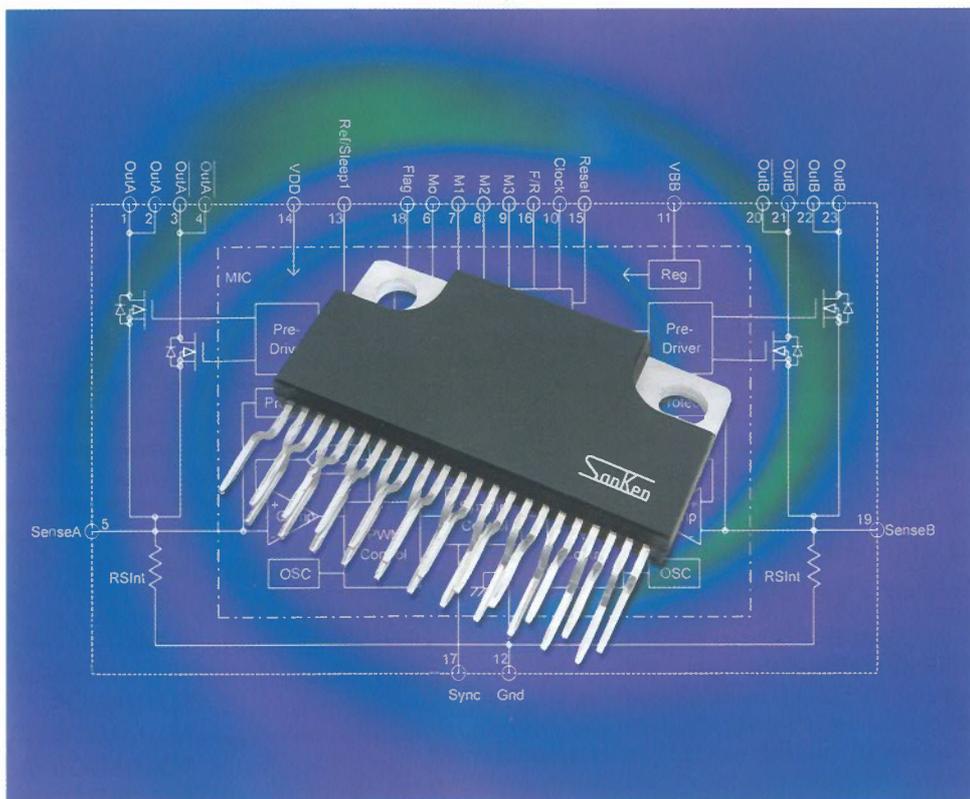
nes TARA, MAX/MIN y FILTRO, además de otras 12 funciones lógicas programables. Permite alimentación: Universal (85-265 Vac) y Baja tensión (10,5-70 Vdc), es totalmente configurable desde PC con software gratuito, Resolución interna ± 15 Bits y Linealización por tramos (10 segmentos).

Cuenta con tres opciones simultáneas: Contacto: 2 Réles, 4 Réles, 4 Optos PNP ó 4 Optos NPN; Comunicación: RS232C, RS485 (Protocolo MODBUS RTU) y Salidas Analógicas: 4-20 mA y 0-10 V. Este nuevo indicador digital perteneciente a la familia MICRA es ideal para solucionar su necesidad de control y/o medición en la empresa.



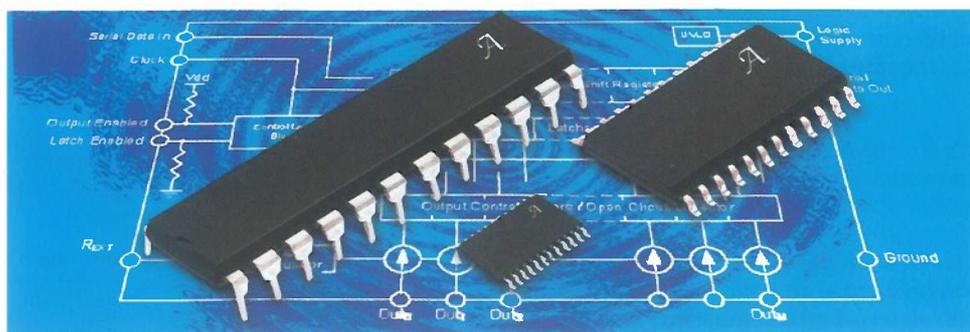
Nuevo driver de motor paso a paso unipolar bifásico

Monolithic presenta el nuevo SLA7070MPR de Allegro MicroSystems, un nuevo driver de motor paso a paso unipolar bifásico diseñado para simplificar el diseño del control, reduciendo el espacio y manteniendo un alto grado de fiabilidad. Incluye una resistencia interna de sensado y un circuito de protección contra cortocircuito de la carga y contra condiciones de trabajo en vacío. Opera con tensiones desde 10V a 44V y rangos de corriente de 1 A, 1,5 A, 2 A y 3 A. Puede ser atacado directamente con tensiones lógicas 3,3V y 5,5V. Las aplicaciones potenciales incluyen control industrial, sistemas de seguridad y máquinas recreativas así como todas aquellas que requieran un control paso a paso compacto y de fácil implementación.



familia A6279 de Allegro MicroSystems, un driver de 16 bits

Monolithic presenta la familia A6279 de Allegro MicroSystems, un driver de 16 bits para LED, de corriente constante y entrada serie que incluye una función de detección de fallo del LED (LED abierto). El A6279 permite el control directo desde un microprocesador a 3,3 o 5V y velocidades de entra-



da de datos de hasta 25MHz. La corriente de los LED's se determina mediante

la selección de una única resistencia permaneciendo ésta constante independien-

temente de las variaciones en las caídas de tensión de los LED's, de la tensión de

alimentación o cualquier otro parámetro que pudiese afectar dicha corriente.

Toppoly lanza una nueva gama de productos

Toppoly Optoelectronics Corp, empresa representada en España por **Anatronic, S.A.**, ha anunciado una nueva gama de productos para teléfonos móviles, PDA, DSC / DVC y dispositivos handheld digitales. Entre los nuevos modelos destacan el panel TFT-LCD LTPS transreflectivo VGA de 3.5" y los paneles AMOLED de 2.5 y 7", así como un panel transreflectivo de 3.5", basados en tecnología System-On-Chip totalmente integrada.

El TFT-LCD LTPS de 3.5 es el primer panel que soporta resolución VGA (480 x 640) en displays transreflectivos, que están especialmente diseñados para PDA high-end, teléfonos PDA y teléfonos inteligentes.

Además de por la resolución VGA, el nuevo panel TFT-LCD LTPS se caracteriza por un brillo de hasta 150 nit, ángulos de visión de hasta 160°, 262 K colores, ele-



Paneles TFT-LCD LTPS y AMOLED para teléfonos móviles, PDA y dispositivos handheld digitales.

vado ratio de contraste de 180:1 y pantalla táctil para facilitar la operación. Este nuevo producto ha sido diseñado para ser compatible con la normativa RoHS.

Los paneles AMOLED han sido desarrollados para displays en

vehículos con elevada calidad de imagen, siendo ideales para navegadores, televisores y sistemas de entretenimiento en coches, soportando WVGA (800 x 600) con resoluciones de hasta 135 ppi y 262 k colores.

Los dispositivos AMOLED son los únicos paneles de este tipo capaces de combinar el circuito de compensación avanzada de píxeles con las tecnologías OLED y COA (Color Filter on Array) de la compañía.

Nuevos MCU Flash AVR con USB y OTG

Atmel® Corporation, empresa representada en España por **Anatronic, S.A.**, anuncia cuatro nuevos MCU Flash AVR® con controlador USB. Los microcontroladores AT90USB han sido diseñados para responder a la amplia variedad de requerimientos de aplicaciones embebidas que necesitan conectividad USB. Existen millones de dispositivos USB en uso para ordenadores y periféricos, por lo que el USB está ganando impor-



Ofrecen una combinación de interfaces versátiles, USB On-The-Go y flexibilidad.

tancia en productos de consumo y equipos industriales. Almacenamiento y transferencia de datos, configuración de producto y actualizaciones de firmware son las aplicaciones esenciales que están dirigiendo esta tendencia.

El AT90USB1286 y el AT90USB646 tienen interface USB para aplicaciones que necesitan comunicar con host USB, mientras que los modelos AT90USB1287 y AT90USB647

cumplen con el estándar USB On-The-Go (OTG) para uso como Dual Role Devices (DRD) en aplicaciones que operan como host o función en el USB. La capacidad de host USB es esencial para dispositivos embebidos que requieren comunicar sin intervención de PC.

El AT90USB1286 y el AT90USB1287 poseen 128 Kbytes de Flash ISP, 8 Kbytes de RAM y 4 Kbytes de EEPROM, en tanto que los

MCU AT90USB646 y AT90USB647 son idénticos pero con la mitad de tamaño de memoria. Todos los dispositivos tienen un cargador boot on-chip que permite ISP a través del bus USB, ofreciendo una flexibilidad inigualable, desde la fase de desarrollo a actualizaciones de campo.

También se ha desarrollado un nuevo controlador USB para soportar todos los modos USB y OTG, pudiendo ser configu-

rado para operar a baja velocidad de 1.5 Mb/s o alta velocidad a 12 Mb/s. La velocidad reducida ofrece soporte para un amplio número de dispositivos de bajo coste como ratones y teclados, mientras que la alta velocidad es suficiente para transferir varios Mb por segundo.

Atmel ofrece una librería de software, así como herramientas de evaluación y depuración.

Ventiladores de gran calidad

Monolithic, S.A como distribuidor de Bi-Sonic, les presenta su gama de ventiladores de gran calidad final, disponiendo de un amplio catálogo con varios tipos de modelos disponibles para los diferentes estándares existen-

tes (desde 25mm a 120mm de lado), tanto en alterna como continua (5-48Vdc; 100-230Vac), disponibles en modelo de bolas y de rozamiento. Todos estos productos ya disponibles según la normativa RoHS Compliance.



Conversores de RS232 a RS485

CelINCOM, Conversores e Interfaces de Comunicaciones, S.L., anuncia la disponibilidad de la familia IC485IP de conversores de RS232 a RS485 de su representada **CTC Union**.

Esta familia de conversores IP tan sólo requiere un cable con un conector apropiado y seleccionar el modo de trabajo DTE/DCE requerido. En concreto el modelo IC485IP-1F, uno de los más

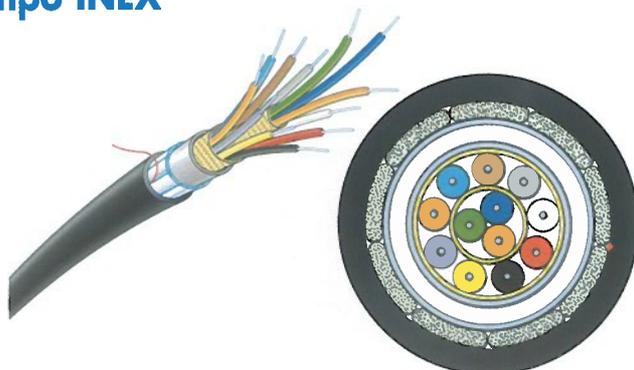
demandados actualmente, es un convertidor de RS232 (con conector DB25 hembra) a RS485 (con cuatro tornillos en Terminal block) que se caracteriza por venir equipado con un dispositivo para seleccionar el modo de trabajo DTE o DCE. El IC485IP-1F permite la selección de simulación y monitorización, control RTS/CTS en half y full duplex, velocidad de hasta 128 Kbps y LED indicadores de alimentación externa DC.



Cable ajustado de interior - exterior tipo INEX

GRUPOPROMAEL presenta su cable ajustado de interior - exterior dieléctrico tipo INEX, que está especialmente indicado para redes LAN o entornos industriales, gracias a sus características de gran flexibilidad, resistencia al aplastamiento y protección contra la humedad estructural fibra a fibra.

Este cable posee un elemento de refuerzo de fibra de vidrio, protección contra roedores mediante hebras de fibra de vidrio, cubierta exterior LSZH (libre de halógenos) y protección contra la humedad de estructura ajustada y waterblocking. Este cable robusto y muy flexible, que no propaga la llama y



ofrece baja emisión de humos, permite una fácil conectorización directa y es muy fácil de pelar (libre de gel).

El cable tipo INEX, que tiene entre cuatro y dieciséis fibras ópticas según el modelo, también destaca

por un diámetro de 4,8 a 6,9 mm, peso de 24 a 46 Kg / Km, tracción máxima instantánea de 900 a 1450

N, tracción máxima permanente de entre 200 y 320 N y radio de curvatura de 50 a 70 mm.

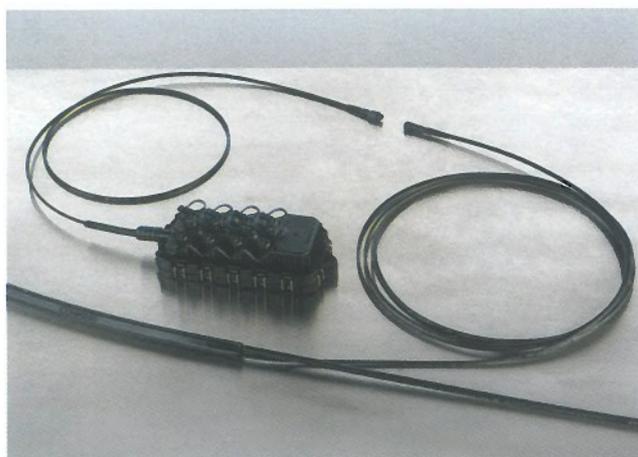
Sistemas OSP Plug & Play para despliegue de fibra en redes LAN exteriores

CMATIC, S.L., empresa dedicada a la distribución de materiales para redes LAN, introduce los sistemas OSP Plug & Play™ de **Corning Cable Systems**, que permiten unas instalaciones más rápidas y económicas en redes de área local (LAN) de planta externa. Los sistemas OSP Plug & Play se componen de cable óptico pre-instalado con puntos de acceso de red con intervalos especificados por el cliente y un conector MT muy resistente. Los puntos de acceso de red MT están formados por un cierre pre-instalado flexible y patentado. Un cable plano se une al punto de acceso de red y se completa con un

conector MT muy resistente y totalmente comprobado. Cuando se requiere conexión de red en la zona de desnivel, el cliente tiene diversas opciones: un conector MTP® de 12 fibras, un dispositivo estándar tipo LC o SC o un terminal multipuerto OptiS-heath™, que se puede configurar con cuatro, seis, ocho o doce puertos de cable. Compatible con aplicaciones subterráneas y aéreas, el Plug & Play OSP System permite que la instalación de red se complete de una forma más eficiente y con velocidades hasta un cincuenta por ciento más rápidas que las instalaciones de campo tradiciona-

les, con un ahorro de hasta el veinte por ciento. Con el sistema OSP Plug & Play el riesgo de fallos se

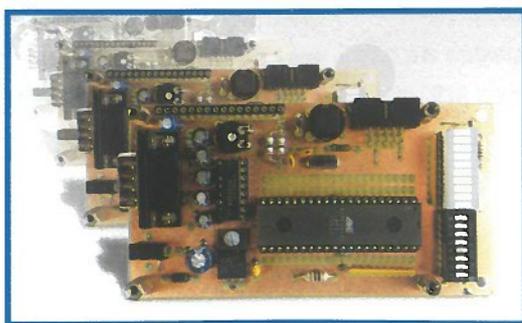
reduce, debido a que cada cable del sistema ha sido comprobado en la fábrica antes de su envío al cliente.



PRÓXIMO NÚMERO PRÓXIMO MES EN ELEKTOR

La mini Tarjeta ATmega

Aunque el microcontrolador de ATmega no es necesariamente mega, sigue siendo absolutamente popular y nos sentimos que realmente no ha recibido la atención que merece. La mini tarjeta de ATmega descrito aquí pone un cambio a eso, y un uso agradable para ello.

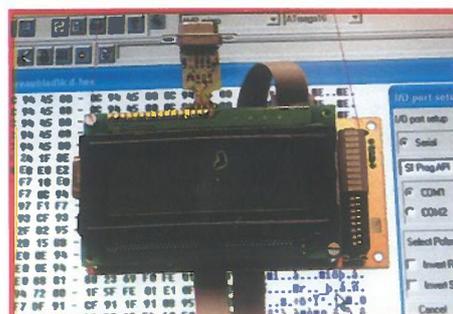


Valla electrificada

¿Los perros y los gatos merodean y cavan en nuestro jardín? Incluso nuestra propia mascota debe ser disuadida de entrar en una área restringida. Este generador para valla electrificada puede ser una respuesta a nuestros problemas. Un simple generador construido alrededor de un 555 y una vieja bobina es suficiente para realizar un circuito con un pequeño desembolso.

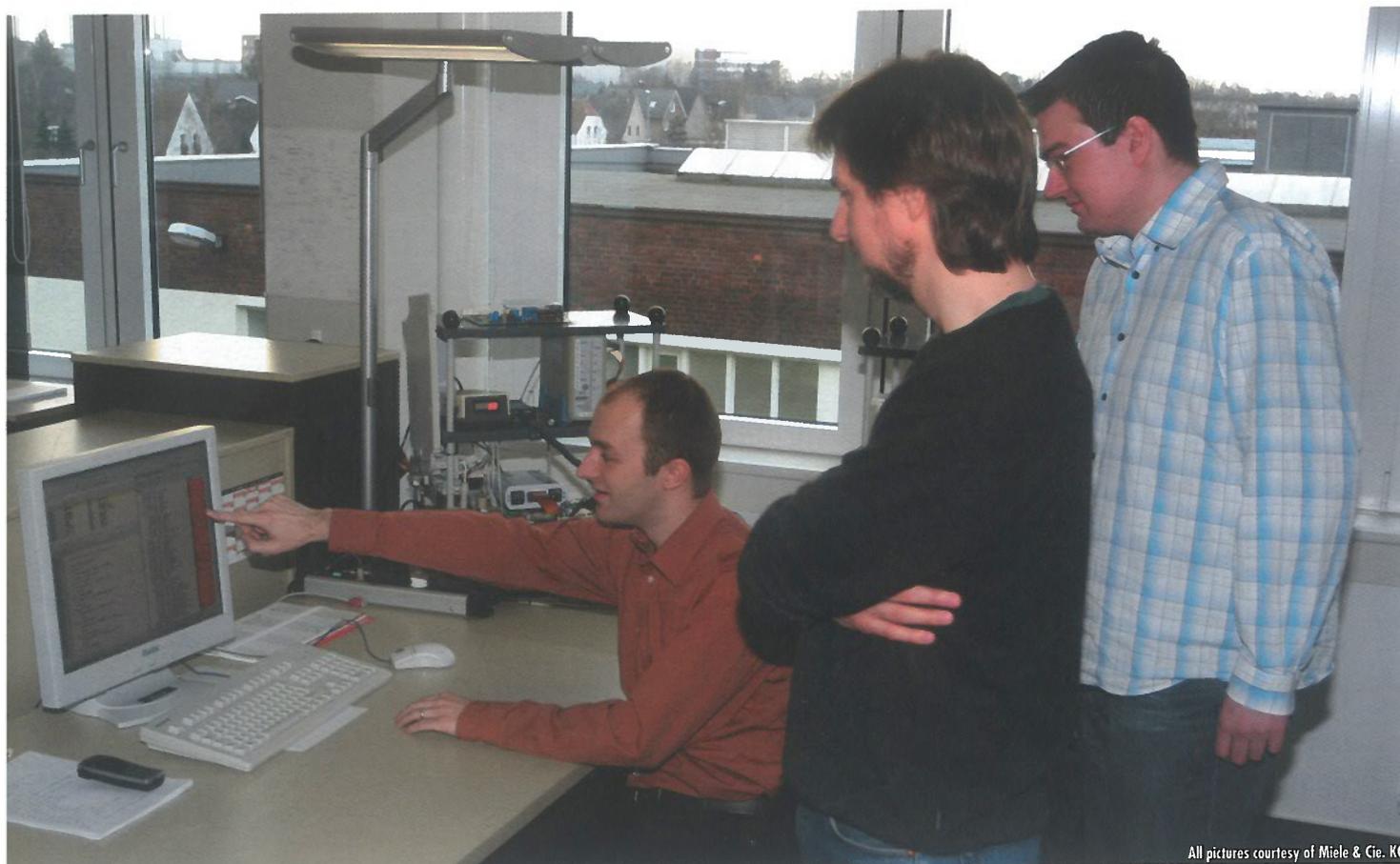
Analizador EOBD único

Usando el conector EOBD, disponible virtualmente en todos los modernos coches, muchos datos del motor se pueden leer, incluyendo la velocidad del motor, potencia, temperatura, mezcla de gasoil y, por supuesto, los códigos de error producidos. En el número de Mayo del 2006 publicaremos una ampliación para el popular Analizador OBD-2 del número especial de verano del 2005. Así, ahora podemos leer datos OBD en tiempo real. Nuestra placa de microcontrolador utiliza un ATmega 16 y los datos se visualizan en el módulo LCD con 4 líneas de 20 caracteres.



Diseños para las

Jens Nickel



All pictures courtesy of Miele & Cie. KG

El omnipresente microcontrolador no solamente aparece en algunos de nuestros proyectos, sino que también está presente en millones de aplicaciones domésticas vendidas a lo largo del mundo. ¿Cómo pueden los profesionales gestionar el diseño y el desarrollo de un nuevo producto anticipando que miles de ellos estarán pronto en la línea de producción? Nos hemos asomado a la casa Miele para ver cómo lo hacen.

Si pedimos a alguien que piense en una ciudad alemana, muy pocos pensarán en Gütersloh. Incluso en la propia Alemania es una ciudad poco conocida. Sin embargo, se trata de una de las más grandes ciudades de fabricantes de electrodomésticos de línea blanca de Alemania. La compañía Miele ha estado produciendo y exportando equipos de limpieza por vacío, lavadoras, secadoras y hornos a más de ciento cincuenta países durante casi cien años. Un frío día de enero viajamos hasta Gütersloh y nos entrevistamos con Reinhild Portmann, la persona encargada de la oficina de prensa de Miele. Esta empresa tiene alrededor de 15.000 empleados y tuvo un total de facturación de unos 2,26 billones de euros en el año 2005. La producción se realizó tanto en Alemania como en Austria, con lavadoras y secadoras que salían

de la línea de producción de Gütersloh. Las prestaciones más relevantes de una lavadora de la casa Miele son su fiabilidad y su longevidad, cualidades que han sido valoradas por los clientes de todo el mundo.

"Nuestras lavadoras han sido diseñadas para tener una vida de trabajo de 10.000 horas, lo que equivale a unos 20 años de servicio", explica Josef Avenwedde, uno de los responsables de la casa Miele. Para remarcar la filosofía de la compañía y cumplir con el compromiso de la calidad, a la cabeza del desarrollo electrónico, vamos a analizar el papel que realiza el departamento de servicio técnico. El diseño de la máquina asegura que el personal del servicio de mantenimiento podría cambiar algunos parámetros operacionales (ver **Figura 1**). Esto proporciona un cierto grado de "adaptación futura"

del diseño, algo que es muy importante para una máquina con unas expectativas de vida de veinte años. Como ejemplo podemos poner el de una nueva familia de detergentes en polvo que se introduzca en el mercado en un futuro próximo y que requiera unas temperaturas de lavado más bajas o una menor cantidad de agua de aclarado. El servicio técnico puede ajustar estos dos parámetros durante la puesta en funcionamiento, para aprovechar esta nueva tecnología. Esta filosofía ha sido muy bien acogida por todos en general, bajo el principio de diseño de "construir con obsolescencia", principio al que también se han acostumbrado los consumidores. Una interfaz óptica para el servicio de mantenimiento, montada sobre el panel frontal de la máquina, permite un acceso adecuado para "volver a programar" las prestaciones de la máquina. Uno de los ayudantes de Josef Avenwedde, Ernst Hokamp (responsables del grupo de productos de lavado) explica que la interfaz de servicio fue añadida durante el diseño de la máquina con un pequeño coste adicional en el que sencillamente se incluía un fototransistor como receptor óptico y se utilizaba un diodo LED para transmitir la información. La competencia en el mercado de los productos de línea blanca es feroz y los fabricantes están ideando continuamente nuevos métodos para reducir costes. La casa Miele produce la mayoría de su trabajo de ensamblado de placas de circuito impreso (PCB), con un 20 % de subcontratación exterior. "Nuestra propia producción debe permanecer competitiva con el resto del mercado", enfatiza Josef Avenwedde.

Pioneros en SMD

En 1989 la casa Miele fue el primer fabricante de gran volumen de electrodomésticos de línea blanca que utilizó componentes con encapsulado SMD. Para aprovechar el ahorro que suponían los componentes SMD tuvo que diseñarse de nuevo el selector de programas rotativo que estaba en el panel frontal de las lavadoras. En aquellos tiempos había muy pocas alternativas de contactos electro-mecánicos que detectasen la posición de conmutación. El laboratorio de desarrollo presentó una versión optoelectrónica que utilizaba una serie de diodos LEDs SMD colocados formando un círculo, con un pequeño tubo que transfería la señal de uno de los diodos LEDs a un fototransistor SMD central, para indicar la conmutación de posición. A mediados de los años 80 se introdujeron los microcontroladores en las máquinas (vea la **Figura 2**). "Originalmente, utilizábamos un microcontrolador para controlar tan sólo el motor" nos comenta Josef Avenwedde. "Antes de esto se utilizaban los triacs y los tiristores junto con un grupo de componentes discretos". El primer microcontrolador que la compañía utilizó en una producción de gran cantidad fue un dispositivo de 8 bits con una memoria ROM de 4 KBytes. En aquellos tiempos, la única opción de programación era utilizar el lenguaje ensamblador. En 1992 la compañía cambió, y empezó a utilizar el microcontrolador de ocho bits, M37451, de la casa Mitsubishi (con un núcleo de 6502). La compañía japonesa también



Figura 1. Los parámetros del programa de una lavadora pueden modificarse mediante una interfaz práctica del servicio de mantenimiento (cubierta en este caso por la copa de succión). Esto permite que el ciclo de lavado de la lavadora pueda adaptarse a los requerimientos de cualquier lavadora más potente (y más eficiente) que aparezca en el futuro.

había desarrollado una herramienta de programación que denominaba "ensamblador estructurado", que incorporaba algunos elementos de lenguajes de programación de alto nivel. Esto permitió alcanzar una productividad mucho

Figura 2. Los avances técnicos en los últimos veinte años han llegado a producir un rápido incremento en las prestaciones que podemos encontrar en las lavadoras y secadoras de la casa Miele. La potencia en los microcontroladores y la capacidad de memoria también han experimentado su correspondiente incremento.

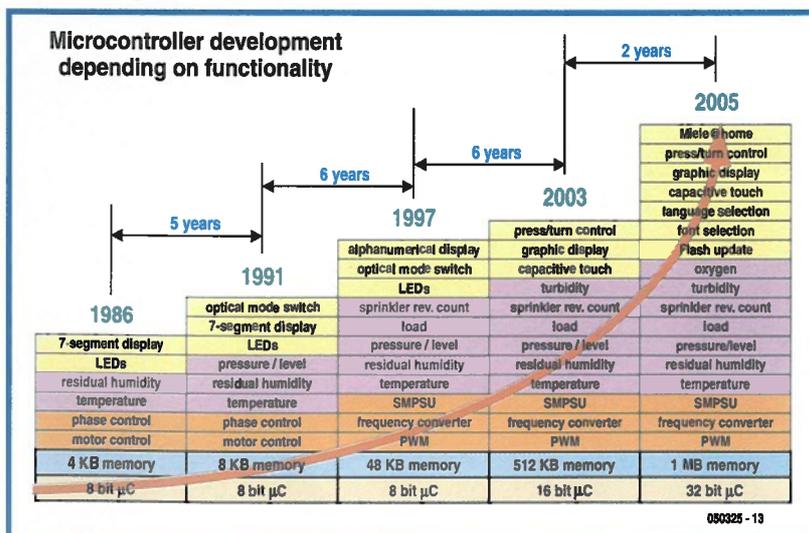




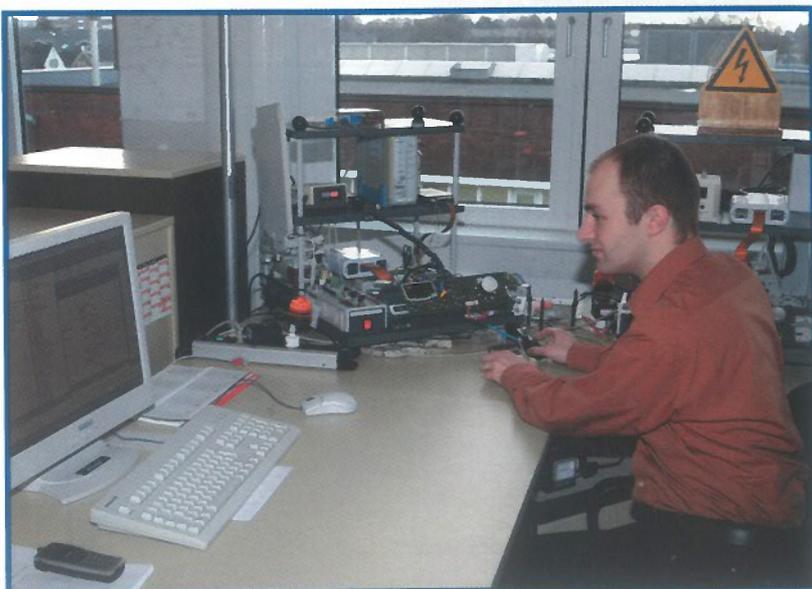
Figura 3. Con una estación de soldadura, un transformador variable y un osciloscopio, el ingenio de componentes Martin Schulze-Hobeling podría trabajar en cualquier departamento de diseño, aunque la panorámica proporciona ciertos indicios del tipo de industria en la que está trabajando.

mayor y fue una mejora sorprendente. Con una sonrisa nuestro acompañante recuerda su primer año en Miele, cuando los ingenieros de programación debían escribir líneas de código utilizando el programa de edición "WordPerfect". Afortunadamente, el proceso de escritura de código hoy día está mucho más estructurado.

C, el lenguaje de las opciones

En el año 2001 se tomó la decisión de utilizar el lenguaje de programación C para todos los desarrollos software, al mismo tiempo que se cambiaba la circuitería y

Figura 4. Andreas Lux está verificando aquí los módulos de programas en un entorno emulador. Los niveles de agua son simulados en la maquinación con la ayuda de una bomba de presión.



se adoptaba el uso de microcontroladores de 16 bits. Fue un poco sorprendente que el procesador elegido fuese el modelo M16C de 16 bits de la casa Renesas, el hermano mayor del microcontrolador de ocho bits R8C. En los modelos de lavadoras más caras este microcontrolador se encargaba del control del motor, de las lecturas de las entradas de los sensores y de los mensajes de salida hacia las pantallas visualizadoras. Para ayudar a mantener los costes de producción lo más bajos posible, así como la cantidad de electrónica implementada, estos elementos fueron integrados en la placa de control principal. Un ejemplo de esto es el sensor de presión de agua construido por la casa Motorola. Este dispositivo controla la cantidad de agua utilizada para cada lavado, dependiendo de la carga de ropa de lavado y de su absorbencia. El sensor de presión está montado directamente sobre la placa del controlador y captura la información del nivel de agua a través de un delgado tubo de plástico. Montado sobre la placa del controlador y al lado del sensor de presión, existe un ASIC que se encarga del acondicionamiento de la señal y de su momento de entrada y salida.

Este método de construcción, donde se utiliza una única placa controladora que incluye el módulo visualizador, se ha aplicado a un gran rango de productos. Cuando un componente falla, sencillamente se sustituye la placa del controlador completa (a menudo el fallo puede repararse en fábrica y la placa vuelve al almacén donde podrá suministrarse como un elemento de reposición). El objetivo del desarrollo explicado aquí tiene un coste de reparación relativamente superior pero los costos de producción se mantienen a niveles mínimos.

Eliminación de los problemas en los programas

La casa Miele también suministra máquinas para usos comerciales. "Estas máquinas están más o menos en uso continuo" nos comenta Ernst Hokamp, "y reciben un trato bastante duro, por lo que realizamos grandes esfuerzos en asegurarnos que sean particularmente fiables y robustas". Esto es cierto para las partes mecánicas; el selector de programas rotativo, por ejemplo, está montado con rodamientos y el mismo cuidado del detalle se aprecia en el diseño de la electrónica. Las lavadoras para uso comercial contienen dos microcontroladores M16C, uno se encarga del funcionamiento de la máquina mientras que el otro es responsable del control. En este caso el trabajo de control del motor se realiza por un circuito integrado especial de la casa NEC.

Cualquier fallo de la máquina en un entorno comercial puede ser bastante costoso, pero esto también es cierto en el mercado de gran consumo, donde la casa Miele tiene una buena reputación en fiabilidad, la cual se puede ver fácilmente dañada si no se mantienen las prestaciones estándar. Basta con un pequeño error en el código de los programas para que el programa de control se quede colgado, algo que sería inaceptable. Ernst Hokamp nos expresó el orgullo que sentía hacia su equipo y una cierta satisfacción, ya que nunca ha tenido que retirar un producto del mercado o traerlo a fábrica para una actualización de sus programas internos. Pasamos por un área en el departamento de diseño donde están colocándose las máquinas modificadas y los nuevos modelos. La mayoría de ellas estaban conectadas a ordenadores que ejecutaban el programa de prueba "DIAdem", de la casa National Instruments. Un operario tenía el trabajo de verificar todas las posibles combinaciones del conmutador a mano. Las secuencias de apertura de la puerta, girar el selector

del programa y presionar botones, todas ellas tabuladas en una hoja de trabajo. El siguiente paso en este procedimiento dura varias semanas de pruebas en campo, donde las nuevas máquinas son instaladas en los hogares de algunos de los empleados para realizar refinamientos adicionales. La etapa siguiente es una fase de marketing donde la máquina se vende localmente en cantidades limitadas. Una de las empleadas mencionó que la casa Miele está realizando de manera continua pruebas en sus máquinas, en sus laboratorios, bajo la vigilancia de una cámara web, que puede verse a través de Internet. En el último momento se verifican las máquinas a las que se les está realizando unas rigurosas pruebas a lo largo de 10.000 horas de test.

Todos los desarrolladores de programas tienen acceso a una librería de rutinas de pruebas y ensayos a las que podrán recurrir si lo necesitan. Cada grupo de productos (lavadoras, cocinas, etc.) dispone de un equipo de entre 10 y 15 ingenieros que gestiona el desarrollo de nuevos productos. En algunos casos puede que tan sólo haya un único ingeniero que se encarga de investigar las necesidades electrónicas y de programa de un nuevo sensor (Figura 3).

Las herramientas profesionales ayudan a detectar y corregir errores

Los ingenieros del grupo de "lavadoras" no están totalmente enfocados hacia la programación. Andreas Lux es un joven ingeniero IT que durante nuestra visita estaba verificando y optimizando el programa de una lavadora ya existente. Su trabajo consistía en un conjunto de placas electrónicas que representaban la circuitería completa de una lavadora. Cualquier cambio en el código que realiza el ingeniero de programación puede pasarse instantáneamente en el compilador y verificarse sobre los circuitos. En este entorno de pruebas también se incluyen los sensores y los actuadores, mientras que los sensores de presión se verifican con la ayuda de unas pequeñas bombas de mano y un manómetro (Figura 4). El entorno Windows del PC en su banco de trabajo es similar al utilizado por sus colegas en el departamento y no es especialmente rápido. Sin embargo, la casa Miele ha realizado grandes inversiones en herramientas software para desarrollos de programas. El código es generado por un ingeniero que utiliza la herramienta "Codewright" de la casa Borland, que también se utiliza como un entorno de desarrollo integrado que hace llamadas a otras herramientas (Figura 5). El paquete de programas "Rhapsody" (de la casa iLogix) también se usa en este departamento. Este programa en particular es capaz de generar código C de manera automática una vez que se ha definido cuál va a ser el comportamiento del programa y se escribe utilizando un tipo especial de diagrama de flujo. Según nos comenta Ernst Hokamp, el código que produce este programa es difícil de leer e incluso Andreas Lux, que es un especialista en lenguaje C, tuvo que admitir que le llevo un cierto tiempo antes de que se sintiese un poco a gusto con el tipo de código elaborado y generado. En este punto tenemos que recordar una cosa, y es que el código es producido por el modelo de diagrama de flujo, de manera que si comenzamos a editar manualmente las líneas de código, nuestro modelo original ya no será válido.

Documentación automatizada

El "Development Assistant for C" (es decir, el "Asistente de Desarrollo para C"), es una herramienta muy útil para mantener el histórico del desarrollo de programas y verifi-

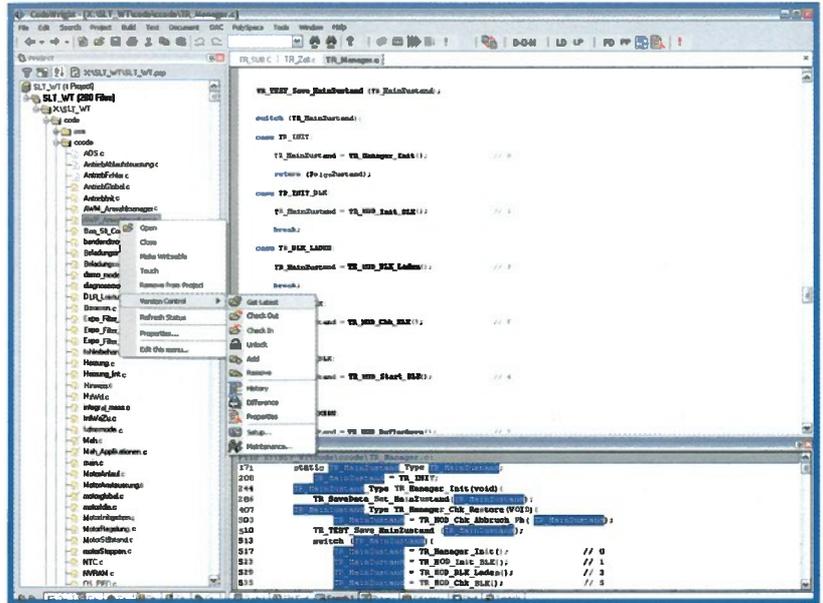
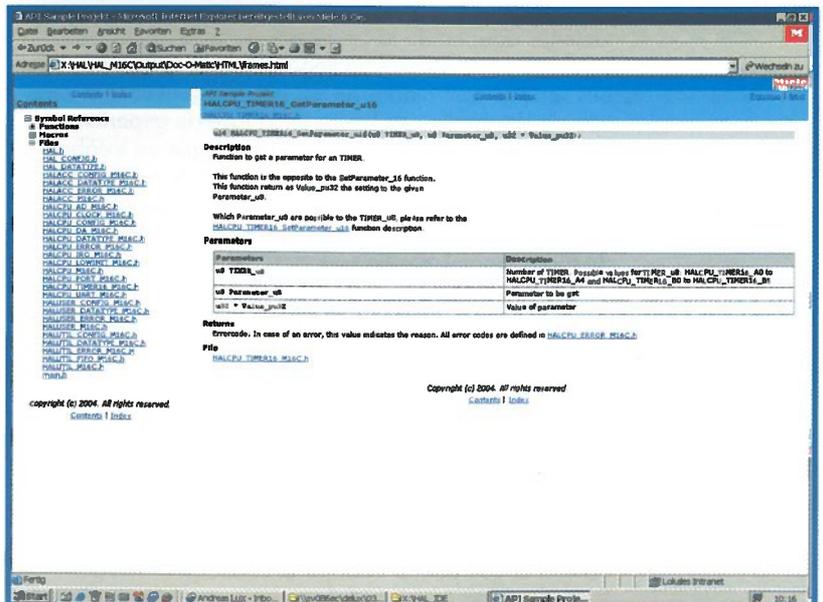


Figura 5. Los desarrolladores escriben el código en lenguaje C con la ayuda del editor "Codewright". Este programa gestiona el control de las versiones de forma automática. La parte izquierda de la ventana muestra los módulos que están en uso.

car que los programas están conformes con la normativa de codificación de la empresa. "Doc-O-Matic" es otra herramienta de programas que prácticamente se explica por sí misma, y es capaz de proporcionar documentación de programas de manera automática, como por ejemplo, para cada función en lenguaje C, detallando todas las variables sobre una hoja en formato HTML que se puede leer y entender fácilmente (Figura 6). En ningún caso este programa es un lujo, ya que no lo ha sido durante las 200.000 líneas de código anteriores generadas por todos los equipos de desarrollo dentro de la compañía. También se utiliza el programa "Polyspace", el cual

Figura 6. El programa "Doc-O-Matic" se encarga de la documentación de los programas.





Algunas cifras

Miele es uno de los pocos fabricantes de electrodomésticos de línea blanca cuya producción interna se produce en su lugar de origen. Esta compañía ensambla alrededor del 80% de todas sus placas. Aproximadamente 180 empleados trabajan en la ciudad de Gütersloh en ocho líneas de producción. La compañía utiliza 520 tipos diferentes de componentes y cada día monta 1,5 millones de ellos. Unas 50.000 placas abandonan la línea de producción cada semana.

Dispone de 14 máquinas para colocar y soldar componentes SMD (el más pequeño de ellos tienen un tamaño de 2,2 mm x 1,5 mm). Cada máquina es capaz de colocar 50.000 componentes por hora. Los componentes de tamaño estándar con terminales son tratados de manera diferente. Antes estos componentes tenían que ser montados y a continuación pasados al través de la ola de soldadura, donde la cara inferior completa de la placa llegaba a alcanzar una temperatura de hasta 250° C y era barrida con una ola de estaño fundido. Los trabajadores actuales de Gütersloh utilizan un sistema de soldadura selectivo que es más efectivo y que proporciona el calor a través de pequeños inyectores o "chimeneas" sólo en la zona de la placa donde se requiere realizar la soldadura. (véase la imagen asociada).

puede predecir errores en tiempo de ejecución durante la compilación de un programa! El sistema IAR proporciona el programa compilador, el "linkador" y el depurador, para desarrollos de programas, mientras que el sistema de funcionamiento en tiempo real "emBos" (de la casa Segger) sólo ocupa entre 1 y 2 KBytes de la memoria interna del controlador. La misma compañía suministra el "Graphical User Interface" (GUI, es decir, Interfaz Gráfico de Usuario), utilizado por los ingenieros de Gütersloh.

Sobrecarga de memoria

No nos sorprende que el código final para el controlador M16C no esté protegido contra lectura de personas no autorizadas o para realizar la programación inversa. Sin embargo, hay que reconocer que existe una pequeña dificultad para reconstruir un listado de código fuente principal a partir de los valores de código máquina almacenados en la memoria ROM. De acuerdo con lo que nos dijo Josef Avenweddes, y por lo que se sabía hasta el momento, aún no habían llegado noticias de alguien que hubiese intentado obtener mayores prestaciones adaptando la circuitería de su lavadora. El siguiente paso lógico que llega desde el uso de los 16 bits sería el controlador de 32 bits. Preguntamos si este tipo de desarrollo estaba ya en la fase de experimentación y Josef Avenweddes nos comentó que ya estaban usándose desde hacía algún tiempo en los hornos. La principal razón para hacer esto no era la potencia de procesamiento adicional sino el incremento del espacio de direccionamiento de memoria del procesador de 32 bits de la casa Renesas. El programa utilizando es muy largo ya que, por ejemplo, la información de la interfaz de visualizador, que se tiene que mostrar en 24 idiomas, también tiene que programarse en una memoria Flash. "Estos datos ocupan una gran cantidad de memoria, y en especial para los controladores de hornos, incluso en las lavadoras esta información ocupa alrededor de 160 KBytes de memoria disponible", nos explica Ernst Hokamp. "Si añadimos los 384 KBytes de código de programa estaremos muy al límite del espacio de memoria con el que puede trabajar el controlador M16C". Todas las máquinas vendidas a lo largo del mundo están

equipadas con el mismo programa que contiene los parámetros específicos de cada país, como la tensión de red y su frecuencia, almacenados de manera independiente en una pequeña memoria EEPROM. En dicha memoria también se almacenan algunos de los parámetros de funcionamiento de las máquinas, tal y como se ha mencionado anteriormente, como pueden ser las revoluciones de lavado, la temperatura, etc., que pueden volver a programarse a través de la interfaz óptica al que tiene acceso el servicio de mantenimiento. "También podemos volver a programar la memoria Flash utilizando esta interfaz, pero esta interfaz utiliza una técnica de "escritura de bloques" que no es muy conveniente si sólo queremos cambiar uno o dos parámetros. En este caso es mucho más fácil almacenarlos en una pequeña memoria EEPROM".

"Los nuevos avances en la industria de la electrónica no siempre son adecuados para nuestras aplicaciones" nos explica Josef Avenweddes. "Si tomamos, por ejemplo, el concepto de sensor "inteligente", la fabricación de un sensor junto con su interfaz electrónica es innecesariamente complejo y costoso, ya que estos dos elementos se producen mediante diferentes tecnologías". Los ahorros en el coste son bastante más elevados si los dos dispositivos son idénticos y están alojados en el mismo encapsulado. En el terreno económico, tampoco podremos encontrar una FPGA o incluso un mini ordenador montado dentro de ninguno de los productos actuales de la casa Miele. "Como equipo, estamos siempre discutiendo sobre nuevas ideas y vigilando constantemente las nuevas tendencias en la tecnología" nos explica el responsable del desarrollo de producto, "pero, por el momento, nuestros clientes nunca verán en nuestros productos ninguna etiqueta que diga "Intel inside" pegada en ninguna de las lavadoras que haya salido de la línea de producción de la casa Miele".

(050325-1)

Enlaces en Internet

www.miele.co.uk <http://eu.renesas.com>
www.zuken.com www.borland.com
www.ilogix.com www.ristancase.com/dac/v40/index.php
www.iar.com www.doc-o-matic.com/index.html
www.polyspace.com www.segger.com

Generación de Onda

John Dobson

En el artículo publicado en el número anterior generábamos una forma de onda de sonido de unos 400 Hz utilizando el Bloque Electrónico de un conversor D/A con bus SPI y una tarjeta de memoria. Usando el programa Flowcode pudimos ver que el diagrama de flujo se convierte primero en un módulo en lenguaje C y, a continuación, en código ensamblador, pero al mismo tiempo podíamos incluir instrucciones en lenguaje C dentro de nuestros programas generados con Flowcode, para aumentar la velocidad de trabajo. En este artículo describimos cómo utilizar el lenguaje C para aumentar la velocidad del generador de la forma de onda de sonido. Simultáneamente vamos a descubrir a nuestros lectores cómo usar el programa Flowcode para aprender más sobre programación en lenguaje C.



Para generar código para el microcontrolador PIC, el programa Flowcode procesa un diagrama de flujo en un cierto número de pasos. Vamos a ver cómo se realiza este proceso:

Paso 1

El programa Flowcode toma inicialmente nuestro diagrama de flujo y genera un programa en lenguaje C a partir del mismo. Como ejemplo de lo

que acabamos de decir vamos a ver un sencillo programa contador (COUNTER1.FCF) con el programa Flowcode y, posteriormente, su equivalente en lenguaje C.

En la **Figura 1** se muestra el programa contador generado con Flowcode: en él declaramos el valor de la variable COUNT a "0" y, seguidamente, tenemos el icono de un lazo. El lazo WHILE 1 declara un lazo sin

fin como "1", es decir, que siempre es verdad. Dentro de este lazo tendremos iconos para incrementar el valor de la variable COUNT en 1, esperar durante un segundo y, posteriormente, sacar el valor de la variable COUNT por el puerto C. Disponemos de un microprocesador PIC del tipo 16F877 con diodos LEDs que cuentan en una secuencia binaria sobre el puerto C.

s en C Bloques electrónicos y Flowcode hacen nuestra programación en lenguaje C

Paso 2

Si abrimos el programa Explorador de Windows y miramos en el directorio donde se encuentra el fichero COUNTER1.FCF del programa Flowcode, veremos que Flowcode ha generado algunos ficheros, entre ellos COUNTER1.C. Éste es el fichero equivalente en lenguaje C al programa COUNTER1 del diagrama de flujo. Si abrimos este fichero con la aplicación de Windows "Notepad" podremos ver el programa que se muestra en la **Figura 2**.

La imagen muestra el programa en dos partes, una al lado de la otra. En primer lugar, el programa Flowcode define algunas de las constantes que el compilador C necesita cuando utiliza sentencias como "char PORTC@0x07;". Esta sentencia define la variable "PORTC" en lenguaje C como variable hexadecimal (de aquí que aparezca "0x") en la dirección 07. De forma similar, la variable TRISC ha sido definida con un valor hexadecimal de 87. TRISC es el registro que almacena la dirección de los datos en el microprocesador PIC. Así, el registro TRISC, que estamos utilizando sobre el microprocesador 16F877, está en la posición de memoria hexadecimal 87. De forma similar, la gran mayoría de los microprocesadores PIC y de circuitos con funciones específicas (tales como la velocidad de reloj, los terminales para la USART interna, etc.) se definen de esta manera. Debemos señalar que en lenguaje C, cuando una línea comienza por "//" indica que es un comentario y que no se trata de una línea de código. En nuestro programa no tenemos ni "macros" ni subrutinas, de manera que esta sección está en blanco, pero existe una variable definida como tipo "CHAR", denominada FCV_COUNT. Ésta es nuestra variable COUNT para el programa Flowcode. Ésta es la primera indicación de cómo podemos utilizar el icono C en el programa Flowcode para introducir código dentro de un programa creado con Flowcode: todas las variables en Flowcode tienen el prefijo "FCV_" cuando se transfieren sobre el

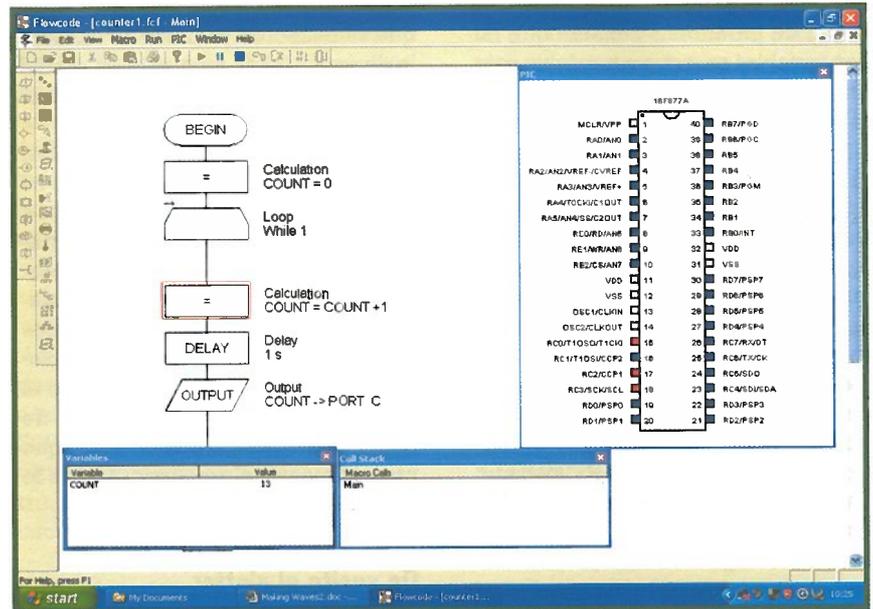


Figura 1. Un sencillo contador en el Flowcode.

compilador C. Esto significa que cuando hacemos referencia a una variable Flowcode en un icono C, también tene-

mos que etiquetar la variable con el prefijo "FCV_". (Como comentario, decir que FCV proviene de FlowCode Varia-

```
counter1.c - Notepad
//Defines for microcontroller
char PORTC@0x07;
char TRISC@0x87;
char PORTD@0x08;
char TRISD@0x88;
char PORTE@0x09;
char TRISE@0x89;

//PIC Functions
#pragma CLOCK_FREQ 19660800
#define P16F877A
#include <system.h>
#define MX_EE
#define MX_EE_TYPE2
const char MX_EE_SIZE = 256;
#define MX_SPI
#define MX_SPI_C
#define MX_SPI_SDI 4
#define MX_SPI_SDO 5
#define MX_SPI_SCK 3
#define MX_UART
#define MX_UART_C
#define MX_UART_TX 6
#define MX_UART_RX 7
#define MX_I2C
#define MX_I2C_C
#define MX_I2C_SDA 4
#define MX_I2C_SCL 3
#define MX_PWM
#define MX_PWM_PORT portc
#define MX_PWM_TRIS trisc
#define MX_PWM_CNT 2
#define MX_PWM_0 2
#define MX_PWM_1 1

//Macro function declarations

//variable declarations
char FCV_COUNT;

//Macro implementations

void main()
{
//PIC initialisation
adcon1 = 0x07;

//interrupt initialisation code
option_reg = 0xc0;

FCV_COUNT = 0;
while(1)
{
FCV_COUNT = FCV_COUNT + 1;
delay_s(1);

TRISC = 0x00;
PORTC = FCV_COUNT;
}
mainendloop: goto mainendloop;
}
```

Figura 2. Equivalente en lenguaje C del COUNTER1.FCF.

ble, es decir, Variable de FlowCode). Después de la declaración de las variables nos encontramos con la primera línea de un programa en C:

```
Void main()
{
```

Ésta es una declaración de función dentro de C que indica que se trata de nuestro programa principal. Es el equivalente al icono START en Flowcode. El par de paréntesis señalan que no hay variables pasadas a través de esta función y la "apertura" del corchete "{" significa el inicio de la función principal. Del mismo modo, podremos ver el correspondiente corchete "}" de "cierre" al final del programa.

Después de esto tenemos dos declaraciones más (para el conversor A/D), que nos permiten declarar estos terminales como entradas analógicas y activar la interrupción del temporizador. Estas declaraciones se hacen dentro del lazo del programa principal para que puedan el usuario pueda modificarlas dentro del programa Flowcode. Después de esta parte tenemos el programa principal propiamente dicho: la variable FCV_COUNT se declara con un valor inicial de "0", de acuerdo con nuestro primer icono en el diagrama de flujo; a continuación nos encontramos con una sentencia "while (1)", seguida por más código dentro de los corchetes abiertos: esto significa que se tiene que ejecutar la rutina dentro de los corchetes durante todo el tiempo de ejecución del programa. Dentro de los corchetes nos encontramos con la sentencia que incrementa la variable FCV_COUNT y nos proporciona un retardo de un segundo. Hay que señalar que estas líneas en lenguaje C acaban con un ";", ya que todas las líneas de código (no de comentario) en lenguaje C siempre tienen que terminar con un ";".

De esta forma tenemos la línea "TRICS=0x00". Primeramente comprobamos que TRICS es el dato de la dirección del registro. Esta línea de código escribe el valor "0" en el registro TRISC, el cual declara todas las líneas del puerto C como salidas. Un valor de "FF" en hexadecimal declararía todas las líneas del puerto como entradas. Nuestra última línea de código en lenguaje C, dentro del lazo, escribe el valor de nuestra variable COUNT sobre el puerto C.

Esta última línea es, en este caso, una cierta medida de seguridad: en el caso de que no dispongamos de un

lazo sin fin en nuestro programa Flowcode, el programa escrito en lenguaje C tomará las riendas en este punto y ejecutará esta línea de manera indefinida. Éste es el equivalente al icono END en el programa Flowcode. Si tenemos un programa en lenguaje C sin un fin de lazo END como éste, tendremos un efecto curioso: el programa contador cargado sobre el microprocesador PIC continuará incrementándose hasta que se desborde y vuelva de nuevo a la dirección 0000, momento en el cual nuestro programa volverá a comenzar de nuevo.

Esperamos que todo esto no sea tan duro como parecía: el lenguaje C es algo misterioso, pero si comenzamos a partir del diagrama de flujo y aportamos una pequeña explicación, el programa escrito en lenguaje C es bastante más comprensible. Tenemos que admitir que todas las declaraciones son difíciles de entender, pero el programa Flowcode se encarga de realizar todas estas tareas, por lo que no tenemos que preocuparnos de lo que significa.

De vuelta al plotter

Volvamos a nuestros objetivos originales de incrementar la velocidad del programa generador de forma de onda que ya escribimos. A su debido tiempo ya compilamos el programa "SINE WAVE GEN,FCF" en la entrega del último mes y echamos una ojeada al fichero en lenguaje C producido. Tenemos que señalar que cada vez que el programa Flowcode utiliza un icono DAC_SEND_CHAR, este icono se transforma en una gran cantidad de líneas de lenguaje C que declaran una serie de variables adicionales y direcciones de referencia que el compilador C necesita. La razón por la cual el programa Flowcode hace esto así es que asume que nosotros no sabemos lo que estamos haciendo (¡gracias a Dios!) y se encarga de configurar todos los parámetros del puerto C para la comunicación serie cada vez que utilizamos el icono macro SPI DAC, tan sólo en el caso en que estemos utilizando las líneas de E/S del puerto C para otras funciones. Por lo tanto, ¿es posible reutilizar algunas de estas líneas de lenguaje C para incrementar la velocidad? Explicaremos esto en dos fases: en primer lugar vamos a sustituir el icono DAC_SEND_CHAR con el equivalente en lenguaje C para estar

seguros de que estamos repitiendo correctamente la función del programa y, en segundo lugar, vamos a intentar modificar el código en lenguaje C para ver si podemos conseguir que sea algo más eficiente.

Volviendo al programa principal, vamos a sustituir sencillamente uno de los iconos DAC_SEND_CHAR con un icono C. A continuación ajustaremos las declaraciones de variables para estar seguros que el icono C captura nuestra variable OUTVAL de Flowcode y, seguidamente, recompilaremos el programa para estar seguros que sigue funcionando correctamente. Esto se consigue sin mayores problemas. Una vez que hemos conseguido realizar esta primera modificación, tendremos que poner en comentario todas las líneas de código de lenguaje C que innecesarias (de las 34 líneas de código C que se generan en un principio) ya que, de las líneas que produce el icono DAC_SEND_CHAR, ¡sólo necesitaremos nueve de ellas! En este caso tuvimos que realizar varios intentos, quitando y poniendo líneas en comentario, hasta que conseguimos dejar las líneas que realmente eran necesarias y que no detenían la ejecución del programa. El programa final tiene el nombre de SINC3.FCF y nuestros lectores pueden volcarlo desde la página web de Elektor. El nombre del

Anteriores en esta Serie

Construyendo Bloques Electrónicos, Enero de 2006.

Bloques Electrónicos y Flowcode, Febrero de 2006.

Bloques Electrónicos - Ahora Ya Podemos (CAN), Marzo de 2006.

Bloques Electrónicos - Generando Ondas, Mayo de 2006.

Los artículos se pueden descargar de forma individual desde nuestra página web.

Para una vista global de los Bloques Electrónicos disponibles recomendamos a nuestros lectores que visiten el apartado "SHOP" en nuestra página web, en

www.elektor-electronics.co.uk

fichero es 065032-11.zip y lo podemos encontrar en las opciones de "Magazine _ Abril 2006".

Sin embargo, por desgracia, nos ha sido imposible conseguir que el conversor D/A vaya más rápido, ya que la limitación de velocidad se modificaba para terminar siendo la velocidad del propio bus SPI. ¡Explicaremos esto un poco más!

Con el código del comando DAC_SEND_CHAR del programa Flowcode, el dato de ocho bits del bus SPI se envía en dos bytes. Los últimos cuatro bits del primer byte contienen los cuatro bits más significativos del dato a transmitir, y los cuatro bits menos significativos del segundo byte contienen los cuatro bits menos significativos del dato a transmitir. Los otros bits de cada byte de datos del bus SPI están reservados para la configuración del circuito integrado, entre otras cosas. El icono DAC_SEND_CHAR del programa Flowcode toma la variable OUTVAL y la procesa de la siguiente manera:

```
dac_val = (FCV_OUTVAL & 0xF0) > 4;
sspbuf = dac_val;
```

```
delay_us(3);
```

```
dac_val = (FCV_OUTVAL & 0x0F) << 4;
sspbuf = dac_val;
delay_us(3);
```

Mirando la parte superior de la rutina, lo primero que hacemos es tomar la variable OUTVAL y hacer una función AND entre su valor y el valor hexadecimal "F0" (11110000 en binario) y, a continuación, hacer un desplazamiento de su contenido en cuatro bits (éste es el motivo de que aparezca "> 4" en la línea). Seguidamente configuramos el registro SSPBUF del circuito integrado con el resultado obtenido y esperamos durante 3 µs. Todo lo que esté colocado en el registro SSPBUF será enviado sobre el bus SPI. La segunda rutina es bastante similar: toma la variable OUTVAL, hace la función AND con el valor hexadecimal "0F", desplaza el contenido en cuatro bits, almacena el resultado en el registro SSPBUF y espera durante 3 µs. La última sentencia es la clave: el registro SSPBUF es el "buffer" serie en el circuito integrado PIC y tarda 3 µs en

que dicho registro vacíe la información que contiene, ya que envía un bit cada vez. Si intentamos escribir de nuevo el buffer en menos de 3 µs estaremos sobrescribiendo el dato a la mitad de la fase en la que se está realizando la operación de envío serie, algo que puede producir unos resultados muy extraños, como que el bus SPI deje de funcionar de manera correcta.

Por lo tanto, tenemos que decir que la frecuencia de nuestro oscilador viene determinada por la velocidad del bus SPI. Cada muestra necesita un tiempo de 6 µs. Esto nos lleva al resultado de que 256 muestras por forma de onda proporcionan un período mínimo teórico de 1,5 ms, lo que da una frecuencia de unos 650 Hz, o bien es el momento de reducir el número de muestras a 64...

(065032-1)

Programas Disponibles para ser Descargados

- COUNTER1.fcf
- SINC3.fcf

Número de Fichero: 065032-11.zip



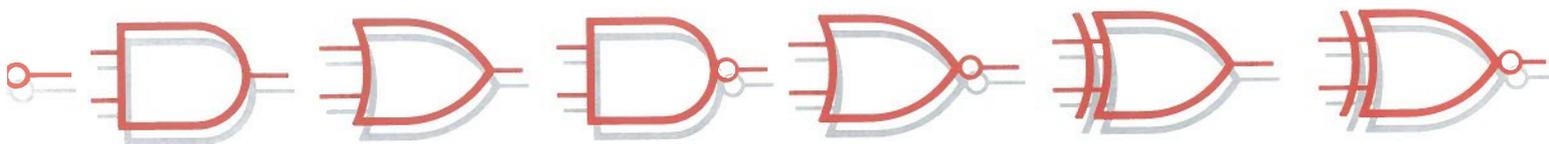
La tienda on-line
que estabas esperando...

- Microcontroladores PIC
- Circuitos Integrados
- Semiconductores
- Optoelectrónica
- Transformadores
- Placas
- Baterías
- Memorias
- Reles
- etc...

www.electan.com

- Envíos por correo contra reembolso.
- Envíos por mensajería.
- Sin pedido mínimo
- Envíos a toda España.

Curso FGPA (1) Los comp



Ésta es la primera entrega de un curso sobre FPGAs, basado en la versátil unidad FPGA y en la placa de prototipos asociada descrita en el último número de nuestra revista. Esta serie de artículos describirá cómo diseñar circuitos digitales y cómo programar una FPGA. En este artículo, que se corresponde con la primera entrega de esta serie, vamos a tratar los componentes básicos de la electrónica digital.

Paul Goossens

La electrónica digital está basada en circuitos que funcionan utilizando tan sólo dos niveles de tensión. Estos niveles son llamados nivel "alto" y nivel "bajo". "Alto" significa que la tensión de la señal es superior que el valor definido previamente, mientras que un nivel "bajo" significa que el nivel de la señal es menor que dicho valor. Habitualmente se utilizan los símbolos "1" y "0" para asignar un nivel "alto" y un nivel "bajo". Normalmente, un "1" se corresponde con un nivel alto mientras que un "0" se corresponde con un nivel bajo. Estas señales digitales pueden procesarse por componentes que, a su vez, generan más señales digitales.

Puerta AND

La electrónica digital está enteramente basada en tan sólo tres componentes elementales. Incluso los circuitos digitales más complicados están compuestos únicamente de estos tres tipos de componentes.

En la **Figura 1** se ilustran estos tres componentes elementales (funciones). Las dos primeras, las funciones AND y OR, pueden tener cualquier número de entradas deseado, mientras que la tercera (el INVERSOR) siempre tiene una única entrada. Cada uno de estos componentes dispone de una única salida.

Vamos a comenzar con la función AND. La salida de este componente estará a "1" sólo si todas sus entradas están a "1". Otra manera de explicar cómo trabaja esta función lógica es la de decir que la salida es "0" si al menos una de sus entradas es "0". Esto quiere decir exactamente lo mismo, lo cual se puede ver claramente en la Figura 1.

Notación

El nombre de cada función se muestra en la parte superior de la Figura 1. La siguiente fila nos muestra el operador booleano. Se puede utilizar la notación booleana para describir la circuitería digital de una manera matemática. Veremos una gran cantidad de notaciones booleanas en esta serie de artículos. Como se puede ver, se trata de una notación relativamente sencilla, tal y como se muestra en los ejemplos de las ecuaciones booleanas de la siguiente fila. Por debajo de las ecuaciones se muestran las tablas de verdad asociadas. Hay una manera muy práctica de describir el comportamiento de un circuito digital y un modo muy sencillo de entenderlo. Los niveles de las entradas se muestra en la columna de la izquierda y los niveles de las salidas en la columna de la derecha. Si miramos lo que sucede con la puerta AND, por ejemplo, podemos ver que la salida está a nivel "1" sólo si todas las entradas están a "1". A menudo es posible reducir el tamaño de la tabla de verdad y hacer que su interpretación sea más sencilla. El

símbolo "X" se utiliza para este propósito. Este símbolo significa "sea cual sea el valor" o "no importa el valor". Si miramos en las tablas de verdad de la parte inferior de la figura, podemos ver que los estados de las entradas A y B de la primera columna no tienen ninguna influencia si la entrada C está a nivel lógico "0", ya que en dicho caso las salidas siempre son "0".

Como última nota a destacar, nos gustaría señalar que aunque las puertas AND y OR de la Figura 1 están dibujadas con tres entradas, el número de éstas puede variar desde dos hasta el que se desee en la práctica actual.

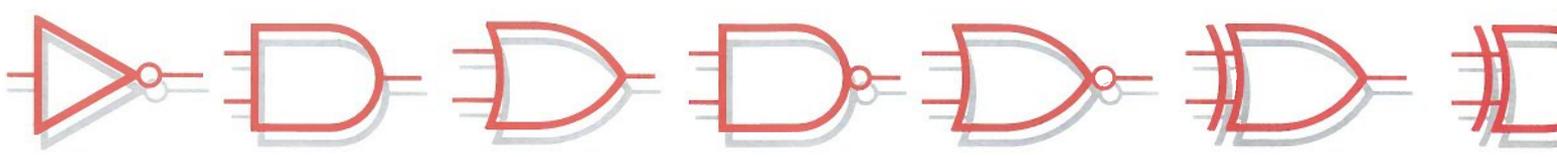
Puerta OR e inversores

La salida de una puerta OR es "1" si al menos una de sus entradas está en el estado lógico alto "1". En otras palabras, la salida de una puerta OR está a nivel lógico "0" sólo si todas sus entradas están en el estado "0". El modelo final de la puerta es el del inversor o puerta NOT. La salida de esta puerta es la inversa de la señal presente en sus entradas. En otras palabras, un nivel lógico "0" en la entrada provoca un nivel lógico "1" en la salida, y viceversa. Esto también se muestra en la Figura 1.

Placa prototipo

Ahora vamos a intentar simular estas puertas en la placa de prototipos. La **Figura 2** muestra el diagrama eléctrico de un circuito sencillo y

Componentes básicos de la electrónica digital. Parte I



vamos a intentar implementar dicho circuito en la FPGA. Utilizaremos las señales provenientes de los cuatro pulsadores de la placa prototipo como entradas, y los siete diodos LEDs de la placa prototipo para mostrar los distintos estados de las correspondientes salidas.

Los ficheros que necesitamos para practicar con este ejemplo están incluidos en los programas para este artículo, los cuales pueden descargarse de manera gratuita de la página

web de Elektor, en www.elektor-electronics.co.uk, en Magazine_April 2006_FPGA Course (1). El programa se puede instalar adecuadamente utilizando el programa de instalación (Setup.exe), incluido en el módulo que se descarga.

En el resto de este artículo vamos a suponer que nuestros lectores ya han instalado los distintos programas en el directorio por defecto (C:\altera\FPGA_course\1\ex1). También vamos a dar por hecho que nuestros

lectores ya tienen instalada la unidad FPGA y la placa de prototipos de acuerdo con las instrucciones recibidas en los artículos asociados.

Haremos doble "clic" con el ratón sobre el fichero "ex1.qpf", en el directorio "C:\altera\FPGA_course\ex1". Esto lanzará la ejecución del programa "Quartus" el cual, a su vez, abrirá el proyecto correspondiente al ejemplo con el que estamos trabajando. Podemos ver el esquema eléctrico en nuestro monitor, que se debe

Función Lógica:	AND	OR	NOT/INV																																																																														
Símbolo Eléctrico:																																																																																	
Símbolo IEC del circuito:																																																																																	
Función Booleana:	*	+	—																																																																														
Ecuación Booleana:	$Q = A * B * C$	$Q = A + B + C$	$Q = \bar{A}$																																																																														
Tabla de verdad:	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>Q</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	Q	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>Q</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	Q	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	<table border="1"> <thead> <tr><th>A</th><th>Q</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	Q	0	1	1	0
A	B	C	Q																																																																														
0	0	0	0																																																																														
0	0	1	0																																																																														
0	1	0	0																																																																														
0	1	1	0																																																																														
1	0	0	0																																																																														
1	0	1	0																																																																														
1	1	0	0																																																																														
1	1	1	1																																																																														
A	B	C	Q																																																																														
0	0	0	0																																																																														
0	0	1	1																																																																														
0	1	0	1																																																																														
0	1	1	1																																																																														
1	0	0	1																																																																														
1	0	1	1																																																																														
1	1	0	1																																																																														
1	1	1	1																																																																														
A	Q																																																																																
0	1																																																																																
1	0																																																																																
	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>Q</th></tr> </thead> <tbody> <tr><td>X</td><td>X</td><td>0</td><td>0</td></tr> <tr><td>X</td><td>0</td><td>X</td><td>0</td></tr> <tr><td>0</td><td>X</td><td>X</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	Q	X	X	0	0	X	0	X	0	0	X	X	0	1	1	1	1	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>Q</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X</td><td>X</td><td>1</td><td>1</td></tr> <tr><td>X</td><td>1</td><td>X</td><td>1</td></tr> <tr><td>1</td><td>X</td><td>X</td><td>1</td></tr> </tbody> </table>	A	B	C	Q	0	0	0	0	X	X	1	1	X	1	X	1	1	X	X	1																																							
A	B	C	Q																																																																														
X	X	0	0																																																																														
X	0	X	0																																																																														
0	X	X	0																																																																														
1	1	1	1																																																																														
A	B	C	Q																																																																														
0	0	0	0																																																																														
X	X	1	1																																																																														
X	1	X	1																																																																														
1	X	X	1																																																																														

060025 - 11

Figura 1. Los componentes básicos mostrados al completo.

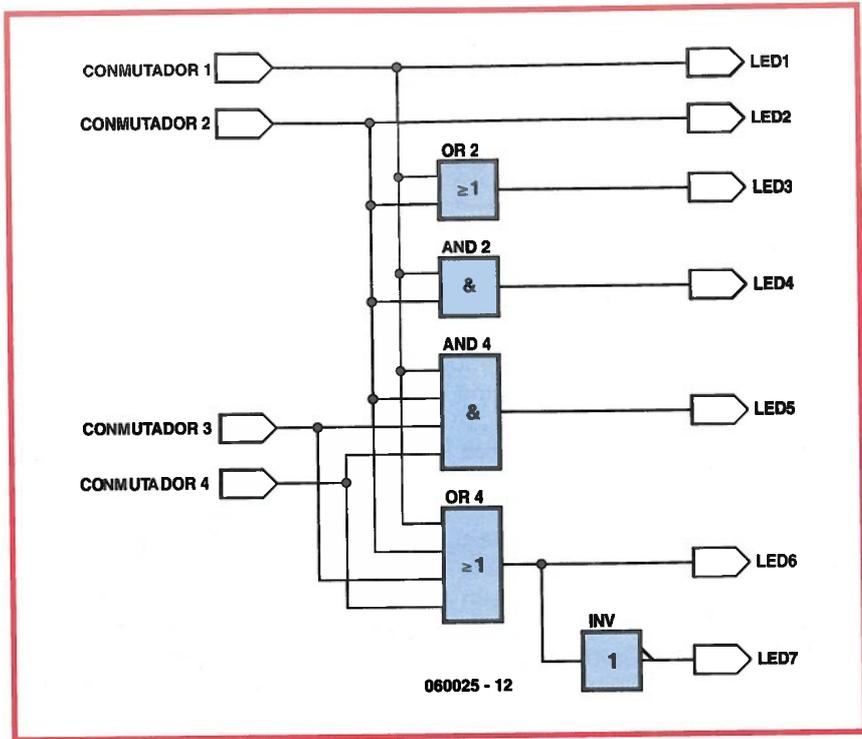
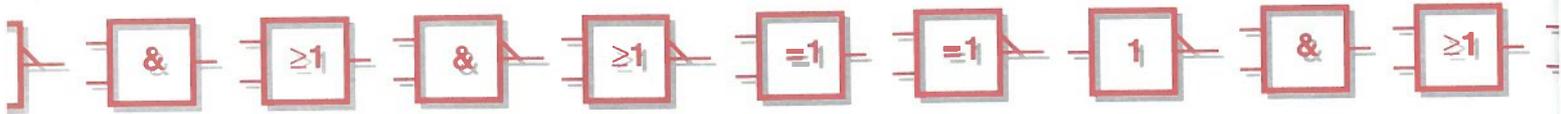


Figura 2. The schematic diagram of our first example.

corresponder con el que se muestra en la Figura 2. Las señales "SWITCH 1" a "SWITCH 4" están conectadas a los terminales de entrada. El programa conecta dichas señales a los contactos asociados de la FPGA, dentro de la unidad FPGA. Naturalmente, lo mismo sucede con las líneas "LED 1" a "LED 7".

El programa "Quartus" puede utilizar el esquema eléctrico para crear un

fichero de programación que, a su vez, puede usarse para configurar este circuito en el interior de la FPGA. Para hacer esto seleccionaremos la opción "Start compilation" dentro de la opción del menú "Processing". Es posible que tarde unos segundos en crear el fichero, pero al final podemos ver un mensaje donde se muestra que la compilación ha sido realizada sin mayores contratiempos. Siguiendo las instrucciones que se die-

ron en el artículo sobre la unidad FPGA, realizaremos la programación de la FPGA usando este fichero. La única diferencia en este caso es que tendremos que utilizar el fichero "ex1.sof" como fichero de programación.

Resultados

Una vez que programada la FPGA, nuestro circuito está listo para usarse. Ahora ya podemos utilizar los cuatro pulsadores para controlar las entradas "SWITCH 1" a "SWITCH 4". Si pulsamos un botón, la señal correspondiente se verá forzada a pasar a nivel alto ("1"). Cuando los botones no están presionados las resistencias R5 a R8 provocan que se aplique un nivel bajo ("0") a las entradas asociadas. Las salidas "LED 1" a "LED 7" controlan los diodos LEDs D8 a D14 a través del circuito integrado IC7. Los diodos LEDs se muestran del "1" al "7" en el esquema eléctrico del circuito.

A partir del esquema eléctrico del circuito en el programa "Quartus", podemos ver que la salida LED 1 está conectada directamente con la entrada "SWITCH 1". Cuando la señal "SWITCH 1" pasa a nivel lógico "1", la salida "LED 1" también cambia inmediatamente a su estado lógico "1". En la práctica, esto significa que el diodo LED 1 se encenderá de manera inmediata cuando presionemos el botón 1. También podemos controlar la salida "LED 2" (y por consiguiente el diodo LED 2) de la misma manera.

La situación es diferente con la salida "LED 3". Esta salida está conectada a la salida de una puerta OR cuyas entradas están conectadas a las líneas "SWITCH 1" y "SWITCH 2". Por lo tanto, este diodo LED se encenderá cuando el pulsador 1 o el pulsador 2 se presionen (o si ambos se presionan al mismo tiempo). Podemos utilizar la línea de salida "LED 4" para estudiar el funcionamiento de una puerta AND. Esta salida solamente se encenderá si el pulsador 1 y el pulsador 2 son presionados al mismo tiempo. Hagámoslo nosotros mismos y podremos ver que el diodo LED se ilumina de acuerdo con las reglas descritas anteriormente.

Ahora podemos utilizar la línea de salida "LED 5" para comprobar

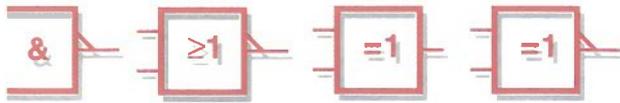
```

LED1 = CONMUTADOR 1
LED2 = CONMUTADOR 2
LED3 = CONMUTADOR 1 + CONMUTADOR 2
LED4 = CONMUTADOR 1 * CONMUTADOR 2
LED5 = CONMUTADOR 1 * CONMUTADOR 2 * CONMUTADOR 3 * CONMUTADOR 4
LED6 = CONMUTADOR 1 + CONMUTADOR 2 + CONMUTADOR 3 + CONMUTADOR 4
LED7 = NOT LED6
OR :
LED7 = (SWITCH1 + SWITCH2 + SWITCH3 + SWITCH4)

'A*B' = 'A AND B'
'A+B' = 'A OR B'
A = 'not (A)'
    
```

060025 - 13

Figura 3. El ejemplo 1 mostrado en una notación booleana.



cómo funciona una puerta AND de cuatro entradas. De acuerdo con las reglas, el diodo LED 5 solamente se encenderá si presionamos los cuatro botones al mismo tiempo. Por su parte, la línea "LED 6" nos muestra cómo funciona una puerta OR de cuatro entradas. En este caso, el diodo LED 6 se encenderá cuando cualquiera de las entradas (los cuatro pulsadores) esté a nivel alto (botón presionado).

Por último tenemos la línea "LED 7". Esta línea nos muestra cómo se puede trabajar con la función NOR. El diodo LED 7 se encenderá cuando el diodo LED 6 esté apagado, lo cual significa que ninguno de los botones tiene que estar pulsado. Para asegurarnos que este circuito está funcionando de manera adecuada, presionaremos cada uno de los botones de manera individual y, realizando las distintas combinaciones posibles, observaremos cuál es la respuesta de cada diodo LED y si se corresponde con lo que se ha descrito anteriormente.

SOBRE EL TEMA: PLACA DE PROTOTIPO PARA FPGA

Placa de Prototipos para FPGA Las FPGAs llegan a la vida

Fai Coosens

VGA
Una salida VGA genera que podemos utilizar para mostrar todo lo que se genera en una pantalla de un ordenador. Todo lo que se realice en esta placa puede ser mostrado en una pantalla externa.
Ver página 24.

Ethernet
Permite conectar a Internet. El sistema permite mostrar los datos de configuración en una pantalla externa y puede ser controlado por un ordenador.
Ver página 24.

USB
Este puerto USB hace que los datos se puedan transferir desde la placa al ordenador, reduciendo el tiempo de configuración.
Ver página 24.

E/S Analógicas
Como alternativa a las E/S digitales, esta placa también ofrece E/S analógicas. Estas E/S analógicas permiten conectar la placa a sensores de temperatura, humedad, etc.
Ver página 24.

Tensión de Alimentación
La placa de prototipos para FPGA necesita una fuente de alimentación de 5V y 20V. El sistema de alimentación de la placa de prototipos para FPGA puede ser controlado por un ordenador.
Ver página 24.

PS/2
Un puerto para el teclado de serie y un ratón.
Ver página 24.

LCD
Un módulo LCD de 2.8" de tamaño con un sistema de control de pantalla por I2C. Este módulo puede ser controlado por un ordenador.
Ver página 24.

E/S de Audio
Con una salida y una entrada de audio de 3.5mm, esta placa de prototipos puede ser utilizada para la salida de audio.
Ver página 24.

Visualizadores
La placa de prototipos para FPGA incluye un módulo de visualización de 7 segmentos para mostrar los datos de configuración.
Ver página 24.

E/S Digitales
Este módulo genera señales de salida de nivel alto y bajo.
Ver página 24.

Diodos LEDs
Permite usar los diodos LEDs para mostrar los datos de configuración.
Ver página 24.

Pulsadores
Permite usar los pulsadores para controlar los datos de configuración.
Ver página 24.

Conmutadores DIP
Utilizados para configurar la placa de prototipos para FPGA.
Ver página 24.

Creación de diseños propios

El objetivo último de este curso es permitirnos diseñar nuestros propios circuitos digitales y programarlos sobre la propia FPGA. Esto significa que, además de entender cómo funciona la circuitería digital, también necesitamos saber cómo se utiliza el programa "Quartus". La única y mejor manera de aprender a utilizar el programa "Quartus" es la de trabajar con él. Así, el mejor modo de comenzar es el de trabajar a través

de su guía de programación. También existe un documento PDF en el directorio "C:\altera\FPGA_course\ex2" con instrucciones paso a paso para dibujar y crear el primer circuito ejemplo. Es importante que nuestros lectores lean este documento en su totalidad y que lo utilicen para dibujar el circuito ejemplo que se muestra en la Figura 2. De este modo, no solamente aprenderemos a utilizar el programa sino que también sabremos cómo configurar el programa para utilizarlo con



¿Está buscando Componentes Electrónicos?

Tenemos las últimas novedades y las mejores marcas en:

Microcontroladores transistores tarjetas diodos placas memorias

Grabadores resistencias conmutadores potenciómetros

Herramientas librería técnica.

Instrumentación laboratorio

Circuitos impresos pre-series desarrollo y montaje

Tenemos una gran exposición a su servicio

Total garantía de nuestros productos, consúltenos sus dudas y nuestros profesionales le asesorarán.

Array Electrónica Profesional S.A.

Envíos a toda España

Envíos a toda España

los circuitos de la FPGA y los circuitos de programación que iremos viendo en Elektor.

Algebra booleana

Las funciones del primer ejemplo también se pueden describir utilizando el álgebra de Bool. La Figura 3 nos muestra cómo se describen con una notación booleana las siete funciones del circuito ejemplo. También podemos utilizar estas ecuaciones para describir el circuito dentro del programa "Quartus". Esto implica el uso de un lenguaje llamado "VHDL", el cual ha sido pensado para describir el funcionamiento de los circuitos digitales. El programa "Quartus" puede utilizar este tipo de descripciones para crear ficheros de programación. En lo que concierne al propio programa, no debemos de preocuparnos de dibujar un circuito en forma gráfica o describirlo utilizando VHDL.

Un ejemplo

Los ficheros Quartus para este ejemplo están localizados en el directorio "C:\altera\FPGA_course\ex3". Una vez en dicho directorio, tan sólo tendremos que "picar" dos veces sobre el fichero "x3.qpf" para lanzar el programa Quartus y cargar el proyecto. Las diferencias con el ejemplo anterior aparecen inmediatamente en pantalla. Como podemos ver en el fichero ex3.bdf, el cual se corresponde con la representación gráfica del circuito, las entradas y las salidas del diseño están conectadas a un gran rectángulo. Este rectángulo procesa las señales de entrada y genera las salidas correspondientes.

"Picando" dos veces sobre el rectángulo se abre el documento que especifica el modo en que funciona dicho rectángulo. Éste puede ser otro esquema eléctrico, pero también puede ser un poco de código VHDL. En este caso, hemos definido las funciones utilizando el código VHDL. En VHDL, todo lo que se escriba después del "-" es interpretado como un comentario. Los comentarios no forman parte del diseño que se está realizando, pero pueden ser muy útiles para la documentación de dicho diseño. Nuestros lectores tendrán que esperar hasta la entrega del próximo número para disponer de una explicación detallada de la estructura de un fichero VHDL. Después de todo, ¿éste sólo es el primer artículo

de la serie!

En la línea 29 del fichero que tenemos abierto damos al diseño un nombre (en nuestro caso, "ex3_VHDL"), el cual está seguido por las definiciones de las entradas en las salidas del diseño. Aquí podemos ver que las señales IN1 a IN4 son entradas (IN). Por su parte, las señales OUT1 a OUT 7 son todas salidas (OUT). Esto es todo lo que diremos por ahora de la estructura de este documento, ya que es la primera vez que hablamos de ello.

Ecuaciones

Sin embargo, las líneas 57 a 63 merecen una explicación adicional. En estas líneas podemos ver ecuaciones similares a las que se muestran en la Figura 3, con la diferencia de que aquí utilizamos el símbolo "<=" en lugar de "=", que es cómo se muestra en la Figura 3. Asimismo observe que los operadores "*" y "+" pueden sustituirse por las palabras "and" y "or" respectivamente.

En la línea 63 hemos escrito la ecuación completa en lugar de utilizar la ecuación "OUT7 <= not (OUT6);", aunque la última ecuación es perfectamente correcta, en teoría. Por desgracia, el lenguaje VHDL no permite usar las señales de salida como entradas para la lógica interna. Más adelante explicaremos a nuestros lectores a qué se debe esto y un sencillo truco para solventar esta restricción. A continuación, una vez más, debe compilar el ejemplo y volcarlo sobre la placa prototipo.

Modificaciones

El directorio "ex4" contiene un documento que explica, paso a paso, cómo dibujar el ejemplo anterior en el programa "Quartus". Podemos utilizar este documento para familiarizarnos con el dibujo de bloques funcionales y con la construcción de un diseño en una estructura jerárquica. Este documento también describe algunas otras características del programa Quartus, de manera que podremos disponer de un punto de ayuda para trabajar a través del ejercicio.

El lenguaje VHDL es extremadamente potente y, en nuestro caso, sólo vamos a tratarlo un poco, lo suficiente como para conocer sus características más básicas. Sin embargo, está claro que diseñar un circuito de esta manera ahorra una gran canti-

dad de tiempo. Por ejemplo, supongamos que hemos decidido que la línea "LED 1" ("OUT1" en el fichero VHDL) sólo debe encenderse cuando los botones 1 y 4 se presionan simultáneamente. Esto nos obliga a modificar el diseño sustituyendo la línea 57 del programa ejemplo con la siguiente línea:

```
OUT1 <= IN1 and IN4;
```

Podemos hacer esta comprobación nosotros mismos, recompilando el fichero del diseño modificado y programándolo de nuevo sobre la FPGA. Si hemos generado el diseño en la forma de esquema eléctrico, tal y como estaba en el primer ejemplo, probablemente nos llevará algo más de trabajo incorporar la modificación sobre el diseño. Es más, el esquema eléctrico de un diseño relativamente grande se puede transformar fácilmente en un conjunto de líneas y puertas lógicas que pueden hacer bastante difícil una comprensión rápida de cómo funciona dicho diseño. El funcionamiento del diseño se entiende de manera mucho más sencilla si lo describimos con el lenguaje VHDL, especialmente si además le añadimos los correspondientes comentarios.

La próxima entrega

Como les hemos prometido, en la siguiente entrega pondremos en funcionamiento el visualizador de siete segmentos. Esto implica el uso de elementos de memoria y la realización de algunos cálculos rudimentarios. Mientras tanto, recomendamos a nuestros lectores que visiten el foro o nuestra página web para tratar sobre temas de las FPGAs, nuestro kit de desarrollo y, por supuesto, nuestro curso sobre FPGAs.

(060025-1)

Los programas para esta entrega están disponibles, de manera gratuita, en la página web de Elektor bajo el código 060025-1-1.zip.

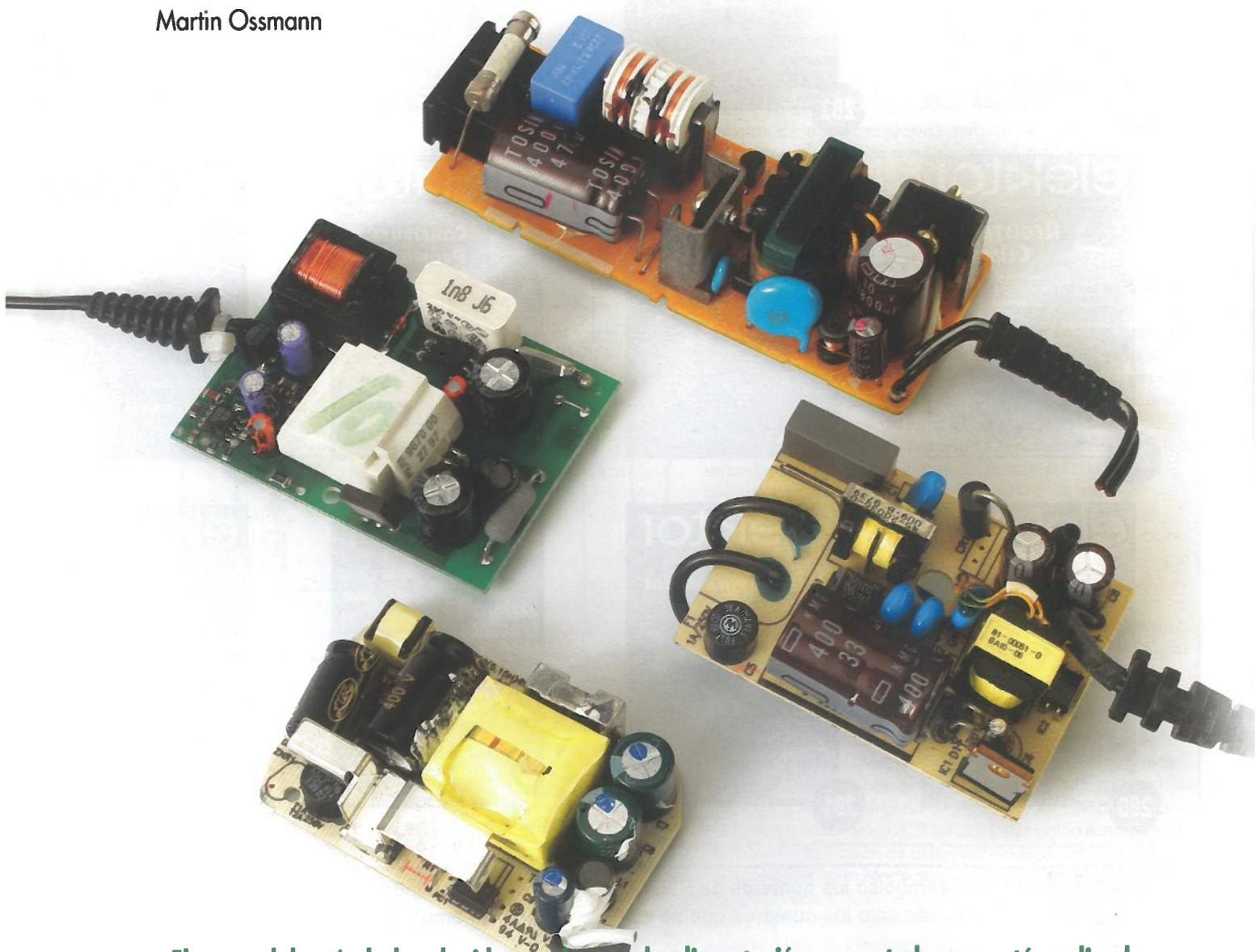
Entregas anteriores de esta serie
Versatile FPGA Module,
 Elektor Mayo 2006.

FPGA Prototyping Board,
 Elektor Mayo 2006.

Manifiesto sobre en Modo Conmutado

Diseños elegantes desde China

Martin Ossmann



El autor del artículo ha elegido una fuente de alimentación conmutada que está realizada sin circuitos integrados de propósito especial, por lo que puede aplicarse la 'ingeniería inversa'. También hemos descubierto diseños desconocidos y sorprendentes de circuitos muy efectivos contruidos con componentes de bajo coste, lo cual puede sernos útil para nuestros propios diseños.

Alimentaciones tado

Las pequeñas fuentes de alimentación, como por ejemplo los cargadores de los teléfonos móviles, incrementan el uso de las topologías de conmutación. El autor ha ido recopilando y analizando unidades de este tipo durante algún tiempo, y ha descubierto que las fuentes de alimentación de un fabricante chino son particularmente interesantes y merecedoras de nuestra atención. Asimismo, veremos algunas estrategias alternativas usadas en otras fuentes de alimentación. Por último, terminaremos este artículo con la construcción de un pequeño proyecto usando la misma idea que en el elegante diseño chino.

Barato, simple y discreto

La fuente de alimentación de red que se muestra en la **Figura 1** está construida con componentes discretos, sin el uso de circuitos integrados de propósito especial. Esto significa que podemos analizar completamente cómo trabaja usando 'ingeniería inversa'. Resulta sorprendente que pueda construirse un circuito efectivo usando unos pocos componentes de bajo coste. En el análisis veremos a cada paso qué aspectos del diseño son importantes, y descubriremos algunas sorpresas en el camino.

Ideas

Una fuente de alimentación de red puede dividirse normalmente en tres bloques funcionales. En la primera etapa, una tensión DC se deriva de la entrada de red. Esta etapa también incluye varias funciones de seguridad y filtrado. La segunda etapa es el circuito de conmutación propiamente dicho, la cual realiza la conversión de tensión y suministra el aislamiento de la red. En la tercera etapa, la tensión de salida es acondicionada y suavizada, y así se obtiene la información necesaria para permitir el control del proceso de regulación de tensión. Primero veremos las especificaciones de la alimentación, que se muestran en la etiqueta de identificación del fabricante:
Tensión de entrada: 100 V a 240 V AC, 50/60 Hz
Tensión de salida: 12 V, max. 1.0 A

La etapa de entrada de la fuente de alimentación está dispuesta tal y como puede verse en la **Figura 2**. El condensador C1 está conectado directamente a la tensión de red sin mayor protección, y debe ser, por lo tanto, un condensador 'clase-X'. Su trabajo es bloquear las altas frecuencias desde la red. La resistencia R1 está incluida para descargar C1. Ordinariamente, las resistencias SMD no son adecuadas para operar con tensiones de red, y así R1 se construye a partir de dos resistencias SMD de 470 k Ω colocadas en serie. Se incluye un fusible para un supuesto problema: un ejemplo sería un pico transitorio de gran energía en la

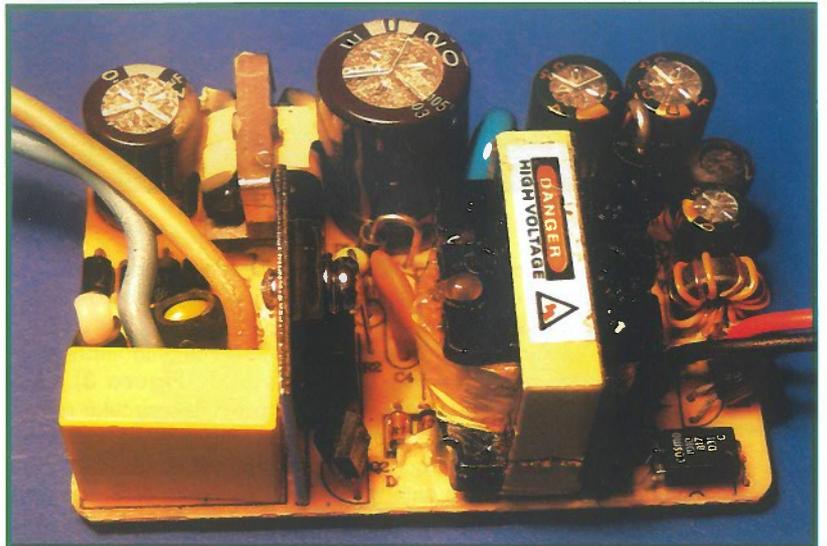
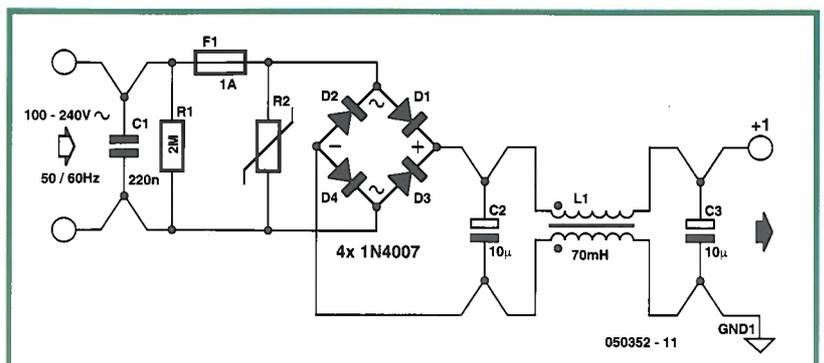


Figura 1. Una fuente de alimentación hecha en China usando componentes discretos.

entrada de la red. Los transitorios menores son absorbidos por VDR R2 (470 V). Cuatro diodos discretos de tipo 1N4007 forman un rectificador para la tensión de red. Algunos fabricantes usan un puente rectificador en un encapsulado de montaje superficial. Los condensadores C2 y C3 juntos constituyen el condensador de reserva, son de tipo 400 V/105° C para asegurar una larga duración a altas temperaturas. Un interesante añadido es el choque L1 modo común ('corriente compensada') entre los dos condensadores, cuya ventaja es que entre los condensadores circula una corriente constante, menor que el pico de corriente en el puente rectificador. El choque, por tanto, puede hacerse más pequeño. Al mismo tiempo, la inductancia de este 'transformador', junto con C2 y C3, efectivamente forma un filtro en modo diferencial. Como una bobina en modo común, cada lado del choque L1 tiene una inductancia parásita de alrededor de 70 mH. Viendo que un transformador tiene una induc-

Figura 2. La etapa de entrada de la fuente de alimentación.



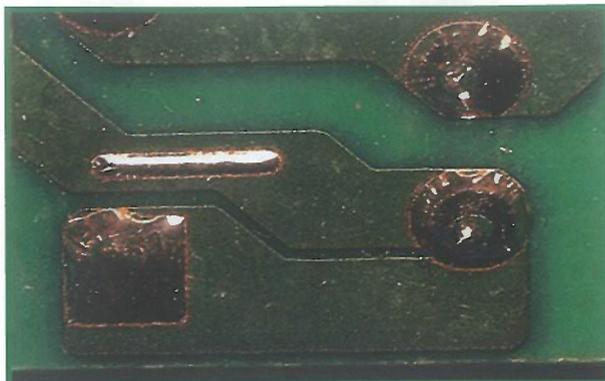


Figura 3. Los cortes en las pistas de la placa de circuito impreso mejoran el desacoplo.

tancia parásita de $460 \mu\text{H}$, éste puede ser adecuado para un choque en modo diferencial. Esta bobina L1 tiene dos funciones de filtrado. Analizando más de cerca la placa de circuito impreso vemos aberturas en varias de las pistas (**Figura 3**). Esto no es un accidente: fuerza a la corriente a circular a través de los condensadores. Usualmente, esta disposición se indica en el esquema del circuito por un bucle en la conexión, como hemos mostrado en la Figura 2. Las

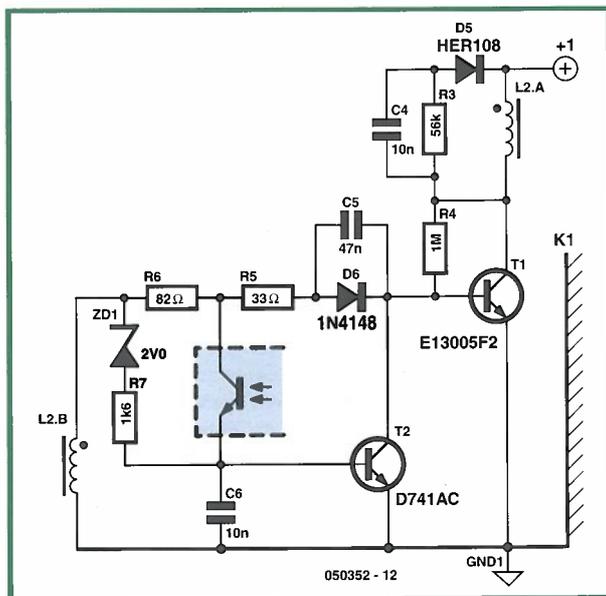


Figura 4. El corazón de la fuente de alimentación.

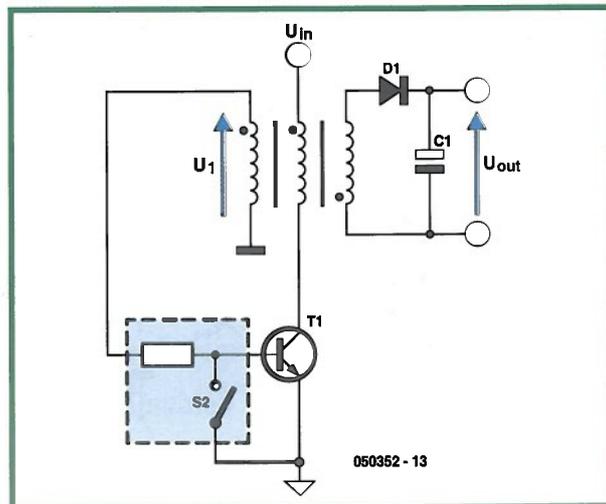


Figura 5. El principio del oscilador flyback auto-oscilante.

características aberturas en las pistas afectan a varios condensadores en el lado primario y secundario del circuito que estamos estudiando. La tensión de alimentación para la parte de conmutación de la alimentación aparece después del filtrado, en C3.

Convertor Flyback

Lo primero que debemos establecer cuando vemos un alimentador de red es la topología básica del circuito. Primero intentaremos encontrar los componentes de potencia individuales, los cuales pueden reconocerse generalmente por su tamaño. Después, podemos intentar identificar los componentes usados (en particular diodos y transistores) realizando medidas o buscando sus hojas de características. Unas sencillas medidas de resistencia tomadas en continua nos pueden ayudar a realizar el cableado del transformador. En este caso descubriremos rápidamente que estamos tratando con un convertidor de retorno, la topología más popular para aplicaciones de baja potencia.

La **Figura 4** muestra el corazón de la fuente de alimentación, sin el circuito secundario. El circuito consta básicamente de dos transistores, pero no es muy breve, como mostraremos en nuestro análisis posterior. El principio básico del convertidor de retorno se muestra con mayor claridad en la **Figura 5**. Los componentes centrales son el 'transformador' con tres bobinados, el transistor T1, y el diodo D1. Ponemos 'transformador' entre comillas porque el componente en cuestión es una bobina con varios bobinados. Al comenzar el ciclo de conmutación no hay circulación de corriente en ninguno de los bobinados de la bobina, y no hay energía almacenada. Entonces T1 se activa y el bobinado primario de la bobina se 'carga' desde la tensión de entrada U_{in} . Por tanto, D1 no está conduciendo. Después de un tiempo, T1 pasa a corte y la corriente es transferida desde el lado primario al secundario. El diodo D1 conduce y la bobina es 'descargada', liberando su energía en el circuito secundario. Esta disposición de circulación de corriente es la que caracteriza la topología del convertidor flyback. La corriente en el secundario cae de forma lineal; cuando alcanza cero, T1 es conmutado de nuevo y se repite el ciclo.

Medidas

Después de tanta teoría, vamos a pasar a realizar algunas medidas prácticas. Debido a que normalmente el circuito está conectado directamente a las altas tensiones de la red, esto no es totalmente directo. Si conectamos el circuito a través de un transformador de aislamiento eléctrico, podemos hacer medidas en el lado primario de la tensión de alimentación usando un osciloscopio normal. En esta situación debemos tener cuidado, ya que no hay conexión a tierra o ninguna otra protección. Como los condensadores electrolíticos C2 y C3 no son particularmente grandes, en la tensión de alimentación existe un rizado de unos 100 Hz, cuyo efecto se puede ver en las trazas del osciloscopio. Para eliminar este efecto usamos un puente rectificador separado y condensadores mayores ($470 \mu\text{F}$) para generar la tensión de alimentación y aplicarla al punto '+1' (ver Figura 4). La **Figura 6** muestra la tensión en la conmutación del transistor T1 (trazo superior, punta de prueba 10:1) y la corriente del colector a través de T1 (trazo inferior, medida usando una pinza de corriente). Para esta medida la tensión de entrada en el punto '+1' fue de 326 V. Con la carga colocada en la salida de 1 A a 12 V la corriente DC de entrada era de 44,5 mA, lo que nos da una eficacia de alrededor del 83%. La frecuencia de

operación era de 95 kHz. Observe que cuando el diodo conduce la caída de tensión en el transistor es igual a la tensión de entrada más la tensión de salida multiplicada por la relación de transformación.

Auto-oscilación

En las fuentes de alimentación de red mayores, la temporización y control de la excitación para los semiconductores de potencia, a menudo son manejados por un circuito integrado. Aquí, todas esas funciones son realizadas por unos pocos componentes discretos. Primero, examinemos la teoría con la ayuda de la Figura 5. El elemento clave es el bobinado del tercer transformador, el cual está acoplado a los otros. Nosotros comenzamos nuestro análisis asumiendo que T1 se acaba de pasar a corte. La corriente de la bobina ahora circula a través del diodo D1 y cae de forma lineal. Tan pronto como la corriente alcanza cero, el diodo se bloquea. La bobina, junto con sus capacidades parásitas y esos otros componentes, forman ahora parte de un oscilador, el cual ejecuta medio ciclo. La tensión en el bobinado secundario cae y eventualmente pasa a negativo. Eso conduce a una tensión positiva que se produce en el tercer bobinado, la cual pone en conducción el transistor T1, haciendo que la tensión en el bobinado primario y secundario suban más, ayudando a mantener corto el tiempo de conmutación. T1 conduce y la corriente de la bobina sube linealmente mientras que la bobina es almacenada en ella. Después de un cierto tiempo que S2 está cerrado (a través de un circuito inteligente), la corriente de base de T1 cae. El proceso de cortar T1, una vez comenzado, se acelera por el circuito de realimentación. En este punto el ciclo comienza de nuevo desde el comienzo.

Todo lo que necesitamos hacer es explicar cómo se corta el transistor T1 después de un cierto intervalo de tiempo. La función de S2 en la Figura 5 la asume el transistor T2 en la Figura 4. Los componentes importantes aquí son R7 y C6. Después de la activación de T1 hay una tensión positiva en L2B relativa a masa. C6 se carga ahora de forma lenta hasta que T2 comienza a conducir. La corriente de colector de T2 roba entonces la corriente de base de T1, pasando T1 a corte. El circuito de realimentación acelera este proceso de conmutación al corte. La **Figura 7** muestra la tensión en la base de T2 (traza inferior) y la corriente de base de T1 (traza superior, medida usando una pinza de corriente). El transistor T1 es activado con una corriente de base de alrededor de 100 mA, la cual permanece constante durante un corto tiempo. Al mismo tiempo, la tensión en la base de T2 sube hasta alcanzar aproximadamente 0,8 V. La corriente de base comienza a caer y brevemente pasa a ser muy negativa, cuando los portadores de carga son eliminados desde la base; T1 entonces pasa a corte y la tensión en la base de T2 comienza a caer.

Inicio

R4 asegura que el circuito arranca correctamente. Debido a que hay una caída alta de tensión en esta resistencia, la hacemos a partir de una combinación de una serie de cuatro resistencias SMD de 240 kΩ. En el estado inicial R4 libera una corriente de base para T1, la cual nos la trae en su región de operación de amplificación lineal. Si R4 es para esta cuestión, la base de T1 no debe tener un camino DC de vuelta a L2B: la combinación de C5 y D6 asegura esto. Ellos permiten que L2B alcance la tensión de base, y también acopla señales AC para realimentación. Esta realimentación positiva asegura que el oscilador arranque. Antes de discutir la función de R3, C4 y D5,

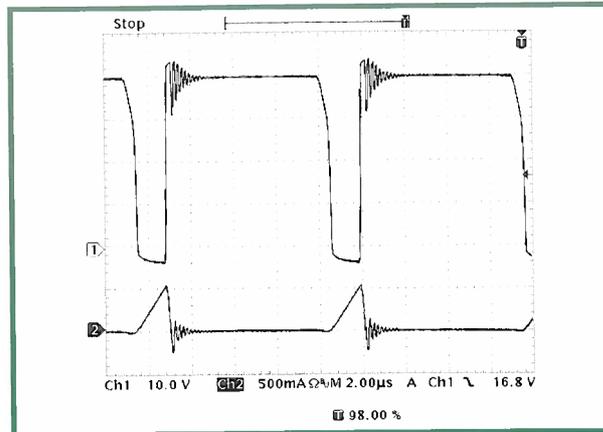


Figura 6. Formas de onda de tensión y corriente en el transistor conmutado.

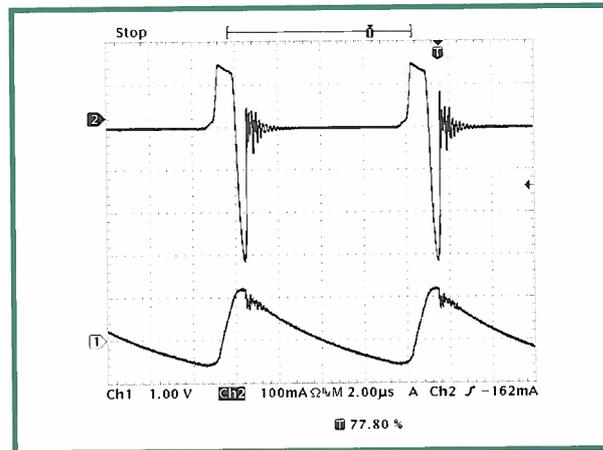


Figura 7. Tensión en la base de T2 (abajo) y corriente en la base de T1 (arriba).

haremos algunas indicaciones en el transformador, que es la componente principal de toda la unidad.

Transformador

El transformador es el único componente diseñado especialmente en el circuito. Analizarlo completamente sin dañarlo es un gran reto. Nosotros comenzamos desoldándolo desde la placa. Necesitamos medir los parámetros del transformador a su frecuencia de operación (100 KHz) para obtener los valores relevantes del circuito. Esto requiere de buenos instrumentos. Las medidas son fáciles de realizar, sin embargo: desde las trazas del osciloscopio de la Figura 6 podemos determinar que la corriente en L1A sube hasta aproximadamente 0,55 A dentro de 1,5 μs con una tensión de $U = 326$ V. Usando la ecuación $U = L \, di/dt$ obtenemos una bobina de $L = 900 \, \mu\text{H}$ aproximadamente. Las medidas con un puente nos da un valor de $L = 800 \, \mu\text{H}$. Si medimos la tensión en los tres bobinados con unas puntas de prueba adecuadamente calibradas también podemos medir la relación del transformador. Para obtener las vueltas totales bobinamos cinco vueltas de fino hilo de cobre esmaltado alrededor del centro del transformador y medimos su relación a los otros tres bobinados a 100 kHz. El resultado muestra que L1A tiene 56 vueltas, L2B tiene 3 vueltas y L2C 11 vueltas.

Esto significa que el corazón tiene un valor AL de aproximadamente $800 \, \text{mH}/56^2 = 255 \, \text{nH/turn}^2$. A_L es la bobina (L) asumiendo un bobinado con $N = 1$ vuelta. Mirando las hojas de características de varios fabricantes (tales como Epcos, Philips, Vogt y Kaschke) encontramos que esto es una buena elección para el núcleo E25/9/6 de Philips, con un agujero de aire de 200 mm. Con una corriente de pico de $I = 0,55$ A en el bobinado primario

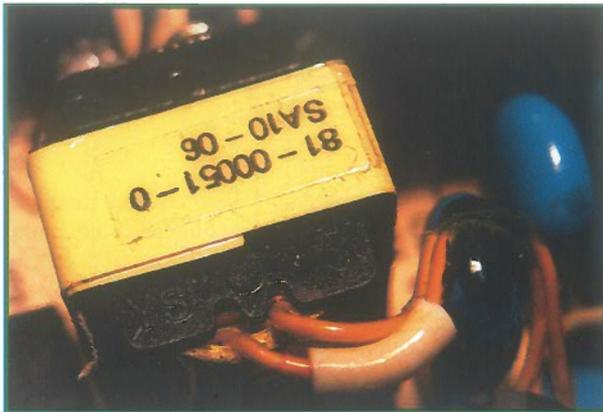


Figura 8. Un transformador miniatura usando cable triple-apantallado.

obtenemos un campo máximo de aproximadamente 200 mT en el núcleo, lo cual es completamente factible con los materiales de las modernas ferritas a 100 kHz. Las pérdidas del núcleo en este pico de corriente son de alrededor de 0,5 W a 0,7 W, un rango razonable para un núcleo de este tamaño desde el punto de vista térmico. Eso completa la descripción eléctrica del transformador, pero hay muchos detalles que parecen decisivos antes de que tengamos un diseño práctico. Son esos detalles los que hacen el diseño de las fuentes de alimentación de conmutación tan difíciles. Por ejemplo, el transformador debe proporcionar aislamiento desde la red con un nivel adecuado de seguridad y las corrientes de fuga se deben mantener dentro de límites aceptables. Con cinta aislante se puede realizar en el transformador el aislamiento y la separación, y pueden encapsularse todas las cosas, como presumiblemente estamos haciendo en este caso. La **Figura 8** nos muestra un pequeño transformador de una fuente de alimentación de 6 V, 1,5 A (9 W). Sólo tiene 13 mm de ancho y 14 mm de alto. El bobinado secundario de este tipo de transformador se hace a menudo usando un tipo especial de cable. El tipo se muestra en la Figura 8 que se conoce como 'cable con triple aislamiento' (disponible por ejemplo en la firma Furukawa). El cable es adecuado para este uso y proporciona aislamiento desde la tensión de red con tres capas de aislamiento independientes. El cable no se suelda normalmente al cuerpo de la bobina, porque las conexiones

estarían demasiado cerca del bobinado primario. El aislamiento de este tipo de transformador de retorno a menudo resulta en una gran inductancia parásita. En el momento en el que en el bobinado secundario haya una sobrecorriente, la energía almacenada en la inductancia parásita se convierte normalmente en un gran pico de tensión en el colector de T1, el cual puede dañarlo.

RCD amortiguadora

Para asegurar que esta energía no daña el transistor T1, se añade una red amortiguadora, formada por R3, C4 y D5. El diodo D5 conduce el pico de tensión para la red RC. El condensador carga y la energía se disipa lentamente en la resistencia. El diodo debe conmutar rápidamente para poder manejar altas tensiones. El condensador tiene una tensión de trabajo de 1 KV, así que puede soportar el pico. El pico de tensión y la oscilación resultante se pueden ver claramente en la Figura 6. La resistencia R3 es de una potencia considerable, presumiblemente de 1 W. Hay pérdidas de potencia en esta parte del circuito, el cual es un buen diseño que trataremos de minimizar.

Secundario

Por último analizaremos el secundario (**Figura 9**). D7 es un diodo Schottky de 3 A que no requiere de refrigeración extra para ayudar a disipar calor. La **Figura 10** muestra un ejemplo de una fuente de alimentación de 2 A donde el diodo se suelda en una posición simple para proporcionar una solución de bajo coste al problema de la disipación. En este ejemplo, el transistor T1 tiene un encapsulado plástico de aislamiento que está cubierto de un pequeño disipador (K1 en la Figura 4), el cual está conectado a masa para reducir EMI. El colector de T1 es uno de los puntos más calientes de todo el circuito. Los condensadores C8 y C9 forman un filtro para el rizado junto con L3. L4 es un choque en modo común. El condensador de clase Y, C7, sirve para mejorar las emisiones EMI del circuito. Nosotros no investigamos porque este condensador está conectado a este punto particular en el bobinado secundario, aunque presumiblemente esto es donde es más efectivo. El diodo zéner ZD2 limita la tensión de salida que debería ser necesaria cuando,

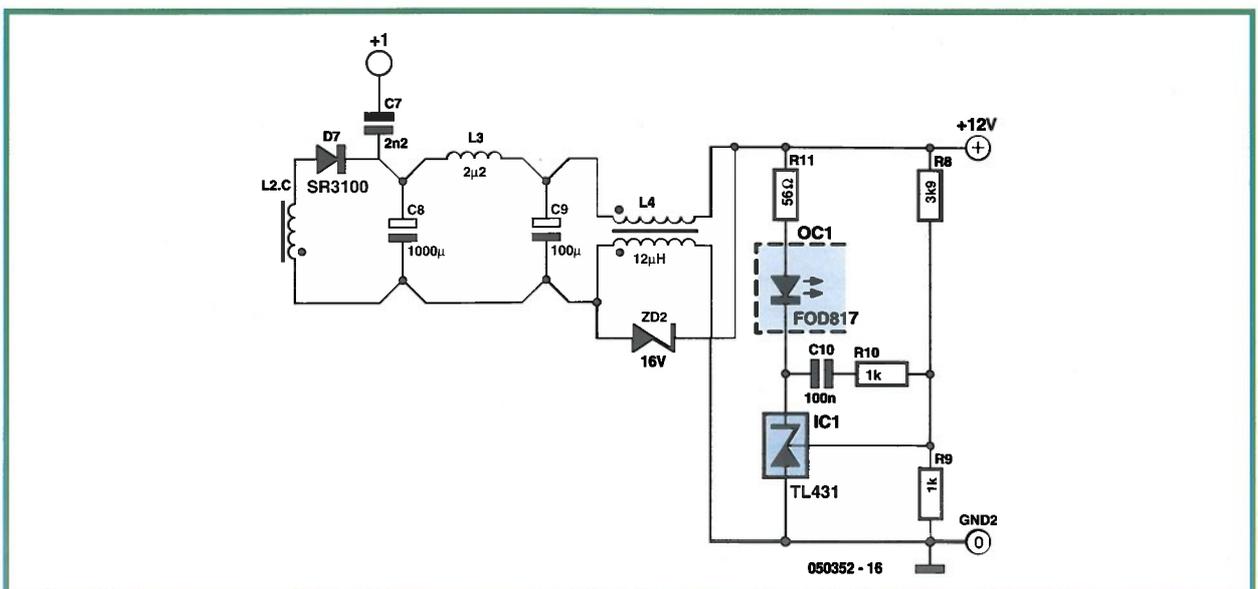


Figura 9. Red amortiguadora RCD.

por ejemplo, la carga se retira de pronto y el bucle de regulación no responde de forma bastante rápida. El diodo zéner ajustable (o regulador shunt) IC1 forma la base del bucle de regulación. Si la tensión de salida excede los 12 V, IC1 comienza a conducir y el LED se ilumina en el optoacoplador OC1. Esto comunica la vuelta al primario donde el fototransistor opera en paralelo con R7. El condensador C6 se carga más rápido y T1 se pone en corte más rápidamente, con el efecto de un menor almacenamiento de energía en la bobina. La cantidad de energía transferida entre el primario y secundario cae y con ella la tensión de salida. La frecuencia de la operación del circuito también está afectada: sin carga (y el convertidor sólo suministra sus propias pérdidas) puede ejecutarse hasta 600 kHz.

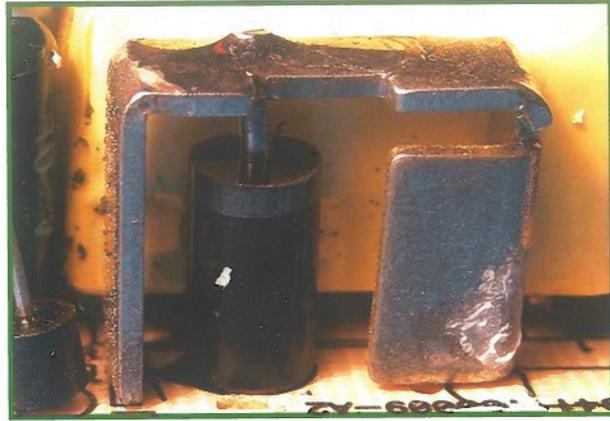


Figura 10. Diodo con dissipador térmico y eléctricamente conductor.

Nuestra versión

Ahora que hemos analizado en detalle un puñado de componentes, naturalmente desearíamos construir un convertidor similar. La **Figura 11** muestra un simple diseño para un convertidor de 24 V a 5 V a 500 mA. Si usáramos un 7805 para hacer esto, disiparía 9,5 W. Incluso si

el convertidor conmutado tiene una eficiencia de sólo el 65 %, funcionará mucho mejor que un regulador serie. En este circuito podemos administrar la red amortiguadora: si los tres bobinados se realizan con la construcción trifilar, el BD237 puede manejar sin problemas picos de tensión.

(050352-1)

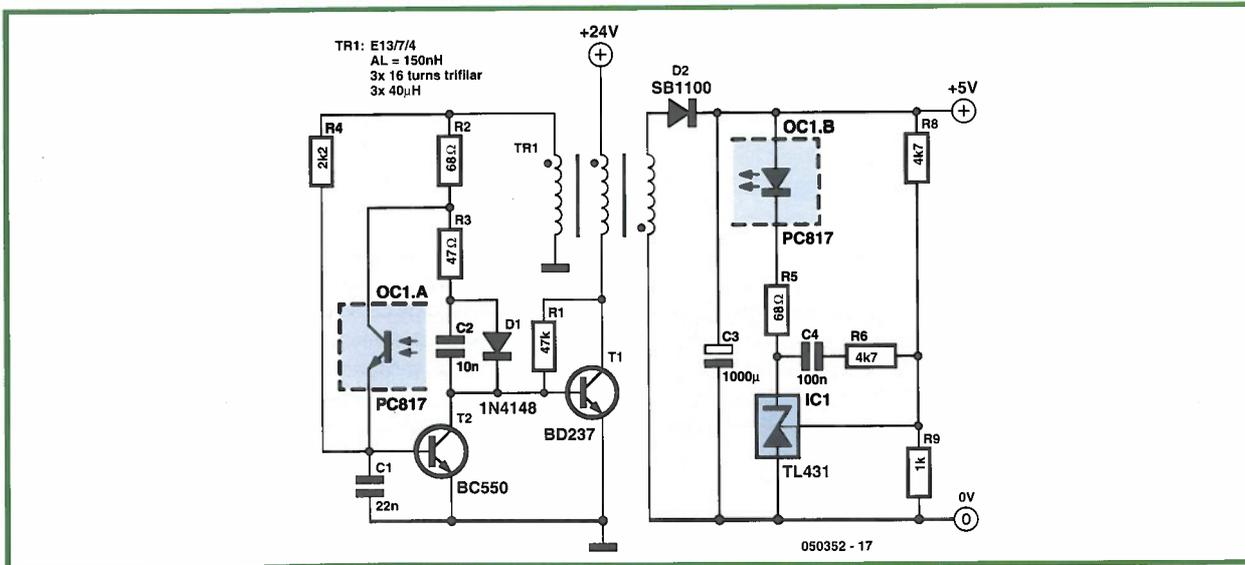


Figura 11. Convertidor reductor de 24 V a 5 V.

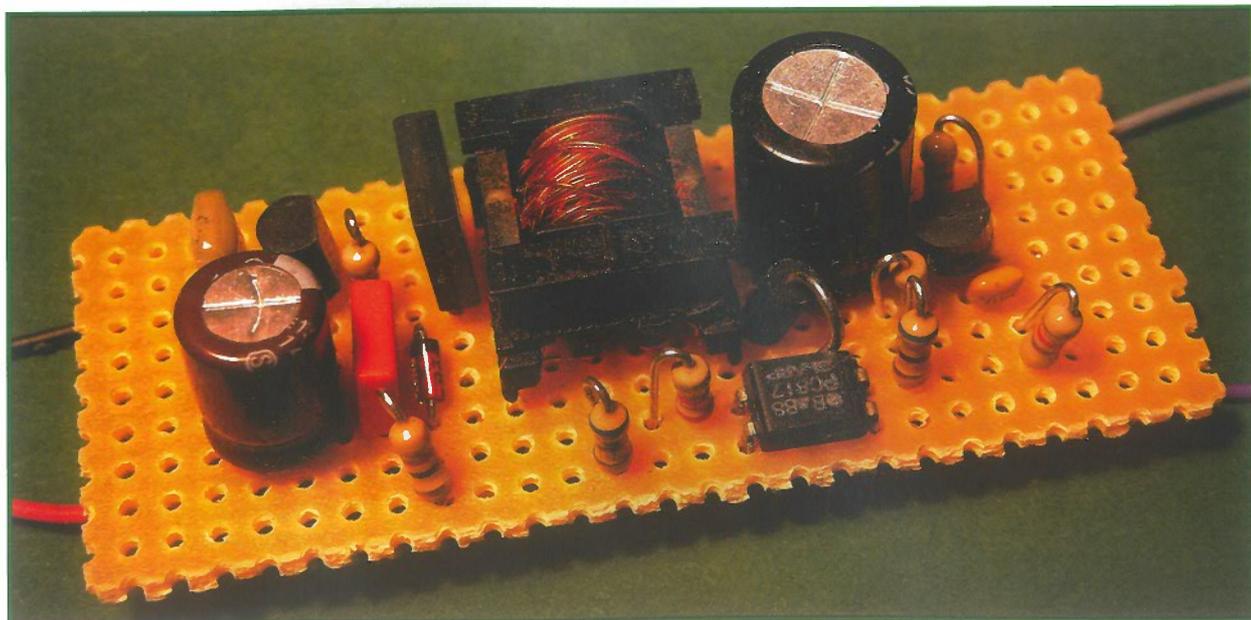


Figura 12. Prototipo de pequeño convertidor.

Simple Caracterizador

Simple caracterizador de pilas AA recargable

Fons Janssen

Las pilas recargables AA de 1,2 V a.k.a, 'pilas de linterna', son muy populares en una gran variedad de aplicaciones portátiles. El número modelos diferentes es enorme y consecuentemente la calidad de esas pilas puede diferir considerablemente. No debe sorprendernos que la capacidad real de las pilas pueda diferir considerablemente de lo que está especificado, y en bastantes casos la capacidad se deteriora enormemente después de un par de ciclos de carga/descarga. Este circuito nos da una visión global de las pilas AA.



El circuito propuesto, en combinación con un PC y algo de software inteligente, puede medir con precisión un ciclo completo de descarga de una pila recargable AA de 1,2 V o AAA, también conocida como de 'linterna' y 'bolígrafo' respectivamente. Desde la curva resultante, la capacidad y resistencia interna se pueden determinar de forma simple.

El circuito está montado con el DS2751, el cual está realmente pensado para la monitorización de la carga de un pack de baterías (Maxim/Dallas lo llama 'battery fuel gauging'). El DS2751 proporciona la clave de los componentes hardware requeridos para estimar la capacidad restante de packs de baterías para integración de baja potencia, medidas de precisión de temperatura, tensión, corriente y acumulación de corriente. Este hardware también se puede usar para grabar con precisión un ciclo de descarga de una simple pila AA (bien NiCd o NiMH).

El circuito

El circuito de la **Figura 1** es muy sencillo. Un PC está conectado a un puerto USB. La energía se extrae de la tensión de bus USB, así que no es necesaria una fuente de alimentación externa. IC1 (un MAX8881EUT33) es un regulador de tensión que genera 3,3 V para IC2

de Pilas AA Recargable

(DS2490S). IC2 convierte el protocolo USB al protocolo de un hilo para permitir al PC comunicarse con IC3 (DS2751E-025), el cual es el componente clave en el circuito. Mide la tensión de la pila en el pin 1 (VIN) a través de R2 y la corriente de pila a través de una resistencia interna de 25 mΩ entre los pines 7 (SNS) y 2 (VSS, GND). El pin I/O programable, PIO (nº 3) permite al software del PC controlar la puerta de alimentación del FET T1. Cuando PIO es alto, T1 se activa y la pila se carga a través de R4. Esto resultará en una corriente de descarga de aproximadamente 1 A. Cuando PIO es bajo, T1 se corta y la pila no se carga.

Software

Se ha escrito un programa en Visual Basic para obtener un ciclo de descarga de una pila. El programa está disponible para ser **descargado de forma gratuita** desde la página web bajo el número **050394-11.zip**. Lo encontraremos bajo el nombre MAGAZINE -> April 2006 -> Caracterizador de pila AA. El software inicializa IC3 y pone en corte T1. Cuando la pila esté completamente cargada se puede insertar y el software presentará la medida de la tensión de la pila (vea la **Figura 2a** para una pantalla de software). Una vez presionado el botón 'iniciar la medida', el software activará T1 y tomará la tensión y corriente de la pila cada segundo. Después de cada punto tomado, T1 se apaga brevemente para permitir la medida de la tensión de pila descargada, la cual también es tomada (vea la pantalla en la **Figura 2b**). Para no dañar la pila, la medida se detiene automáticamente tan pronto como la tensión de la pila cae por debajo de 0,8 V. Después de que la medida se ha detenido, la corriente acumulada muestra cuánta carga se ha consumido de la pila, proporcionando una indicación de la capacidad real. La medida también se puede detener pulsando sobre el botón 'Stop Measurement' -Parar la medida-. Pulsando en 'Save data' -Guardar dato- el dato tomado se puede guardar en disco para posteriores análisis.

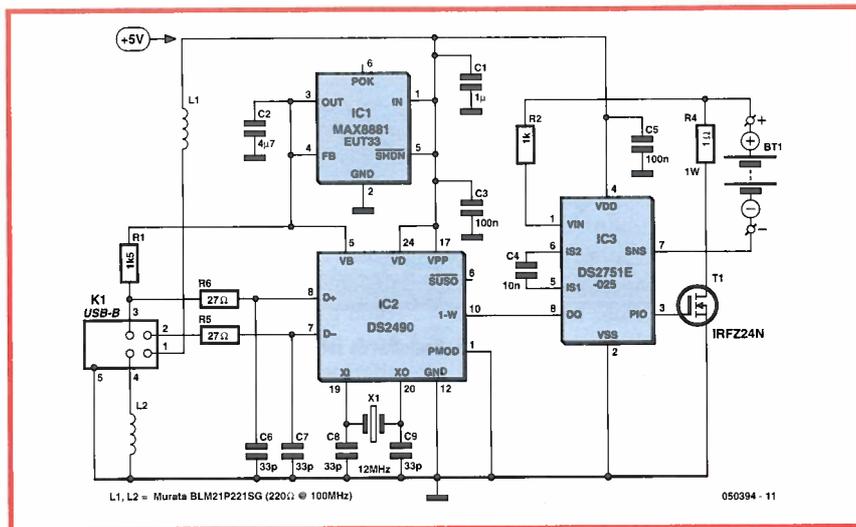


Figura 1. Esquema del circuito para la caracterización de la descarga de una pila AA.

Procesamiento de datos

El dato se puede analizar usando Microsoft Excel. Un ejemplo se muestra en la **Figura 3**. El fichero tomado se puede importar a través de File_Open y seleccionando después ficheros de tipo: 'all files' -todos los ficheros-. A continuación elija el fichero de datos, escoja 'delimited' y presione 'finish'. Vera cuatro columnas que representan los datos almacenados:

- **ACR [mAh]**, la corriente acumulada en mAh;
- **V1 [V]**, la tensión de la pila descargada en voltios;
- **V2 [V]**, la tensión de la pila cargada en voltios;
- **I [A]**, la corriente de descarga en amperios.

Las columnas 2, 3, y 4 se pueden usar para determinar la resistencia interna de la pila mediante esta ecuación:

$$R_{\text{cell}} = [(V_{\text{UL}} - V_{\text{L}}) / I_{\text{L}}] - R_{\text{sense}}$$

Que se puede simplificar así

$$[(V1 - V2) / I] - 25 \text{ m}\Omega$$

donde

- V_{UL} = tensión de descarga;
- V_{L} = tensión de carga;
- I_{L} = corriente de carga.

Póngase la fórmula de Excel '= (B3-C3)/D3-0.025' en la pila E3 y cópiese al resto de columnas hasta el último punto tomado en E. Ahora podremos hacer una gráfica X-Y para mostrar el resultado. Escoja la columna A como valores de X y las otras columnas como valores Y. veremos la tensión/corriente/resistencia como una función de corriente de descarga. En la fotografía introductoria puede ver un ejemplo.

Drivers para un hilo

Para habilitar la comunicación entre el PC y el DS2490S, necesitamos instalar los drivers de un hilo. Pueden descargarse desde la página web de Maxim/Dallas a través del siguiente enlace: www.maxim-ic.com/products/ibutton/software/tmex/.

Después de la instalación, ejecutaremos 'default 1-wire net' para encontrar el DS2490S conectado y seleccionarlo como el 'default 1-Wire net'.

Otras pilas

El circuito se puede usar para otras pilas de tamaño AA o AAA. También es posible realizar una medida de más pilas en serie. Siempre debemos asegurarnos que no se exceden los valores máximos. Escogeremos una resistencia de carga, de forma que la corriente de descarga no

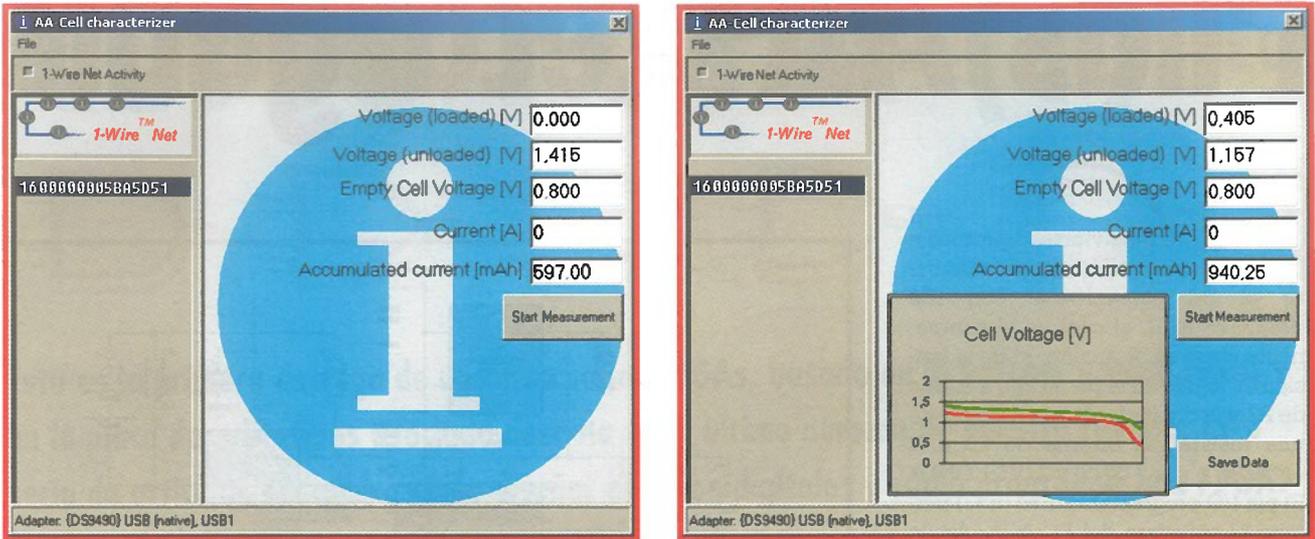


Figura 2. Pantallas de software (izquierda: arranque, derecha: durante la medida).

Precaución
 Todos los tipos de baterías mencionados en este artículo pueden suministrar elevadas corrientes de corto-circuito muy peligrosas. Por ello, debemos evitar daños y/o situaciones peligrosas debidas a sobrecargas de la batería, sobrecalentamiento y explosión. Usaremos sólo los cargadores adecuados.

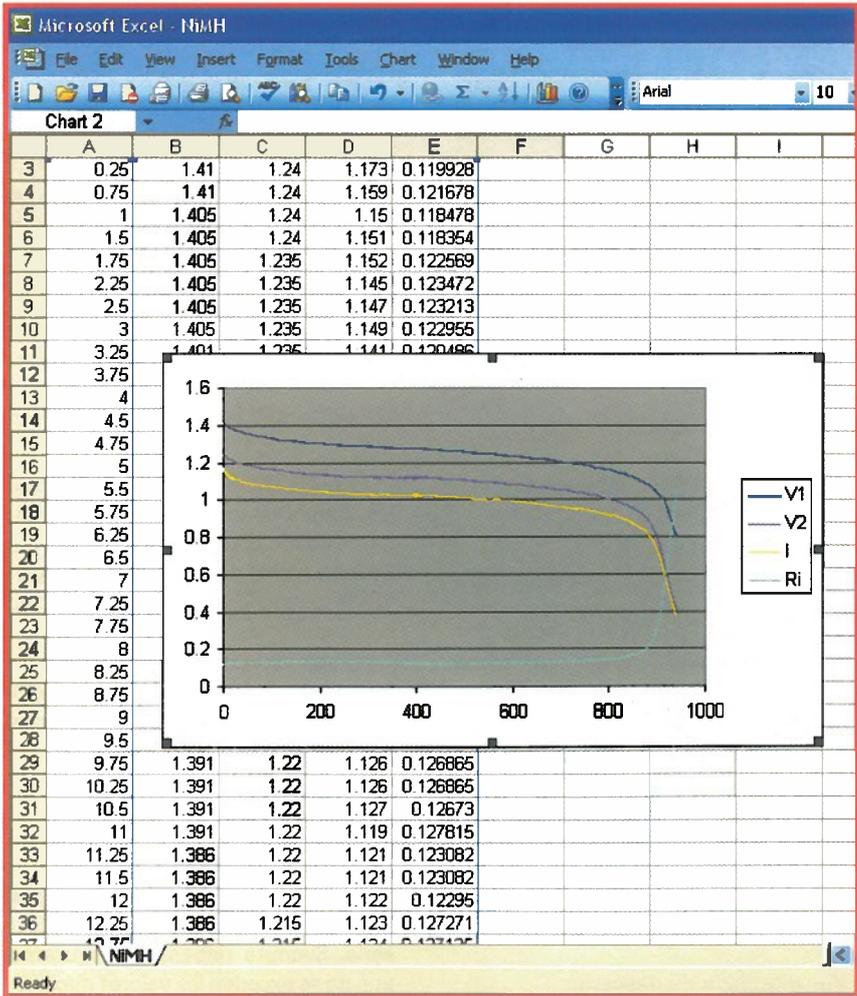


Figura 3. Ejemplo de una hoja Excel que contiene los datos de una batería capturados por el circuito.

sobrepase los 1,9 A en el rango de escala completa del DS2751E-025. También nos cercioraremos de que V_{in} en el pin 1 del DS2751E-025 no excede el valor máximo de 4,5 V. Usaremos un divisor resistivo si es necesario. Es recomendable limitar la corriente de descarga a aproximadamente C mA para evitar dañar a la pila o a nosotros mismos. C es la capacidad nominal en mAh, que normalmente aparece impresa en la pila.

Dentro del DS2751

El DS2751 fue desarrollado originalmente para añadir la funcionalidad 'fuel-gauging' a los packs de baterías. La **Figura 4** muestra un diagrama de bloques funcional del DS2751. A través de DQ, un host puede leer/escribir los registros usando el protocolo de un hilo. Esos registros contienen el dato medido, información de estado, y también alguna información dedicada que se puede almacenar por el host (por ejemplo, fecha de fabricación, número de serie, etc). El DS2751 utiliza un método llamado 'contador de columb' para medir la cantidad de carga que va dentro y fuera de la pila. Ésta se alcanza integrando digitalmente la precisión de la corriente medida a través de la resistencia de sensado. El resultado se almacena en el aCR (registro de corriente acumulada) y se puede usar por el host para calcular la capacidad sobrante. Debido a que la capacidad también se ve afectada por la temperatura de la pila, se ha añadido un sensor de temperatura. El DS2751E-025 tiene una resistencia interna de sensado de 25 mOhm (la

DS2751E se usa en combinación con una resistencia de sensado externa). La corriente se almacena como un valor de 12 bits con una resolución de 0,625 mA y un rango a fondo de escala de $\pm 1,9$ A. La corriente acumulada se almacena como un valor de 16 bit con una resolución de 0,25 mAh y $\pm 8,2$ Ah a fondo de escala. La tensión medida tiene un máximo de 4,5 V, así que se puede usar para un pack de batería con una pila de Li-ion, e incluso para tres pilas de NiMH en serie.

Construcción

Se diseñó una PCB miniatura a simple cara para el proyecto, que se muestra en la **Figura 5**. La placa lista para ser usada está en el Servicio de Lectores bajo el código 050394-1. Todos los ele-

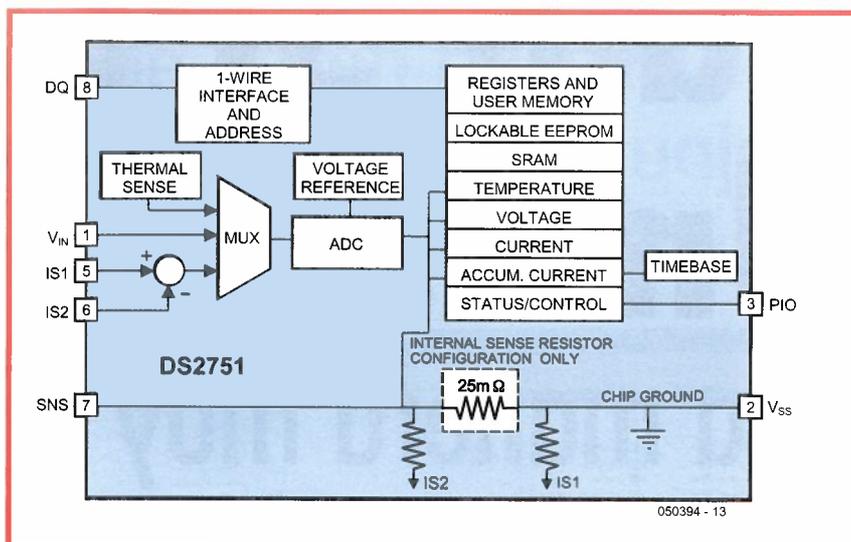


Figura 4. Esquema funcional del DS2751 (cortesía Maxim/Dallas).

Sabíais que

Maxim/Dallas nos ha informado que los lectores de Elektor que deseen construir el proyecto del caracterizador de pilas AA pueden pedir muestras gratuitas de los integrados principales desde esta url:

www.maxim-ic.com/samples

Sin embargo, el suministro de muestras es competencia exclusiva de Maxim/Dallas, y puede que no sean posibles los envíos a todos los destinos.

mentos SMD van a la cara de soldadura de la placa, dejando sólo el oscilador, el FET de potencia IRFZ24N y el conector USB-B en la parte superior.

(050394-1)

Fons Janssen
(Maxim Integrated Products Inc., Benelux)
fons_janssen@maximhq.com

Más información

Interface Serie para bus de un hilo,
Elektor Junio 2002.

Hojas de características del MAX8881:
<http://pdfserv.maxim-ic.com/en/ds/MAX8880-MAX8881.pdf>

Hojas de características del DS2490:
<http://pdfserv.maxim-ic.com/en/ds/DS2490.pdf>

Hojas de características del DS2751:
<http://pdfserv.maxim-ic.com/en/ds/DS2751.pdf>

Herramientas software de 1-hilo/
iButton: <http://www.maxim-ic.com/products/1-wire/software/>

LISTADO DE COMPONENTES

Resistencias:

Todas SMD, encapsulado tipo 0805
R1 = 1kΩ5
R2 = 1kΩ
R3 = 10kΩ
R4 = 1Ω 2W
R5,R6 = 27Ω

Condensadores:

Todas SMD, encapsulado tipo 0805
C1 = 1μF
C2 = 4μF7
C3,C5 = 100nF
C4 = 10nF
C6-C9 = 33pF

Bobinas:

L1,L2 = BLM21P221SG (Murata) (220Ω @ 100MHz)

Semiconductores:

T1 = IRFZ24N
IC1 = MAX8881EUT33 (Maxim/Dallas)
IC2 = DS2490S (Maxim/Dallas)
IC3 = DS2751E-025 (Maxim/Dallas)

Varios:

K1 = conector USB, montaje PCB, Tipo B
Soporte de batería para una pila miniatura de linterna AA
PCB, código de pedido **050394-1**, véase la página del Servicio de Lectores
Disco, software PC, código de pedido **050394-11**

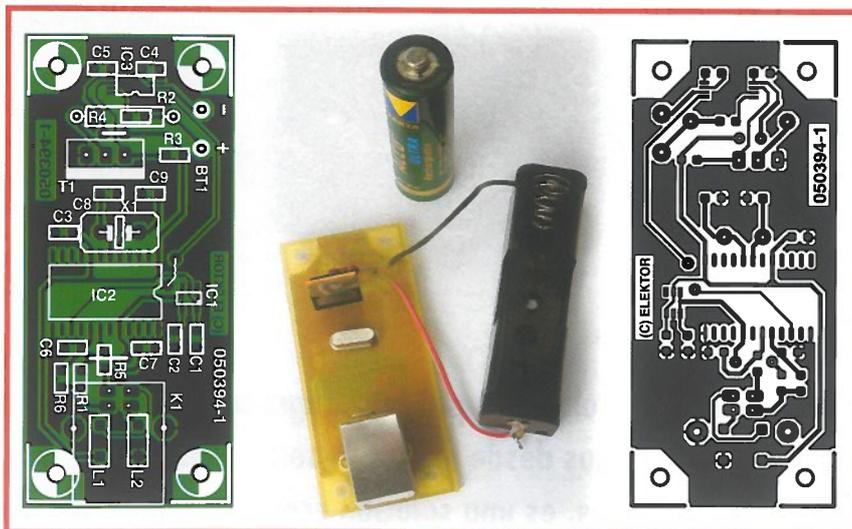
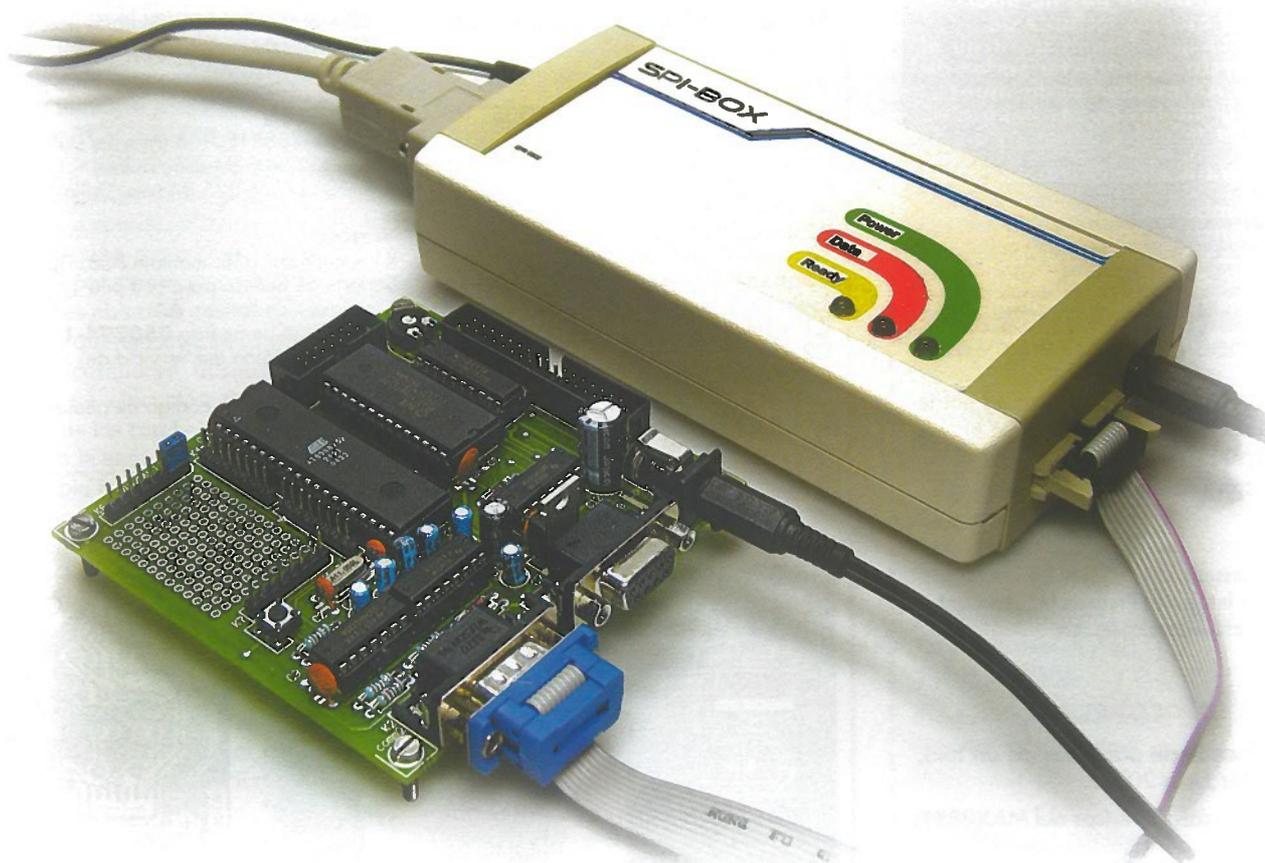


Figura 5. La cara de cobre y la del montaje de componentes de la pequeña placa del circuito que hemos diseñado para el analizador de batería.

Caja SPI Universal

Una manera muy rápida de programar microcontroladores

Rainer Reusch



Muchos microcontroladores pueden programarse a través del bus SPI. La posibilidad de trabajar con este bus desde el puerto serie de un ordenador, añadiendo sólo un par de componentes extras, es una solución económica pero lenta. Una alternativa mejor consiste en utilizar la caja SPI que se describe en este artículo, la cual proporciona una velocidad aceptable incluso cuando se usa con un dispositivo conversor USB a serie.

Una búsqueda por Internet nos proporcionará cierta cantidad de juegos de instrucciones que nos mostrarán cómo utilizar un par de componentes adicionales para hacer la conexión entre un puerto serie o un puerto paralelo y el "Serial Peripheral Interfaz" (es decir, "Interfaz Serie para Periféricos"), o bus SPI [1]. Para la programación ocasional de un microcontrolador este tipo de solución es indudablemente muy aceptable. Sin embargo, el uso de estas interfaces tiene un par de desventajas.

La primera es que la velocidad de transferencia de datos sobre la interfaz serie es relativamente lenta y puede llevarnos a un ciclo de desarrollo (programa, prueba, programa, prueba y así sucesivamente), donde tendremos que hacer una parada cada vez que sea necesario descargar nuevos datos sobre el microcontrolador, lo cual puede conducir a una situación cada vez más irritante. Además, la situación puede empeorar si tienen que utilizarse conversores USB a serie, disponibles comercialmente (los ordenadores portátiles modernos, por ejemplo, ya no soportan las interfaces prehistóricas tales como los puertos serie y paralelos). En este caso los bytes sólo pueden enviarse hacia el microcontrolador de manera muy lenta, debido a que la latencia que provoca el uso del protocolo USB hay que añadirla en cada transacción. Esto proporciona un tiempo adicional que se tiene que sumar al tiempo total que se utiliza en volver a programar el microcontrolador.

La opción del puerto paralelo es decididamente más rápida, pero el uso de este puerto lleva consigo el riesgo de producir daños en la placa base del ordenador: descargas estáticas o fallos en la conexión de los circuitos que pueden ser peligrosos. Normalmente, el puerto paralelo no lo proporciona un circuito integrado interfaz independiente, lo que significa que es posible que sea necesario un controlador especial, de manera que podamos conseguir que el puerto tenga un comportamiento significativamente diferente del que normalmente se requiere. Como ya hemos mencionado anteriormente, los ordenadores portátiles ya no disponen de un puerto paralelo en su configuración, motivo por el que hemos desestimado esta solución.

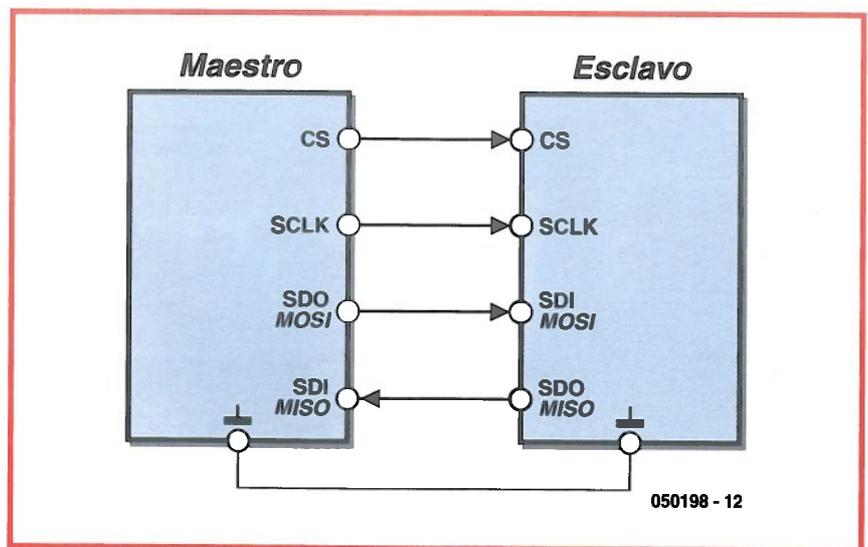


Figura 1. El bus SPI dispone de cuatro conexiones entre el dispositivo maestro y el esclavo.

Por lo tanto, un circuito dedicado, como puede ser la caja SPI que se describe en este artículo, solucionaría todas estas desventajas de un modo relativamente sencillo.

SPI

Un puerto SPI bidireccional conecta un dispositivo "maestro" y un dispositivo "esclavo" sobre cinco hilos: cuatro señales de datos más una de masa (vea la Figura 1). En nuestro caso, el dispositivo maestro es la propia caja SPI conectada al ordenador, mientras que el microcontrolador que va a ser programado ejerce el papel de dispositivo esclavo. La interfaz está diseñada expresamente para realizar la comunicación entre un microcontrolador y un componente periférico, así como para la programación de los microcontroladores en sus circuitos finales ("programación en el propio sistema"), y puede alcanzar una velocidad de transferencia de datos que llega hasta los 5 Mbits/s. El sistema del bus SPI, que fue diseñado por la casa Motorola, no es un bus estándar, pero debido a su sencillez y al amplio abanico de circuitos integrados compatibles con el bus SPI, este bus está disponible en componentes de una gran variedad de fabricantes. La gama completa de dispositivos abarca desde los microcontroladores y memorias hasta los conversores A/D, sensores e interfaces a otros sistemas de bus tales como el bus CAN. Por tanto, la caja SPI es adecuada no solamente para la

programación de microcontroladores, sino también para controlar circuitos integrados compatibles con el bus SPI, tales como memorias EEPROM y conversores A/D.

Los procesos de transferencia de datos se realizarán como se describe a continuación. Si el dispositivo maestro desea transmitir un dato hacia un dispositivo esclavo, activa la línea de la señal "chip select" ("selección de dispositivo", CS, activa a nivel bajo). La casa Motorola también hace referencia a esta señal como la línea SS (selección de esclavo). A continuación el dispositivo maestro configura un reloj en la línea SCLK y transfiere los datos de salida de manera síncrona sobre la línea SDO (Serial Data Out, es decir, datos de salida serie), la cual está conectada a la línea SDI (Serial Data In, es decir, datos de entrada serie). La casa Motorola también referencia la conexión de la línea SDO del dispositivo maestro con la línea SDI del dispositivo esclavo con las siglas MOSI (Master Out, Slave In, es decir, maestro salida, esclavo entrada). El flujo de datos en la dirección opuesta se produce de manera similar, es decir, la información va desde la línea SDO del dispositivo esclavo hacia la línea SDI del dispositivo maestro, que la casa Motorola denomina MISO (Master In, Slave Out, es decir, maestro entrada, esclavo salida). El bus SPI funciona con los ya tradicionales niveles lógicos de 0 y + 5 V. Si el dis-

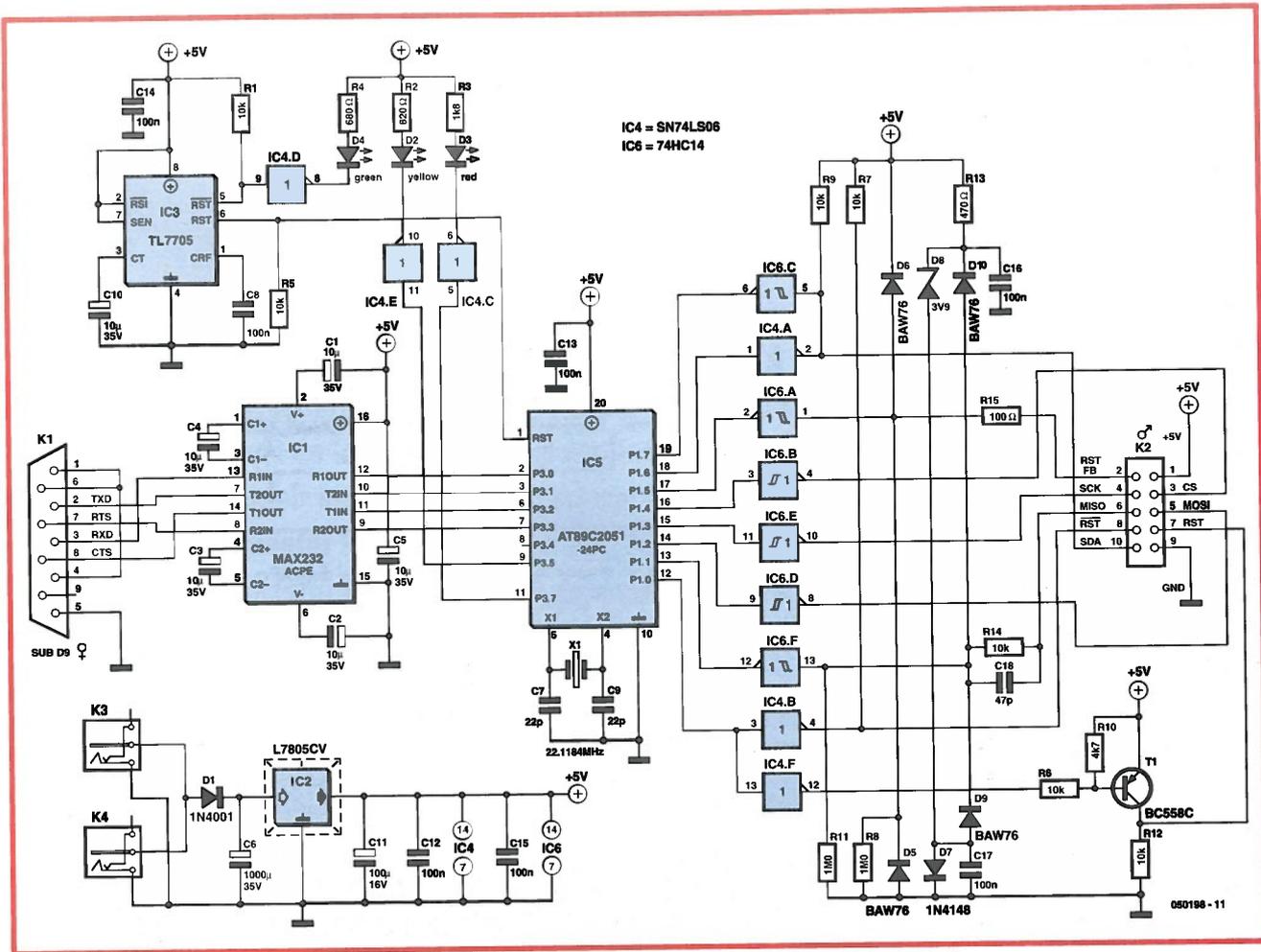


Figura 2. El circuito de la caja SPI está compuesto tan sólo por componentes estándar junto con un microcontrolador de la casa Atmel.

positivo maestro está comunicado con diferentes dispositivos esclavos, se requiere una línea CS independiente para cada elemento esclavo, mientras que las líneas de datos están conectadas en la forma de un bus tradicional a todos los dispositivos esclavos. Todo es bastante sencillo, en teoría.

Preguntas y respuestas

La pregunta que nos queda por resolver es: ¿por qué es tan lento el uso de un puerto serie para controlar un bus SPI? La respuesta es bien sencilla, para simular las señales del bus SPI no solamente debemos controlar las señales RX y TX del puerto serie, sino que también deben de poder manejarse, de manera individual, las distintas señales que realizan el control de flujo de información. La circuitería de la interfaz serie no estaba optimizada para este modo de funciona-

miento y, como resultado de ello, no es posible conseguir una alta transferencia de datos de esta manera. Las medidas realizadas sobre ordenadores en la "vida real" muestran que el mínimo ancho de pulso que puede obtenerse en una línea de control de flujo de un puerto serie está alrededor de los 40 µs. Este tiempo se corresponde con una frecuencia máxima de trabajo de, aproximadamente, 12 KHz. Puesto que se tienen que generar tanto las señales de datos como la señal de reloj, la máxima frecuencia de reloj que se puede alcanzar en la práctica es incluso inferior, lo que da como resultado unos tiempos globales de varios segundos, incluso si tan sólo tenemos la necesidad de transferir unos pocos cientos de bytes de información. La situación puede ser peor si queremos utilizar dispositivos comerciales conversores de bus USB a bus serie. En estos dispositivos, el tiempo de latencia

asociado con las señales de control está en el rango de los milisegundos, lo que da como resultado unos tiempos de programación excesivamente largos.

Por lo tanto, pensamos que una mejor opción es utilizar el puerto serie para la transmisión de datos para la que fue concebido y, posteriormente, realizar la conversión de salida entre la información RS 232 y el bus SPI, utilizando un microcontrolador como interfaz. El microcontrolador es capaz de generar señales muy rápidas en sus salidas y en esto se basa la idea que es el corazón de nuestra caja SPI.

Se puede comprar un dispositivo de programación de microcontroladores ya hecho, equipado con el bus SPI. Por ejemplo, la casa Atmel dispone del "AT 89 SPI Tool". Entonces, ¿por qué molestarnos en hacer nuestra caja SPI cuando ya hay una totalmente fabricada en el mercado? La ventaja de una solución hecha en

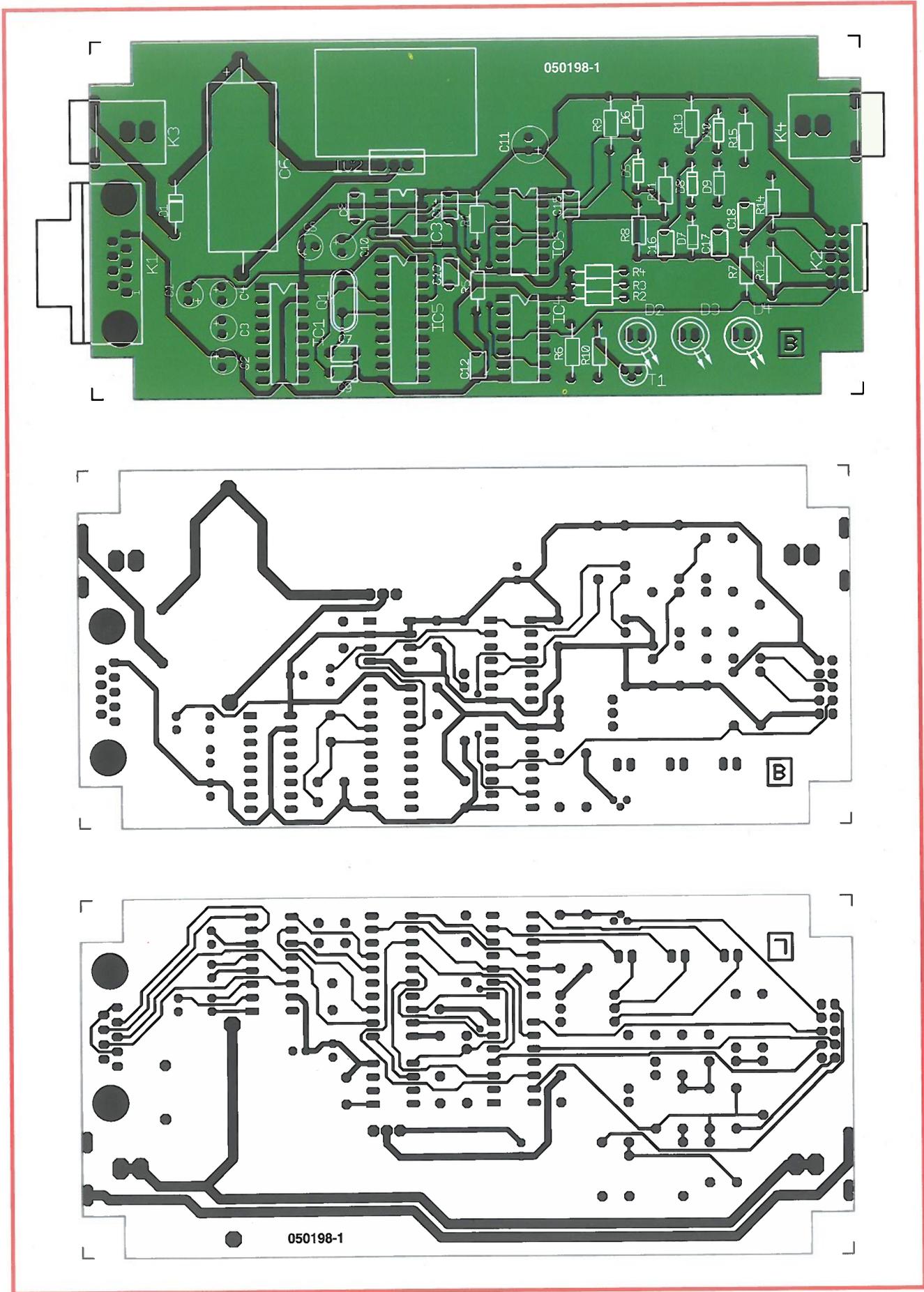


Figura 3. La placa de circuito impreso de doble cara para la caja SPI está montada utilizando únicamente componentes con terminales "normales".

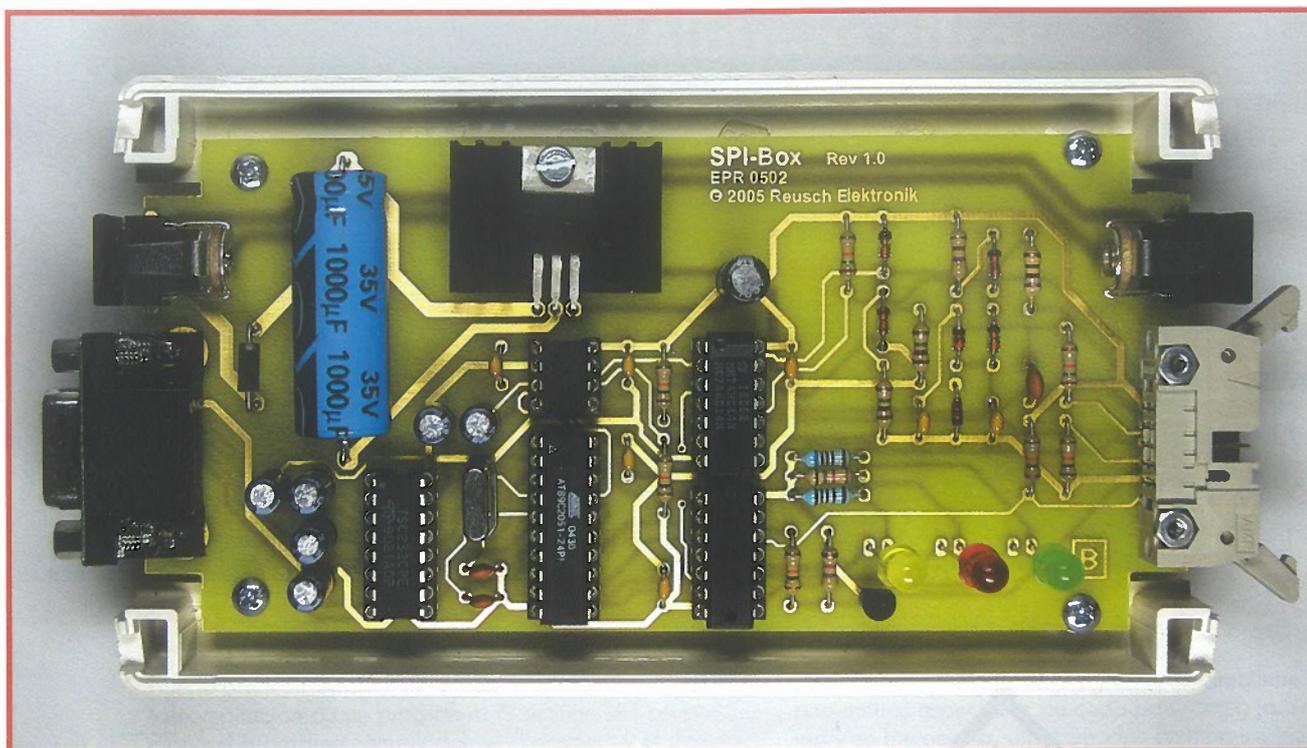


Figura 4. La placa de circuito impreso del prototipo con todos los componentes montados.

casa está ligada principalmente a su flexibilidad: por ejemplo, si necesitásemos programar otro tipo de microcontrolador nos veríamos en la necesidad de tener que comprar un nuevo programador para utilizar los microcontroladores RISC de la casa Atmel (por ejemplo). Otra ventaja es que conseguimos obtener una comprensión más profunda de cómo fun-

ciona la programación de los microcontroladores haciendo el montaje nosotros mismos y, además, la Caja SPI es más barata que la unidad que tendríamos que comprar.

La caja en sí misma

Como ya hemos mencionado anteriormente, la caja SPI es un tipo de

convertor de información serie a SPI. En la **Figura 2** se muestra el esquema eléctrico del circuito cuyo corazón es el integrado IC5, un microcontrolador del tipo AT 89C2051 de la casa Atmel, compatible con el 8051. Si utilizamos un cristal de cuarzo de 22,1184 MHz (X1), podemos obtener una velocidad de transferencia de datos exactos de

LISTA DE MATERIALES

Resistencias

- R1, R5, R6, R7, R9, R12, R14 = 10kΩ
- R2 = 820Ω
- R3 = 1kΩ
- R4 = 680Ω
- R8, R11 = 1MΩ
- R10 = 4kΩ
- R13 = 470Ω
- R15 = 100Ω

Condensadores

- C1-C5, C10 = 10µF condensador electrolítico de 35 V radial
- C6 = 1000µF condensador electrolítico de 35 V radial
- C7, C9 = 22pF
- C8, C12-C17 = 100nF
- C11 = 100µF condensador electrolítico de 16 V radial

C18 = 47pF

Semiconductores

- D1 = 1N4001
- D2 = LED, amarillo de baja corriente
- D3 = LED, rojo de baja corriente
- D4 = LED, verde de baja corriente
- D5, D6, D9, D10 = BAW76
- D7 = 1N4148
- D8 = Diodo zéner de 3,9 V y 400 mW
- IC1 = MAX232ACPE
- IC2 = 7805
- IC3 = TL7705ACP
- IC4 = 74LS06
- IC5 = AT89C2051-24PC, programado, con código de pedido **050198-41***
- IC6 = 74HC14
- T1 = BC558C

Varios

- X1 = Cristal de cuarzo de 22,118 MHz
- Bu1, Bu2 = Conector de adaptador de red para conector de 2,1 mm, en

ángulo recto para su montaje en placa de circuito impreso

- K1 = Conector Sub-D de 9 terminales macho en ángulo recto para montaje en placa de circuito
- K2 = Conector tipo "boxheader" de 10 terminales en ángulo recto para montaje en PCB
- Zócalos para los circuitos integrados IC4, IC5 e IC6
- Radiador para el circuito integrado IC2 (encapsulado TO-220)
- Caja de la casa OKW, del tipo TOPTEC 154F
- Adaptador de tensión de red con conector de 2,1 mm macho (negativo en el anillo exterior), de 8 - 15 VDC y de 300 mA
- PCB, Placa de Circuito Impreso con código de pedido nº **050198-1**, disponible en PCB Shop
- Programas para ordenador con los ficheros para el AT 89S en binario y en hexadecimal: descarga gratuita con la referencia N°: 050198-11 de página web

115.200 bits/s en el puerto serie y, al mismo tiempo, el microcontrolador puede trabajar a una velocidad lo suficientemente rápida como para conseguir velocidades de transferencia de datos elevadas en el lado del bus SPI.

La conexión entre el microcontrolador y el ordenador se consigue utilizando un convertidor de nivel del tipo MAX 232 (IC1). Para reducir al mínimo la posibilidad de errores en la gestión de las señales de control, este componente se utiliza habitualmente en el puerto serie y es el motivo por el que la señal RTS también se tiene en cuenta en el circuito integrado IC1.

Las conexiones con el bus SPI (ampliado) se toman a través de nueve inversores (IC4 e IC6) y del transistor T1, cableado como inversor. Los circuitos de estabilización, formados por los diodos D5 a D10, evitan posibles daños en los componentes debidos a los altos niveles de tensión o a la descarga de cargas estáticas. Los restantes inversores de conector abierto controlan tres diodos LEDs que realizan la función de indicadores de estado. Como podemos ver en la inspección del conector K2, en el lado del bus SPI de la unidad no solamente se ofrecen las cuatro señales SPI habituales, sino que también se han añadido un par de señales adicionales que pueden ser muy útiles cuando intentamos controlar dispositivos SPI externos.

La función de Reset la proporciona el circuito integrado IC3 y asegura que se encienden en los circuitos integrados en un estado definido, al mismo tiempo que este circuito integrado controla la línea de alimentación de +5 V. Una de sus salidas controla el diodo LED verde, el cual se ilumina cuando la unidad está lista para su funcionamiento. Si el diodo D4 no se enciende puede deberse a dos motivos, porque la tensión de alimentación está ausente, o porque la tensión presente en la línea de +5 V no tiene un valor que esté dentro de las tolerancias requeridas para un funcionamiento estable.

El diodo LED D3 se ilumina cuando los bytes de datos están enviándose a través de la conexión SPI. El diodo LED amarillo D2 se controla por el ordenador a través del programa que está almacenado en el microcontrolador (IC5). Cuando la unidad se

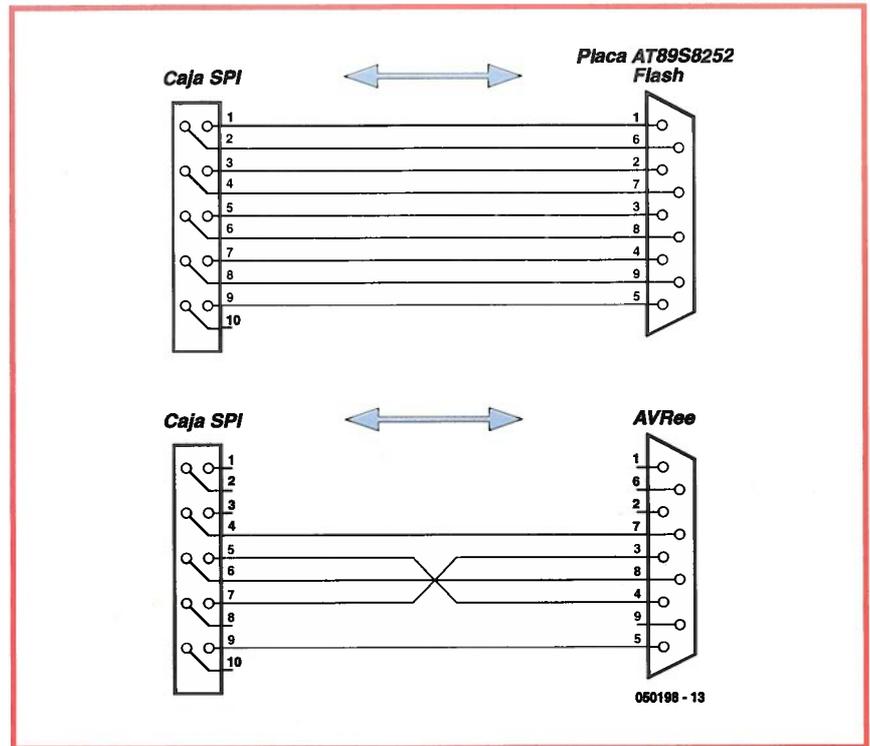


Figura 5. Distribución de terminales de los cables de la interfaz utilizado con la caja SPI para la Placa Microcontroladora Flash 89S8252 y para la Placa AVRee de Elektor.

enciende, este diodo LED está apagado. El programa de control que corre sobre el ordenador y que controla la unidad enciende este LED

para indicar que la conexión entre el ordenador y la caja SPI se ha establecido de manera correcta. Cuando se procesa un comando, el circuito inte-

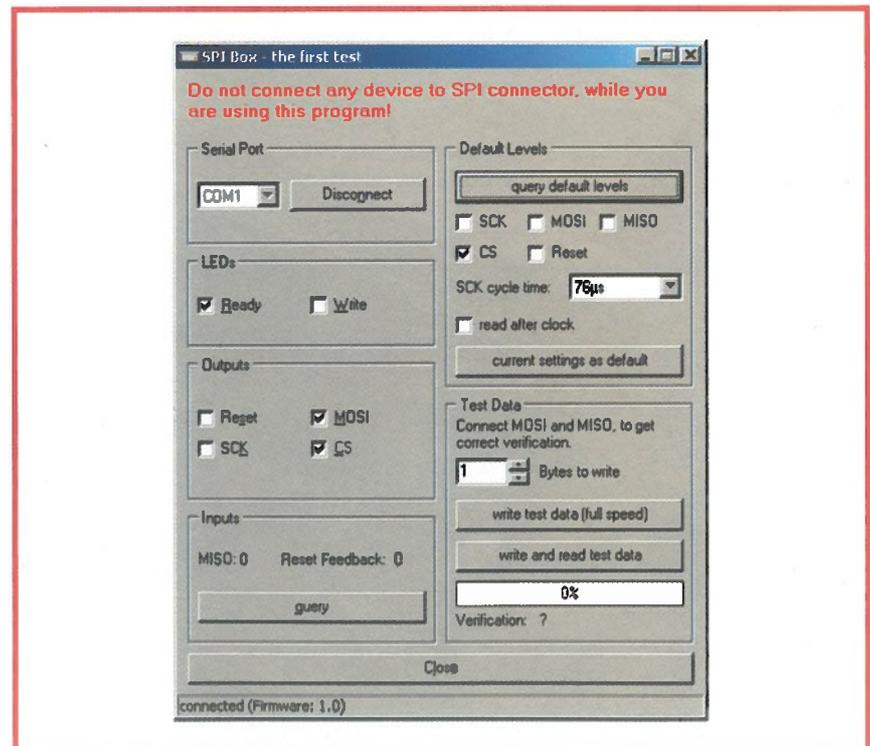


Figura 6. Cuadro de diálogo del programa de prueba.

grado IC5 apaga este diodo LED brevemente. Si el diodo permanece apagado, esto significa que aún está esperando bytes desde el ordenador para completar el comando.

Fuente de alimentación y montaje

La unidad se debe alimentar con una tensión continua (DC) comprendida entre + 9 y + 15 V, proveniente, por ejemplo, de un adaptador de tensión de red comercial de los que existen en el mercado. Por conveniencia, la placa de circuito impreso (véase la **Figura 3**) está equipada con dos conectores de alimentación, de manera que cuando se conecta un circuito externo a la unidad, ambos pueden alimentarse a partir de la misma fuente de alimentación, utilizando el cable adecuado, con dos conectores de baja tensión.

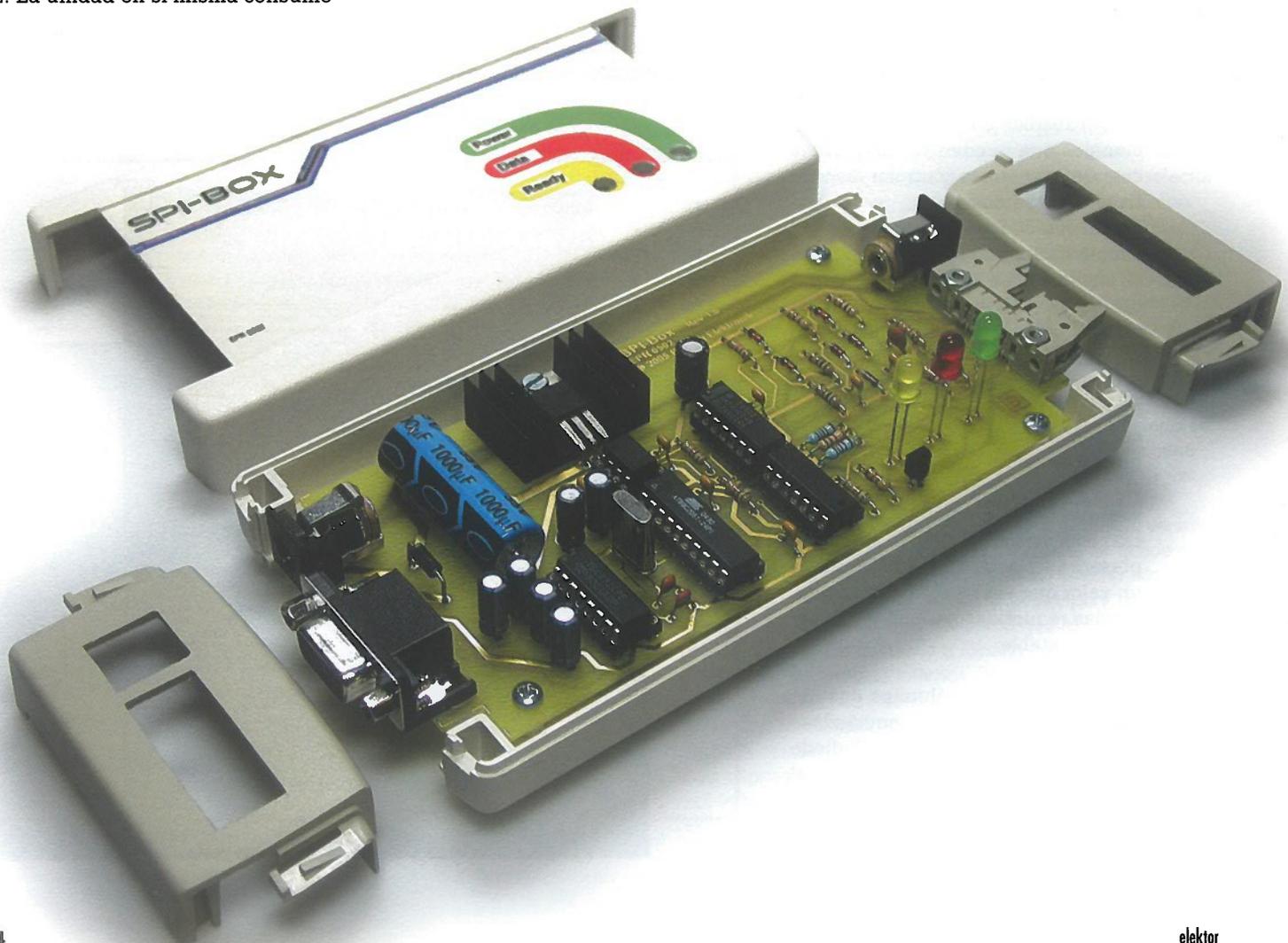
La tensión de alimentación regula el circuito integrado IC2, el cual proporciona una tensión de salida estabilizada de + 5 V. Esta tensión de alimentación también está disponible en el terminal 1 del conector K2. La unidad en sí misma consume

La caja SPI y el programa

Incluso la caja SPI más bonita jamás construida es inútil sin el adecuado programa de control (basado en Windows). Por lo tanto, para que nuestros lectores puedan escribir su propio programa interfaz hemos puesto a su disposición en nuestra página web de Elektor Electronics, además de los programas de descarga, un completo documento en formato PDF que contiene una gran cantidad de información adicional detallada, capturas de pantalla y tablas. La página web del autor de este artículo también incluye más información sobre este proyecto y ofrece la oportunidad de comprar un paquete de programas adicionales. La base de todos los programas generados en nuestra programación es la librería de Windows "SPIBox.dll", la cual se debe de instalar junto con el programa de prueba.

menos de 100 mA. Si se utiliza la tensión de + 5 V para alimentar un circuito externo, se recomienda colocar un radiador al circuito integrado IC2. Por otro lado, el montaje del circuito no tiene demasiadas complicaciones. La placa de circuito impreso es de doble cara, pero sólo se han utilizado componentes con terminales normales, es decir, no es necesario soldar componentes SMD a mano. Para el circuito integrado IC5, en particular, se recomienda utilizar un zócalo, ya que este microcontrolador debe estar progra-

mado antes de que pueda usarse en el circuito (la programación dentro del sistema no es posible). La **Figura 4** nos muestra la placa prototipo con los componentes ya montados, la cual encaja perfectamente en la caja OKW que se muestra en las fotografías y que se especifica en la lista de materiales. Sobre dicha caja se deben realizar las correspondientes aberturas para tener acceso a los distintos conectores (tanto en el panel frontal como en el panel posterior), así como los agujeros necesarios para poder ver el estado de los tres diodos LEDs.



Las conexiones

Se conectará la unidad al ordenador utilizando un cable serie comercial de los que existen en el mercado (con conector Sub-D de nueve terminales, en un extremo conector macho y en el otro extremo conector hembra, con un cableado pin a pin).

El dispositivo que va a programarse a través del bus SPI tiene que conectarse a K2 en la unidad, utilizando un cable con la correspondiente y adecuada conexión. Las señales adicionales sobre el conector K2 simplifican la conexión hacia la ya muy popular Placa Microcontroladora Flash AT 89S8252, publicada en Elektor en Febrero de 2002 [3], ya que las señales en los terminales 1 a 9 del conector K2 tienen una correspondencia directa con los nueve terminales del conector Sub-D de la placa en cuestión. Por supuesto, otros sistemas de desarrollo necesitarán un cable especial para poder trabajar con ellos. En la **Figura 5** se muestra la distribución de terminales requerida para la Placa Microcontroladora Flash y para la Placa AVRee publicada en Elektor en Mayo de 2003 [4].

Verificación y prueba

Cuando tuvimos que probar la unidad nos encontramos con la típica situación de "el huevo y la gallina", ya que no disponíamos del correspondiente sistema de programación para el circuito integrado IC5 o, como mínimo, un sistema de programación sencillo. El microcontrolador debe programarse antes de que podamos utilizar la caja SPI para programar cualquier otro dispositivo. En Internet podemos encontrar muchos diseños para realizar sencillos programadores, pero para evitar el esfuerzo de tener que construir un dispositivo de este tipo y solucionar así el problema, en la tienda de Elektor disponemos de microcontroladores ya programados para nuestros lectores (el código de pedido es **050198-41**). Si deseamos programar el microcontrolador nosotros mismos también tenemos los correspondientes ficheros (en la forma de ficheros binarios y hexadecimales, junto con un fichero "readme" que contiene los bytes de configuración para el dispositivo) dentro de un fichero zip, con la referencia **050198-11**, que nuestros lectores pueden descargar de forma gratuita de la página web de Elektor en [\[nics.co.uk\]\(http://www.elektor-electronics.co.uk\). Una vez que la placa de circuito impreso ha sido equipada y montada con todos los componentes y, además, hemos realizado una inspección visual detallada de la unidad, podemos conectar la unidad al ordenador utilizando el cable serie de extensión. Cuando se aplica la tensión de alimentación se encenderá el diodo LED verde, así como los diodos LEDs rojo y amarillo que lo harán brevemente.](http://www.elektor-electro-</p></div><div data-bbox=)

El siguiente paso será instalar los programas sobre el ordenador. Podemos encontrar el programa "SPI-Box_Setup.exe" en el fichero Zip **050198-11** en la página web de Elektor. Como alternativa, también podemos encontrar el programa en la página web del autor de este artículo [2].

Cuando el programa instalador se ha ejecutado, deberíamos encontrar el programa "SPIBox_Setup.exe" en el directorio "C:\Program Files\SPI-Box". Los ficheros "SPI-Box_Test.ENG" y "SPI-Box.dll" estarán también en el mismo directorio.

La Figura 6 nos muestra la ventana que se abre cuando se ejecuta el programa de prueba. Si se ha seleccionado el puerto serie correcto y se pulsa el botón "Connect" ("Conectar"), se encenderá el diodo LED verde y la línea de estado indicará el número de versión del programa interno. Esto nos muestra que la caja SPI está trabajando de manera correcta. Para verificar el funcionamiento de la parte SPI de la unidad podemos conectar un voltímetro o, mejor, un osciloscopio en los correspondientes terminales del conector K2.

En la misma ventana, dentro de la sección "Default Levels" ("Niveles por Defecto"), es posible seleccionar si las señales SPI SCK, MOSI y MISO son activas a nivel alto o a nivel bajo. Normalmente, la selección correcta es a nivel alto. Si algunas señales individuales están invertidas (como las señales MOSI y MISO en la Placa Microcontroladora Flash), es aquí donde se deben realizar los cambios correspondientes para tener la polaridad lógica adecuada. Las pequeñas cajas de selección de las señales "CS" y "Reset" controlan el estado de las correspondientes señales de salida: las dos señales de reset están controladas al mismo tiempo. La unidad también permite que se pueda seleccionar la velocidad de las ope-

raciones de lectura/escritura en el bus SPI. En el programa de prueba esto se hace utilizando la configuración del "tiempo del ciclo de SCK" (SCK cycle time). Además, también existe una función de escritura pura que funciona a una velocidad considerablemente superior. Para las operaciones de lectura es posible seleccionar cuándo será muestreado el nivel de la señal MISO, durante el pulso de reloj o después de éste. Las configuraciones básicas almacenadas en la unidad pueden ser leídas y modificadas utilizando el programa de verificación. Para conseguir que cualquier cambio se haga efectivo es necesario escribirlo de nuevo en la unidad utilizando el botón de "current settings as default" (es decir, "configuración actual por defecto"). Las configuraciones se mantienen durante todo el tiempo que la tensión de alimentación está aplicada a la unidad.

Cuando se completa la verificación podemos proceder a usar el programa actual de programación, el cual ofrece un rango considerablemente más amplio de posibilidades. La página web de Elektor y la página web del autor de este artículo ofrecen una gran cantidad de información adicional y de consejos sobre este tema, vea el apartado de texto "La caja SPI y el programa".

(050198-1)

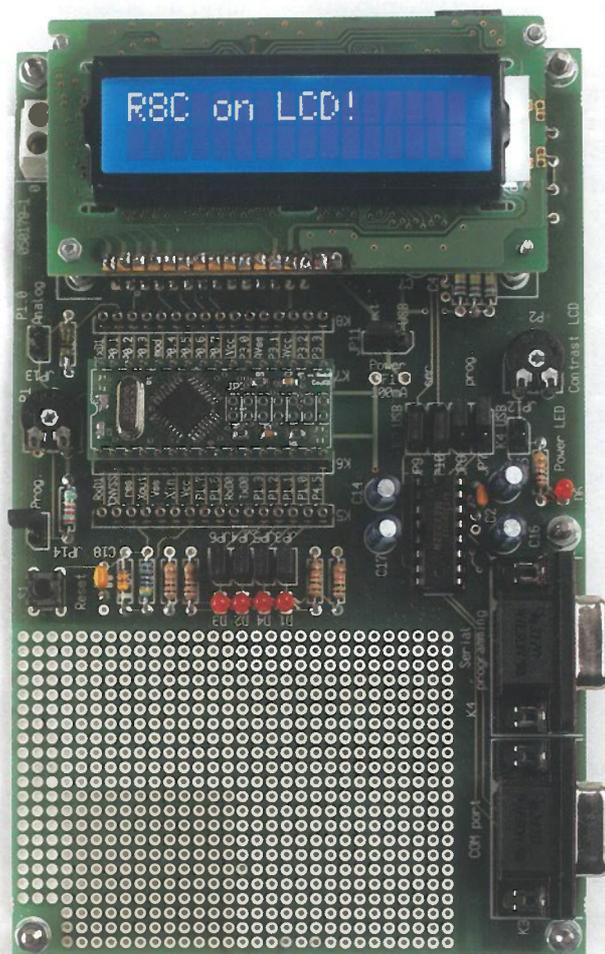
Información Adicional:

[1] http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[2] <http://reweb.fh-weingarten.de/elektor/en/projects/spibox/index.html>

Funciones de Con

Uso del circuito R8C/13 en nuestra



En la entrega del anterior número ya introdujimos la Placa de Aplicación R8C/13 de la casa Renesas. En esta entrega de la serie de artículos R8C vamos a describir cómo utilizar dicha placa con nuestras propias aplicaciones. Así, la pantalla LCD, el conversor A/D y la interfaz serie, son los elementos claves en este artículo.

Burkhard Kainka

La descripción sobre cómo utilizar el compilador y el depurador de esta placa está disponible en una sección dedicada al microcontrolador R8C, en nuestra página web de Elektor (www.elektor-electronics.co.uk). Además, en el foro encontrará comentarios adicionales enviados por nuestros lectores y que se han "publicado" en el apartado "R8C Tom Thumb", así como una buena selección de avisos y trucos relacionados con todos los posibles problemas que puede encontrar en la práctica. Por primera vez un número significativo de lectores han comenzado a utilizar un kit de un microcontrolador más o menos al mismo tiempo. Naturalmente, aplicando la famosa "ley de Murphy", los problemas han aparecido bastante rápidamente, sin embargo, lo bueno es que cuando un nuevo problema aparece, existe una gran probabilidad de que alguien haya encontrado ya una solución. Por lo tanto, es muy recomendable ojear el foro sobre el R8C/13, en el que también puede descubrir sencillos proyectos desarrollados por lectores para lectores, y de los cuales podremos

aprender bastantes cosas. Algunos de ellos tienen un nivel de sofisticación técnica que va bastante más allá de los límites de este artículo.

Este artículo pretende ser intencionalmente sencillo, para asegurar que incluso los lectores con relativa poca experiencia puedan utilizar la placa del microcontrolador sin demasiados problemas. Esperamos que nuestros lectores disculpen las "violaciones" ocasionales realizadas sobre el buen estilo de programación en lenguaje "C". Por ejemplo, todas las funciones descritas en este artículo están contenidas en un único listado de código fuente en C. La función "main" incluye una serie de lazos sin fin en los que cada uno realiza una función completamente diferente. En cada caso solamente se ejecutará el primer lazo habilitado de la serie. La ventaja de todo esto es que nos permite un amplio abanico de posibilidades con un único proyecto. Todo lo que tenemos que hacer cuando queremos intentar algo nuevo es insertar o borrar unos pocos caracteres de comentarios. El fichero del proyecto (lcd.zip) puede descargarse de la página web de Elektor y contiene todas las funciones que se van

Control para el R8C

aplicaciones

a ver en esta entrega.

El control de la pantalla LCD

Este módulo utiliza seis bits del puerto 0 de la placa, ya que trabaja con el modo de interfaz de cuatro bits que se usa habitualmente, lo que también significa que sólo son necesarias las cuatro líneas de datos superiores D4 a D7 (ver Figura 1). Las cuatro líneas de menor peso (D0 a D3) normalmente se dejan abiertas, sin embargo, esto puede producir algunos problemas con ciertos módulos visualizadores, por lo que en la placa de nuestra aplicación han sido llevadas a masa. Asimismo, el módulo necesita señales en las líneas "E" y "RS". La línea de control (RS) selecciona también el modo de trabajo de datos o el modo de trabajo de comandos. Las dos líneas restantes del puerto PO sirven para otros propósitos que deberán tenerse en cuenta cuando estemos programando nuestras propias aplicaciones. La línea PO.1 también ha sido designada como AN6 y se usa como una entrada analógica para la realización de medidas. La línea PO.0 se utiliza como la salida TXD1 y la emplea la interfaz de depuración.

Para poder controlar la pantalla de visualización, lo primero que tendremos que hacer es configurar las líneas PO.2 a PO.7 como salidas. Sin embargo, a diferencia de lo que sucede con el puerto P1, este proceso requiere algo más que escribir tan sólo la sentencia `pd0 = 225` en la línea de código correspondiente, ya que el acceso al registro de dirección de datos para el puerto PO tiene una protección especial. Para retirar esta protección se debe introducir la instrucción `prc2 = 1` inmediatamente antes de cada acceso:

```
prc2 = 1;           // Permite el acceso al puerto pd0
pd0 = 0xfd;        // Puerto LCD, bit 1 dejado como
// entrada del conversor A/D
```

La dirección de los datos es conmutada por medio de bits 1111101 = 0xfd. Esto significa que un bit a 1 permanece como una entrada y un bit a 0 es una salida de la línea TXD1. Cuando utilizamos el programa depurador KD30, el microcontrolador transmite datos hacia el ordenador a través de esta línea del puerto. Sin embargo, podemos ignorar este bit en las salidas subsiguientes ya que la inicialización de la interfaz serie aísla a este bit de cualquier otro acceso directo al puerto. Esto significa que una instrucción de salida como `po = 0`, dejará las líneas PO.0 y PO.1 sin ningún cambio. El siguiente listado del programa muestra la función clave para enviar un byte, la cual se denomina "lcddata". Un valor "data" del tipo carácter sin signo (lo que significa un byte con un rango de valor de 0 a 255) es el que se pasa a la fun-

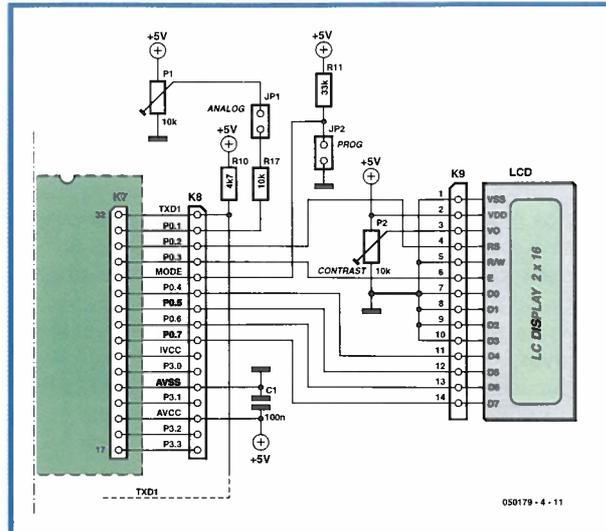


Figura 1. Conexión de un módulo LCD al puerto PO.

ción, donde "data" es un nombre de variable arbitrario que puede ser intercambiable. Esta función no nos devuelve ningún valor (es decir, está vacía), lo que significa que el flujo de datos se produce solamente en una dirección. Esto es similar a un procedimiento en el lenguaje Delphi o una subrutina (sub) en Visual Basic. El lenguaje C sólo tiene funciones, de manera que los procedimientos solamente son funciones que no devuelven ningún valor. Si queremos volver a llamar a la función más tarde en el programa, tan sólo tendremos que escribir el nombre de la función con el valor deseado entre paréntesis, como puede ser 'lcddata (65)'. Esto explica también porque los bytes en lenguaje C son referidos como caracteres (char), y un número puede representar un carácter así como un número ordinario.

```
static unsigned char Port0;
```

```
void lcddata(unsigned char data)
{
    delayus(100);
    Port0 = data & 0xF0;
    Port0 = Port0 + 0x04; //RS = 1
    p0 = Port0;
    p0 = Port0 + 0x08; //E = 1
    asm("nop");
    p0 = Port0; //E = 0
    Port0 = ((data & 0x0f) << 4);
    Port0 = Port0 + 0x04; //RS = 1
    p0 = Port0;
    p0 = Port0 + 0x08; //E = 1
    asm("nop");
    p0 = Port0; //E = 0
    delayus (100);
}
```

Dentro de la función 'lcddata', el contenido de la variable es procesado y sacado a través de los cuatro bits superiores del puerto 0 en dos pasos. El estado

actual se almacena siempre en la variable Port0 de tipo carácter sin signo estática. La designación "estática" significa que varias funciones pueden acceder a la variable. El primer paso es sacar los cuatro bits superiores.

El símbolo "&" representa la operación lógica 'AND'. Por lo tanto, la expresión 'data & 0xF0' nos proporciona los cuatro bits superiores, con los cuatro bits inferiores colocados a 0. En el segundo paso, el dato es desplazado a la izquierda en cuatro bits para poder conseguir la salida de los cuatro bits de menor peso sobre el puerto. En la línea E se genera un corto pulso después de la obtención de cada bloque para sincronizar el dato con el módulo LCD. Una vez que los dos bloques de cuatro bits han sido transferidos de este modo, el módulo visualizador mostrará el carácter correspondiente. Antes de enviar la siguiente salida se recomienda utilizar una breve pausa. Más adelante se describe con bastante detalle la función 'delayus'.

Por supuesto, el módulo LCD debe estar inicializado de manera adecuada antes de que cualquier carácter pueda presentarse en pantalla. Una vez que se ha aplicado la tensión de alimentación por primera vez al módulo, la fila superior está en oscuro y la fila inferior está iluminada. En este momento se deben enviar una serie de bytes de comandos al módulo para su inicialización. La función 'lcdctrl' es similar a la función 'lcddata', con la única diferencia de que la línea RS permanece a nivel bajo en este caso. De este modo, el módulo visualizador interpretará las señales en las líneas de datos como bytes de comandos.

```
void lcdctrl(unsigned char data)
{
    Port0 = data & 0xF0;
    Port0 = Port0 + 0x01;
    p0 = Port0;
    p0 = Port0 + 0x08;    //E = 1
    asm("nop");
    p0 = Port0;          //E = 0
    Port0 = ((data & 0x0f)<<4);
    Port0 = Port0 + 0x01;
    p0 = Port0;
    p0 = Port0 + 0x08;    //E = 1
    asm("nop");
    p0 = Port0;          //E = 0
    delayus (100);
}
```

Durante la fase de inicialización se transfieren varios bytes de comandos. A lo largo de este proceso es esencial respetar los retardos especificados en la hoja de datos del módulo visualizador. Para este propósito hemos escrito una sencilla función de retardo con una resolución no muy fina de alrededor de un microsegundo. Podemos generar un retardo que va desde 1 μ s hasta los 65.000 μ s, pasando un valor de 16 bits sin un signo positivo o negativo.

```
void delayus (unsigned int micros)
{
    unsigned int i;
    for (i = 0; i < micros; i++)
    {
        asm("nop");
    }
}
```

Si nuestros lectores desean estudiar la función retardo con detalle, pueden insertar las pocas líneas de código que se muestran a continuación en la función 'main'. Con esta modificación podremos ver señales de onda cuadrada en todas las líneas del puerto P1. Con un tiempo de retardo de 1.000 μ s, cada pulso debe tener una duración exacta de 1 mseg. Si utilizamos un osciloscopio para realizar una medida podremos ver que la función de retardo es en ese momento alrededor de un 10% más lenta. Esto significa que no debemos utilizar dicha función para la programación de cualquier tipo de función de reloj, aunque su precisión es totalmente adecuada para el propósito, que ha sido diseñada.

```
pd1=255;
while (1)
{
    p1=255;
    delayus(1000);
    p1=0;
    delayus(1000);
}
```

La fase de inicialización que estamos realizando para el módulo LCD está formada por varios comandos de acceso combinados con retardos cortos y largos. Si nuestros lectores desean conocer los detalles le recomendamos que consulten las hojas de características del módulo LCD o las distintas aplicaciones sencillas que ya han sido descritas para otros microcontroladores. Lo único importante aquí es saber que el módulo LCD se ha inicializado para trabajar en modo de cuatro bits con dos filas de caracteres. Además, cualquier carácter que haya sido enviado previamente al módulo LCD se borrará.

```
void initlcd (void)
{
    delayus(15000);
    lcdctrl(0x28);
    delayus(5000);
    lcdctrl(0x28);
    delayus(1000);
    lcdctrl(0x28);
    delayus(1000);
    lcdctrl(0x0c);
    delayus(1000);
    lcdctrl(0x01);
    delayus(5000);
}
```

Por último podemos verificar la operación de salida actual. Por ejemplo, podemos escribir las siguientes líneas de código en la función 'main'.

```
initlcd();
lcddata (65);
lcddata (66);
lcddata (67);
lcddata (68);
lcddata (69);
lcddata (70);
```

Los caracteres ASCII correspondientes a los números desde 65 a 70, los cuales se corresponden con las letras mayúsculas "ABCDEF", deberán aparecer en la pantalla del visualizador. Podemos escribir cualquier texto que deseemos de esta manera tan sencilla. Un método mucho más conveniente es utilizar la función 'lcd_text', que es capaz de procesar cualquier cadena de caracteres completa. Por ejemplo, sería tan sencillo como escribir

"lcd_text ("Hola mundo")". El último carácter de una cadena de caracteres debe estar siempre seguido por un 0, el cual se emplea para indicar que todos los caracteres han sido transferidos. La marca de exclamación (!) se utiliza para indicar que va a hacerse una operación lógica NOT. Así, el lazo será atendido tantas veces como caracteres hayan sido leídos y que NO sean un cero.

```
void lcd_text (char text[20])
{
    unsigned int i;
    i = 0;
    while (!(text[i] == 0) & (i < 20))
    {
        lcddata (text[i]);
        i = i + 1;
    }
}
```

Sin duda, los valores numéricos son incluso más interesantes que las cadenas de caracteres. Por esta razón hemos escrito una pequeña función denominada 'lcd_integer', que nos permite enviar números enteros en el rango comprendido entre 0000 a 9999. Esto no cubre el rango completo de un valor entero, pero la función está pensada para usarse con valores medidos de 10 bits, provenientes del conversor A/D, los cuales pueden tener un rango de 0 a 1023.

```
void lcd_integer (unsigned int data)
{
    unsigned char byt;
    byt = data / 1000;
    data = data - byt * 1000;
    lcddata(byt + 48);
    byt = data / 100;
    data = data - byt * 100;
    lcddata(byt + 48);
    byt = data / 10;
    data = data - byt * 10;
    lcddata(byt + 48);
    lcddata(data + 48);
}
```

Esta función se explica casi por sí misma. La variable de tipo carácter 'byt' almacena el valor de cada dígito que va llegando. Como los códigos ASCII que van del 48 al 57 representan los dígitos del 0 al 9, se debe añadir la constante 48 al contenido de la variable 'byt' para obtener el código ASCII adecuado. La operación de división de entero sin resto trabaja con un único dígito cada vez. Por ejemplo, la división 9.654 / 1.000 da como resultado el valor de 9; si en este momento restamos 9.000 del valor 9.654, el resto es 654. A continuación, podemos extraer el siguiente dígito y así sucesivamente hasta el final.

Si se hace la llamada a esta función con un número, por ejemplo 'lcd_integer (5678)', la función escribirá el número (en este caso 5678) en el visualizador. Pero, ¿se colocará este dato en el visualizador? El módulo LCD tiene un puntero de direcciones automático que se desplaza en una posición después de cada salida. Si está inicializado a la posición cero en la línea superior la línea uno ocupa las direcciones de 0 a 19 dentro del módulo LCD, mientras que la línea dos ocupa las direcciones que van desde el valor 64 al 83. Si queremos escribir un carácter en una posición particular, tendremos que enviar el comando 0x80 y la dirección correspon-

diente dentro del módulo. Además, se necesita utilizar de nuevo un pequeño retardo después de esta operación.

```
void lcd_pos (unsigned int Row, unsigned int Column)
{
    lcdctrl (0x80 + Column-1 + 0x40*(Column-1));
    delayus(100);
}
```

La función 'lcd_pos' coloca el puntero en la posición especificada por la variable "Row" (1 ó 2) y la variable "Column" (1 a 20). Ahora podemos sacar sencillamente cualquier dígito en la posición que se desee:

```
lcd_pos (2,4);
lcd_integer(1234);
```

El conversor A/D

¿Qué tenemos que hacer cuando queremos controlar un nuevo microcontrolador y no estamos familiarizado con él? Naturalmente, lo primero que se nos ocurre es buscar ejemplos de aplicación ya realizados en las hojas de características. En este caso, tenemos un ejemplo de aplicación disponible en el CD-ROM que se suministra para el módulo R8C, en el directorio "AD Converter" bajo el directorio "Application Notes R8C". Con unas pocas modificaciones y simplificaciones menores, la función que se presenta en este ejemplo puede usarse para el conversor A/D interno del R8C/13. Existen dos grupos de entradas analógicas, las cuales están localizadas en las líneas del puerto 0 (de AN0 a AN7) y del puerto 1 (de AN8 a AN11). En nuestro caso sólo utilizamos el grupo de nivel más bajo.

```
unsigned int ad_in(unsigned char ch)
{
    adcon0 = 0x80 + ch; //Grupo Port P0
    adcon1 = 0x28; //Modo de 10-bit
    adst = 1; //Inicio de la conversión
    while(adst == 1){} //Espera para hacer una conversión A/D
    return ad; //Valor A/D
}
```

Tenemos que realizar ciertas configuraciones en los registros de control 'adcon0' y 'adcon1' antes de que podamos utilizar el conversor. Después de esto, podemos empezar a realizar una conversión, esperar hasta que se haya completado y, por último, leer el resultado en el registro 'ad'. Se llama a la función 'ad_in' utilizando el canal deseado (de 0 a 7) como un parámetro. Debemos señalar que las entradas analógicas están numeradas en el orden opuesto a las líneas del puerto PO_0 a PO_7. Esto significa que la entrada analógica AN6 está localizado en la línea del puerto PO_1.

Después de haber conseguido realizar una conversión de una medida, la función devuelve un resultado de tipo entero sin signo, con el valor de la medida comprendido dentro del rango de 0 a 1023.

```
while(1)
{
    lcd_pos (2,4);
    lcd_integer(ad_in(6));
    delayus(50000);
}
```


medida como un valor real en unidades de voltios. Esto también nos sirve para ilustrar que incluso los cálculos verdaderamente más complejos pueden realizarse fácilmente con el módulo R8C. El programa compilador enlaza automáticamente todo con las librerías necesarias, que incrementan en alguna medida el tamaño del código del fichero. Esto no es ningún problema, ya que disponemos de bastante espacio reservado.

```
void voltmeter()
{
  float u;
  unsigned char c;
  u = (float) ad_in(6);
  u = u / 1023.0 * 5.0;
  c = u;
  lcddata (c+48);
  sendTxd0 (c+48);
  lcddata (0x2e);
  sendTxd0 (0x2e);
  u = u - c;
  u = u * 10;
  c = u;
  lcddata (c+48);
  sendTxd0 (c+48);
  u = u - c;
  u = u * 10;
  c = u;
  lcddata (c+48);
  sendTxd0 (c+48);
  u = u - c;
  u = u * 10;
  c = u;
  lcddata (c+48);
  lcddata (0x20);
  lcddata (0x56);
  sendTxd0 (c+48);
}
```

```
sendTxd0 (0x20);
sendTxd0 (0x56);
sendTxd0 (0x0d);
}
```

La función 'voltmeter' calcula primero el valor medido en voltios y, a continuación, separa los dígitos individuales para su salida. El punto decimal se inserta como el carácter especial '0x2e'. Cada dígito individual se escribe en la pantalla LCD y se transmite a través de la UART0. Al final de la transmisión se añade un carácter de espacio y la unidad de medida ('V'). Además, se añade un carácter CR (retorno de carro) para la salida de la interfaz serie, de manera que se inicie una nueva línea en el programa emulador de terminal. Por otro lado, para la pantalla LCD la posición del primer carácter se especifica de nuevo antes de que se haga llamada a la función de medida, de manera que cada nueva medida sobrescriba la anterior.

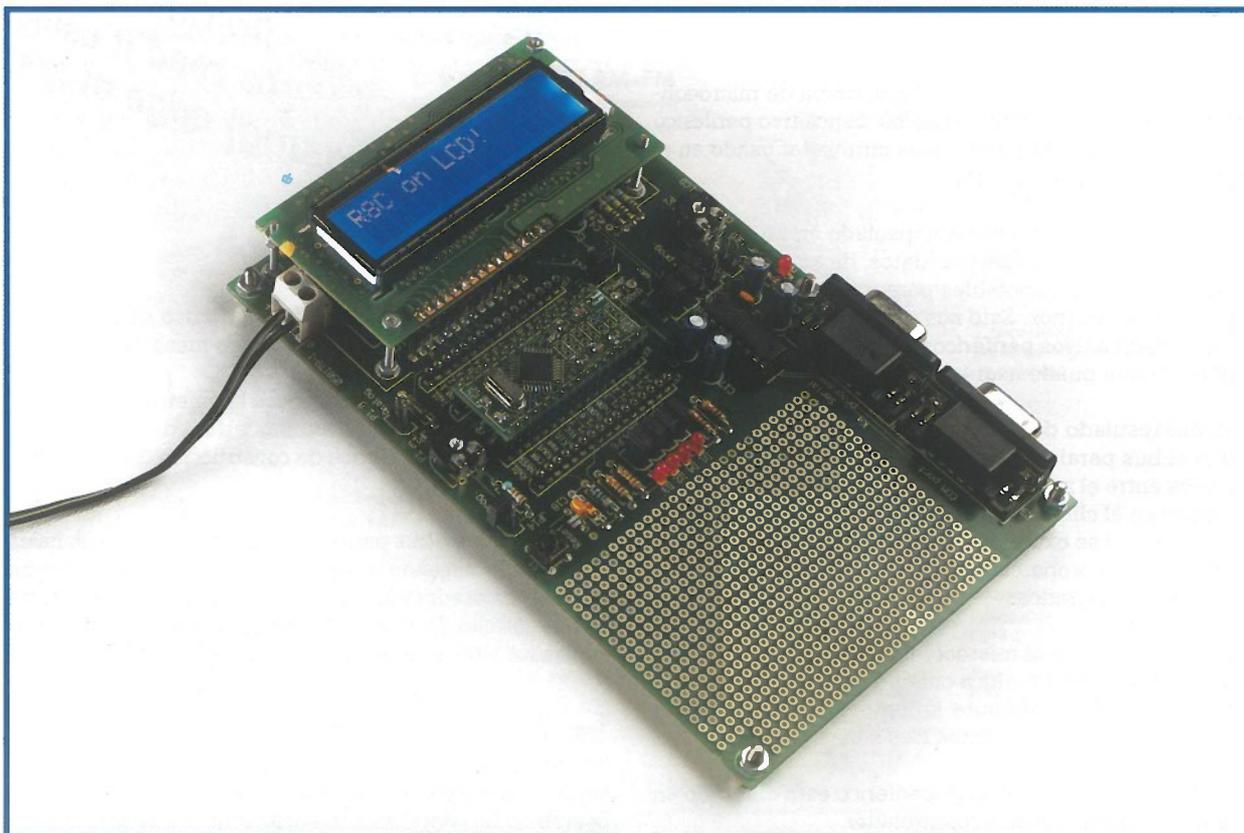
```
while (1)
{
  lcd_pos (1,10);
  voltmeter();
  delayus (50000);
}
```

El resultado se muestra en el programa emulador de terminal con los siguientes valores:

```
4.628 V
4.633 V
4.633 V
...
```

¡Estupendo! ¡Todo funciona!

(050179-4)



Comunicaciones

Transferencia de datos con reloj

La transferencia de datos serie síncrona entre un microcontrolador y dispositivos periféricos ha estado de moda en los últimos años, especialmente porque la velocidad de datos que se ha alcanzado es de hasta 10 MBit/s.

La mayoría de las grandes compañías de semiconductores como Analog Devices, Atmel, Burr-Brown, Maxim/Dallas, Microchip, entre muchas otras, disponen de un amplio rango de dispositivos periféricos con interfaces serie de alta velocidad realizados para comunicación para corto rango a un microcontrolador. No hay duda de que esta técnica de interface tiene muchas ventajas sobre la interface paralelo tradicional. En este artículo veremos los principios básicos y ofreceremos algunos comentarios útiles con consideraciones de este tipo de interface para nuestro siguiente diseño.

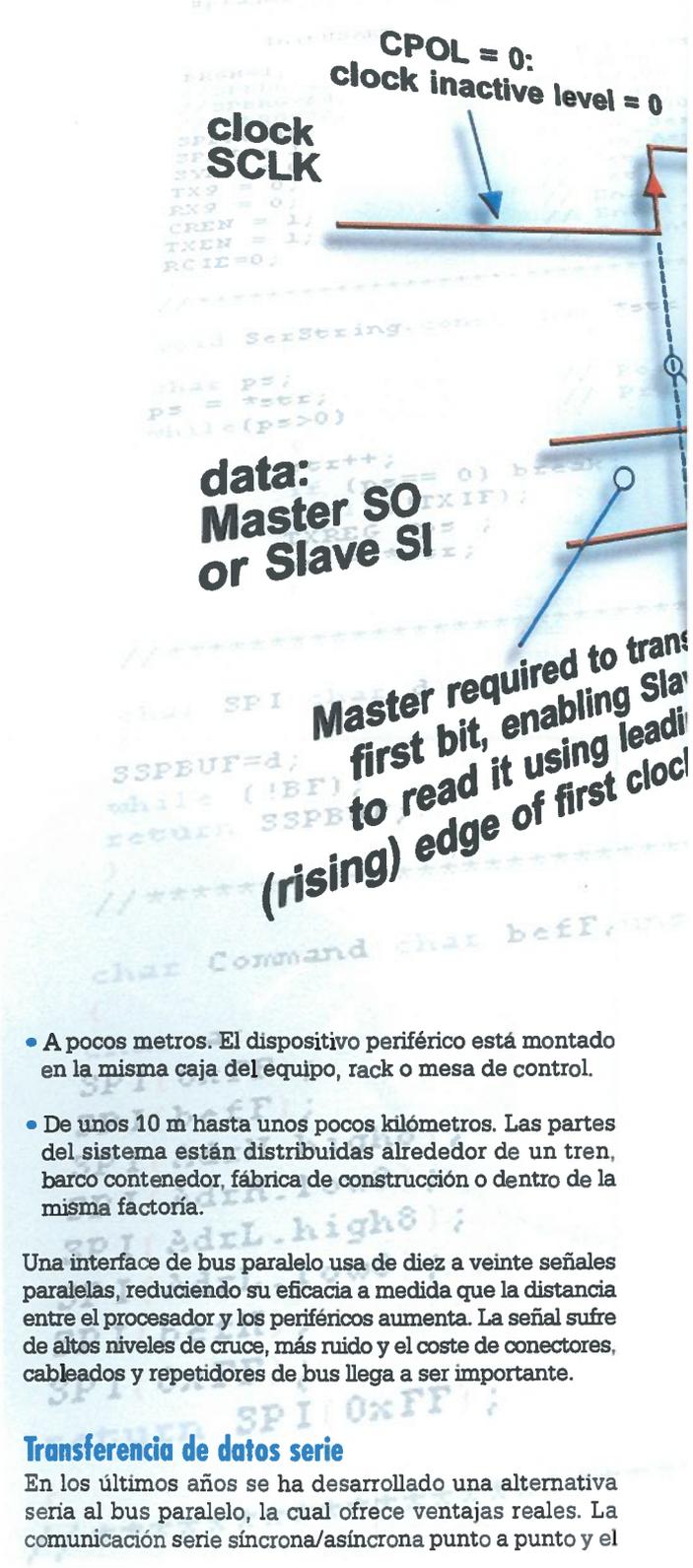
El problema básico

Cuando es necesario extender un sistema de microcontrolador existente, la elección de un dispositivo periférico depende mucho del tipo de microcontrolador usado en el sistema:

- El microcontrolador está encapsulado en un gran chip que lleva el bus del sistema (datos, dirección y control) con las señales disponibles para usar por dispositivos periféricos externos. Esto nos permite un amplio espectro de dispositivos periféricos con una interface de bus paralelo que puede usarse con el controlador.
- El encapsulado del microcontrolador es tan pequeño que el bus paralelo sólo puede darse a escala microscópica entre el microprocesador y los dispositivos integrados en el chip; no hay pines suficientes para que el bus paralelo se extienda a los pines del chip. Los pines se usan para proporcionar conexiones I/O para los dispositivos integrados.

La distancia entre el microcontrolador y el dispositivo periférico también es crítica cuando consideramos el bus del sistema. Generalmente la longitud del bus estará comprendida en una de estas tres zonas:

- De 1 a 10 cm. El dispositivo periférico está montado en la misma placa del microcontrolador.



- A pocos metros. El dispositivo periférico está montado en la misma caja del equipo, rack o mesa de control.
- De unos 10 m hasta unos pocos kilómetros. Las partes del sistema están distribuidas alrededor de un tren, barco contenedor, fábrica de construcción o dentro de la misma factoría.

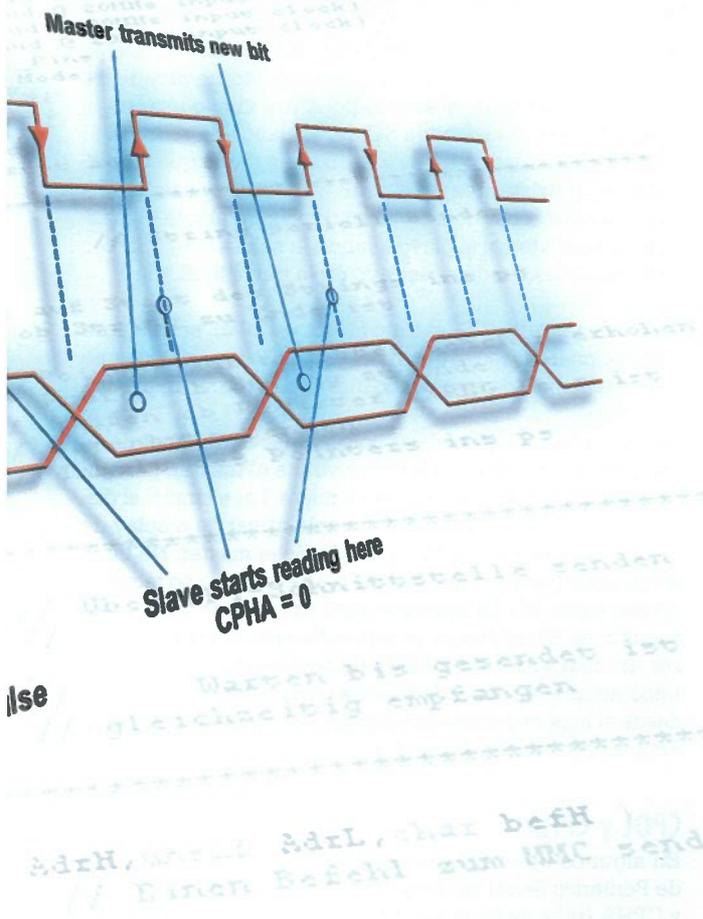
Una interface de bus paralelo usa de diez a veinte señales paralelas, reduciendo su eficacia a medida que la distancia entre el procesador y los periféricos aumenta. La señal sufre de altos niveles de cruce, más ruido y el coste de conectores, cableados y repetidores de bus llega a ser importante.

Transferencia de datos serie

En los últimos años se ha desarrollado una alternativa seria al bus paralelo, la cual ofrece ventajas reales. La comunicación serie síncrona/asíncrona punto a punto y el

de Datos Serie

Prof. Dr. Bernd vom Berg
y Peter Groppe



La velocidad de mejora de la interface serie estándar convierte esto en un problema menor que se compensa por las numerosas ventajas que ofrece el sistema serie. Un moderno disco duro con la interface serie ATA puede transferir datos más rápidamente que una vieja interface paralelo de 16 bits.

Bits y PCs

Un concepto importante en la transferencia de datos serie síncronos entre un transmisor y receptor es la sincronización (Figura 1). El transmisor pone cada bit en la línea de datos serie uno tras otro a una velocidad predefinida. ¿Cómo sabe el receptor cuándo debe leer cada bit? Es vital no perder un simple bit o leer el mismo bit dos veces e interpretarlo como dos bits.

El receptor requiere información que le diga cuándo un bit válido está en la entrada de datos serie y se puede leer. La solución más sencilla para el transmisor es proporcionar una señal indicadora adicional cuando el dato es válido y se puede leer. Esta señal adicional es el reloj de datos enviado en una línea separada desde el transmisor al receptor.

El receptor consulta la línea de reloj y su procesamiento, traducido a nuestra forma de pensar, podría ser algo así:

Cuando se produce un flanco de subida en la señal de reloj es porque hay un bit válido en la entrada de datos serie y entonces puedo leerlo para saber si es un 1 o un 0 y almacenar su valor. Cuando hay un flanco de caída en la línea no hago nada porque el transmisor usa este flanco para sacar un nuevo bit.

bus de sistema síncrono/asíncrono. La transferencia de datos entre dispositivos no sucede en paralelo donde un byte o varios bytes de información se deberían transferir por una simple operación en lugar de dividir los bytes en bits individuales y enviarlos en serie un bit cada vez. La interface serie reduce drásticamente el cableado entre periféricos y también permite que ampliar la interface de forma bastante económica con sólo añadir sencillos desplazadores de nivel o drivers de línea entre dispositivos. Por último, la interface es relativamente sencilla para implementarla en hardware con un microcontrolador con dos o cuatro pines de reserva de I/O digitales (y con la ayuda de un poco de software), de manera que pueden usarse como un puerto de una interface serie. Una desventaja de este tipo de interfaces es la velocidad de transferencia de datos. Los 8 bits que se transfieren en una interface paralela suceden con sólo un flanco de reloj, pero en una interface serie, el byte se envía de bit en bit, lo que requiere de 8 ciclos de reloj. Sin embargo, la veloci-

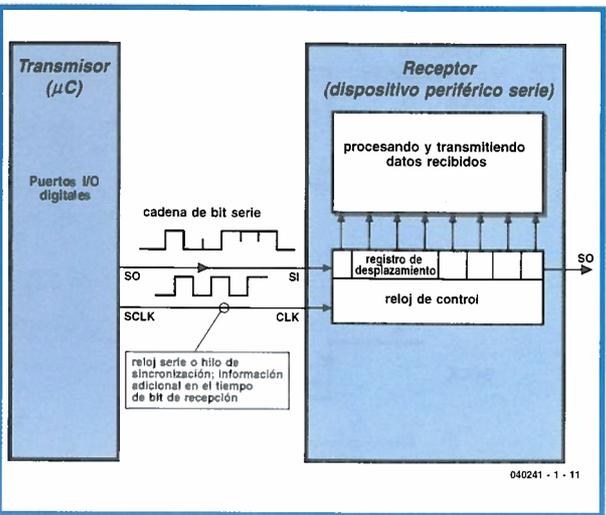


Figura 1. Sincronización de transferencia de datos entre el transmisor y el receptor.

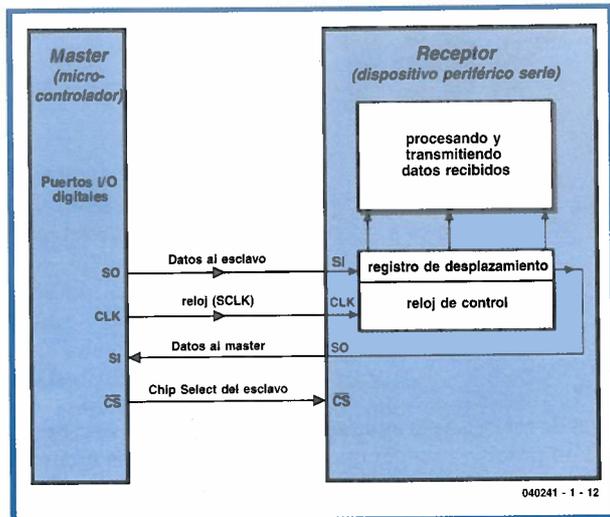


Figura 2. Transferencia de datos síncrona serie.

Los bits recibidos son ensamblados uno a uno en un registro de desplazamiento o posición de memoria hasta que se han recibido todos y entonces el esclavo interpretará y actuará en el dato. Para implementar la interface síncrona serie es necesario tener al menos una señal de datos y una señal de reloj (más una de retorno de tierra). Una técnica alternativa de transferencia de datos serie es el método asíncrono. Este sistema no requiere una señal de reloj. Esta interface se usa en el sistema de bus CAN, pero es más usual en los puertos serie COM de los ordenadores, que típicamente utilizan un chip UART (Transmisor/Receptor Asíncrono Universal) para pasar los datos a formato serie desde bytes. Este artículo, sin embargo, hace referencia a sistemas síncronos.

Transferencia de datos serie síncrona

La Figura 2 nos muestra una configuración típica punto a punto de transferencia de datos serie síncrona entre un dispositivo maestro -master- y otro esclavo -slave-. Cada estación puede enviar datos hacia y desde el otro usando las dos líneas de datos separada, pero sólo una estación proporciona señal de reloj cada vez que el dato viaja en ambas direcciones. Esta estación se conoce como master y normalmente contiene un microcontrolador. La velocidad del reloj determina la velocidad de transferencia de datos serie (Bit rate) y se elige para que no sea más rápida que la velo-

cidad máxima permitida por el master o el esclavo. Describiremos la secuencia de transferencia de datos:

- La señal de reloj (S) CLK (Serial Clock) se produce en el master. Esta señal de reloj la toma el registro de desplazamiento del esclavo.
- El master envía datos serie desde el pin de la SO (Salida Serie) el cual está conectado al pin SI (Entrada Serie) del dispositivo esclavo. Aquí se almacena en el registro de desplazamiento usando el reloj serie. Una vez que todos los bits se han desplazado en el registro, el dispositivo esclavo lee la información, que pueden ser palabras de datos o comandos.
- El esclavo puede enviar datos al master si éste lo requiere. El esclavo carga el byte de retorno al registro de desplazamiento y éste lo pone (de nuevo usando el reloj master) de la salida SO del esclavo a la entrada SI del master. Cuando el esclavo no tiene datos que enviar, los datos existentes en el registro son desplazados. Es un canal de comunicación full-duplex en el que el master envía datos al esclavo usando la salida SO y a la vez recibe datos desde el esclavo en la entrada SI.
- La cuarta señal, \overline{CS} (el Chip Select esclavo), se usa para seleccionar un esclavo particular cuando hay más de uno en el sistema. Lo veremos con más detalle en otro momento.

La salida SO en el esclavo sólo es necesaria cuando el esclavo necesita enviar datos de vuelta al master, pero en algunas aplicaciones esto no se requiere. Por ejemplo, el dispositivo esclavo puede ser un simple driver de display de LED y sólo necesita información desde el master. No tiene información para enviar y en este caso el esclavo no requiere de una salida SO. La interface serie síncrona consta de tres conexiones. Sin embargo, en otra aplicación el esclavo puede ser un dispositivo de medida de temperatura controlado remotamente en serie, de manera que reciba los comandos desde el master y envíe de vuelta la lectura de temperatura. Esta configuración requerirá de cuatro conexiones.

CPOL y CPHA

En algunos interfaces serie estándar (como la Interface de Periférico Serie) las etiquetas CPOL (polaridad de reloj) y CPHA (reloj de fase) se usan para definir qué flancos de reloj son importantes para la lectura y escritura de datos serie. La interface de chips de este tipo de bus generalmente tiene dos entradas digitales CPOL y CPHA (que también pueden ser bits en un registro de configuración) que permiten cambiar el formato de los chips de la interface serie. Esta flexibilidad asegura que el periférico pueda configurarse para interactuar con un formato de bus existente. CPOL y CPHA pueden ser 1 ó 0, lo que nos da un total de cuatro posibles formatos de interface. Antes de que tenga lugar la transferencia de datos, la señal de reloj SCLK puede estar en uno de los posibles estados de reposo definidos por el parámetro CPOL -Polaridad de reloj- (Clock Polarity):

- CPOL = 0: En estado de reposo (inactivo) la señal de reloj es baja (0 V) así que al comienzo de la transferencia de datos serie síncrona comienza con un flanco de reloj de subida.
- CPOL = 1: El estado de reposo de la señal de reloj es alto (+5 V) así que el inicio de la transferencia de datos serie síncrona comienza con un flanco de reloj de caída.

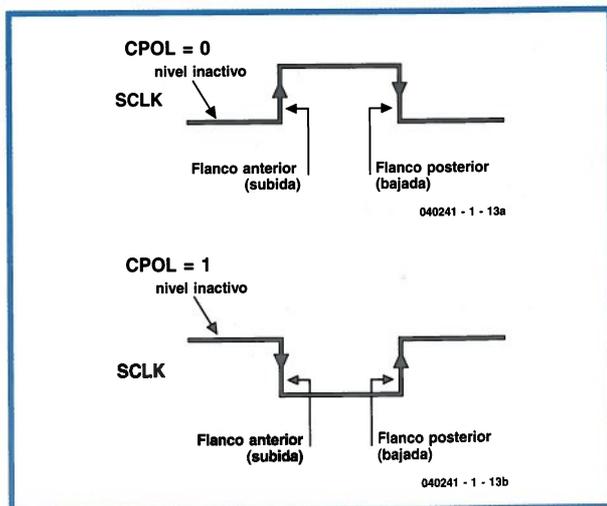


Figura 3. Flancos de subida y bajada del pulso de reloj.

Tabla 1. Los cuatro modos de transferencia de datos serie síncrona del master al esclavo.

CPOL	CPHA	Modo síncrono (también conocido como Modo SPI (Serial Peripheral Interface))
0	0	Modo 0 El esclavo lee datos en el flanco de subida (anterior) de la señal de reloj. La salida de datos del master en el flanco de bajada (posterior) del pulso de reloj anterior.
0	1	Modo 1 El esclavo lee los datos en el flanco de bajada (posterior) de la señal de reloj. La salida de datos del master en el flanco de subida (anterior) del pulso de reloj anterior.
1	0	Modo 2 El esclavo lee datos en el flanco de bajada. La salida de datos del master en el flanco de subida del pulso de reloj anterior.
1	1	Modo 3 El esclavo lee los datos en el flanco de subida de la señal de reloj. La salida de datos del master en el flanco de caída del pulso de reloj actual.

En un sistema síncrono, cuando vamos a transferir datos, la señal de reloj de la interface tiene dos características importantes: su flanco de subida y su flanco de bajada; esos dos eventos se usan para mover datos y leer el valor de ese dato. En todos los sistemas síncronos es importante encontrar la relación entre la cadena del reloj y los datos, de manera que sepamos cuándo pueden salir los datos y cuándo se pueden leer. En un enlace serie entre un dispositivo master y un esclavo enviamos datos desde su salida serie (SO) a su entrada serie (SI) en el esclavo. La comunicación exitosa sólo ocurrirá cuando sepamos:

- Qué flanco de reloj debería usar el esclavo para leer datos en su pin SI.
- Qué flanco de reloj debería usar el master para poner cada nuevo bit en su pin SO que va a leer el esclavo en el siguiente flanco de reloj.

De nuevo hay dos posibilidades gobernadas por el estado del segundo parámetro CPHA.

Esto también es importante para determinar la definición de la señal CPOL, porque determina cómo son el flanco anterior -leading- y el flanco posterior -trailing- de la señal de reloj (Figura 3).

Cuando CPOL = 0 el flanco anterior es de subida y el flanco posterior de bajada. Con CPOL = 1, lo inverso es cierto. Tanto CPOL como CPHA afectarán a la lectura y escritura de datos serie del master y del esclavo.

• **CPHA = 0**

El esclavo usa el flanco anterior del reloj para leer el bit de dato en su entrada SI. Cuando CPOL = 0 el flanco ante-

rior es de subida. Cuando CPOL = 1 el flanco anterior es de bajada. El master debe usar el flanco posterior del reloj precedente para poner el bit de datos en su salida SO.

• **CPHA = 1**

El esclavo usa el flanco de bajada del reloj para leer el bit de dato en su entrada SI. Cuando CPOL = 0 el flanco de reloj es de bajada. Cuando CPOL = 1 el flanco es de bajada. El master debe usar el flanco de subida del reloj precedente para poner el bit de datos en su salida SO.

Usando esos parámetros hay otras cuatro posibles variantes en el camino en el que los datos se transfieren y la relación reloj/datos. (Tabla 1).

Es evidente que el master y el esclavo se deben configurar idénticamente en uno de los cuatro modos para que ocurran las comunicaciones confiablemente. Es importante establecer qué variante del interfaz está utilizando el dispositivo esclavo para poder programar correctamente el master para comunicarse con él. En el caso donde un periférico se agrega a un bus existente debe ser establecido el modo en que el bus está utilizando CPOL y CPHA para que en el chip periférico se configuren correctamente. Un ejemplo práctico de pequeña ayuda para hacer las cosas un poco más claras:

Asumiendo que usamos el modo 0 (CPOL = 0, CPHA = 0) en la Figura 4, consideraremos tres puntos clave:

1. La señal de reloj en estado de reposo es baja (0 V).
2. El esclavo lee el bit de datos serie en el flanco de subida del reloj. El master también debe haber

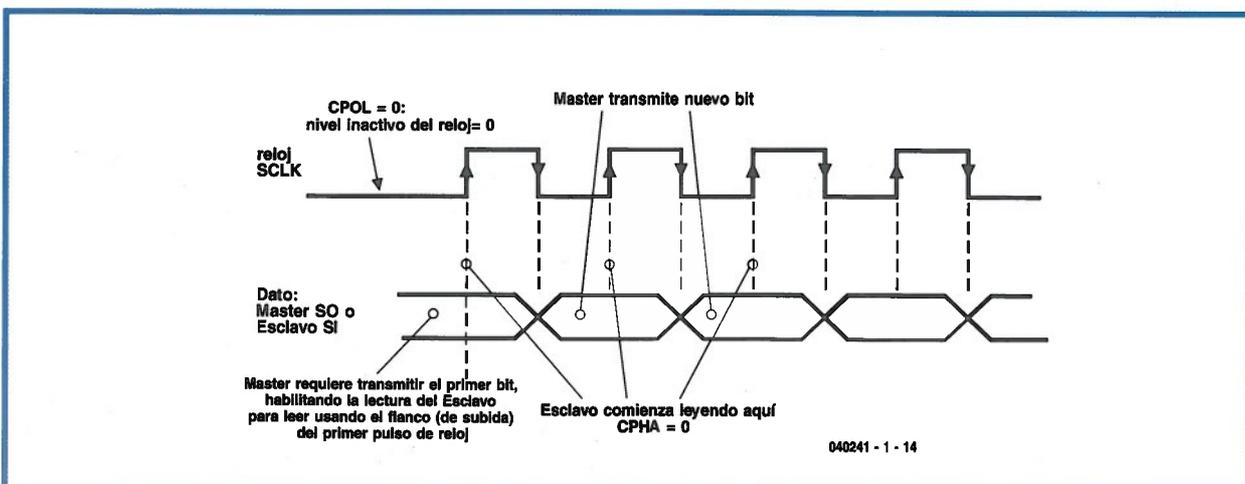


Figura 4. Transferencia de datos serie síncrona en modo 0.

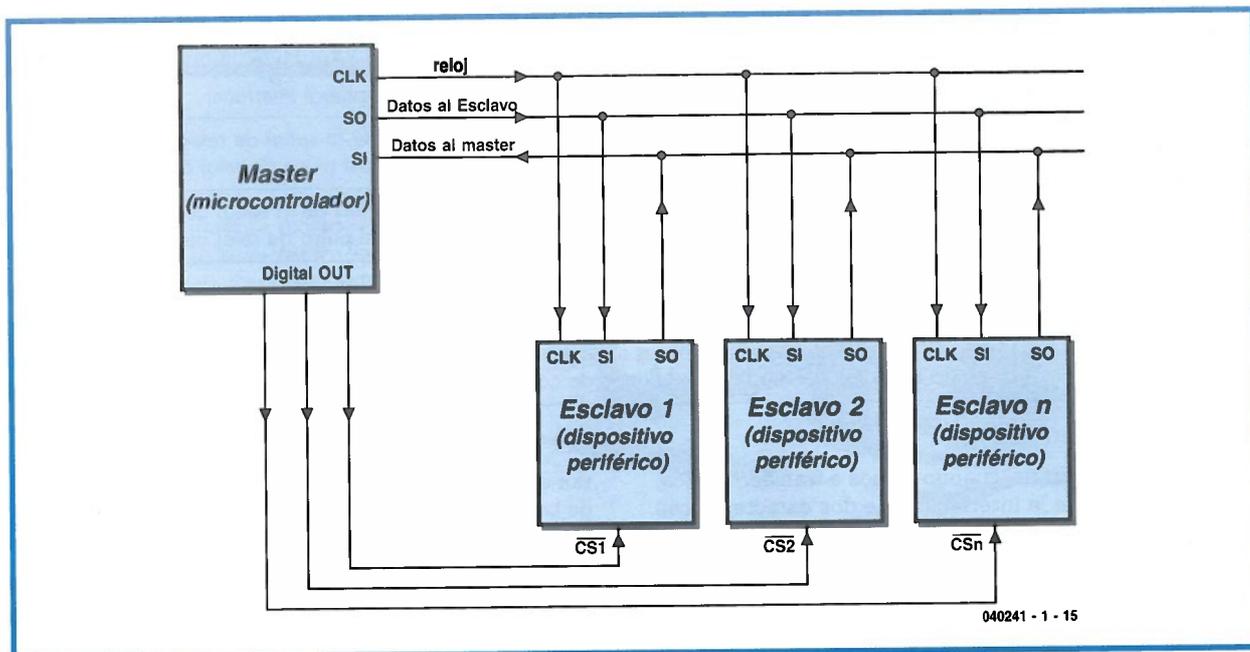


Figura 5. Una expansión en el bus serie.

enviado su bit de datos en el flanco previo de reloj, así que ahora es estable en la entrada SI de los esclavos.

3. El master cambia el bit de datos serie en el flanco de bajada del reloj. Esto permite bastante tiempo para estabilizar el nivel del bit de datos antes de que el esclavo lo lea en el siguiente flanco de subida. La señal de reloj es generada en el master.

Para una transferencia exitosa de datos serie asíncronos es importante estudiar las hojas de características de cualquier dispositivo periférico, para estabilizar las convenciones y responder a las tres cuestiones más básicas:

1. ¿Cuál es el estado de reposo (inactivo) (nivel bajo o alto) de la línea SCLK?
2. ¿Qué flanco de reloj (subida o bajada) usará el esclavo para leer datos en la línea de datos?
3. ¿Qué flanco de reloj (subida o bajada) usará el maestro para escribir datos en la línea de datos?

Una vez aclarados estos tres puntos, debería ser bastante sencillo programar correctamente el microcontrolador. Cuando estudiamos las hojas de características que describen cualquier dispositivo periférico siempre usamos los términos CPOL, CPHA, modo (SPI), flanco de subida y flanco de bajada. Para caracterizar el comportamiento de la interface es importante tener claro lo que quieren decir, antes de comenzar a trabajar en el software de comunicaciones.

La transferencia de datos en otra dirección desde un master o esclavo (desde la salida del esclavo SO a la entrada master SI) ocurre exactamente de la misma forma, porque el registro de desplazamiento del master y esclavo están conectados en un bucle, ya que la salida serie del esclavo está unida a la entrada serie del master y la señal de reloj es común. Cuando el esclavo lee datos en el flanco de subida del reloj cambiará su salida de datos en el siguiente flanco y así sucesivamente.

El bus serie síncrono

La configuración punto a punto puede expandirse añadiendo más dispositivos esclavos y conectando las señales de datos y reloj en paralelo con la mostrada en la Figura 5. Ahora que todos los esclavos comparten la misma señal de datos y reloj, es necesario para el master poder seleccionar qué esclavo envía o recibe datos. El master genera un chip select (\overline{CS}) activo a nivel bajo (normalmente). La transferencia de datos a o desde una estación esclava sólo puede ocurrir cuando su chip select está activo. Una vez que los datos se han transferido, el \overline{CS} pasa a nivel alto, el dispositivo esclavo se desactiva y los drivers de salida del bus pasan a estado de alta impedancia (o desactivados en colector abierto).

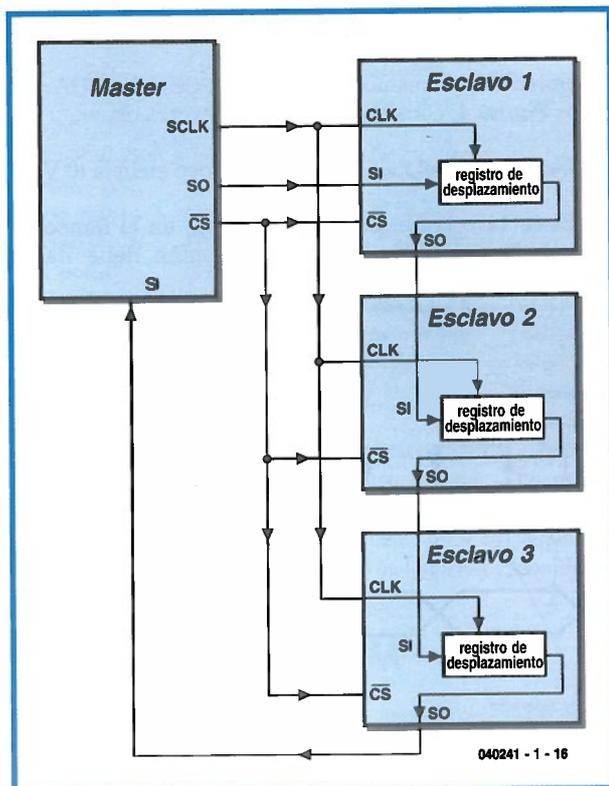


Figura 6. Disposición de esclavos en cascada.

Tabla 2. Ventajas y desventajas de los métodos de comunicación síncronos esclavos

Sistema de bus estándar	
Ventajas:	Cada esclavo es direccionable de forma individual. Adecuado para esclavos sin una salida SO. La transferencia de datos de cada esclavo puede ocurrir a la máxima velocidad de transferencia sin un sobrecalentamiento adicional del envío de datos a otros esclavos.
Desventajas:	Cada esclavo requiere una línea I/O dedicada o señal de chip select desde el master.
Sistema de bus en cascada	
Ventajas:	Cada esclavo adicional requiere sólo de una señal CS. La transferencia de datos a todos los esclavos ocurre de forma simultánea, por ejemplo, todos los esclavos leen los nuevos datos en el mismo punto cuando el CS está inhabilitado. Con un mínimo gasto de hardware, una buena solución si la velocidad de transferencia de datos no es la mayor prioridad.
Desventajas:	Sólo es adecuada para esclavos con una salida SO. Transferencias de datos lentas, especialmente si sólo necesita datos un esclavo. Es necesaria para que los datos lleguen a través de todos los esclavos. El esclavo con la menor velocidad de transferencia requiere que todo el sistema caiga hasta su velocidad.

La salida serie (SO) del esclavo permite que los datos sean enviados al master, pero si esto no es necesario, podemos dejarlo abierto y el siguiente byte de datos simplemente será desplazado y sobre-escribirá el byte de datos existente. Una vez inhabilitada la señal de CS, el esclavo puede leer el nuevo valor del dato.

El dato generalmente se transfiere en bytes con el más a la derecha, o Bit Menos Significativo (LSB) primero, o el más a la izquierda, o Bit Más Significativo (MSB) primero. La velocidad de transferencia de datos está gobernada por la velocidad de reloj producida por el master. Es importante que esta velocidad de reloj no sea tan rápida que el esclavo no pueda sacar y estabilizar su dato lo suficientemente rápido y que el master pueda leerlo de manera fiable, antes de que suceda el siguiente flanco de reloj. Si el procesador genera la señal de reloj puede ser necesario introducir bucles de espera en la rutina de temporización del reloj, para reducir la velocidad del reloj, aunque esto no es particularmente eficaz para el tiempo del procesador. Si reducimos demasiado la velocidad del reloj puede impactar en la respuesta total de todo el sistema, de manera que se ganará muy poco, porque la velocidad de transferencia es excitada por flanco. Es importante asegurar que se genera el número correcto de flancos de reloj para transferir todos los bits.

Esclavos en cascada

Hace años, un niño podía pasar el tiempo recogiendo flores para hacer un collar y, en particular uniendo tarimas para hacer cadenas, aunque eran tiempos más inocentes. Ahora, con la protección medioambiental y la introducción de la Gameboy de Nintendo, las cosas han cambiado. Varios dispositivos juntos en cascada, tal y como se muestra en la **Figura 6**, forman una cadena en configuración margarita, de manera que si se rompe el enlace a uno de los esclavos, todos los esclavos de más abajo pasan a desconectados, justo como en una cadena margarita real. En esta configuración, los registros de datos de todos los esclavos están conectados en serie y comparten la misma señal de chip select (\overline{CS}). Cada esclavo tiene una señal de salida serie conectada a la entrada serie de la siguiente etapa. Los datos del master son enviados a la entrada serie del primer esclavo y la salida serie del último esclavo se puede conectar de nuevo al master si es requerido o alternativamente dejando abierto el circuito. Desde el punto de

vista de los masters esta configuración parece un registro gigante de desplazamiento donde los datos de cada esclavo ocupan bytes sucesivos en la cadena de datos. En este ejemplo asumimos que tres esclavos están conectados y cada esclavo tiene un registro de desplazamiento de 8 bits. Para enviar datos al esclavo 3 el master:

1. Habilita \overline{CS}
2. El reloj saca 8 bits de datos para el Esclavo 3.
3. El reloj saca 16 Bits de forma que los 8 Bits para el Esclavo 3 son puestos en la posición correcta.
4. Inhabilita \overline{CS}

Aquí puede ocurrir un problema: si los esclavos 1 y 2 no necesitan ningún nuevo dato, ¿qué datos deberían enviarse para esos 16 bits? Algunas interfaces implementan un byte especial NOP (no operación) que se usa para sacar fuera la cadena de datos y que no tenga efecto cuando el esclavo lo lee como \overline{CS} inhabilitado. En nuestro ejemplo será necesario enviar los siguientes bits:

1. Habilita \overline{CS}
2. El reloj saca 8 nuevos bits de datos para el Esclavo 3.
3. El reloj saca dos NOPs de 8 bit para el esclavo 2 y 1.
4. Inhabilita \overline{CS}

Después de transferir datos, cuando \overline{CS} está inhabilitado, sólo el esclavo 3 tiene nueva información y otros dos esclavos no responderán a las NOPs. Un método alternativo consiste en enviar los datos existentes a los esclavos 1 y 2, de manera que cuando la transferencia esté completada, el dato del registro de desplazamiento será el mismo que antes. Los datos para cualquier número de esclavos pueden enviarse durante la transferencia de datos proporcionando que están insertados en la posición correcta en la cadena de datos.

Los esclavos en cascada a menudo se usan para almacenar caracteres de datos para cada posición en un gran display de LED (muchos caracteres). En principio no hay límite para el número de esclavos que se pueden poner en cascada de esta manera, pero en la práctica cada posición de esclavo adicional incrementa el tiempo que tarda en visualizarse el display. La **Tabla 2** nos indica algunas de las principales ventajas y desventajas de las configuraciones de chip esclavo.

(040241-1)

Conmutador DC controlado por RS-232

W. Pieczarek

Si desea aplicar un control de ordenador a una válvula solenoide u otra carga DC no es necesario usar una tarjeta de medida cara. Un transistor de potencia en la interface del PC puede hacer el trabajo de un 'canal de control' muy bien, y además a bajo coste. Como ejemplo mostramos el uso de la interface serie RS-232.

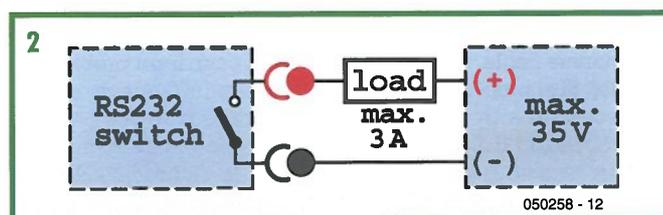
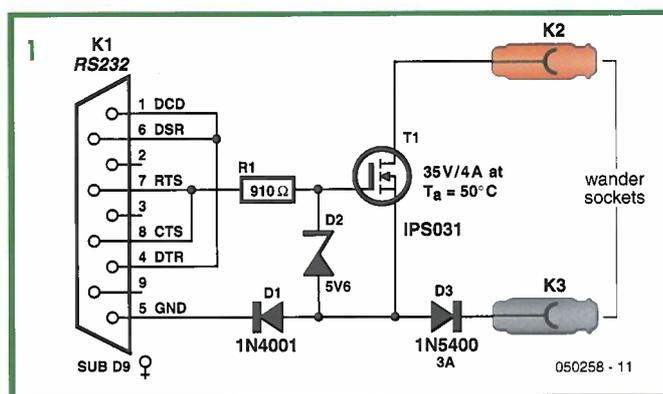
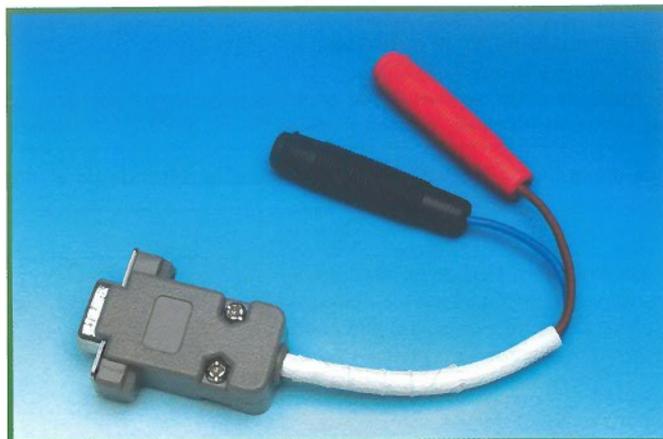
Cuando se abre en el PC un canal serie (puerto COM), la línea RTS (listo para enviar) se pone a nivel alto, y pasa de nuevo a bajo cuando el canal está cerrado. De esta forma puede implementarse un sencillo control on/off de una carga de potencia abriendo y cerrando un puerto COM.

La tensión nominal que hay en las líneas RS-232 está entre el rango ± 15 V. En algunos PCs este valor puede ser un poco más bajo, ± 12 V o incluso ± 8 V. El circuito debe considerar los diferentes valores que podemos encontrarnos en los diversos PCs que se construyen, además de la tensión negativa del nivel bajo de la interface RS232.

El transistor de potencia que se usa aquí es un MOSFET de Internacional Rectifier (IRF) modelo IPS031 en encapsulado TO220. Sus principales especificaciones se pueden añadir de la siguiente forma:

- Protección de sobre-temperatura integrada (corta a 165°C);
- Limitación de sobre-corriente (establecida a 12 A);
- Protección de descarga electrostática (4 kV);
- Bajo RDS (on) a unos 60 mW (0,06 W).

Este MOSFET corta corrientes de hasta 4 A sin un disipador, a una temperatura ambiente de hasta 50°C.



La **Figura 1** nos muestra la forma del conmutador DC controlado por RS232 y conectado entre el PC y la carga DC para poder encender y apagar. Esta disposición se basa en la asunción de que hay un aislamiento eléctrico entre el PC y el circuito DC que está apagándose y encendiéndose. También es importante para que la carga

DC se conecte con la polaridad correcta.

El circuito de la **Figura 2** consta de un conector hembra sub-D de nueve pines y hasta cinco partes más. El nivel alto RS232 en RTS está limitado un valor seguro de unos 5,6 V por los componentes R1 y el diodo zéner D2. A un potencial negativo en RTS (Bajo), el diodo D1 se bloquea y junto con

D2 protege la entrada del FET.

El diodo D1 está incluido para bloquear la tensión negativa de la línea RS232. El diodo D3 protege el transistor y la interface contra tensión de polaridad inversa aplicada a los terminales de carga DC. Usando un diodo 1N5400 como se muestra en el esquema, la corriente de carga se debería mantener por debajo de 3 A.

El circuito funcionará de forma fiable con todo el rango de tensión RS232, incluyendo 'seudo-RS232' ó 0 V/+5 V, i.e., niveles TTL como los usados en algunos PCs portátiles. Sin embargo, con tensiones de control -demasiado- bajas, necesitaremos llegar a incrementar la resistencia 'on' del transistor. El hardware se fija fácilmente en un conector sub-D, tal y como muestra la fotografía del autor del prototipo (**Figura 3**).

Usando esta configuración pueden alcanzarse frecuencias de conmutación de hasta 30 Hz, o incluso mayores, dependiendo del PC y la carga de la CPU. Para frecuencias mayores, sin embargo, Windows producirá un incremento de la fluctuación (variación de los instantes de conmutación). Como ya mencionamos, lo que necesitamos es que el canal COM del PC active el conmutador de la carga, y lo pueda apagar de nuevo. En QBasic esto apenas supone unas pocas líneas de programa más.

Encendido -on-:

```
OPEN "COM1:9600,N,8,1"
FOR OUTPUT AS #1
```

Apagado -off-:

```
CLOSE #1
```

Los ejemplos superiores asumen que COM1 está disponible en el ordenador.

(050258-1)

Tension de referencia con una diferencia

Rainer Reusch

Si necesitamos una tensión de referencia para obtener una temperatura estable en un circuito analógico, la elección habitual es un dispositivo con una referencia integrada, tal como el LC385 o REF02. Una alter-

nativa interesante es el denominado 'diodo de referencia'.

Los diodos zéner con una tensión menor de unos 6 V, como su nombre sugiere, dependen del efecto zéner. Estos dispositivos tienen un coeficiente de temperatura negativo. Los diodos zéner con una ten-

sión de unos 6 V, por otro lado, explotan el efecto avalancha y su tensión de ruptura se incrementa con la temperatura. Los diodos de referencia ocupan el espacio entre el zéner y el efecto avalancha: cuando operan a la corriente adecuada, dichos diodos tienen una

tensión de ruptura que es independiente de la temperatura.

Hay un amplio rango de diodos de referencia, incluyendo el 1N821(A), 1N823(A), 1N825(A), 1N827(A) y el 1N829(A). Todos ellos tienen una tensión nominal de 6,2 V y requieren una corriente

típica de 7,5 mA. Los diferentes tipos difieren en la tolerancia de su tensión de ruptura y su coeficiente de temperatura. El 1N821 tiene una tolerancia de tensión de 96 mV y un coeficiente de temperatura de 0,01 %/K. En el caso del 1N829 esos valores están alrededor de 5 mV y 0,0005 %/K: considerablemente más preciso, pero también, por supuesto, más caro. Sin embargo, es posible construir una tensión de referencia estable con la temperatura usando el 1N821: es simplemente una cuestión de proporcionar una calibración a través de una corriente de operación ajustable. La resistencia dinámica de los diodos es 15 Ω (10 Ω en el caso de los modelos con sufijo 'A').

En lo que se refiere a usar un diodo de referencia para construir una tensión de referencia cuya salida sea independiente de la carga y temperatura cuando precisamente necesitamos mantener la corriente de operación requerida (aquí 7,5 mA), incluso sobre un amplio rango de temperaturas, ¿qué mejor forma de lograrlo que usando el diodo para producir la corriente? Ésta es una solución elegante si la usamos en el circuito descrito aquí.

La operación del circuito depende de la base teórica del amplificador

operacional con realimentación negativa. El amplificador operacional ajusta continuamente, o por lo menos intenta ajustar, su salida, para que no haya tensión diferencial entre sus entradas. Si no tenemos en cuenta la tensión de alimentación, una tensión de salida de 0 V debería satisfacer este requerimiento, lo que nos daría una tensión de salida de alrededor de -1 V (con el diodo polarizado en directo). Sin embargo, si el circuito funciona a partir de una tensión de alimentación simple, el amplificador operacional pasará a nuestro estado deseado con una tensión de salida positiva. En este caso, el diodo está polarizado en inverso. Los valores del componente mostrados en el esquema del circuito nos dan una tensión de salida de 10,4 V. Este valor es el producto de la tensión de ruptura de 6,2 V y la relación entre R1 y R2. En R3 cae una tensión de 4,2 V, dando exactamente la corriente requerida de 7,5 mA. En principio es posible, cambiando los valores de los componentes, obtener cualquier tensión de salida mayor que la tensión de ruptura del diodo, aunque es esencial seleccionar el valor de R3 en cada caso para obtener la corriente correcta. Se deberían usar las resistencias de película de metal. En este

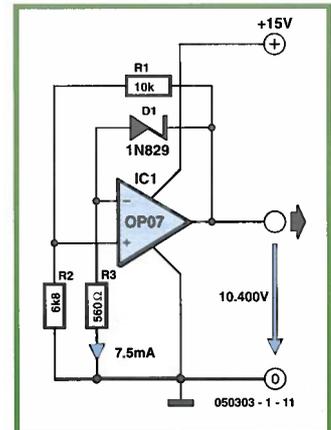
circuito no debemos usar amplificadores operacionales comunes, ya que estos tienen una tensión de offset que puede variar con la temperatura. El amplificador operacional de precisión y bajo coste OP07 que sugerimos aquí da unos resultados excelentes, y no es necesario el ajuste de la tensión de offset. La elección de los valores de componentes para el circuito se hace en dos pasos simples. Para el tipo 1N89x (A) tenemos:

$$R1 / R2 = 6.2 \text{ V} / (U_A - 6.2 \text{ V})$$

$$R3 = (U_A - 6.2 \text{ V}) / 7.5 \text{ mA}$$

Hemos comprobado el circuito usando un amplificador operacional OP07, un 1N829 y una resistencia de película metálica del 1 % de tolerancia con un coeficiente de temperatura de 100 ppm. Sobre un rango de temperatura de 5° a 40° C la variación de la tensión de salida será de unos pocos microvoltios. Colocando el circuito junto al hielo de un congelador y a un secador de pelo se produce una variación de alrededor de 2 mV; merece la pena que este experimento se haga con el OP07 colocado fuera del rango de temperatura especificado.

Para el circuito mostrado, la tensión de alimentación no debería ser menor de 12 V. Sobre este



valor la tensión de salida permanece absolutamente constante. El diodo de referencia es una alternativa económica a un dispositivo de referencia de precisión. El circuito en su totalidad toma una corriente de unos 10 mA y no puede competir en ese terreno con las referencias de las bandas tales como el LM385. El circuito es menos adecuado para usar en aplicaciones alimentadas a batería. La misma idea también se puede usar para diodos zéner ordinarios con tensiones de 5,6 ó 6,2 V. Con una corriente considerable, es posible alcanzar la salida más estable con una simple resistencia en serie.

(050303)

¿Como cablear un núcleo toroidal?



Ton Giesberts

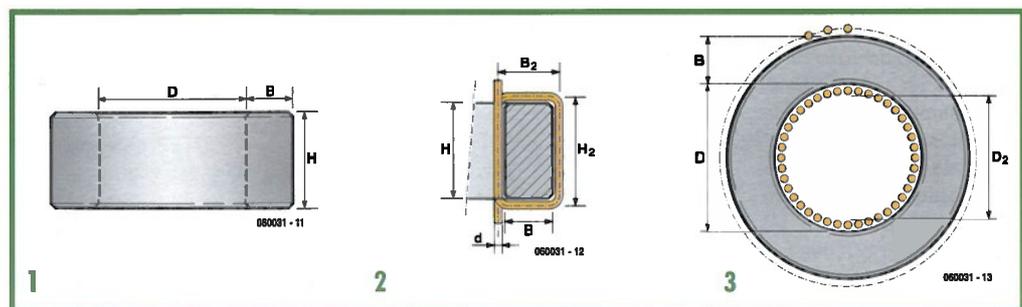
La parte de la electrónica más terrible es quizás el núcleo toroidal. Para todo aquel que lo haya bobinado sabrá perfectamente de qué estamos hablando. Particularmente si nos referimos a algo más que unas pocas vueltas. A continuación describiremos cómo tratar esto de la mejor forma posible. Primero mediremos las

dimensiones más importantes del toroide con un nonio. Esto es necesario porque en la práctica las dimensiones pueden variar más de unos pocos milímetros de lo publicado en las hojas de características. Las tres dimensiones importantes son (Figura 1) la altura (H), el espesor (B) y el diámetro interior (D). Para evitar cualquier equivocación, el espesor es igual a la mitad del diámetro externo menos el diámetro interno. Además, también es importante el diámetro del hilo y necesita-

mos tener en cuenta cuál es el espesor del aislamiento del hilo de cobre esmaltado, que típicamente va de 0,05 a 0,1 mm. Cada vuelta requiere una corrección de dos veces la altura más el espesor. No debemos olvidar tener en cuenta el tamaño del hilo, particularmente si es muy fino. Usando la línea central como referencia (Figura 2):

$$H2 = H + d$$

$$B2 = B + d$$



La longitud del hilo por vuelta es entonces:

$$2 \times (H2 + B2)$$

En lo que se refiere a saber el número de vueltas necesario para rellenar una capa, es necesario calcular el número máximo de vueltas por capa.

Debido al espesor del hilo, el diámetro interior efectivo se reduce (Figura 3). La curvatura hace que esta pérdida sea mayor que el sim-

ple tamaño del alambre. Es bastante posible que casi nunca consigamos hacer un buen cableado alrededor del núcleo, particularmente con hilo gordo. Para asegurarnos, quitaremos dos veces el diámetro del hilo desde el diámetro interior:

$$D2 = D - 2d$$

El número de vueltas es entonces la circunferencia del diámetro efectivo dividida por el espesor del hilo:

$$(\pi \times D2) / d$$

Por tanto, la longitud total para una capa es:

$$(\pi \times D2) / d \times 2 \times (H2 + B2)$$

Dependiendo de la diferencia de potencial entre los extremos del bobinado, el diámetro interior efectivo podría ser incluso más pequeño porque requiere de una distancia de aislamiento mínima. Las alternativas son usar un núcleo mayor o un

cable más fino. Esto último no es a menudo una opción porque se requiere de un dato de corriente.

Otro problema es el bobinado de múltiples capas. Cuando hagamos el cálculo, simplemente podemos doblar el diámetro del cable.

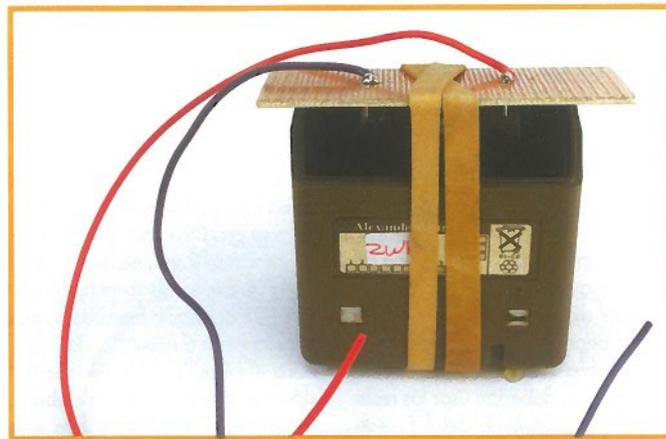
Por último, no debemos olvidar añadir los sobrantes finales de la bobina a la longitud total, de otra forma será muy difícil realizar la conexión a la bobina para trabajar con ella.

(080031-1)

Soportes de batería DIY

Luc Lemmens

Ocasionalmente existe la necesidad de cargar una batería de un portátil, teléfono móvil o PMR portátil usando una fuente de alimentación de laboratorio ordinaria. Podría ser que no tuviéramos el cargador o, como es habitual en el caso de los teléfonos móviles, que no dispongamos del adaptador de alimentación o que éste no nos pueda suministrar suficiente corriente como para descargar la batería. Si queremos cargar una batería de esta forma, siempre tenemos que proceder cuidadosamente y necesitamos saber lo que podemos hacer y lo que no con un tipo particular de batería. Sin embargo, esas consideraciones particulares están fuera del alcance de este artículo.



Casi todas las baterías tienen superficies de contacto planas para sus conexiones. ¿Cómo podemos conectarlas entonces a la fuente de alimentación? Podríamos soldar cables a ellas, si el material de contacto lo permite, pero eso a menudo

no es posible o deseable. En nuestro laboratorio usamos un método extremadamente sencillo y universal: un trozo de placa prototipo, unos pocos pines PCB, y unos grandes trozos de tiras de goma. Los pines de PCB se sueldan en un

trozo de placa separados a una distancia que se aproxima de centro a centro a la de los contactos de una batería. Los cables que van a la fuente de alimentación se sueldan a los pines de la PCB en la capa de cobre de la placa. Todo este montaje se hace al contrario de la batería, de forma que las conexiones se hacen contactos. Se colocan un par de fuertes trozos de goma alrededor de la batería y el conector que la sujeta en su lugar, proporcionando suficiente presión para dejar nuestras manos libres para operar con la fuente de alimentación.

Por último, recordamos que siempre debemos estar atentos al cambiar una batería, ya que a veces desconocemos el modelo exacto, lo que requiere atender especialmente a la corriente de carga y tensión.

(080032-1)

Limpieza del aluminio con bicarbonato de sosa

Luc Lemmens

Hemos recibido un sorprendente truco de un radio aficionado para eliminar fácilmente el óxido y la suciedad del aluminio ayudándonos del bicarbonato de sosa a.k.a, levadura en polvo. Al parecer, un poco de bicarbonato de sosa disuelto en agua es un potente limpiador, por ejemplo, para las veletas de los viejos condensadores de aluminio de los sintonizadores. Una rápida búsqueda en Google con las palabras 'bicarbonato de sosa para limpieza' nos muestra que esta sustancia es una alternativa verde a los pro-



ductos de limpieza. La solución de cuatro cucharadas de bicarbonato de sosa en un litro de

agua es adecuada para una gran cantidad de propósitos de limpieza. Podemos encontrar

recetas más detalladas usando el bicarbonato de sosa como ingrediente para limpiar toda clase de cosas.

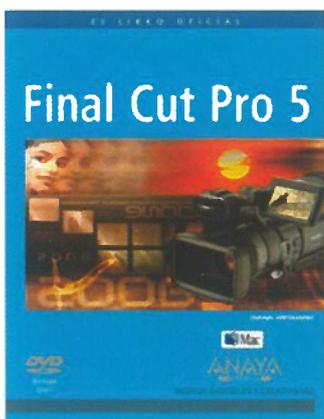
Nosotros no teníamos un condensador de sintonía a mano para probarlo, pero experimentamos con una taza de café de aluminio y verdaderamente funciona sorprendentemente bien.

(080033-1)

Más información en

www.frugalfun.com/bakingsoda.html
www.frugalfun.com/cleaners.html
<http://parents.berkeley.edu/advice/household/jewelry.html>
<http://ezinearticles.com/?Baking-Soda-for-Cleaning&id=10168>

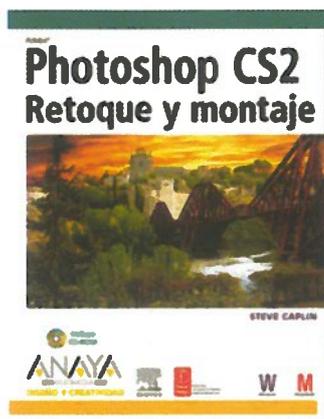
Final Cut Pro 5
 Por Diana Weynand
 ISBN 84-415-1983-8
 480 páginas
 Editorial Anaya
 Multimedia



Final Cut Pro ofrece uno de los paquetes de edición de vídeo más potentes y precisos del mercado. Sus herramientas funcionan prácticamente en cualquier formato, desde DV y HDV nativo hasta HD sin compresión. Actualizado a la versión 5, este manual proporciona toda la información necesaria para garantizar la máxima velocidad y calidad de los proyectos, crear títulos de impacto y realizar tareas de grabación.

Contiene ejemplos prácticos que han sido tomados de proyectos reales, tanto en NTSC como en PAL, para demostrar los distintos usos de la aplicación. El DVD-ROM adjunto contiene archivos con lecciones y materiales para practicar durante más de cuarenta horas.

Photoshop CS2. Retoque y montaje
 Por Steve Caplin
 ISBN 84-415-1966-8
 480 páginas
 Editorial Anaya
 Multimedia

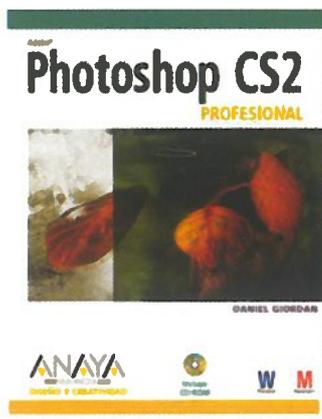


Realizar montajes para lograr composiciones auténticas es uno de los objetivos esenciales de cualquier artista digital. Las técnicas para conseguir fotomontajes impactantes se perfeccionan continuamente fomentando la expectación de los creativos.

Este libro le enseña el tratamiento de todo tipo de imágenes con rapidez y economía de medios. Todos los términos y referencias aparecen adaptados a la última versión CS2. Cada capítulo contiene ejercicios prácticos que desarrollan técnicas del mundo real.

Aprenderá a utilizar las nuevas herramientas de Photoshop CS2 e interesantes trucos tales como envejecer fotografías, generar atmósferas sugerentes e incorporar texturas. Se incluye un CD-ROM con imágenes de alta resolución y libres de derechos.

Photoshop CS2. Profesional
 Por Daniel Giordan
 ISBN 84-415-1991-9
 432 páginas
 Editorial Anaya
 Multimedia



Photoshop es la herramienta preferida por diseñadores y artistas para crear composiciones digitales. Para aquellos que ya conocen sus herramientas, el desafío es la forma de combinarlas.

La diferencia entre este libro y otros muchos realizados sobre Photoshop El objetivo es convertir imágenes simples en obras de arte cargadas de emotividad. Descubrirá qué aspectos hacen espectacular una imagen y cómo el color y los añadidos textuales generan impacto en una composición.

Este manual presenta más de una docena de vistosas y provocativas imágenes deconstruidas individualmente para que descubra cómo usar tecnologías de la nueva versión CS2. Incluye un CD-ROM con fotografías originales utilizadas por el autor, imágenes terminadas y otras herramientas de gran utilidad.

InDesign CS2. Técnicas esenciales
 Por Wayne Rankin,
 Mike McHugh
 ISBN 84-415-1967-6
 288 páginas
 Editorial Anaya Multimedia



Adobe InDesign es hoy en día una de las herramientas de trabajo más utilizadas en la edición y diseño de publicaciones profesionales gracias a su flujo de trabajo eficiente e integrado.

Con este libro aumentará sus habilidades como diseñador, descubriendo todo su talento. Los ejemplos tratan muchas de las situaciones a las que los creativos se enfrentan diariamente, como por ejemplo, la creación de un portafolio interactivo con botones, vídeos, y sonidos.

Los consejos y trucos de los expertos le ayudarán a crear proyectos profesionales. El CD adjunto contiene plantillas, así como todos los archivos utilizados en los ejemplos del libro.



JUNIO 2006

CONDICIONES GENERALES

Los circuitos impresos, carátulas autoadhesivas, ROMs, PALs, GALs, microcontroladores y disquetes que aparecen en las páginas de ELEKTOR se encuentran a disposición de los lectores que lo requieran. Para solicitarlos es necesario utilizar el cupón de pedido que se encuentra en las páginas anexas.

Este mismo cupón también puede utilizarse para efectuar pedidos de los libros de la colección de ELEKTOR (en versión original inglesa).

- Los items marcados con un asterisco (*) tienen una vigencia limitada y su disponibilidad solo puede garantizarse durante un cierto periodo de tiempo.

- Los items que no se encuentran en esta lista no están disponibles.

- Los diseños de circuitos impresos se encuentran en las páginas centrales de la Revista. En ocasiones y por limitación de espacio no se garantiza la publicación de todos los circuitos. En estos casos los lectores interesados pueden solicitar los diseños, utilizando el mismo cupón de pedido y les serán enviados a su domicilio contra reembolso de 500 pts. (incluidos gastos de envío).

- Los EPROMs, GALs, PALs, (E)PLDs, PICs y otros microcontroladores se suministrarán ya programados. Los precios y las descripciones de los diferentes productos están sujetos a cambios. La editorial se reserva el derecho de modificar los precios sin necesidad de notificación previa. Los precios y las descripciones incluidas en la presente edición anulan los publicados en los anteriores números de la Revista.

FORMA DE ENVIO

Los pedidos serán enviados por correo a la dirección indicada en el cupón de las páginas anexas. Además los lectores pueden formular pedidos por teléfono llamando al número 91 327 37 97 de lunes a viernes en horario de 9,30 a 14 h y de 16 a 19 h. Fuera de este horario existe un contestador telefónico preparado para recoger las demandas. Los gastos de envío serán abonados por el comprador, tal como se indica en el cupón.

FORMA DE PAGO

Todos los pedidos deberán venir acompañados por el pago, que incluirá los gastos de envío, tal como se indicó anteriormente.

El pago puede realizarse mediante cheque conformado de cualquier banco residente en territorio español, giro postal anticipado, tarjeta VISA (en este caso debe indicarse la fecha de caducidad, domicilio del propietario de la tarjeta y firma del mismo).

Nunca se deberá enviar dinero en metálico con el pedido. Los cheques y los giros postales deben ser nominativos a la orden de VIDELEC S.L.

SUSCRIPCIONES A LA REVISTA Y EJEMPLARES ATRASADOS

Las suscripciones o pedido de números atrasados, si se encuentran disponibles, se realizarán a LARPRESS, C/ La Forja, nº 27 - 28850 Torrejón de Ardoz (Madrid). Telf: 91 677 70 75, Fax: 91 676 76 65. E-mail: suscrip@larpress.com

Los precios de ejemplares atrasados son de 3,60 € más gastos de envío.

COMPONENTES UTILIZADOS EN LOS PROYECTOS

Todos los componentes utilizados en los proyectos ofrecidos en las páginas de la Revista se encuentran generalmente disponibles en cualquier establecimiento especializado o a través de los anunciantes de este ejemplar. Si existiera alguna dificultad especial con la obtención de alguna de las partes, se indicará la fuente de suministro en el mismo artículo. Lógicamente los proveedores indicados no son exclusivos y cualquier lector podrá optar por su suministrador habitual.

CONDICIONES GENERALES DE VENTA

Plazo de entrega: El plazo normal será de 2-3 semanas desde la recepción del pedido. No obstante no podemos garantizar el cumplimiento de este periodo para la totalidad de los pedidos.

Devoluciones: Aquellos envíos que se encuentren defectuosos o con la falta de alguno de los componentes podrán ser devueltos para su reposición, solicitando previamente nuestro consentimiento mediante llamada telefónica al número (91) 3273797 en horario de oficina. En este caso la persona que llame recibirá un número de devolución que deberá hacer constar al devolver el material en un lugar bien visible. En este caso correrá por nuestra cuenta el gasto de envío de la devolución, debiéndolo hacer así constar el remitente en su oficina postal. A continuación se le enviará nuevamente el pedido solicitado sin ningún gasto para el solicitante. En cualquiera de los casos anteriores, solo se admitirán las devoluciones en un plazo de tiempo de 14 días contados a partir de la fecha de envío del pedido.

Patentes: Algunos de los circuitos o proyectos publicados pueden estar protegidos mediante patente, tanto en la Revista como en los libros técnicos. La editorial LARPRESS no aceptará ninguna responsabilidad derivada de la utilización inadecuada de tales proyectos o circuitos para fines distintos de los meramente personales. Copyright: Todos los dibujos, fotografías, artículos, circuitos impresos, circuitos integrados programados, disquetes y cualquier otro tipo de software publicados en libros y revistas están protegidos por un Copyright y no pueden ser reproducidos o transmitidos, en parte o en su totalidad, en ninguna forma ni por ningún medio, incluyendo fotocopiado o grabación de datos, sin el permiso previo por escrito de Editorial LARPRESS.

No obstante, los diseños de circuitos impresos si pueden ser utilizados para uso personal y privado, sin necesidad de obtener un permiso previo.

Limitación de responsabilidad: Todos los materiales suministrados a los lectores cumplen la Normativa Internacional en cuanto a seguridad de componentes electrónicos y deberán ser utilizados y manipulados según las reglas universalmente aceptadas para este tipo de productos. Por tanto ni la editorial LARPRESS, ni la empresa suministradora de los materiales a los lectores se hacen responsables de ningún daño producido por la inadecuada manipulación de los materiales enviados.

CONSULTORIO TECNICO

Existe un Consultorio técnico telefónico gratuito a disposición de todos los lectores. Este servicio se presta todos los lunes y martes laborables en horario de 17 a 19 h.

El número de teléfono para consultas es el 91 375 02 70.

Codigo Precio (€)

E312 MAYO 2006

Placa de Aplicación para el R8C/13:

- Placa montada	050179-92	96,00
- PCB	050179-1	27,35
- LCD con contraluz	050179-72	14,39
- Poly-LED display	050179-73	51,00

Placa de Prototipos para FPGA:

- Placa montada	050370-91	S/P
-----------------	-----------	-----

Módulo FPGA Versátil:

- Módulo montado plug-on	040477-91	342,00
• Suscriptores		353,00
• No suscriptores		

E311 ABRIL 2006

Controlador de Motor sin Escobillas:

- ST7MC1, programado	050157-41	7,54
----------------------	-----------	------

Un Tom Thumb de 16 Bits:

- R8C Starter Kit	050179-91	16,45
-------------------	-----------	-------

E310 MARZO 2006

Adaptador de Fuente de Alimentación para Portátil de 95 Vatios:

- PCB	050029-1	59,00
-------	----------	-------

Controlador Automático para Ventana de Ático:

- Disco, PIC código fuente y hex	050139-11	10,28
- PIC16F84A-20/P, programado	050139-41	26,00
- Módulo LCD 2x16 caracteres	030451-72	14,00
- Módulo PLED 2x16 caracteres	030451-73	51,00

Horno de Soldadura de Oja para SMD:

- Disco, PIC código fuente y hex	050319-11	10,28
- AT89C52/24J, programado	050319-41	15,00
- Módulo LCD 2x16 caracteres	030451-72	14,39
- Módulo PLED 2x16 caracteres	030451-73	51,00

Conmutador Temporizado para Lavadora:

- PCB	050058-1	18,00
- Disco, PIC código fuente y hex	050058-11	12,44
- PIC16F84, programado	050058-41	26,00

E309 FEBRERO 2006

De la "A" a la "D" por medio del USB:

- PCB	050222-1	16,00
- IOW24-P, programado	050222-41	19,00

Supervisor de teléfono:

- PIC16F628-20/P, programado	050039-41	17,00
- CD-ROM, PIC hex & source codes, LCM First Server	050039-81	14,00

E308 ENERO 2006

Control remoto por teléfono móvil:

- PCB	040415-1	13,00
- Disco, PIC código fuente y hex	040415-11	11,00
- PIC16F84A-20/P, programado	040415-41	21,00

E307 DICIEMBRE 2005

27C512 Emulador:

- Disco, software de proyecto	030444-11	11,00
- EPM7064SLC84-15, programado	030444-31	55,00
- AT90S8515-4PC, programado	030444-41	30,00

Pequeño Coloso:

- Disco, código fuente del PIC	040267-11	11,00
- PIC12F675-C/P, programado	040267-41	6,00

Cerradura Flash para PCs:

- PIC16F628A-I/SO, programado	050107-41	10,00
- CD-ROM, software de proyecto	050107-81	14,00

E306 NOVIEMBRE 2005

Medidor ESR/C:

- PCB, desnudo	040259-1	25,00
- Disco, PIC código fuente y hex	040259-11	11,00
- PIC16F877-20/P, programado	040259-41	31,00
- PIC16F84A-20/P, programado	040259-42	21,00
- Kits de piezas sin caja	040259-71	127,00
- Caja	040259-72	14,00

Medidor de distancia manual:

- ATmega8-16PC, programado	040015-41	31,00
- Disco, software de proyecto	040015-11	11,00

E305 OCTUBRE 2005

Adaptador MP3 para TV:

- PCB, desnudo	054035-1	8,00
- Disco, software de proyecto	054035-11	10,50
- EPM7064SLC44-10, programado	054035-31	55,00

Cerradura codificada con un botón:

- Disco, código fuente y hex	040481-11	10,50
- PIC16F84, programado	040481-41	14,00

Contador de Frecuencia de 1 MHz:

- Disco, software de proyecto	030045-11	10,50
- AT90S2313-10PI, programado	030045-41	11,00

	Código	Precio (€)
E297 FEBRERO 2005		
BUS casero-I²C:		
- PCB	040033-1	22,56
- Disco, código fuente y hex	040033-11	9,12
E296 ENERO 2005		
Filtros de Supresión de Ruido:		
- PCB	030217-3	25,00
E295 DICIEMBRE 2004		
Amplificador de Clase T de 2 x 300 W Clarity:		
- PCB	030217-2	30,00
E294 NOVIEMBRE 2004		
Amplificador Clase-T 2x300 W:		
- Placa amplificador con SMD premontado; núcleos para L1 y L2	030217-91	59,00
Dado rodante:		
- Set de PCB'S (6 caras)	040248-1	26,00
- Disco, código fuente y hex	040248-11	9,00
- AT89C2051-12PI, programado	040248-41	16,00
Cuchillo del Ejército Suizo:		
- Set de PCB'S: MCU, RS232, USB	030448-1	17,71
- Set de discos (a+b), todo el software del proyecto	030448-11	12,12
- AT89S8252-24PC, programado	030448-41	48,00
E293 OCTUBRE 2004		
Analizador R/C:		
- Disco, PIC código fuente	030178-11	9,12
- PIC16F627-4/CP, programado	030178-41	19,00
Convertidor USB Controlado a través de HTML:		
- Disco, programas ejemplo	044034-11	9,12
E292 SEPTIEMBRE 2004		
Micro Servidor Web con placa MSC1210:		
- Placa microprocesador, ensamblada y comprobada	030060-91	119,00
- Placa de expansión de red, ensamblada y comprobada	044026-91	78,00
- Paquete combinado (incluyendo 030060-91 + 044026-91 y todos los artículos)	044026-92	195,00
- Placa de expansión de red, sólo PCB	044026-1	20,00
E291 AGOSTO 2004		
Multi Programador:		
- PCB	020336-1	16,45
- Disco, firmware y código fuente	020336-11	9,12
Pocket Pong:		
- Disco, software PIC	030320-11	9,12
Router de vías:		
- PCB	030403-1	20,19
- Disco, software PC Y PIC	030403-11	9,12
- PIC16F877-20/P, programado	030403-41	40,02
Operador Silencioso:		
- Disco, PIC software	030209-11	9,12
- PIC16F84-10P, programado	030209-41	27,41
E290 JULIO 2004		
Diseño de Nuestro Propio Circuito Impreso:		
- PCB	030385-1	29,00
Preamplificador de Gama Alta Controlado Digitalmente (2):		
- PCB placa principal	020046-1	16,00
- PCB placa de relés	020046-2	14,00
- PCB placa de alimentación	020046-3	12,50
- Disco	020046-11	9,00
- PIC18LF452-V/L, programado	020046-41	52,00
Medidor de Velocidad y Dirección del Viento:		
- Disco, software del proyecto	030371-11	9,00
- PIC16F871, programado	030371-41	33,00
E289 JUNIO 2004		
Construya su Propio Receptor DRM:		
- PCB	020148-1	16,00
- Disco, proyecto ejemplo	020148-11	9,12
Caja De Música y el Sonido de Pandora:		
- PCB	030402-1	20,00
- Disco, código fuente y hex	030402-11	9,12
- PIC16F871/P, programado	030402-41	38,50
Explorador de VHF de Banda Baja:		
- PCB	020416-1	16,50
E288 MAYO 2004		
Construya su Propio Receptor DRM:		
- PCB	030365-1	17,00
- Disco, programa DRM.exe	030365-11	9,00
Cerradura Codificada:		
- PCB	020434-1	14,40
- Disco, código fuente y hex	020434-11	9,00
- PIC16F84A-4P, programado	020434-41	27,41
Multicanal Seguro para Modelos Controlados por Radio:		
- Disco, código fuente	020382-11	9,00
- AT89C52-24JI, programado	020382-41	16,24

	Código	Precio (€)
Medidor de Frecuencia Multifunción:		
- PCB	030136-1	17,50
- Disco, software del proyecto	030136-11	9,00
- AT90S2313-10PC, programado	030136-41	15,00
E287 ABRIL 2004		
Reloj Digital con Alarma:		
- Disco, PIC código fuente y hex	030096-11	9,11
- PIC16F84-04/P, programado	030096-41	28,36
iAccess:		
- Disco set, código fuente y control	020163-11	14,02
- AT89S8252-12PC, programado	020163-41	25,94
Sencillo Inversor de Tensión de 12V a 230V:		
- PCB	020435-1	16,00
Conmutador Controlado por Tacto:		
- Disco, PIC código fuente	030214-11	9,12
- PIC12C508A04/S08, programado	030214-41	11,49
E286 MARZO 2004		
Placa flash 64-K 80C552:		
- PCB	030042-1	17,45
- Disco, misc. software del proyecto	030042-11	9,12
- 29F010, programado	030042-21	18,66
- GAL 16V8D15QP, programado	030042-31	11,38
Registrador Climático:		
- PCB	030076-1	14,44
- Disco, software Windows	030076-11	9,12
Codificador FMS para Simulador de Vuelo:		
- PCB	030066-1	19,03
- 87LPC767BN, programado	030066-11	31,74
Ruleta a Diodos Led:		
- PCB	030168-1	33,00
- Disco, código fuente y hex	030168-11	9,12
- 89C2051-12PC, programado	030168-41	15,02
E285 FEBRERO 2004		
Receptor de Control Remoto en FM:		
- PCB	034044-1	19,00
Cronómetro de Proyectos:		
- Disco, códigos fuente y objeto	020350-11	9,12
- PIC16F84-10P, programado	020350-41	24,52
Descubriendo el motor paso a paso (II):		
- Disco, código fuente	020127-11	9,12
Generador de Reloj Universal:		
- Disco, código fuente	020395-11	9,12
Enlace RS232 sin hilos:		
- PCB	030204-1	16,24
E284 ENERO 2004		
Contador de revoluciones para modelos de radio-control:		
- PCB	024111-1	33,00
- Disco, código fuente y hex	024111-11	9,00
- 89C2051-12PC, programado	024111-41	16,00
Visualizador de Texto con Desplazamiento:		
- Disco, código fuente y hex	020407-11	9,00
Convertor USB analógico:		
- PCB	020374-1	14,00
- Disco, códigos hex y software Windows	020374-11	9,00
- PIC16C765, programado	020374-41	25,00
E283 DICIEMBRE 2003		
Generador de Señal de RF con DDS:		
- PCB, generador	020299-1	22,00
- PCB, control/alimentación	020299-2	23,00
- AT90S8515 8PC, programado	020299-41	57,00
Detector de metal por inducción balanceada:		
- PCB	020290-1	17,00
E282 NOVIEMBRE 2003		
Generador de imágenes ATV:		
- Disco, código fuente y hex	020295-11	9,12
- AT90S8515-8PC, programado	020295-41	28,37
- AT90S1200-12PC, programado	020295-42	25,70
Interruptor remoto mediante teléfono DTMF:		
- PCB	020294-1	22,00
- Disco, software del proyecto	020294-11	9,12
- PIC16F84A-20/P, programado	020294-41	27,50
Display de Cristal Líquido con Bus I²C:		
- PCB	030060-2	14,00
PICProg 2003:		
- PCB	010202-1	17,00
- Disco, software Windows	010202-11	9,12
- PIC16F874-20/P, programado	010202-41	44,00
Central de Medida de Precisión (2):		
- Placa ensamblada y comprobada	030060-91	68,00
Preamplificador a válvulas (I):		
- PCB, placa amplificador	020383-1	22,00
- PCB, placa alimentación	020383-2	21,00
- PCB, placa I/O	020383-3	19,00