



Preguntas para hoy

- ¿Qué es un FPGA?
- ¿Qué puede hacer por mí?
- ¿Qué aplicaciones?
- ¿Precio? ¿Herramientas gratis?

FPGA

domando a la bestia

¡Última edición impresa de Elektor!

¡Únete a la comunidad digital de Elektor ahora!
www.elektor.es/miembro



Microcontroladores PIC® USB de 8 bit sin cristal que reducen los costes del sistema y el consumo de energía

La precisión del reloj del 0,25% permite conectividad USB, suprimiendo así la necesidad de un cristal externo



Los microcontroladores USB con el coste más bajo y el tamaño más pequeño de Microchip tienen de 14 a 100 patillas y son los primeros microcontroladores de 8 bit en integrar control de LCD, reloj/calendario en tiempo real alimentado mediante batería y USB en un solo chip.

Los microcontroladores PIC® USB más recientes de Microchip incorporan fuentes de reloj interno con una precisión del reloj del 0,25% para permitir conectividad USB sin cristal externo. También son los primeros microcontroladores USB en combinar entre 14 y 100 patillas junto a periféricos integrados y hasta 128 KB de Flash. La tecnología eXtreme Low Power (XLP) también logra un consumo de energía a partir de 35 µA/MHz en modo activo y de 20 nA en modo dormido.

El coste más bajo y el tamaño más pequeño

Los microcontroladores PIC16F145X ofrece conectividad USB y sensado táctil capacitivo, así como una amplia variedad de periféricos integrados que ocupan una superficie a partir de 4x4 mm.

Sensado táctil de altas prestaciones con USB

Los microcontroladores PIC18F2X/4XK50 integran una Unidad de Medición de Tiempo de Carga (Charge Time Measurement Unit, CTMU) y trabajan a una tensión de 1,8 V a 5 V, tienen patillas compatibles con los microcontroladores PIC18 ya existentes, facilitando así la adaptación a unas mayores prestaciones.

USB con control de LCD y reloj/calendario en tiempo real con Vbat

La familia PIC18F97J94 ofrece conectividad USB con control de LCD y un reloj/calendario en tiempo real (RTCC), todo ello en un solo microcontrolador PIC® de 8 bit.

EMPIECE CON 3 PASOS SENCILLOS:

1. Escoja los periféricos y el número de patillas que mejor se adapten a su aplicación
2. Utilice las pilas USB y los drivers de software gratuitos para acelerar el diseño
3. Empiece a desarrollar con kits de desarrollo de bajo coste

Para mayor información, visite: www.microchip.com/get/eu8bitUSB



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

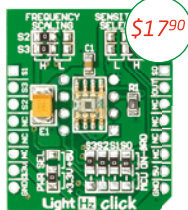
El nombre y el logo de Microchip, MPLAB y PIC son marcas registradas de Microchip Technology Incorporated en EE.UU. y en otros países. PICDEM es una marca de Microchip Technology Incorporated en EE.UU. y en otros países. Las restantes marcas citadas pertenecen a sus respectivas compañías. © 2012, Microchip Technology Incorporated. Todos los derechos reservados. DS31039A, ME10495Pa10.12



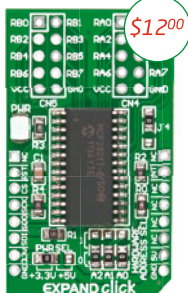
\$49⁰⁰



\$18⁰⁰



\$17⁹⁰



\$12⁰⁰



\$10⁰⁰



\$16⁹⁰



\$16⁰⁰

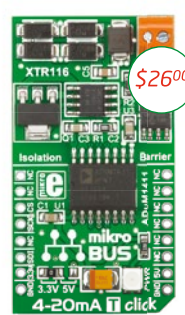
un

CONECTOR



para TODOS PLACAS click

Solo pon la placa Click™ necesitas en el innovador nuevo conector mikroBUS™ y está lista para trabajar directamente.



\$26⁰⁰



\$25⁰⁰



\$21⁰⁰



\$9⁰⁰



\$45⁰⁰



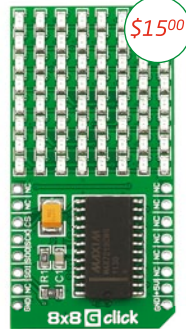
\$19⁰⁰



\$49⁰⁰



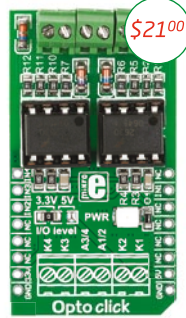
\$18⁰⁰



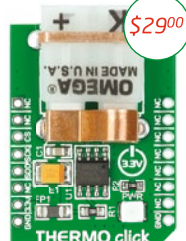
\$15⁰⁰



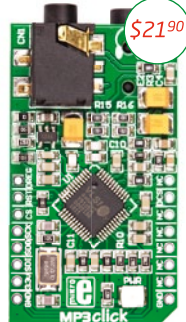
\$41⁹⁰



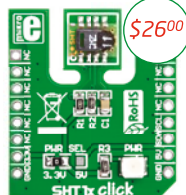
\$21⁰⁰



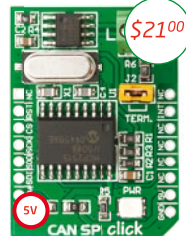
\$29⁰⁰



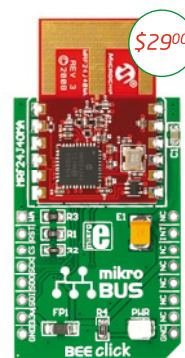
\$21⁹⁰



\$26⁰⁰



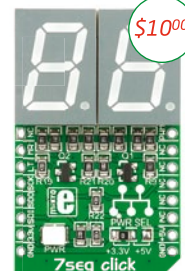
\$21⁰⁰



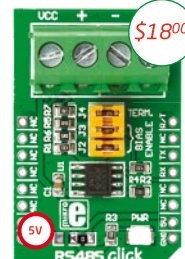
\$29⁰⁰



\$13⁹⁰



\$10⁰⁰



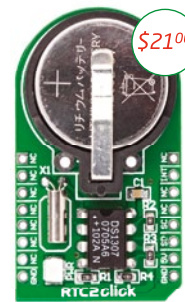
\$18⁰⁰



\$8⁰⁰



\$21⁰⁰



\$21⁰⁰



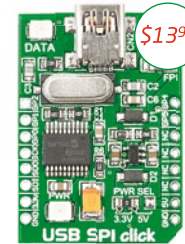
\$44⁰⁰



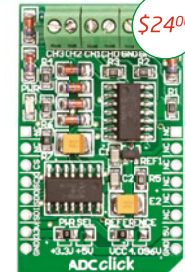
\$19⁰⁰



\$24⁰⁰



\$13⁹⁰



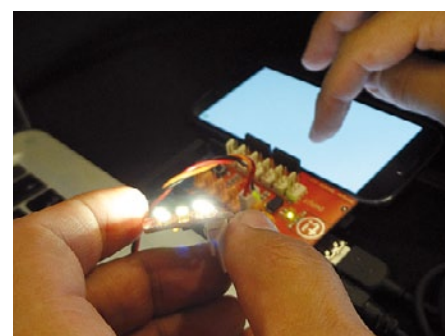
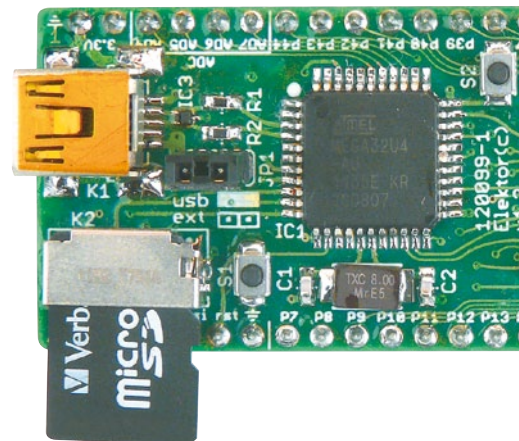
\$24⁰⁰

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

www.libstock.com

CONSIGUELO AHORA
www.mikroe.com

NOTA: Con cada placa Click se incluyen dos conectores header 2x8 en el paquete.



● Comunidad

14 El Mundo de Elektor

- Medidor de Radicación Mejorado de Elektor en el ConSat-1... pronto
- Medidor de Radicación Mejorado de Elektor @ Universidad de Oxford ... pronto 2
- Elektor Monster Formant en YouTube
- Lost in the Electronic Maze
- Fiesta del 25 (19hex) Aniversario de Circuit Cellar
- Poeta RF



● Proyectos

18 Árbol de Navidad con LED activo

Los vivos patrones visuales generados por este chisme controlado por un PIC no tienen fin, especialmente si montas dos.

22 Diseña tu propio integrado

Una introducción optimista a los FPGAs, seguida de una descripción del Sistema de Desarrollo de FPGA DIY de Elektor.

30 Cable USB-IO24

Medir, controlar y regular fácilmente con el PC. Con el cable presentado aquí pueden leerse y transferirse hasta 24 señales digitales mediante el puerto USB de un PC.

44 Simplificando Linux Embebido (6)

Este mes conectamos la placa Linux de Elektor a la típica red doméstica utilizando una aplicación de servidor escrita por nosotros mismos.

50 Arduino en marcha (3b)

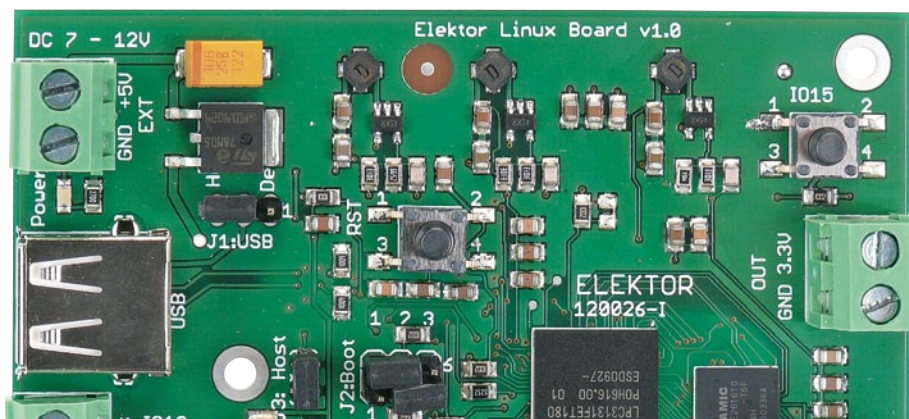
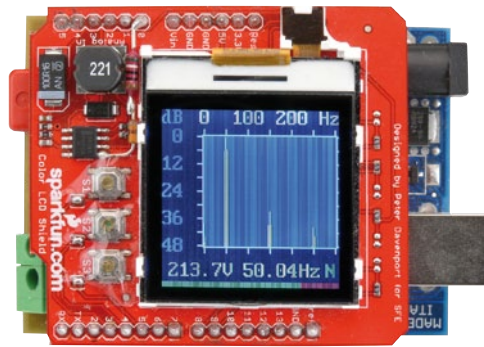
¿Puede to smartphone Android leer sensors directamente? Probablemente no, y aquí es donde un Arduino Mega se convierte en útil.

58 Electrónica para Principiantes (10)

En esta décima y última parte damos un vistazo a circuitos básicos de radio frecuencia (RF).

68 Analizador de tensión de red con Arduino

Este circuito realiza una comprobación con rigurosa calidad de la red de AC a partir de un enchufe. Dispone de FFT y mini analizador de espectro gracias a una placa Arduino.



● Labs

- 12 ¿Qué se cuece en Elektor Labs?**
Clemens Valens nos cuenta lo que se destila, huele y se quema en los laboratorios de Elektor, incluyendo en su web.
- 40 ¿Qué se está cocinando?**
Un vistazo al prototipo de la nueva SMP-SU de Elektor.
- 42 ¡Pequeño pero matón!**
La historia de una capacidad perdida en un FET que hacía que un micro no se pudiera programar.
- 43 ¿¡Planos de masa perdidos!?**
Gracias a un despiste de uno de nuestros diseñadores de circuitos impresos aprendimos lo útil que es el PCB Visualizer, una herramienta para chequear las placas antes de producirlas.
- 56 Consejos sobre componentes**
Controlador de ventilador dependiente de la temperatura.

● Industria

- 8 Noticias y Nuevos Productos**
Un vistazo mensual a lo último en productos y componentes electrónicos.



● Tech the Future

- 64 El Internet de las Cosas (IoT)**
Tres expertos en la industria y la investigación electrónica charlan sobre los orígenes del IoT y su impacto en la electrónica de hoy en día. Serie editada por Tessel Renzenbrink.

● Magazine

- 76 Retrónica: El Grabador de Cinta Nagra IV**
El Nagra, uno de los grabadores de cinta portátiles más célebres del mundo, fue una vez un dispositivo profesional, desarrollado para aplicaciones móviles profesionales; para su uso en broadcast o en grabación sincronizada de audio; para películas y TV. Serie editada por Jan Buiting
- 80 Hexadoku**
Nuestro rompecabezas mensual con un toque electrónico.
- 86 El próximo mes en Elektor**
Un vistazo a los artículos seleccionados para la siguiente edición.

Volumen 33, Número 390, Diciembre 2012
Depósito Legal: GU.3-1980 31/12/2006
ISSN 0211-397X

Editor:
Elektor International Media Spain, s.l.
Jerez de los Caballeros, 2
28042 - Madrid, España
Teléfono: +34 91 101 9395
Fax: +34 91 101 9396
Internet: www.elektor.es

La revista está disponible en quioscos, librerías y tiendas minoristas de electrónica, o mediante suscripción.
Elektor se publica 10 veces al año con edición doble para Enero/Febrero y Julio/Agosto.

Suscripciones:
Elektor International Media Spain, s.l.
Apartado de Correos 62011
28042 - Madrid, España
Teléfono: +34 91 101 9395
Internet: www.elektor.es/miembros
Email: suscripciones@elektor.es

Oficinas Centrales:
Elektor International Media b.v.
P.O. Box 11 NL-6114-ZG Susteren
The Netherlands.
Teléfono: +31 (0)46 4389444,
Fax: (+31) 46 4370161

Distribución:
S.G.E.L.
Valdelaparra, 29
28108 - Alcobendas - Madrid - España
Teléfono: +34 91 657 6900

Publicidad:
Elektor International Media Spain, s.l.
Apartado de Correos 62011
28042 - Madrid, España
Teléfono: +34 91 101 9395
Fax: +34 91 101 9396
Internet: www.elektor.es
Email: publicidad@elektor.es
Tarifas y condiciones de publicidad disponibles
bajo petición.

Derechos de autor
Los circuitos descritos en esta revista son exclusivamente para uso doméstico. Los derechos de autor de todos los gráficos, fotografías, diseños de circuitos impresos, circuitos integrados programados, discos, CD-ROM's, portadores de software y los textos de los artículos publicados en nuestros libros y revistas (que no sean anuncios de terceros) están registrados por Elektor International Media BV y no pueden ser reproducidos o difundidos de ninguna forma ni por ningún medio, incluidas fotocopias, escaneos o grabaciones, parcial o totalmente sin la previa autorización escrita del Editor. También será preciso disponer del citado permiso antes de almacenar cualquier parte de esta publicación en sistemas de recuperación de cualquier naturaleza. Los circuitos, dispositivos, componentes, etc., descritos en esta revista pueden estar protegidos bajo patente. El Editor no acepta responsabilidad alguna en ausencia de identificación de la citada patente(s) u otra protección. La presentación de diseños o artículos implica que el Editor está autorizado a modificar los textos y los diseños presentados y a utilizar los contenidos en otras publicaciones y actividades de Elektor International Media. El Editor no garantiza la devolución del material a él enviado.

Renuncia
Los precios y descripciones de los productos relacionados con la publicación están sujetos a modificación. Excluidos errores y omisiones. Las opiniones expresadas a lo largo de los distintos artículos, así como el contenido de los mismos, son responsabilidad exclusiva de sus autores. Así mismo, el contenido de los mensajes publicitarios es responsabilidad de los anunciantes. Los precios y descripciones de los elementos relacionados con la publicación están sujetos a cambios. Están excluidos los errores u omisiones.

© Elektor International Media b.v. 2012
Printed in the Netherlands

¡Adiós al papel!

Si, es verdad ¡El día ha llegado! ¡Este es el último ejemplar de Elektor impreso en papel!

Han pasado 33 años desde que la primera edición en español de Elektor llegara a los quioscos. Empezaban los 80 y el papel era entonces el rey. Nadie se planteaba otro medio mejor para divulgar la información. La idea de una sociedad sostenible, donde los bosques sirvieran solo para regalarnos oxígeno y no para producir papel, estaba más cerca de las novelas de ciencia ficción que de la realidad cotidiana.

Hoy las cosas han cambiado y, en mi opinión, para bien. Ya no es preciso imprimir miles de ejemplares - para después tirar a la basura la mitad de ellos - ni esperar a que lleguen al quiosco o a tu casa. Ya no se perderán, ni se agotarán, ni tendrás que estar pendiente de la fecha en que sale a la venta para no perdértelo. Esa tecnología que tanto nos gusta, nos ofrece hoy los medios para llevar la información a cualquier punto del globo de forma inmediata, sin esperas, sin ataduras ni barreras físicas. ¡Te conectas a Internet y listo!

La era digital está aquí, y por mucho que cueste creerlo, relegará en muy poco tiempo al papel impreso a casos especiales, a ejemplares de colección y a ediciones de lujo. Quizá resistan un poco más algunas famosas revistas del corazón de enorme difusión, pero no pasará mucho tiempo antes de que encontremos tablets en lugar de revistas en las salas de espera o en las peluquerías.

Para aquellos que creéis que abandonar el papel es un error, debo confesaros que hasta hace unos meses tampoco yo me planteaba leer por placer nada que no estuviera impreso sobre papel; me resultaba extraño - incluso me incomodaba - pensar en leer una novela, revista o periódico en otro soporte. Pero el azar quiso que Papa Noël me trajera una 'tablet' las pasadas navidades. Desde el punto de vista tecnológico me atraía pero, la verdad, no lo veía muy útil. Hoy, diez meses después, he pasado todas mis suscripciones a formato digital, me he suscrito al periódico en formato digital y compro libros digitales. Van conmigo donde quiera que vaya y puedo leerlos en cualquier lugar - si, en cualquier lugar, incluso dónde estás pensando. ¡Probadlo, os engancharéis! Hoy decimos adiós al papel, pero solo a eso. Iniciamos una nueva etapa con la misma energía y los mismos objetivos que cuando nació Elektor, solo cambiamos el medio.

¡Cuento contigo, Elektorízate!
Eduardo Corral, Editor

El Equipo

Editor:	Eduardo Corral (e.corral@elektor.es)
Redacción Internacional:	Harry Baggen, Thijs Beckers, Jan Buiting, Wisse Hettinga, Denis Meyer, Jens Nickel, Clemens Valens
Equipo de diseño:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser
Diseño gráfico y preimpresión:	Giel Dols, David Márquez, Mart Schroijen
Director online:	Daniëlle Mertens
Director de marca:	Wisse Hettinga
Director general:	Don Akkermans



Spain

Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es



USA

Hugo Vanhaecke
+1 860-875-2199
h.vanhaecke@elektor.com



Germany

Ferdinand te Walvaart
+31 46 4389417
f.tewalvaart@elektor.de



France

Denis Meyer
+31 46 4389435
d.meyer@elektor.fr



Netherlands

Harry Baggen
+31 46 4389429
h.baggen@elektor.nl



United Kingdom

Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com



Italy

Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it



Sweden

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com



Brazil

João Martins
+55 11 4195 0363
joao.martins@editorialbolina.com



Portugal

João Martins
+351 21413-1600
joao.martins@editorialbolina.com



India

Sunil D. Malekar
+91 9833168815
ts@elektor.in



Russia

Nataliya Melnikova
+7 (965) 395 33 36
Elektor.Russia@gmail.com



Turkey

Zeynep Köksal
+90 532 277 48 26
zkoks@beti.com.tr



South Africa

Johan Dijk
+27 78 2330 694 / +31 6 109 31 926
j.dijk@elektor.com



China

Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

Nuestras redes



VOICE  COIL

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION



audio  PRESS



Te conectamos con



Empresas Colaboradoras



AudioXpress

www.cc-webshop.com57



Microchip

www.microchip.com 2



Design Spark

www.designspark.com17



Mikro Elektronika

www.mikroe.com 3



Eurocircuits

www.elektorpcbservice.com21



National Instruments

www.ni.com/es80



Hameg

www.hameg.com78



REICHEL

www.reichelt.es11

¿No eres patrocinador aún?

Ponte en contacto con nosotros en publicidad@elektor.es (+34 911019395)
para reservar tu propio espacio en la próxima edición de nuestra revista

Los nuevos SMSC JukeBlox® 3.1 SDK y CX875 Wi-Fi® Network Media Module de Microchip ofrecen la solución AirPlay® más sencilla y económica

Microchip anuncia la nueva generación de su plataforma de conectividad SMSC JukeBlox® Wi-Fi®, que incorpora el kit JukeBlox 3.1-AAP (JB3.1-AAP) SDK (Software Development Kit) y el nuevo módulo CX875 (Wi-Fi Network Media Module). Esta expansión de la plataforma aporta un software de conectividad de alta integración y especialización así como un módulo CX875 Wi-Fi de coste optimizado y totalmente certificado que cuenta con el nuevo procesador DM875 (Network Media Processor) de bajo coste y 8 MB de SDRAM para reducir en un 20% los costes de la lista de materiales. Gracias a esta reducción de costes y a su fácil utilización, esta plataforma JukeBlox de última generación permite disponer de una nueva categoría de sistemas inalámbricos de streaming de audio compatibles AirPlay® con un precio en el mercado a partir de 149 dólares.

Las mejoras introducidas en JB Connect de Microchip han simplificado la tecnología de configuración de la red Wi-Fi, junto con las sencillas funciones de configuración de la red que caracterizan a iOS de Apple, convierten a esta expansión de la tecnología JukeBlox en la solución de uso más sencillo y manejable para el diseño de productos AirPlay existente hasta la fecha. Su fácil configuración, junto con los avances en las prestaciones de Wi-Fi y los tiempos de arranque del sistema, hace que el streaming de Wi-Fi sea más robusto y mejora la experiencia del usuario en su conjunto. La plataforma JukeBlox de última generación de Microchip también proporciona un soporte integral a los nuevos iOS 6 e iPhone® 5, incluyendo sus características de streaming y control de música para AirPlay. El nuevo módulo CX875 (Wi-Fi Network Media Module) de Microchip se basa en su nuevo procesador de bajo coste SMSC DM875 (Wi-Fi Network Media Processor). Este procesador de triple núcleo integra un DSP y, con el nuevo software JB DSP 2.0, puede añadir las prestaciones del proceso de señal digital para mejorar la funcionalidad de audio. Algunos ejemplos son las mejoras en los bajos y las optimizaciones de la señal acústica integradas en el propio chip, que disminuyen los costes de la lista de materiales al reducir o eliminar la necesidad de un DSP por separado. Entre las

mejoras introducidas en JB Fast Boot se encuentran una reducción de los tiempos de arranque a partir del estado totalmente apagado (power-down) al estado "listo y conectado" en 5-10 segundos, liderando así el mercado de productos CE conectados a Wi-Fi. Esta solución con módulo certificado Wi-Fi completo permite desarrollar con un menor riesgo, facilita la fabricación y acelera el plazo de comercialización al combinar un subsistema Wi-Fi y RF totalmente integrado y numerosas precertificaciones estándar.

El software JB3.1-AAP SDK también se podrá ejecutar en los actuales módulos de la serie CX870, así como en el kit MK2 (Manufacturing Kit

2) anunciado previamente, una

plataforma de diseño de referencia de base

para altavoces Wi-Fi

totalmente integrada y de

bajo coste. El JB3.1-AAP SDK ofrece

mejoras en los interfaces de programación

de aplicación (application programming interfaces, API), así como herramientas que simplifican el desarrollo y la personalización de productos. Muchas de las nuevas características del nuevo JB3.1 se pueden introducir mediante cambios en el software, aprovechando así las inversiones del cliente en las plataformas de producto ya existentes. El SDK ofrece bibliotecas para las capas más altas de software, middleware para streaming de medios, acceso a contenidos, navegación y control del sistema, archivos de configuración para funciones de control remoto y mucho más.

La plataforma JukeBlox sigue ofreciendo una gama completa de codecs de audio, protocolos de radio Internet, conocidas aplicaciones de música, diversas opciones de conectividad y los principales estándares de interoperatividad. JukeBlox también incorpora actualizaciones de firmware a prueba de fallos que permiten a los usuarios finales la instalación de nuevas funciones a lo largo del tiempo y sin necesidad de cambiar el hardware. Las muestras del CX875 con JB3.1-AAP se encuentran disponibles de forma inmediata, mientras que los pedidos para producción se aceptan en diciembre de 2012.

www.microchip.com

XP Power introduce Fuentes de 400 y 600W para ambientes severos y aplicaciones con equipos herméticos y sellados, con un 90% de eficiencia y refrigeradas por conducción



XP Power, distribuido en España y Portugal por VENCO Electrónica, S.A., anuncia los modelos de fuente de alimentación CCH400 y la CCH600. Una serie de fuentes AC/DC de 400W y 600W, refrigeradas por conducción, con salidas de tensión simples, ideales para ambientes severos. Con su alta eficiencia de hasta el 90%, las series CCH generan menos calor que las

se encuentran en el mercado actualmente, sobretodo muy necesario en aplicaciones críticas con equipos herméticos y sellados.

Se ha diseñado de tal forma, que todos los componentes de la fuente están en contacto con la base metálica, permitiendo que el calor se disipe por el chasis del equipo o a través de los radiadores del mismo. No se necesita pues ventilación forzada.

Midiendo solo 214 x 102 x 43 mm (8.43 x 4.02 x 1.69 inches), las series compactas CCH son hasta un 50% mas pequeñas que otros modelos de potencias similares del mercado, dejando mas espacio libre en el equipo del cliente, o pudiéndose diseñar equipos mas compactos y pequeños.

Con su alta eficiencia, el radiador a utilizar puede ser mas pequeño, reduciendo también el tamaño del footprint.

Ambas series 400 W CCH400 y la 600 W CCH600 tienen un rango Universal de entrada de 90 a 264 Vac, sin reducción de potencia, haciendo este modelo adecuado en cualquier diseño/equipo de cualquier país del mundo. Las salidas simples disponibles son +12, +24, +28 or +48 Vdc.

Esta familia cumple con la MIL-STD-461 para emisiones y la STD-810 para vibraciones/choques, por tanto hacen esta fuente recomendada para aplicaciones militares COTS, como también para aplicaciones Industriales y equipos comerciales utilizados en ambientes externos y severos.

También cumplen con la normativa standard EN/UL

60950-1 para equipos TI y cumple con la EN55022 level B para conducidas, y level A para emisiones radiadas. Incluyen una serie de señales de control como remote On/Off, remote sense, AC OK y alertas por sobrettemperatura y paro. Y dispone de una tensión extra de standby de +5 Vdc. Permite compartir corriente para paralelar con otras unidades y compartir con la carga hasta 3 unidades CCH. Estas series CCH tienen un rango amplio de temperatura de -40 a +70°C, con una temperatura máxima en la base de +85°C.

Las Fuentes CCH están disponibles en el distribuidor local especializado VENCO Electrónica S.A. en España y Portugal, como también en Farnell, element14, y tienen una garantía de 3 años.

www.xppower.com

Publicidad

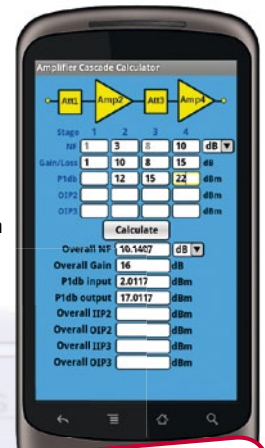
Elektor RF & Microwave Toolbox

➡ para Android

La RF & Microwave Toolbox contiene 43 herramientas de cálculo y conversión para RF, microondas y electrónica en general. Ya seas profesional, radioaficionado, astrónomo o aficionado, esta app pone la mayoría de las herramientas importantes al alcance de tus dedos.

Destacados:

- Amplificador en Cascada (NF, Gain, P1db, OIP2, OIP3)
- Convertidor de intensidad de campo y densidad de potencia (W/m², V/m, A/m, Tesla, Gauss, dBm, W)
- Calculadora de trazado de PCB (impedancia/dimensiones)
- Atenuador PI y T
- Temperatura de Antena (Kelvin)
- EMC (EIRP, ERP, dBμV/m)
- Diseño de filtros (Butterworth, Chebyshev, prototipo)
- Y más



¡Descarga tu app ahora!



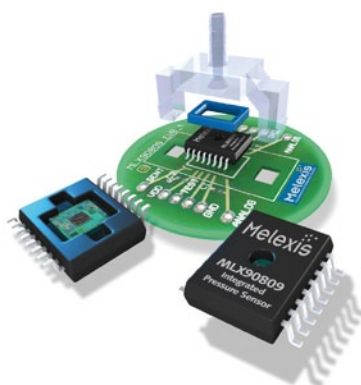
Más información en

www.elektor.es/rf-app

Los robustos sensores de presión MEMS de Melexis ofrecen alta precisión y sencillo uso gracias a su innovadora tecnología

Melexis presenta el primer modelo de su nueva línea de sensores de presión MEMS encapsulados para el segmento comercial. El MLX90809 aprovecha los más de 10 años de experiencia en el desarrollo de sensores de presión a medida para exigentes aplicaciones en el automóvil. Este dispositivo es un sensor de presión muy preciso, homologado según AEC Q100 y especialmente dirigido a aplicaciones de 1 Bar. La tecnología MEMS piezorresistiva de la propia compañía, junto a un elemento de sensado de alta precisión, su etapa de entrada analógica de bajo ruido y un convertidor A/D sigma-delta de 16 bit, ofrecen un nivel superior de prestaciones. La sofisticada cadena analógica proporciona la amplificación y la compensación de offset necesarios para el elemento de sensado. La unidad de microcontrolador de 16 bit incorporada se encarga de la compensación de temperatura y aporta los mecanismos de diagnóstico que exigen las aplicaciones críticas desde el punto de vista de la seguridad.

La memoria EEPROM totalmente programable que integra ofrece soporte a diferentes configuraciones, como el ajuste independiente de funciones de



diagnóstico en caso de sobretensión o subtensión de alimentación así como en caso de sobrepresión o subpresión, además de ajustes de filtros digitales seleccionables para reducir aún más el ruido de salida o bien para disminuir el tiempo de respuesta del sensor. También permite almacenar los datos de compensación del sensor y la identidad de la unidad.

“Dado que la tecnología MEMS utilizada es totalmente compatible con los procesos CMOS estándar, hemos podido fabricar un sensor completamente monolítico en el cual el elemento de sensado y su circuitería de acondicionamiento de señal se procesan en la

misma oblea”, explica Laurent Otte, Director de Marketing de Producto para Sensores de Presión en Melexis. “Esto concede al dispositivo una enorme ventaja respecto a los otros sensores de presión no integrados, cuyo acondicionamiento de señal se encuentra normalmente a varios milímetros del elemento de sensado y por tanto ofrece una integridad de señal inferior y una susceptibilidad mucho mayor a la interferencia electromagnética (EMI). Ello nos permite asimismo reducir la superficie de la pastilla de semiconductor a menos de la mitad de que nuestra generación anterior de sensores de presión integrados basados en MEMS”. El dispositivo sensor MLX90809, que se suministra en un encapsulado muy robusto para montaje superficial de 16 patillas moldeado en plástico, puede entregar los datos de presión por medio de una tensión analógica de salida proporcional a la tensión de alimentación o bien utilizando el protocolo digital SENT. Su rango de temperaturas de funcionamiento es de -40 °C a +150 °C, por lo que se puede adaptar a los entornos más adversos del automóvil.

www.melexis.com

TSL4531 – Sensor de luminosidad con visión fotópica y rechazo de infrarrojos

El nuevo TSL4531 de AMS es mucho más que un sensor de luminosidad porque su visión fotópica le permite evaluar exactamente la misma sensibilidad que el ojo humano. Su capacidad de rechazo de infrarrojos permite aplicarlo sin necesidad de ningún filtro externo. Tampoco necesita cualquier proceso de calibrado.

El TSL4531 es un verdadero sensor de luz ambiente por las siguientes características:

- Visión fotópica con rechazo de infrarrojos

- Mide la luminosidad sin saturación hasta 220kLux
- Sofisticado filtrado de ruidos y de parpadeo AC
- Salida digital por bus I2C con lectura directa en Lux
- Diferencia entre luz solar, incandescente, fluorescente o de LED

Las aplicaciones de este pequeño sensor con tan solo 2x2mm incluyen: sistemas de alumbrado por LEDs, sistemas de interfaz usuario por pantalla TFT, OLED, LED, etc. , y también apli-



caciones que necesitan medir la luz ambiente tal como percibida por el ojo humano, para equilibrar alumbrados y ahorrar energía.

www.ams.com



- ✓ Más de 40 años de experiencia
- ✓ Más de 40.000 productos en almacén
- ✓ Sin suplemento por cantidad mínima
- ✓ Envío rápido en 24h

+49 (0)4422 955-333

+49 (0)4422 955-360

¡Calidad profesional a precios de descuento!

Tecnología de la carcasa

Técnica de medición



Carcasa del sistema

- para acoger tarjetas de circuito impreso en formato europeo
 - 3 unidades de altura: 112,8 mm de profundidad 236 mm
- Abertura de 10": 203,8 mm
Abertura de 19": 427,8 mm

GEH SG 1-10 **46,65**
GEH SG 1-19 **66,50**

Serie Hammond 1550W, IP66

- Carcasa de aluminio con diseño robusto
- Cinta de espuma autoadhesiva, 4 tornillos de acero inoxidable



		L x A x A (mm)
1550WQ	4,10	60 x 55 x 30
1550WA	4,10	89 x 35 x 30
1550WB	5,40	115 x 64 x 30
1550WD	6,10	115 x 64 x 55
1550WC	6,95	115 x 90 x 55
1550WE	10,65	171 x 121 x 51

Funda de la carcasa de gran calidad

- protegida contra el polvo
- protegida contra chorros de agua



BOPLA ET-205	4,80	52 x 50 x 37
BOPLA ET-206	7,15	65 x 50 x 37
BOPLA ET-208	8,95	98 x 64 x 38

Más selecciones: <http://rch.it/60>

Carcasa pequeña de plástico

- Carcasa estándar
- 6 boquillas de fijación para platinas



GEH KS 21	0,98	72 x 50 x 21
GEH KS 28	0,98	72 x 50 x 28
GEH KS 35	1,10	72 x 50 x 35
GEH KS 42	1,10	72 x 50 x 42



Artículos de taller

Conector enchufable

Suministro eléctrico



NEUTRIK NE-8FDP

9,65



Casquillo de brida

- Ethernet
- RJ45 en RJ45

Conector RCA estándar

- con protección antiflexiones

CSP RT	rojo
CSP GE	amarillo
CSP WS	blanco

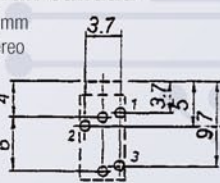
por cada **0,09**



Todos los conectores RCA <http://rch.it/6P>

Manguitos tiradores de montaje de gran calidad

- 2,5 mm
- estéreo



LUM 1501-03 **0,69**

El surtido Lumberg <http://rch.it/6Q>

Manguitos de montaje profesionales ajustables

- 6,35 mm
- mono/estéreo



NEUTRIK NJ-3FP **5,50** Carcasa de metal
NEUTRIK NJ-3FPB **5,95** Carcasa negra cromada

Tecnología de red y de PC

Herramientas de desarrollo

Tecnología satélite y de TV



ARDUINO UNO **27,95**

Panel de microcontrolador

Plataforma Prototipo Open Source basada en Atmel ATmega328

- 14 interfaces I/O digitales, 6 entradas analógicas
- Oscilador de cuarzo de 16 MHz, encabezamiento ICSP, reinicio



Paneles de microcontrolador



ARDUINO MEGA **49,50**



WizNet W5100 Ethernet

- basado en ATmega 328
- 13 pines digitales I/O (4 sirven para la salida PWM)

ARDUINO ETHERNET

49,50

Carcasa Raspberry Pi profesional

- diseño práctico — montaje perfecto
- Conexión intuitiva de los cables
- Señal de estado
- 4 pines de carcasa
- Tamaño de la platina: de hasta 85 x 56 mm

Suministro sin platina

TEK-BERRY

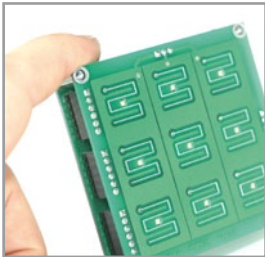
4,95



Para consumidores: son válidas las normativas de revocación legales. Todos los precios se indican en € con el IVA legal incluido, a partir del almacén de Sande, más gastos de envío para toda la cesta de la compra. Son válidas exclusivamente nuestras condiciones generales (en www.reichelt.de/agb, en el catálogo o a petición). Salvo venta. Todos los nombres de productos y logos son propiedad del fabricante correspondiente. De manera similar las ilustraciones. Quedan reservados los fallos de impresión, errores y cambios en los precios. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (HRA 200654 Oldenburg)

Pago internacional a través de
PayPal

Clemens Valens
(Elektor.Labs)



No os lo perdáis

Cuando era estudiante, los profesores a menudo decían que era duro mantenerse al día de los avances de la tecnología. Yo pensaba "¿de verdad lo es?", ya que yo no encontraba ninguna dificultad en consumir toda la información que encontraba. Pero ahora que ya estoy perdiendo el último pelo de mi cabeza y que mis músculos se están haciendo débiles, debo admitir que no es una tarea fácil el saber constantemente lo que está pasando en cada rincón del mundo de la tecnología. Afortunadamente, no estoy solo y alguna gente está incluso peor que yo. El otro día compartí un paseo en coche con un colega incluso más viejo que yo. Como él era el que conducía, yo iba mirando las señales del camino y, cuando ya estábamos cerca de nuestro destino, puse delante de mí el teléfono para mirar el Google Maps. El móvil estaba "dormido" por lo que tuve que activarlo primero y, para ello, tengo que deslizar mi dedo sobre la pantalla dibujando un patrón concreto como firma. A esto se le denomina "llave patrón". Mi colega me vio hacer esto y estaba tan sorprendido que casi nos damos con otro coche. De nuevo en el camino, me pregunto si lo que acababa

de hacer era lo último en 'gadgets' ('artilugios'). Le dije que no lo era y que sabía de gente que lo llevaba usando durante años. Le dije que, realmente, tenemos un proyecto en la página web .LABS que muestra cómo construir nuestra propia "llave patrón". Mi chofer se quedó profundamente impresionado. Aunque divertido, justo tres días después recibí un mensaje de la persona que estaba haciendo este proyecto en .LABS, anunciando que ya estaba listo. Incluso habían enviado un video promocional ¿que tuve la suerte de ver? No había estado en la página de su proyecto hacía algún tiempo por lo que lo busqué de nuevo y ahora fui yo el que quedó profundamente impresionado. Un reportaje claro y agradable, fotografías de alta calidad, video excelente, exactamente como nos gusta. Su presentación hizo que quisiese inmediatamente construir una por mí mismo. No os sorprenderá leer más sobre este proyecto, próximamente, en la revista impresa.

Por supuesto, sería estupendo si todos los proyectos en .LABS estuviesen tan bien presentados pero, en ningún caso, es algo obligatorio. Si no disponéis del equipo necesario para realizar fotos con calidad de estudio, podéis usar vuestro teléfono móvil. Si no tenéis el tiempo para dibujar un esquema eléctrico en un paquete CAD profesional, sencillamente hacéis un borrador en un trozo de papel y le hacéis una copia escaneada. La única cosa en la que tenéis que preocuparos es en tomaros el tiempo necesario para enviarnos proyectos claros y grandes ideas. Daos a conocer y os buscaremos.

Echad una ojeada al proyecto de llave patrón y construir el vuestro propio.

www.elektor-projects.com/project/android-style-capacitive-sensing-pattern-lock.12407.html

El origen de las especies

Me encanta Wikipedia, ¿a vosotros no? Mirad lo que descubrí hace algún tiempo:

"Psocoptera son un tipo de insectos que se conoce comúnmente como psocópteros o psocódeos o piojos de los libros. Aparecieron por primera vez en el período Primario, hace entre 295 y 248 millones de años. Muchas de estas especies sólo han sido descritas en los últimos años.

El suborden más grande es el de la Psocomorfa, con unas 3600 especies en 24 familias clasificadas desde la subespecie más pequeña, la Bryopsocidae (2 especies), hasta la numerosa (con unas 900 especies). La Psocomorfa es conocida por tener una antena con 13 segmentos. Esta especie tiene dos (o tres) tarsos segmentados. Esta condición es constante (por ejemplo, en la Psocidae) o variable (por ejemplo, en la Pseudocaeciliidae) dentro de las familias."

Si, al igual que yo, tampoco entendéis la mayor parte de esto, estad seguros que se está hablando de bichos. Así pues, ¿por qué lo mencionamos aquí? Porque recientemente la subespecie de la Psocomorfa se ha ampliado con una nueva especie, la más ruidosa, el sintetizador de música PSoCaMorph. Primero lo descubrimos en .LABS y creo que ha sido muy divertido. El objetivo del proyecto es diseñar un sintetizador de música analógico con apariencia antigua, usando tecnología actual de Sistemas Programables en Chip (Programmable System-on-Chip o PSoC). Como probablemente habéis adivi-

nado desde la primera letra de este artículo, las familias de chinches son muy modulares; las especies tienen alas, patas y segmentos de antenas, junto

con otras partes del cuerpo. Los sintetizadores de música analógicos son también modulares. Están constituidos de uno o más módulos VCO, VCF y VCA y de algunas otras partes. Los insectos pueden estar recogidos por unas membranas especiales al igual que los sintetizadores analógicos tienen generadores envolventes. Los circuitos integrados, al igual que las chinches, son pequeños y modestos pero tienen muchas patas. La Psocoptera apareció hace unos 250 millones de años, de igual forma que los sintetizadores analógicos son también bastante antiguos. Así pues, un sintetizador modular en un circuito integrado es casi una chinche, ¿de acuerdo?

En cualquier caso, lo que quiero decir es que vosotros también podríais entrar en la 'entomología' usando el proyecto PSoCaMorph de .LABS. Si lo hacéis, podéis ganar incluso un Kit de Desarrollo PSoC 5, todo lo que tenéis que hacer es contribuir.

El PSoCaMorph vive aquí.

<http://www.elektor-projects.com/project/the-psocamorph.12269.html>





La elección del Editor

Nuestro editor ha seleccionado un cierto número de proyectos para .LABS que deben ser publicados en un futuro próximo. Lamentablemente, para algunos de estos proyectos nos hemos encontrado que el remitente original (OP, 'original poster') no ha respondido a nuestros mensajes. Así pues, si enviáis un proyecto, por favor, comprobad de manera regular la cuenta de correo electrónico que habéis facilitado a .LABS. No podremos publicarlos si no llegamos a ponernos en contacto con vosotros.

A continuación mostramos una selección de proyectos que hemos pensado que sería interesante es el que nos gustaría publicar en la revista impresa:

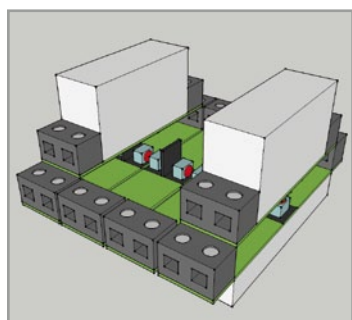
Grabador de datos de contador Geiger con interfaz WLAN

El OP Marcus Vohburger está trabajando en un sistema completo para medir y monitorizar la radiactividad. Según él, construir un contador Geiger es en realidad bastante sencillo. Por eso le ha añadido algunos elementos extra.

www.elektor-projects.com/project/geiger-counter-data-logger-with-wlan-interface.12264.html



elektor **e** labs



El caso 'Tensión de red': Relé Programable y Monitor de Energía

El OP John Hind escribe sobre este proyecto: más que un escándalo político de US, ésta es una idea para "desmenuzar" cinco de los circuitos del número doble de julio y agosto de 2012, y crear un circuito "filtro de la tensión de red" realmente flexible, combinando una multitud de controles, medidas, protecciones y funciones de registro con un coste mínimo.

www.elektor-projects.com/project/mains-gate-programmable-relay-energy-monitor.12461.html

Sencillo Instrumento de Viento MIDI

Este proyecto está basado en experimentos anteriores hechos por el OP que acabó en un sistema demasiado complejo. Si publicamos este proyecto sería una versión simplificada y modernizada. Podríamos hacerlo con algo de ayuda que nuestros lectores nos haría llegar sobre un instrumento de viento que corta el aire hacia sensor de presión.

www.elektor-projects.com/project/simple-midi-wind-instrument.12277.html



Instrumento de Dirección y Velocidad del Viento

Alojado a forma de bocadillo entre dos condensadores de gas y montado en una pequeña PCB con sólo cuatro componentes: un transistor específico de sobrecalentamiento y tres resistencias NTC. El viento soplará la columna de aire caliente producida por el transistor fuera del centro. Esto cambiará la información capturada por las resistencias NTC. Claro, ¿no creéis?

www.elektor-projects.com/project/wind-direction-and-speed-instrument.12480.html

El Mundo de Elektor

Jan Buiting, Editor UK/US

Medidor de Radiación Mejorado de Elektor en ConSat-1... dentro de poco

Hola. Soy Michelle Boyce, represento a Space-Concordia.Ca. Recientemente hemos construido un Cube Satellite (ConSat-1) para el 'Desafío de Diseño de Satélite Canadiense (Canadian Satellite Design Challenge).

En su interior, una de las cargas es un Detector de Radiación de Estado Sólido basado en artículos de:

- Mide Rayos Gamma Rays con un Detector de Radiación por Fotodiodo que usar un BPW34, hecho por Burkhard Kainka, Junio de 2011.
- Medidor de Radiación Mejorado para radiación Alfa, Beta y Gamma, hecho por Burkhard Kainka, Noviembre de 2011.

El proyecto era parte de un Programa de Participación Educativa con Space Concordia.

Saludos cordiales de Michelle Boyce (Canadá).

Nuestro proyecto Medidor de Radiación fue un éxito desde el primer día. Mirando este mensaje desde Canadá los miembros de la plantilla de PR de Elektor realizar una búsqueda global de la expresión "cobertura global" en todos los documentos de la empresa y la sustituyeron por "cobertura global y *más allá*". Mientras tanto, se le ha pedido a Michelle que escriba un artículo para Elektor.

Medidor de Radiación Mejorado de Elektor de la Universidad de Oxford... dentro de poco 2

Hola Thijs, trabajo en el departamento de Física de la Universidad de Oxford. Hemos construido un circuito similar al vuestro para montarlo sobre un globo de helio y poder medir los rayos cósmicos. Compramos el circuito detector de radiación (hemos intentado con varios kits diferentes) con la esperanza de probarlo en nuestro laboratorio. Tenemos acceso a muchas fuentes diferentes pero no hemos conseguido que vuestro kit funcione. Creo que lo he montado y apantallado correctamente pero no pudo encontrar lo que funciona mal dentro de él. Actualmente se dispara con el comparador configurado a L68 (antes no se disparaba). La señal en K4 es de unos 500 mV(pp) en el osciloscopio.

Ahora tenemos poco tiempo pero, personalmente, me gustaría ver que este kit funciona. Para probar este circuito con nuestras fuentes, ¿sería posible que nos prestaseis un circuito que funcione (si disponéis de uno)? O, ¿podríamos enviaros el que hemos construido (está en una caja de metal)?

Jeff Lidgard (UK)

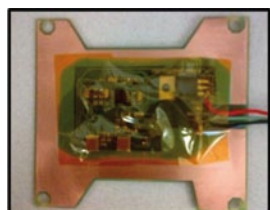
¿Qué?, ¿OU y 'no funciona'? Un prototipo funcionando está en camino para ti, Jeff.

El Formant Monster de Elektor triunfa en YouTube

Siegfried Brückner probablemente tenga la constelación más grande del mundo de módulos Formant de Elektor, en el planeta y en el espacio. Estimamos que el instrumento tiene, aproxima-



damente, 3 metros de ancho por 1,20 de alto. Está conectado a un secuenciador, a un pedal de bajos y a un teclado. Un total de, aproximadamente, 110 módulos Formant, 500 mandos, y docenas de latiguillos rojos producen este inimitable sonido del sintetizador Formant [1]. Siegfried nos escribió: "La fase #3 del proyecto comprende más de 160 módulos de los libros 1 & 2 Formant de Elektor, así como circuitos especiales y discretos. Todo está conectado por



Perdido en el Laberinto Electrónico

Crear un puzzle que tenga una única solución es tan duro como resolverlo. Ver la página 78 de Elektor en la edición de julio y agosto de 2012 sobre el lector Klavs Rommedahl y el creador del puzzle Sadettin Commert.

Hola amigos de Elektor,
No estoy triste ni cabreado por no haber ganado, pero si totalmente perdido con la respuesta dada como Entrada 2.

Es posible, pero con los pequeños conocimientos de mi cerebro, me he dado cuenta que ¿la Entrada 1 también sería una posible solución? Esta parte la he tomado de Wikipedia y espero que estemos de acuerdo en el hecho de que un zéner es un diodo normal en la dirección de la flecha :-)?

La foto muestra un dibujo a mano indicando mi solución de Entrada 1. ¿Sería posible que hicierais un rápido dibujo a mano de la solución de la Entrada 2 y verificar si estoy en lo cierto o no? Y si es que no, ¿podrías indicarme donde me he perdido?

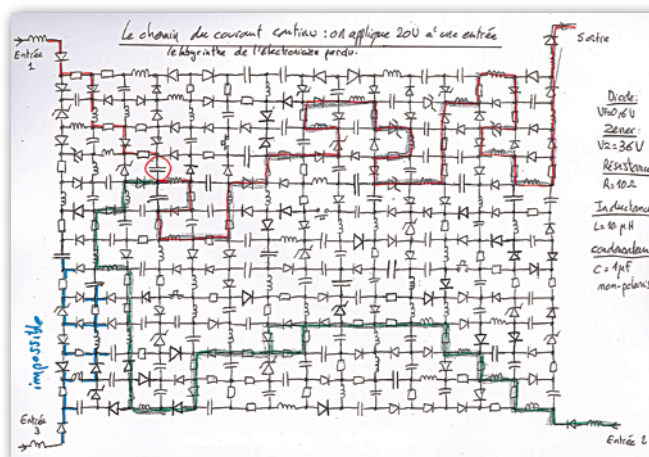
Sinceramente, Klavs Rommedahl (Dinamarca)

(Traducido del francés)

Queridos Jan y Klavs

Lo que ha pasado es que la versión impresa del laberinto era una primera versión que contenía un error. Os adjuntamos el fichero pdf que contiene la versión correcta. ¡Una resistencia debería haber sido un condensador! Así pues, el dibujo erróneo permitía dos soluciones posibles.

KR, Sadettin Commert (Francia)



un módulo de pared. Se ha incorporado un interfaz MIDI de 8 vías, haciendo que el Formant se controle sobre MIDI y CV/Gate.

El nuevo teclado de Formant tiene ahora más teclas, con una 4a y 5ª octavas y, además, un pedal de 13 llaves. Cada teclado / pedal tiene un interfaz C/V.

Para ser capaces de tratar otros sonidos en el Formant, se han integrado unos interfaces MIDI adicionales. Esto permite que los teclados puedan tocar en mono así como de forma polifónica (con extensor). Además, los sonidos del extensor pueden ser modificados como se requiera. En abril de 2012 fui capaz, por primera vez, de mostrar mi Monster Formant a los amigos especialistas en el evento Happy Knobbing, en Rheinland Pfalz, Alemania, cuando el tema del año era: generación de sonidos usando un sistema modular [2]. Por desgracia, el sonido grabado estaba por debajo del estándar. Mi siguiente presentación será en Munich, el 17 de noviembre." El propio Formant de Elektor fue rescatado de un contenedor y en ocasiones se toca en el Sótano Elektor [3]. ¡Auriculares encendidos!

[1] http://youtu.be/_5cRr7og4Vc

[2] <http://youtu.be/ynQyGT6t9tU>

[3] <http://youtu.be/JOhxLE3-ecI>

 Fiesta del 25 (19_{hex}) Aniversario de Circuit Cellar



versaciones, no sólo con miembros actuales de la plantilla de CC, sino que también lo hizo con muchos de los renombrados columnistas y editores de proyectos como Ken Davidson, Dave Tweed, Robert Lacoste, Ed Nisley, y Jeff Bachiochi. Con los CEOs, SFOs, etc... también pudo escuchar conversaciones que pronto derivaron a temas electrónicos y embebidos. Queremos dar las gracias más sinceras a los anfitriones de la fiesta, Steve Ciarcia y Hugo Van Haecke

 Recital de RF

Mi agradecimiento al equipo de la Granja de Robert Frost, Derry, NH, por premiarme con un lápiz por citar sólo una línea de poesía de RF: "Se podría hacer peor que ser un libertino de abedules". De vuelta a mi despacho en la central de Elektor, ya uso el lápiz para garabatear en las correcciones y para dibujar esquemas eléctricos de circuitos cuando nuestros servidores principales se han caído.

Los días 28 y 29 de septiembre de 2012 marcaron el 25 aniversario de la formidable revista hermana de Elektor, Circuit Cellar (CC). Nuestro editor tuvo la fortuna de ser invitado a la cena en New Castle NH, donde hubo discursos y con-

Publicidad

Todo un laboratorio de electrónica en un solo dispositivo USB



Con NI myDAQ, tu ordenador se convierte en el instrumento. Al instalar el controlador de hardware NI ELVISmx dispondrás de: Multímetro Digital, Osciloscopio, Generador de Funciones, Analizador Bode, Analizador de Señales Dinámicas, Generador de Forma de Onda Arbitraria, Generador de Salidas Digitales y Lector de Señales Digitales.

Ocho instrumentos de ingeniería para trabajar como y donde tu quieras.

¡NI myDAQ por solo 149 Euros!
(incluyendo licencias de estudiante de LabVIEW y Multisim)

Contenido del paquete:

Dispositivo NI myDAQ, DVD de instalación que contiene NI ELVISmx y NI DAQmx, así como las versiones para estudiante de LabVIEW y Multisim con sus licencias correspondientes, Cable USB, Puntas de prueba para el multímetro, Cable de audio (3,5 mm), Conector de tornillos con destornillador NI, Tarjeta informativa de iniciación, Tarjeta informativa de programas de ejemplo de TI, Bandeja de almacenamiento reutilizable.



Más información y pedidos en: www.elektor.es/myDAQ

¡Atención! Para poder adquirir el NI myDAQ deberás acreditar tu condición de estudiante.





DESIGNSPARK

NEW. FREE. **DESIGNSPARK PCB** **VERSION 4**

COMPREHENSIVE NEW LIBRARIES.
EASIER COMPONENT SELECTION & QUOTATION.
NEW PCB PROTOTYPING SUPPORT.

Discover today at
www.designspark.com

UNIQUE
RESOURCES BY 

Árbol de Navidad con LED activo

Con un amplio espectáculo de luces y posibilidades de interconexión

Willem Tak
(Holanda)



Hoy en día no se conciben los artículos de Navidad sin electrónica. Antiguamente utilizábamos una cadena de bombillas incandescentes que parpadeaban como mucho pero ahora podemos elegir entre muchos tipos de iluminación LED con efectos lumínicos incorporados. Pero un verdadero electrónico puede hacer esto mejor y con más posibilidades, como es este árbol de Navidad que con sus 56 LED ofrece un amplio espectáculo de luces de más de un cuarto de hora.

Se acerca la época en que hay que bajar del desván los adornos de Navidad e ir a por un árbol de Navidad (real o de imitación). Afortunadamente el árbol se viste muy pocas veces con velas de verdad, hace mucho fueron sustituidas por mini bombillas incandescentes o incluso LED. Esto mismo sucede con el resto de adornos de Navidad, que llevan dentro cada vez más LED controlados por medio de un circuito electrónico. El autor del árbol de Navidad electrónico que describimos aquí quiso construir un mini árbol de Navidad que fuese un poco más exclusivo que los productos estándares de los hipermercados. Esto dio como resultado un circuito que

sólo consta de un microcontrolador y 56 diodos luminosos. Los LED se encuentran en una placa en forma del clásico árbol de Navidad, tal y como se puede observar en la foto de cabecera. Como es habitual en estos circuitos, ¡La inteligencia está en el software! Este se encarga de que el microcontrolador encienda el árbol de Navidad en los patrones más interesantes (ver el video en [1]). Incluso es posible interconectar dos de estos árboles, para que los patrones generados puedan ir moviéndose entre ellos. El resultado es un señuelo asombrosamente efectivo si se coloca en el alfeizar de la ventana. Y no es sólo varios efectos simples todo lo que hay - la dura-

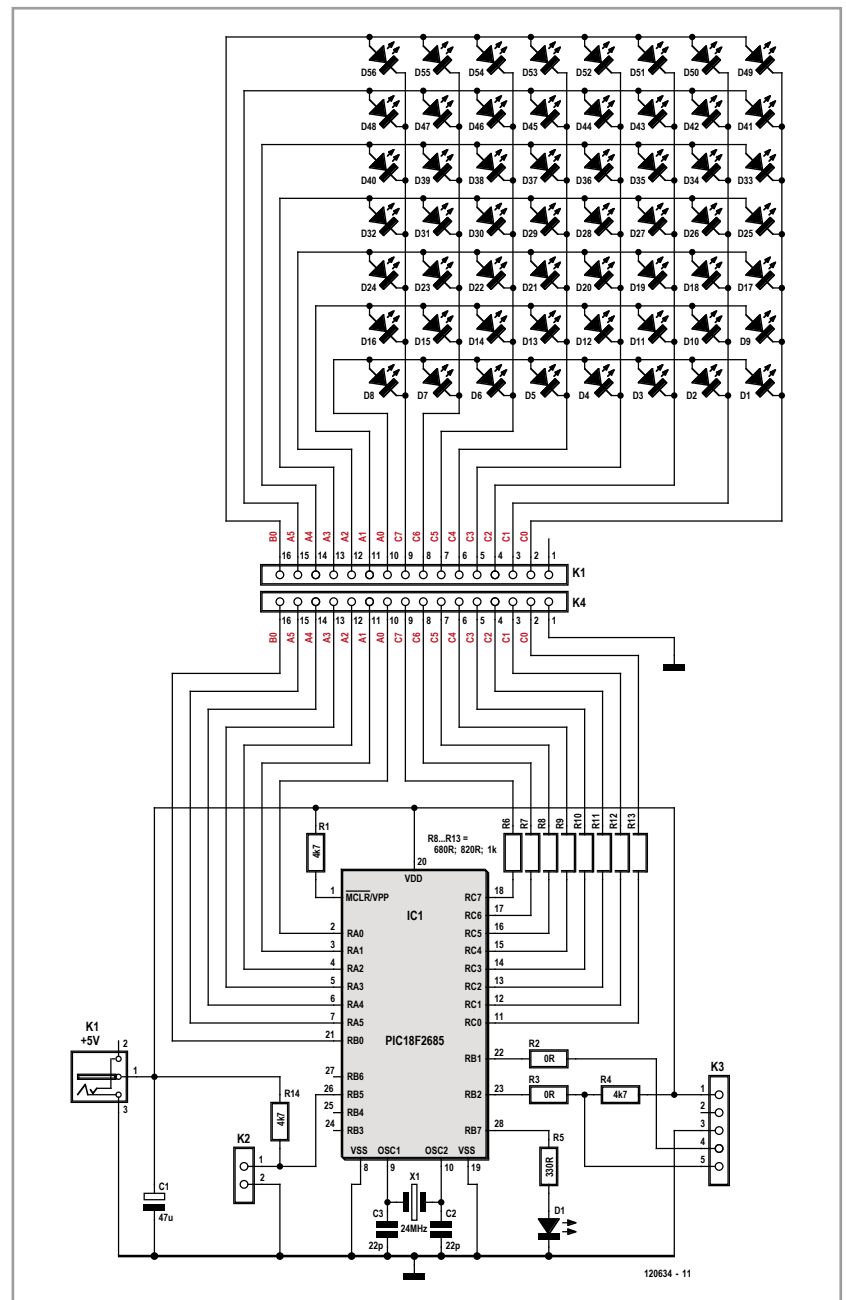
ción del ciclo antes de que empiece de nuevo todo el espectáculo, iasciende nada menos que a 40 minutos!

Poca electrónica, mucha memoria

Como aquí los efectos ópticos juegan un papel importante y la intención no es que aparezcan los mismos patrones al cabo de poco tiempo, el autor optó por un microcontrolador con una gran cantidad de memoria. La elección recayó sobre el tipo 18F2685-E/SP de Microchip, un microcontrolador equipado con 96 KB de memoria flash (por lo tanto 96 KB de espacio para programación). Dicha memoria está casi llena con el programa actual, por ahí también esa duración tan larga de más de 50 minutos. Un cristal de cuarzo externo corriente de 24 MHz sirve de reloj para el microcontrolador (resultó que el reloj interno de 8 MHz no era lo suficientemente rápido para algunos patrones).

El propio circuito (**figura 1**) es muy sencillo: El micro controla los LED directamente en una matriz de 8x7. Las 7 columnas de cada 8 LED se conectan alternativamente con la tensión de alimentación positiva mediante los terminales A0...A5 y B0. Además las salidas del puerto C conectan las filas de cada 7 LED con la masa a través de resistencias. Así que los niveles en las salidas C determinan que LED se encienden de una columna, mientras que las salidas A y B se encargan de la multiplexación.

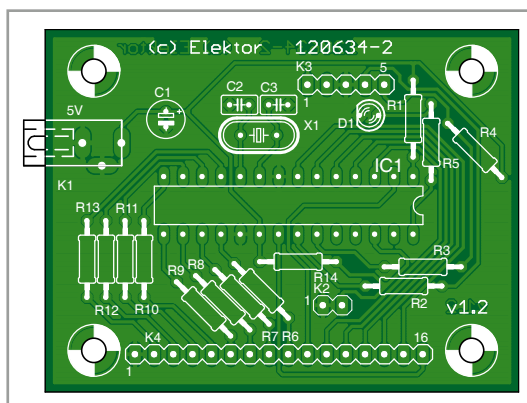
Debido a la configuración matricial (cada LED se enciende sólo 1/7 parte del tiempo) y las corrientes limitadas que puede proveer el microcontrolador, los LED activados no se iluminan con mucha claridad, pero el propósito del circuito no es para ser utilizado en un entorno con demasiada luz ambiental (por ejemplo, un salón iluminado de forma agradable). El autor utilizó LED que iluminan a su alrededor de la compañía SBL [2] y consiguió así un resultado aceptable. Si no se utilizan LED normales, sino LED brillantes especialmente con un alto rendimiento óptico, se obtienen buenos resultados. Las salidas del controlador soportan un máximo de 25 mA cada una. Esto significa que cada LED recibe unos 3 mA debido a la multiplexación; incluso en la oscuridad podría ser demasiado. Las ocho resistencias de los terminales del puerto C determinan la corriente de cada LED. Dependiendo de la claridad deseada se puede optar por valores de 680 Ω , 820 Ω ó 1 k Ω en las resistencias. A través de la conexión sync (terminales 22 y 23,



interconectados a K3) se pueden interconectar dos árboles de Navidad. Determinados patrones saltan de un árbol a otro. Para conseguir este efecto los árboles se interconectan a través de los terminales 22 y 23 de forma cruzada. El software determina si un árbol es el ejemplar de la derecha o de la izquierda. La versión I simboliza el árbol de la izquierda y la R el de la derecha. Los efectos se escribieron de tal manera que el árbol de la izquierda también se encuentra a la izquierda visto de frente.

Figura 1.
Casi no puede ser más sencillo: un microcontrolador controla directamente 56 LED puestos en una matriz. La mayor parte del trabajo está en un extenso programa que se encarga del control.

Figura 2.
La placa del microcontrolador con algunos componentes pasivos. Un electrónico hábil también puede construirlo en un trozo de placa de pruebas.



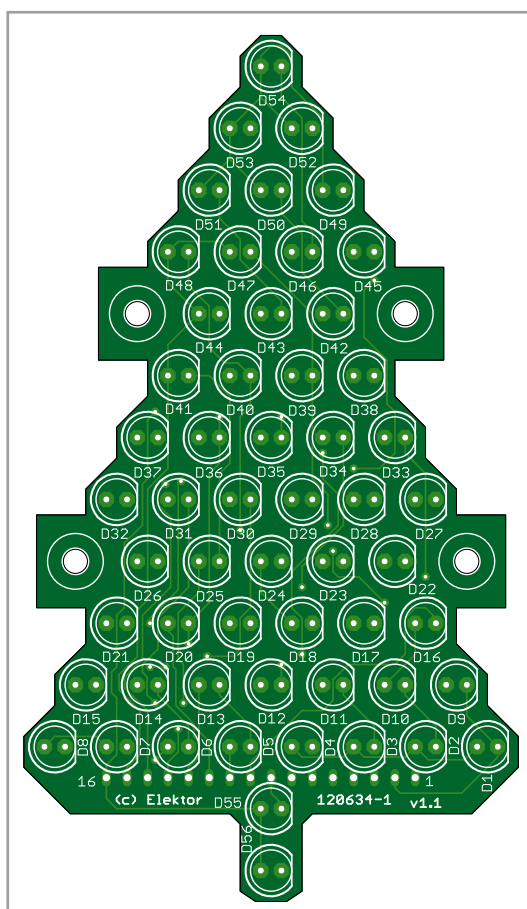
No es necesario que el LED de status esté conectado al terminal RB7. Al conectar la tensión de la alimentación o después de un reset parpadeará rápidamente varias veces. Así se puede comprobar el funcionamiento correcto del PIC y su contenido.

Para la alimentación lo mejor es utilizar un adaptador de red de 5 V, cualquier adaptador estándar es suficiente para, por lo menos, 2 árboles. Con dos árboles interconectados puedes juntar los hilos de alimentación con las líneas de sincronización (todos están disponibles en el conector K3). En las **figuras 2 y 3** puedes ver las placas impresas diseñadas para los árboles de Navidad. Se pueden adquirir a través de Elektor [1]. La construcción es sencilla, sólo se han utilizado componentes de hilo pasante. Para la conexión entre el microcontrolador y la placa del display es mejor utilizar un trozo de cable plano de 16 hilos.

El software de este circuito se descargará de forma gratuita del sitio web de Elektor [1]. Hay versiones diferentes para el árbol de la izquierda y de la derecha, tenlo en cuenta durante la programación. Si construyes un único árbol, no importa que software cargues en el PIC. En el sitio web también hay un video de demostración del autor y los ficheros del diseño de las placas impresas para aquel que quiera hacerse sus propias placas. Si los patrones ofrecidos por defecto no son suficientes y quieres crear tus propios efectos, puedes trabajar con el código fuente disponible. Pero con una duración del programa de más de 50 minutos, los patrones lumínicos pre-programados no te aburrirán rápidamente...

(120634)

Figura 3.
Así es la placa (de doble cara) con los LED en forma de árbol de Navidad.



Enlaces Web

[1] www.elektor.es/120634

[2] www.superbrightleds.com

Lista de materiales

Placa controladora

Resistencias:

R1,R4,R14 = 4k7
R2,R3 = puente de hilo
R5 = 330 Ω (opcional)
R6...R13 = 680 Ω (ó 820 Ω ó 1 k, ver texto)

Condensadores:

C1 = 47 µF/16 V

C2,C3 = 22 pF cerámico

Semiconductores:

D1 = LED rojo, 3 mm (opcional)
IC1 = PIC18F2685-E/SP (Microchip)

Varios:

X1 = cristal de cuarzo 24 MHz
K1 = conector de alimentación (por ejemplo CUI PJ-031DH)
K2 = conector SIL de 2 vías (opcional)

K3 = conector SIL de 5 vías

K4 = conector SIL de 16 vías (o un trozo de cable plano de 16 hilos)

Pulsador opcional para modo y reset

Placa 120634-1 (ver [1])

Placa LED

D1...D56 = LED de 5mm de alta eficiencia

K1 = conector SIL de 16 vías (ver K4 de la placa controladora)

Placa 120634-2 (ver [1])

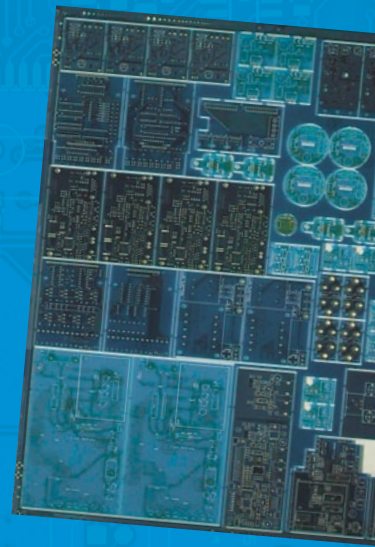
The European reference for PCB prototypes and small series

Easy-access online pooling services cut your board costs

- No tooling charges
- No minimum order charge - order from 1 PCB
- Instant online ordering without pre-payment

New added-value PCB services get your designs into production faster

- PCB Visualizer immediate online data check
- Online advice and tips on design for manufacture



PCB proto – dedicated engineering prototype service, fast and low-cost

- 1 or 2 PCBs in 2, 3, 5 or 7 working days
- DRC-checked and fully finished with 2 soldermasks and 1 legend, 150µm technology
- 1 x 100 x 80mm in 7WD - 2 layer 47,03 € - 4 layer 95,52 €
- 2 x 100 x 80mm in 7WD - 2 layer 36,89 € each - 4 layer 74,76 € each

Prices quoted include 21% Belgian VAT but exclude transport costs

STANDARD pool – the widest range of pooling options in Europe

- 1-8 layers 150µm technology PCBs
- from 2 working days

TECH pool – all the benefits of pooling for high-density PCBs

- 2-8 layers 100µm technology PCBs
- from 4 working days

IMS pool – aluminium-backed PCBs for high heat-dissipation (LED) applications

- Single layer Insulated Metal Substrate PCBs
- from 3 working days

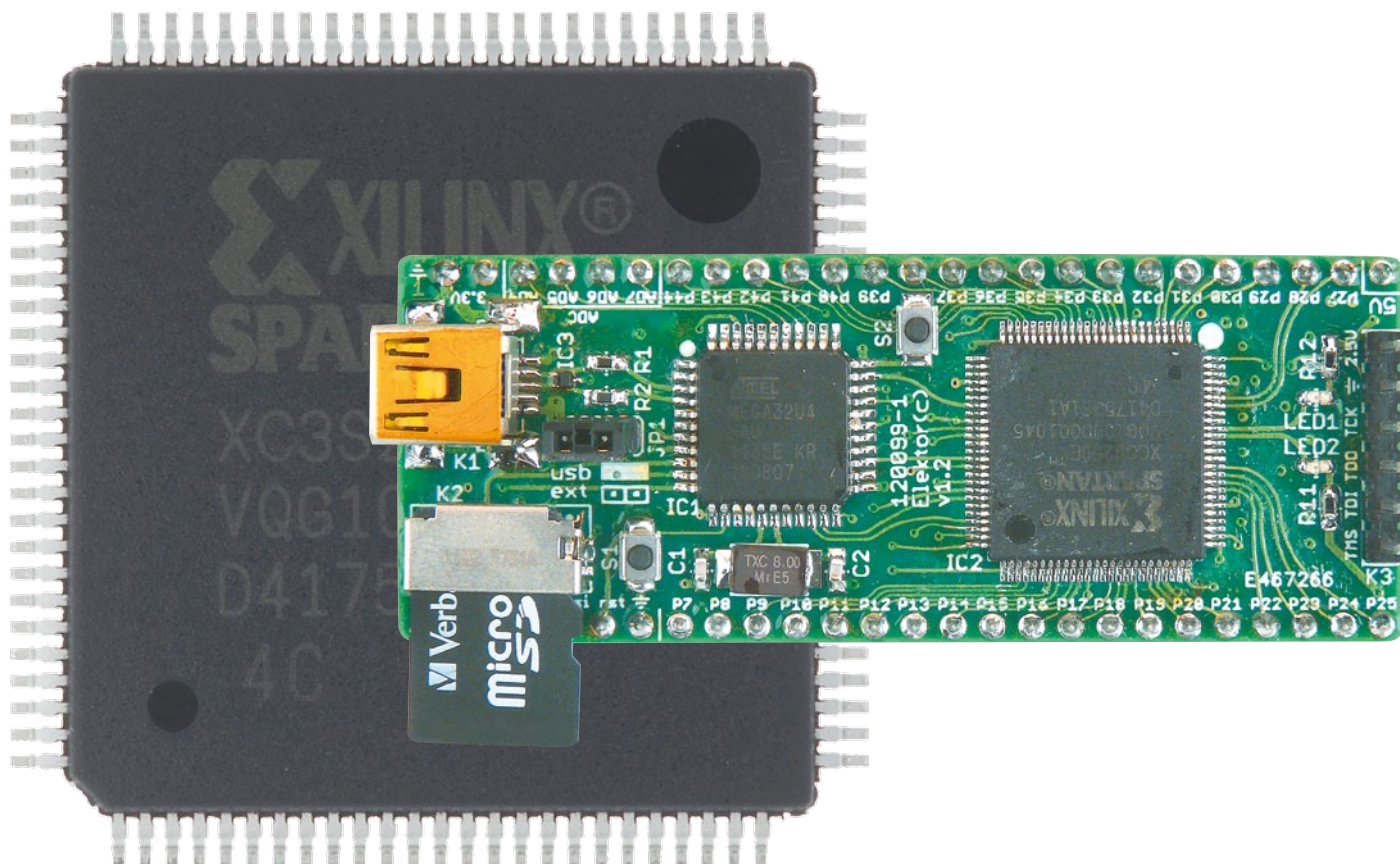
On demand – multi-option non-pooling service for specialist needs

- 1-16 layers down to 90µm technology
- from 2 working days

www.elektorPCBservice.com

Diseña tu propio integrado

Trabajar de forma sencilla con una FPGA potente



Texto:
Clemens Valens
(Elektor.Labs)

Diseño:
Raymond Vermeulen
(Elektor.Labs)

Uno de los componentes más polifacéticos y complejos de la electrónica actual es sin duda la FPGA, un laberinto de puertos y circuitos con la que se puede configurar tu propio circuito digital en un integrado, como si fuese de verdad tu propio chip. En esta serie mostramos cómo cualquier electrónico puede iniciarse fácilmente en este tipo de lógica programada, con la ayuda de una placa FPGA de pruebas diseñada en el laboratorio de Elektor.

¿Miedo a las FPGA? Que va, ¡ya no con la ayuda de la placa FPGA de Elektor!

FPGA es una abreviatura de *Field Programmable Gate Array*, lo que significa, traducido literalmente, "Matriz de Puertas Programable en Campo". Una mejor traducción es probablemente

"Lógica Configurable por el Usuario", ya que de eso tratan en realidad las FPGA. Una FPGA contiene una gran cantidad de puertas, biestables y elementos de memoria (y también alguna

que otra cosa) que el usuario puede configurar hasta convertir en un circuito. De aquí parte directamente la gran diferencia entre FPGA, microprocesadores y microcontroladores: una FPGA no ejecuta un programa como lo hace un procesador, hay que ver a una FPGA como si fuera un circuito.

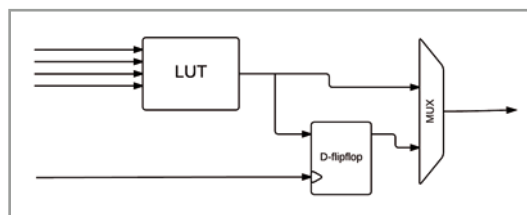


Figura 1.
Una sencilla versión de un diagrama de bloques de una celda lógica dentro de una FPGA.

El principio: una matriz sencilla

Inicialmente la lógica programable consistía en una matriz de varias decenas de puertas cuyas entradas y salidas se podían interconectar más o menos a gusto del usuario. Así se podían ubicar fácilmente complejas funciones lógicas en un único integrado en vez de una placa grande con muchos integrados de lógica sueltos. Una ventaja añadida de este diseño es la posibilidad de adaptar rápidamente funciones mediante la reprogramación del integrado, sin que sea necesario cortar pistas en la placa o colocar nuevas conexiones. Este tipo de lógica programable se parece mucho a una ROM (*Read-Only Memory*) y una EPROM (*Erasable Programmable Read-Only Memory*) que ofrecen posibilidades parecidas. Por poner un ejemplo sencillo tomamos una puerta AND con dos entradas A y B y una salida Q. Esta tiene la siguiente tabla de verdad:

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Si renombros ahora las entradas A y B en A1 y A0 respectivamente, y la salida Q en D0, entonces obtienes una línea de datos con la que puedes direccionar cuatro posiciones en la memoria. Llena estas cuatro posiciones con los valores 0, 0, 0, y 1 y obtendrás la función AND en memoria. También puedes llenar la memoria con algo diferente y utilizar más (o menos) direcciones. Así puedes construir funciones complejas de una forma sencilla.

Sin embargo, lo que no puedes hacer en una memoria es guardar el resultado de la función en otra parte de la memoria. Aunque una memoria contenga muchos registros, no puede acceder a ellos por sí misma. Sólo el usuario puede escribir y leer el resultado en ella.

Por eso se amplió la lógica programable con registros en donde se puedan guardar los resultados para ser utilizados en otra parte del

integrado. Con registros nos referimos a sencillos biestables, es decir, un elemento de memoria de 1 bit. En la **figura 1** se puede ver la versión simplificada de una celda lógica que se utiliza en una FPGA.

Se puede hacer un montón de cosas con biestables, sobre todo si hay muchos. Puedes crear, por ejemplo, contadores, registros de 8 bits o registros de una anchura cualquiera. También se pueden hacer registros de desplazamiento, de modo que se realicen conversiones de paralelo/serie (y viceversa) o retardar datos. En resumidas cuentas, una matriz de puertas lógicas está bien, pero empieza a ser realmente interesante si se añaden elementos de memoria.

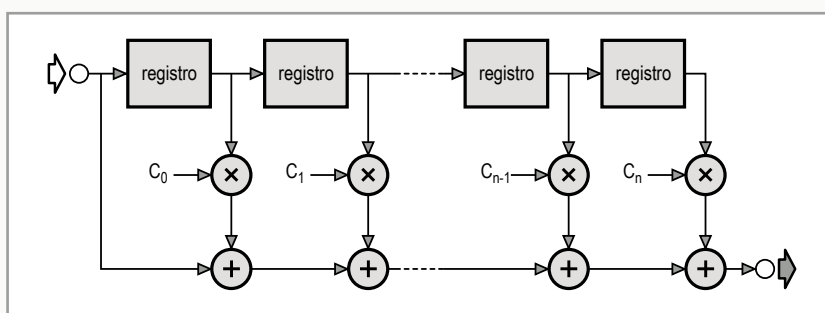
Más complejo: más posibilidades

Durante los años pasados, la lógica programable se hizo cada vez más compleja, tal y como se puede observar en el término CPLD, que es una abreviatura de *Complex Programmable Logic Device*. La tecnología que avanzaba hizo también posible - o mejor quizás - necesario diseñar los componentes de tal manera que se pareciesen cada vez más a la RAM (*Random Access Memory*) en vez de a la ROM. Una RAM no puede guardar datos sin tensión y lo mismo sucede para este tipo de lógica programable: El integrado pierde su configuración cada vez que se desconecta la tensión. Así que fue necesario poder configurar el integrado durante el arranque sin tener que utilizar un aparato especial de programación, como si estuviese en el campo. Así surgió la FPGA. Para que la configuración fuese más fácil, el fichero de configuración se carga a menudo en una memoria externa, que la FPGA lee durante el arranque. Lo cual hace pensar mucho en sistemas de microprocesadores que carguen también su programa en una memoria externa. Por otra parte existen actualmente FPGA que sí pueden guardar su configuración.

Con el curso de los años, los fabricantes de FPGA han conseguido juntar una cantidad gigantesca de puertas y biestables en un único integrado. Esta avanzada integración hace que pueda haber

Figura 2.

Un filtro digital consta de una línea de retardo, multiplicadores y sumadores. Aquí se ha omitido la señal de reloj que se encarga de la sincronización. La línea de retardo se puede construir con registros, las constantes se guardan en registros y los multiplicadores y sumadores constan de funciones lógicas y registros. Todo esto se puede implementar de forma sencilla en una FPGA, lo que explica por qué las FPGA son tan adecuadas para aplicaciones de procesamiento digital de la señal.



nuevas aplicaciones y velocidades de datos extremadamente altas, y se encarguen también de nuevos problemas, como es la organización en el integrado. La interconexión aleatoria de varios millones de puertas requiere un montón de 'hilos'. Algunas de estas conexiones pueden llegar a ser largas, por ejemplo, si se tiene que interconectar la salida de una puerta en la esquina superior de la izquierda con una entrada en la esquina inferior de la derecha. Las distancias pueden llegar a ser problemáticas si la velocidad de datos es alta y por lo tanto la longitud de pulso es corta. Entonces aparecerán retardos, los flancos empezarán a no seguir el paso y será cada vez más difícil mantener las puertas sincronizadas entre sí.

Esto ha llevado a una arquitectura global de la FPGA que está repartida en bloques de funciones. La ubicación de estos bloques en el integrado ha sido cuidadosamente elegida, de modo que puedan ser utilizados de forma óptima en

la mayoría de las aplicaciones estándar. Para algunas aplicaciones específicas hay a veces funciones especiales que hacen que una FPGA sea distinta a otra. Los bloques que actualmente se encuentran en cualquier FPGA (pero bajo nombres diferentes), son:

- Bloques lógicos
- Bloques de E/S
- Bloques de cálculo
- Bloques de memoria
- Bloques de reloj

Aparte de estos bloques funcionales comunes también puedes encontrar cosas como bloques de procesador, memoria flash, aparatos periféricos como los controladores de Ethernet, controladores de memoria, ADC y controladores para la comunicación serie (SPI, I²C, etcétera). Con este tipo de funciones adicionales, la FPGA se inclina más hacia un sistema completo y se habla a veces de un System-on-Chip (SoC) con una FPGA. Se pueden interconectar los bloques mediante un sistema flexible de conexiones configurables.

Los bloques

Repasamos la lista de bloques funcionales de arriba abajo.

Bloques lógicos

Los bloques lógicos ocupan la mayor parte de la FPGA y forman la 'tela' de la FPGA (*fabric* en inglés), sobre la que se pinta como si fuera la aplicación. El ejemplo que se ha explicado al principio de este artículo corresponde a este tipo de bloques. Un bloque lógico es un circuito configurable de puertas y registros, basado en una denominada LUT (*Look-Up Table*). Una LUT es exactamente igual al ejemplo de una puerta AND, pero generalmente algo más grande, con más

Tabla 1. Los bloques funcionales del Spartan 3E XC3S250E-4VQG100C de Xilinx.

Puertas	250.000
Bloques lógicos	612
Bloques de E/S	66, de los cuales 7 sólo de entrada
Bloques de cálculo	12 multiplicadores
Bloques de memoria	12 para un total de 221.184 bits (216 Kbit)
Memoria distribuida	39.168 bits (38,25 Kbit)
Bloques de reloj	4

El número de bloques de E/S depende del encapsulado del integrado, nosotros hemos utilizado la variante VQ100 con 100 terminales. Un bloque lógico, en términos de Xilinx, un Configurable Logic Block (CLB), consta de cuatro slices (rodajas) que contienen cada una dos LUT de 16 bits, donde la mitad puede servir de RAM distribuida y la otra mitad como funciones lógicas. Cada placa contiene también dos biestables, de modo que cada CLB dispone de 8 biestables en total (ver también **figura 2**).

entradas. Es un registro de x número de bits, por ejemplo dieciséis, con el que puedes realizar una función lógica. También es posible utilizar la LUT como registro de desplazamiento y como memoria RAM, entonces se habla de RAM distribuida (para indicar la diferencia con los bloques de memoria). Añadiendo varios multiplexores y un poco de lógica se amplía la flexibilidad y un bloque lógico podrá hacer algunos cálculos.

Como la FPGA dispone de muchos bloques de este tipo, puedes colocar varios en paralelo y construir un multiplicador o un sumador. Un filtro digital consta de varios elementos de memoria, algunos multiplicadores y un sumador, esto también vale para una transformada digital de Fourier (**figura 2**). Así que un bloque lógico dispone de todos los ingredientes básicos para construir filtros digitales o transformadas de Fourier. Con un único bloque no se puede hacer, pero con varios se consigue bastante bien. Esto hace que la FPGA sea especialmente apta para el procesamiento digital de la señal. Colocada entre un convertidor A/D y un convertidor D/A, la FPGA puede hacer cosas muy curiosas con audio, video, radio, etc. Y luego con mas canales al mismo tiempo.

Actualmente existen FPGA con más de un millón de bloques lógicos de este tipo (Xilinx Virtex-7 XC7V2000T, 1,2 millones de LUT y 2,4 millones de biestables). Este integrado contiene casi siete mil millones de transistores. En comparación, el procesador Intel Xeon Westmere-EX de 10 núcleos contiene 'sólo' 2,6 mil millones de transistores.

Bloques de E/S

Los bloques lógicos del integrado, igual que otros bloques funcionales, tienen que comunicar con el mundo exterior a través de los terminales del integrado (el anteriormente nombrado XC7V2000T tiene 1200). Esta comunicación pasa a través de los denominados bloques de E/S.

Un bloque de E/S es más que únicamente un par de transistores en una configuración push-pull. Este tipo de bloques soporta varios estándares de comunicación (DDR, PCI, LVTTTL, LVCMOS, LVDS, etc.), tanto el 'normal' (*single-ended*) como el simétrico (*differential*) son capaces de alcanzar velocidades de datos extremadamente altas. Para eso vienen equipados con líneas de retardo, biestables simétricos y convertidores paralelo/serie

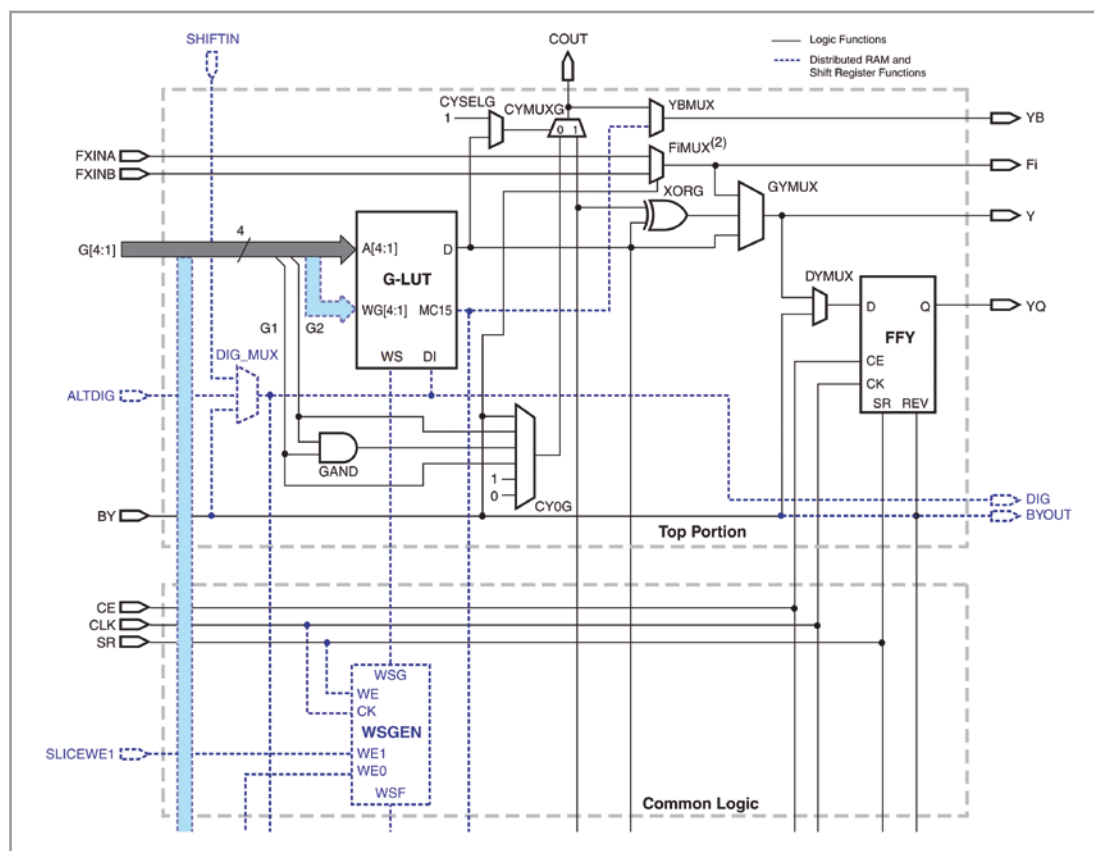


Figura 3.
El diagrama de bloques de un trozo de slice. La parte que falta es la 'Bottom Portion' que es en gran medida idéntica a la parte del rectángulo punteado denominado 'Top Portion'.

El circuito

Tal y como se puede observar en el diagrama de bloques (**figura 4**), el circuito consta principalmente de un ATmega32U4, una FPGA Spartan3e, una parte de alimentación y un zócalo para una tarjeta microSD.

Es de la FPGA de lo que aquí se trata (ver también el esquema real de la **figura 5**). Hemos optado por el XC3S250E-4VQG100C, un tipo de la serie Spartan3e de Xilinx. Tiene 250.000 puertas de sistema, más de 5500 celdas lógicas, 612 unidades lógicas y la frecuencia máxima asciende a 572 MHz (ver más en **tabla 1**). Para conseguir que las medidas de la placa fueran lo más reducidas posibles, se buscó una FPGA con un encapsulado pequeño. La elección recayó en un encapsulado con terminales (SMD). El integrado utilizado contiene 100 terminales de los que 66 son de E/S (suficiente para la mayoría de las aplicaciones). La cantidad de lógica es suficiente incluso para proyectos bastante complejos.

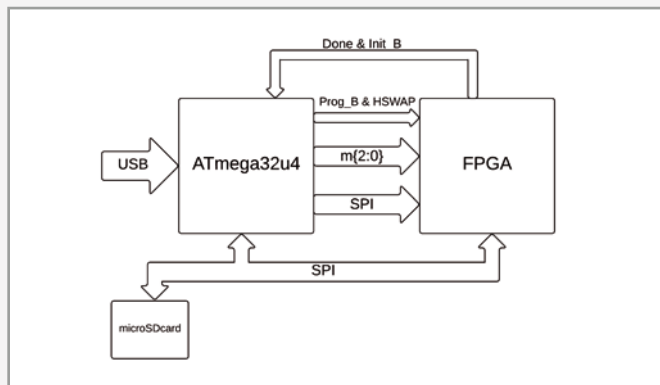


Figura 4. El esquema de bloques de nuestra placa FPGA de pruebas muestra que el circuito consta principalmente de una FPGA, un microcontrolador y un conector de tarjeta microSD.

El microcontrolador del tipo ATmega32U4 es el ‘manitas’ de la placa. Se encarga de la comunicación con el PC a través de USB, ofrece un puerto COM virtual y permite que el PC trate la tarjeta microSD como si fuera un dispositivo de almacenamiento masivo. También lee la tarjeta microSD y programa la FPGA. Se ha optado por este controlador porque combina un ATmega conocido con hardware USB. Si bien es cierto que es un micro lento de 8 bits, sirve perfectamente para esta aplicación. El firmware del microcontrolador está disponible en el sitio web de Elektor [1], en la placa suministrada por nosotros viene ya pre-programado.

Se ha elegido el zócalo de la tarjeta microSD por sus reducidas dimensiones en la placa impresa. Está conectado para ser utilizado en modo SPI. La mayoría de las tarjetas microSD lo

soportan. Las dos E/S sin utilizar están conectadas a 3.3 V a través de resistencias pullup para evitar que surjan estados indefinidos. Hay un interruptor de detección de tarjeta que indica si hay tarjeta en el zócalo.

La parte de alimentación es bastante amplia para un circuito de este formato. Esto se debe principalmente a que la FPGA requiere de tres tensiones de alimentación: 3.3, 2.5 y 1.2 V. La tarjeta microSD y el microcontrolador utilizan también los 3.3 V. Se han añadido condensadores adicionales para una buena estabilidad y desacoplamiento de las diferentes tensiones de alimentación.

La placa se alimenta a través de la conexión USB si el puente JP está colocado. Esta tensión de 5 V también está disponible en P26 de K5 en el borde de la placa, para uso externo. ¡Ten cuidado! Las conexiones de E/S de esta placa no soportan 5 V.

Al revés también funciona. Si omites el puente JP1, puedes alimentar el circuito localmente a través de P26 de K5 con una tensión de +5 V, de modo que la conexión USB no es obligatoriamente necesaria. A causa del gran número de señales y planos de alimentación fue necesario diseñar una placa de cuatro capas (**figura 6**) para este proyecto. Pero descuida, en Elektor puedes pedir una placa completamente ensamblada por un precio muy razonable [1].

Un método de configuración particular

Normalmente no se puede programar las FPGA de forma fija, cada vez que se arranca hay que volver a configurarlas. Esto se hace en la mayoría de las placas FPGA de desarrollo con la ayuda de un PC y un programador JTAG (lo que aquí también es posible a través de K3). En configuraciones permanentes con una FPGA se utiliza generalmente una memoria flash conectada al bus SPI. Entonces la FPGA lee durante el arranque la configuración de la memoria flash.

Aquí lo hacemos de forma muy diferente (ver **figura 3**). Utilizamos un microcontrolador para colocar la configuración en la FPGA de una manera parecida a SPI. Esto es el denominado modo SerialSlave. Normalmente se utiliza este método en grandes sistemas que trabajan con varias FPGA, pero también pueden ser utilizadas con una única FPGA.

El microcontrolador puede comunicar con la tarjeta microSD a través de SPI. Para poder utilizar esta tarjeta en un sistema operativo (SO) tiene que estar formateado en el formato FAT16 o FAT32. Por lo tanto, el microcontrolador tiene que ser capaz de trabajar con el sistema de ficheros para poder leer la tarjeta microSD. Para permitir que se pueda acceder a esta tarjeta desde el SO hace falta que el microcontrolador pueda interpretar los denominados comandos SCSI que ejecuta a continuación.

Gracias a esta configuración, el flujo de trabajo es muy sencillo. Cuando hayas creado un fichero de configuración en el software Xilinx ISE, lo colocas en la tarjeta microSD mediante el explorador de Windows. La placa aparece en el explorador como disco USB con

una letra del disco. Sólo hace falta renombrar el fichero a 'config.bin'. Después puedes desmontar el disco en el SO, resetear la placa y la nueva configuración se cargará cuando la arranques.

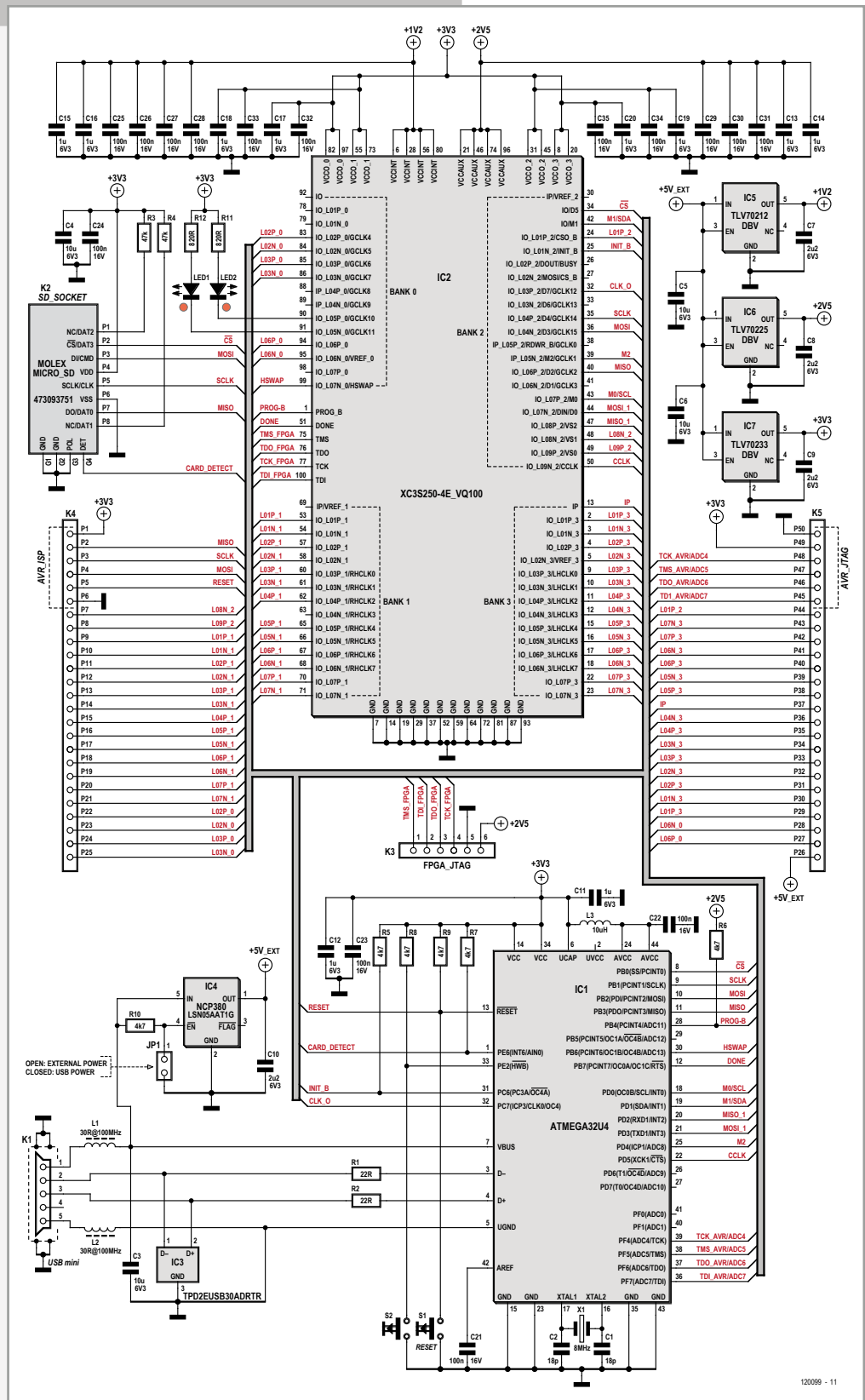
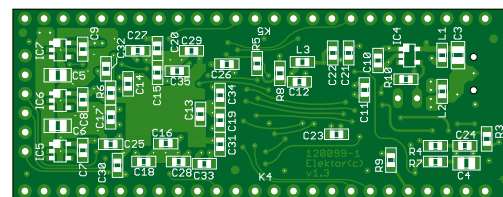
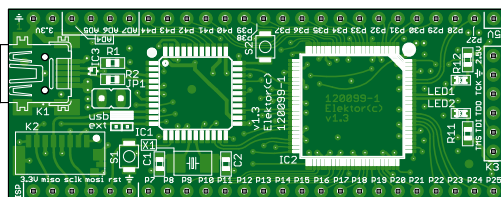


Figura 5.
El esquema de la placa de pruebas. El esquema completo cabe justo en una página, en realidad todo está alojado en una placa de tan solo 65 x 25 mm.

Gracias a la utilización de cuatro capas de cobre fue posible realizar todas las conexiones en una superficie de una placa tan pequeña. La placa se entrega completamente ensamblada.



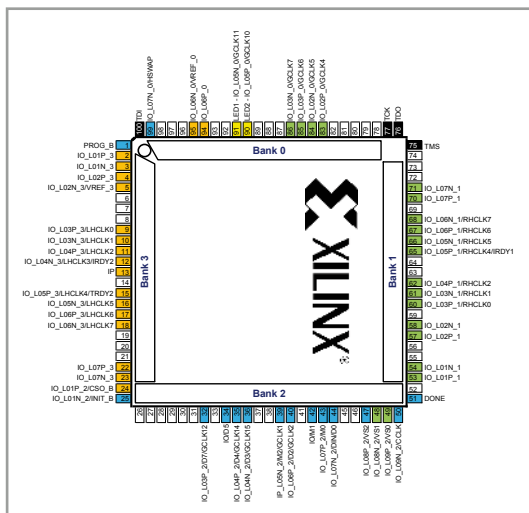
electromagnética dentro de unos límites, el usuario debe configurar la pendiente y la corriente máxima de cada salida.

Como los usuarios de las FPGA utilizan bloques lógicos para calcular, los fabricantes integraron también bloques de cálculo en sus productos. Generalmente se trata de la multiplicación, pero algunos fabricantes van más allá y añaden funciones DSP, con las que se pueden hacer cálculos como $a + (b \cdot c) \rightarrow a$ (multiply-accumulate) de forma sencilla.

Las FPGA contienen, aparte de la RAM distribuida en forma de LUT, bloques de RAM cuyo tamaño asciende a megabits. La integración de la memoria en el integrado hace posible velocidades de procesamiento de datos extremadamente altas. Por eso se pueden realizar también sistemas de microcontroladores completos en una simple FPGA. Equipando los bloques de RAM con dos puertos independientes, también pueden servir de buffers de datos (FIFO) para, por ejemplo, sincronizar sistemas que funcionan a diferentes velocidades. La anchura de palabra de la memoria es variable, de modo que se pueden utilizar bloques tanto de 1 bit como de, por ejemplo, 32 bits mezclados entre sí.

Aunque en teoría es posible implementar las funciones de forma asimétrica, quiere decir sin señal de reloj, no suele ser un buen planteamiento para sistemas complejos. Debido a las diferentes distancias que tienen que recorrer las señales, surgen diferencias en el tiempo y los flancos que salieron a la vez no llegan al mismo tiempo a su destino. Así es como surgen estados intermedios indebidos en el circuito que conducen a errores. La solución es mostrar las señales en determinados tiempos, como si se tomase una foto del estado

La distribución de los terminales de la FPGA utilizada. Los terminales de color naranja y verde están conectados a los conectores de la placa, los de color azul forman el interfaz con el microcontrolador. Los LED están conectados en los terminales de color amarillo, los terminales de color negro forman el interfaz JTAG que se puede utilizar a través de un conector aparte.



Las patas de los conectores de ampliación. Los terminales de color naranja y azul corresponden a los de la figura 6. El microcontrolador se puede programar a través de estos conectores. Los terminales del puerto AVR JTAG también pueden ser utilizados como entradas analógicas.

1	3.3V		GND	50
2	MISO		AVR_3V	49
3	SCLK	TCK_AVR/ADC4		48
4	MOSI	TMS_AVR/ADC5		47
5	RESET	TDO_AVR/ADC6	AVR_JTAG	46
6	GND	TDI_AVR/ADC7		45
7	IO_L08N_2/VS1	IO_L01P_2/CS0_B	24	44
8	IO_L09P_2/VS0	IO_L07N_3	23	43
9	IO_L01P_1	IO_L07P_3	22	42
10	IO_L01N_1	IO_L06N_3/LHCLK7	18	41
11	IO_L02P_1	IO_L06P_3/LHCLK6	17	40
12	IO_L02N_1	IO_L05N_3/LHCLK5	16	39
13	IO_L03P_1/RHCLK0	IO_L05P_3/LHCLK4/TRDY2	15	38
14	IO_L03N_1/RHCLK1	IP	13	37
15	IO_L04P_1/RHCLK2	IO_L04N_3/LHCLK3/IRDY2	12	36
16	IO_L05P_1/RHCLK3	IO_L04P_3/LHCLK2	11	35
17	IO_L06P_1/RHCLK6	IO_L03N_3/LHCLK1	10	34
18	IO_L06N_1/RHCLK7	IO_L03P_3/LHCLK0	9	33
19	IO_L07P_1	IO_L02N_3/VREF_3	5	32
20	IO_L07N_1	IO_L02P_3	4	31
21	IO_L02P_0/GCLK4	IO_L01N_3	3	30
22	IO_L02N_0/GCLK5	IO_L01P_3	2	29
23	IO_L03N_0/GCLK6	IO_L06N_0/VREF_0	95	28
24	IO_L03P_0/GCLK6	IO_L06P_0	94	27
25	IO_L03N_0/GCLK7		5V	26

en cada momento. Si bien es cierto que ralentiza el circuito, mantiene sincronizados todos los flancos. La señal de reloj y los biestables hacen posible esta sincronización. Sin embargo, para las señales de reloj cuentan las mismas leyes que para cualquier otra señal, así que surge retardo atravesando el integrado. Los fabricantes de los integrados FPGA encontraron una solución a esto incorporando todo tipo de posibilidades de retardo y desplazamiento en los bloques de reloj. También se pueden distribuir las señales de reloj a través de redes especiales de reloj.

Otra razón para utilizar bloques de reloj es cuando partes del sistema trabajan a diferentes velocidades de reloj (denominadas dominio de reloj). Para eso se pueden configurar los bloques de tal manera que generen diferentes frecuencias partiendo del mismo reloj de entrada. Estas frecuencias pueden ascender muy alto, hasta los GHz, lo que, por supuesto, puede beneficiar la velocidad de un circuito.

En la realidad

Después de esta introducción general es interesante ser un poco más práctico y jugar un poco con una FPGA. Para eso Raymond Vermeulen, diseñador de Elektor, diseñó una pequeña placa de pruebas FPGA que elimina una gran desventaja de la mayoría de la mayoría de placas de prueba FPGA, es decir, el intrincado método de programación de la configuración de la FPGA. Las herramientas software para la configuración de una FPGA son bastante complejas y se necesita a menudo un dispositivo externo de programación JTAG para cargar el fichero de configuración en el integrado. Raymond pensó en una solución donde el PC ve la placa FPGA como si fuera una memoria USB (en realidad es la tarjeta microSD que se encuentra en la placa). Sobre ella se puede copiar el fichero de configuración como cualquier otro fichero. Al arrancar de nuevo la placa se carga el fichero desde esta memoria USB en la FPGA. Así se ha solucionado de una forma elegante el espinoso problema de los controladores especiales para el hardware, también especial, de programación.

Hasta el próximo mes

Como la configuración de las FPGA es una materia compleja, vamos a intentar explicar claramente cómo se puede llegar a obtener una aplicación interesante a través de una serie de artículos. Si todo va bien, el próximo mes explicaremos

Se puede adquirir la placa FPGA de pruebas completamente ensamblada en Elektor por un precio de tan sólo € 59,95 más gastos de envío.



Ver www.elektor.es/120099

primero cómo se instala el Xilinx ISE Webpack gratuito y cómo utilizarlo para hacer parpadear los LED en la placa de pruebas. Para matar el tiempo hasta que salga el próximo artículo te aconsejamos descargar ya el software [2]. Esta versión de Webpack es gratuita y tiene un tamaño de más de 6 GB. Para aquellos que tengan una conexión lenta de internet es aconsejable pedir el DVD gratuito en Xilinx (**figura 9**). Normalmente se recibe dentro de dos semanas en casa.

(120099)

Enlaces Web

- [1] www.elektor.es/120099
- [2] <http://www.xilinx.com/support/download/index.htm>



Figura 9.
El DVD gratuito de Xilinx que contiene el ISE Webpack y otras herramientas FPGA.

Cable USB-IO24

Medir, controlar y regular fácilmente con el PC

Dr. Ing. Uwe Altenburg (Alemania)

Con el cable presentado aquí pueden leerse y transferirse hasta 24 señales digitales mediante el puerto USB de un PC. Opcionalmente también se pueden capturar hasta ocho señales analógicas, generar señales PWM o controlar un servo. En el artículo presentamos un sencillo protocolo de control del cable, así como la programación en el PC, en lenguaje C#.

Hoy en día existen ordenadores en cualquier formato concebible: ya sean de sobremesa, portátiles o tabletas. Esto también ha tenido un impacto positivo en los precios. Aquellos que hace algunos años tenían que utilizar una mini-torre para controlar la iluminación del acuario, hoy en día quizá utilicen un box PC y hayan desarrollado un bonito

entorno de usuario con controles táctiles. Sin embargo, conectar nuestros propios dispositivos electrónicos a un PC mediante USB no es algo del todo simple, y a la hora de generar señales críticas en el tiempo, el PC sigue sin ser nuestra primera opción. Además, hemos de preocuparnos por el protocolo de comunicación, así como elegir primero el lenguaje de programación más apropiado. ¡Os presentamos una solución para estos casos!



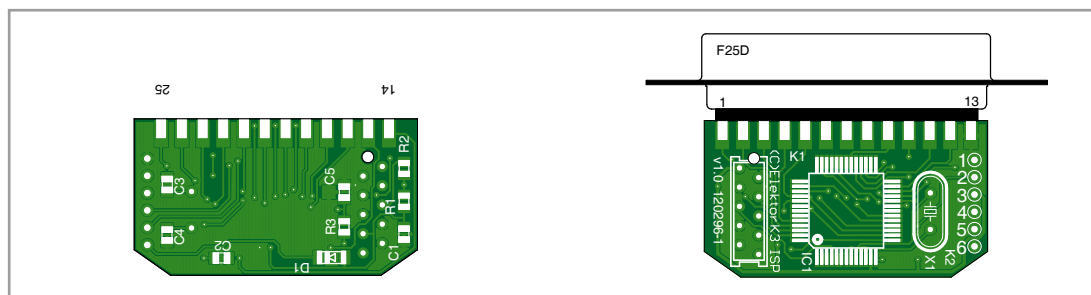


Figura 1.
La tarjeta cabe en el propio encapsulado del conector sub-D.

Lista de materiales

Resistencias (SMD 0805, 5 %):

R1 = 100 k Ω
R2 = 4k7
R3 = 100 Ω

Condensadores:

C1, C2, C5 = 100 nF, 50 V, 10 %, X7R, SMD 0805
C3, C4 = 22 pF, 50 V, 5 %, NP0, SMD 0805

Semiconductores:

D1 = LL4148 Minimelf
IC1 = R8C/25 SMD TQFP52

Varios:

Cable adaptador USB/TTL, TTL-232R-5V-WE, disponible en Elektor: 080213-71 [6]
K1 = conector DSub-25 hembra, para montaje sobre placa (77SD B25S Amphenol)
K3 = conector MicroMatch de 10 pines, para montaje sobre placa con los puertos debajo (8-215079-0 TE Connectivity/Amp)
X1 = cuarzo de 18,432 MHz, 18 pF, 50 ppm
Encapsulado para conector DSub-25, Flip Top (A-FT 25 Assmann)

Opcional:

Conector MicroMatch hembra de 10 pines (8-215083-0 TE Connectivity/Amp)

Circuito

El fabricante inglés FTDI ofrece desde hace varios un cable adaptador USB/TTL (TTL-232R-5V-WE [1]), también disponible en Elektor [2]. Gracias a este cable, que junto con las señales Rx/D, Tx/D, RTS y CTS también dispone de alimentación a 5 V, resulta muy sencillo conectar un microcontrolador al PC. El cable dispone de un convertidor USB-serie FT232R, encapsulado en el propio conector USB. Se nos ocurrió colocar en el otro extremo del cable un conector sub-D de 25 pines, en el cual podíamos integrar un microcontrolador. El que utilizamos aquí, de 8 bits, es un derivado de la familia R8C, ya utilizada en Elektor en otras muchas ocasiones. El R8C25 [3] dispone de una memoria flash de 64 kB y 3,5 kB de RAM, suficientemente grande. Con su convertidor analógico de 10 bits, así como una potente estructura de temporizadores, resulta ideal para tareas de control. Los 25 pines del conector sub-D son suficientes para acceder desde el exterior a tres puertos de entrada/salida del controlador, así como a la conexión de masa necesaria. En el encapsulado del conector hay sitio de sobra para una pequeña placa de 37x20 mm (**figura 1**). Junto al microcontrolador, en esta placa encontramos un cuarzo de 18,432 MHz, un conector para la programación, así como unos pocos componentes discretos. Está planificado suministrar

la placa ya montada, puede encontrarse más información al respecto en la página web de este artículo [6]. En esta placa sólo hará falta soldar el cable de FTDI y albergar todo en una carcasa adecuada.

Cabe destacar algunas peculiaridades sobre el diseño del circuito (**figura 2**). El microcontrolador se alimenta a través del cable USB. Mediante el diodo D1 se conecta la línea RTS del cable USB-serie con el pin de reset del microcontrolador. De esta manera es posible realizar un reset del hardware directamente por el puerto USB. Las señales Tx/D y Rx/D del cable USB-serie se utilizan para la comunicación entre el PC y el microcontrolador.

Características

- 24 entradas/salidas digitales, o alternativamente hasta:
- 8 entradas analógicas (resolución de 10 bits)
- 8 salidas PWM (resolución de 10 bits)
- 8 señales para servos RC (resolución de 10 bits)
- 4 entradas de contador (16 bits hacia arriba/abajo)
- 2 Integrados MAX7219 para controlar hasta 128 LEDs

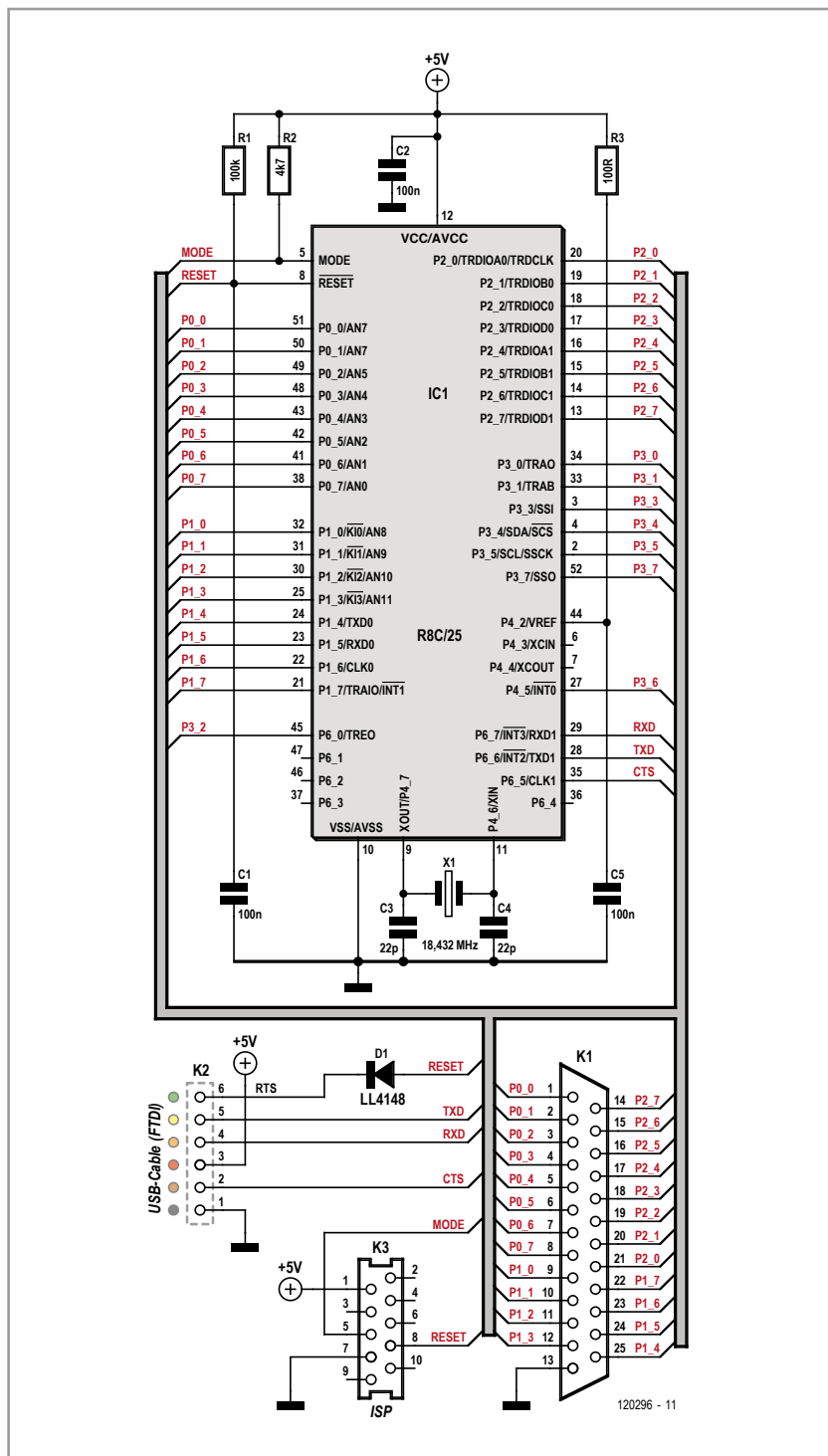


Figura 2. Junto con el microcontrolador sólo necesitaremos un oscilador de cuarzo y unos pocos componentes discretos.

Los puertos P0 a P2 del microcontrolador son accesibles desde el conector sub-D (**figura 3**), es decir, la mayoría de señales. Los pines de Port0 pueden configurarse como entradas analógicas, con los cuales muestrear hasta ocho señales. La entrada para la tensión

de referencia del convertidor analógico/digital es estándar, utilizando un circuito RC junto con la alimentación a 5 V, la cual permite un rango de medida de 0 a 5 V, con suficiente resolución. En caso de que necesitemos más precisión a la hora de medir señales analógicas, también podemos incorporar otra tensión de referencia, con un integrado del tipo LM4040.

Port1 ofrece alternativamente hasta cuatro entradas de interrupción, con las cuales pueden realizarse, contadores rápidos, por ejemplo. Aparte, en Port1 encontramos un puerto serie, capaz de operar en los modos UART y SPI.

En Port2 se encuentran las salidas de ambos timers de 16 bits del R8C25. Con ellos es posible generar hasta ocho señales PWM al mismo tiempo. Una señal PWM puede utilizarse, por ejemplo, para el control de motores de CC o generar señales analógicas. Por esta razón, nuestro cable USB-IO24 ofrece muchas posibilidades! Pero, ¿y en cuanto al software?

Firmware

El autor ha diseñado un firmware para el controlador, haciendo que sea muy fácil operar desde el PC (el controlador de la placa montada ya viene cargado con este firmware). La transferencia de archivos entre PC y microcontrolador se realiza a 115200 baudios con 8 bits de datos, 1 bit de stop y sin paridad (115200,8,N,1).

En lugar de un protocolo de datos binario con una estructura fija (que utilizara por ejemplo un tipo y longitud de datos determinados, así como un checksum), para la comunicación bastan sencillos comandos de texto. Esta idea resulta flexible y fácil de entender; aparte, la comunicación también es transparente. Y para los comandos de configuración no necesitaremos nada más que un simple programa terminal, siendo totalmente independiente del sistema operativo utilizado.

Tras el reset, el microcontrolador se inicializará. No hay mucho que hacer aquí ya que el funcionamiento exacto de los puertos y el timer se configura previamente por comandos. Únicamente han de inicializarse la generación del reloj y, como es natural, la conexión por puerto serie con el PC. Además, el programa principal consta de un bucle infinito: el intérprete de comandos (**listado 1**). Un comando es una secuencia de caracteres ASCII seguidos de un CR (Carriage Return) al final. Realmente también pueden introducirse

muchos comandos seguidos y cerrar sólo el último con un CR. Esto da mayor flexibilidad a la aplicación. La cadena de caracteres *X10 P0.0=1 P0.0=0 <CR>* se trata de un comando válido y significa: “pon el puerto 0.0 10 veces a 1 y resetéalo”.

La función `ReadCmd()` se encarga de leer una secuencia de comandos hasta alcanzar un CR. De dicha secuencia, se eliminan los espacios con `SkipBlanks()` y después se hace una llamada a la función `Execute()`. Si la función ha sido capaz de ejecutar todos los comandos con éxito, devuelve un valor positivo, en caso contrario devuelve un cero; posteriormente, de acuerdo con esto, se envía OK o ERROR al PC.

La implementación de una secuencia de comandos puede llevar algo de tiempo hasta completarse. En ese tiempo se envían muchos caracteres a través del puerto serie del PC. A una tasa de transferencia de 115200 baudios, ise tardan 90 µs en transferir un carácter! La recepción se hace mediante una rutina de interrupción, en la cual se almacenan los caracteres en un buffer de anillo. `ReadCmd()` se encarga de leerlos del buffer.

Listado 1. Rutina principal con el intérprete de comandos.

```
// --- Main ---
VOID main()
{
    InitCrystal();           // init crystal
    InitUart();              // init uart
    EI;                      // ints

    SendString("\xCPIOCable V1.5\r\nOK\r\n");// version

    for (;;)                // endless..
    {
        ReadCmd();           // read command
        SkipBlanks();        // remove spaces

        if (Execute())       // execute commands
            SendString("\r\nOK\r\n");
        else
            SendString("\r\nERROR\r\n");
    }
}
```

Los posibles comandos se distinguen gracias al primer carácter. De esta manera, el comando de puerto comienza con una *P*, la lectura de las entradas analógicas con una *A*, el ajuste de un contador con una *C*, etc., donde se permiten

Listado 2. Interpretación de los comandos de puerto.

```
// --- Macros ---
#define Digit(p)    (*p++ - '0')           // get a digit
#define GetIf(p,c)  (*p == c ? p++, 1 : 0) // get if char

// --- Read port ---
BYTE GetPort(BYTE nPort)
{
    switch (nPort)
    {
        case 0: return p0;                // p0 input
        case 1: return p1;                // p1 input
        case 2: return p2;                // p2 input
    }
}

// --- Write port ---
VOID SetPort(BYTE nPort,BYTE nBits,BYTE nMask)
{
    BYTE nSet = nBits & nMask;

    switch (nPort)
    {
        case 0: p0 = p0 & ~nMask | nSet; break; // p0 output
        case 1: p1 = p1 & ~nMask | nSet; break; // p1 output
        case 2: p2 = p2 & ~nMask | nSet; break; // p2 output
    }
}

// --- Port command [Pn|Pn=v|Pn.b=v|Pn~|Pn.b~] ---
BOOL PortCmd()
{
    BYTE nPort = Digit(pCmd);              // port

    BYTE nMask = 0xFF;
    BYTE nPin  = 0;

    if (GetIf(pCmd, '.'))                  // '.'
    {
        nPin = Digit(pCmd);                // pin
        nMask = 1 << nPin;                // mask
    }

    if (nPort < 3 && nPin < 8)              // check
    {
        BYTE nValue;

        if (GetIf(pCmd, '='))              // '='
        {
            nValue = GetValue() << nPin;    // value
            SetPort(nPort,nValue,nMask);    // set port
        }
        else if (GetIf(pCmd, '~'))          // '~'
        {
            nValue = GetPort(nPort);        // get port
            SetPort(nPort,~nValue,nMask);   // set invert
        }
        else
        {
            nValue = GetPort(nPort) & nMask; // get port
            nValue = nValue >> nPin;         // get value
            SendValue(nValue);              // send value
        }

        return TRUE;
    }

    return FALSE;
}
```

Tabla 1. Guía de referencia de los comandos soportados			
A..	An	$n = 0 \text{ a } 7$	Se configura un pin como entrada analógica y devuelve el valor analógico en un rango de 0 a 1023.
C..	$Cn, Cn = x$ $Cn+, Cn-$	$n = 0 \text{ a } 3$ $x = 0 \text{ a } 65535$	Se devuelve el valor actual del contador CNTn o también puede fijarse un valor entre 0 y 65535. Con + o - se especifica el sentido de la cuenta.
P..	$Pn, Pn = x$ $Pn.b, Pn.b = y$ $Pn\sim, Pn.b\sim$	$n = 0 \text{ a } 2$ $b = 0 \text{ a } 7$ $x = 0 \text{ a } 255$ $y = 0 \text{ a } 1$	Se lee o escribe un puerto de 8 bits (o un único pin). Con ~ se invierte el puerto o el pin.
S..	$Sn, Sn = x$	$n = 0 \text{ a } 7$ $x = 0 \text{ a } 1023$	Se programa uno de los pines PWM como salida de señal para servos RC. Para centrar el servo se introduce $x = 512$.
W..	$Wn, Wn = x$	$n = 0 \text{ a } 7$ $x = 0 \text{ a } 1023$	Se programa uno de los pines PWM como salida de señal PWM (Modulación en Ancho de Pulso). El ancho del pulso puede ajustarse en un rango de 0 a 1023.
D..	$Dn =$ x, x, x, x, x, x, x, x	$n = 0 \text{ a } 1$ $x = 0 \text{ a } 255$	Control de un MAX7219, en el cual pueden conectarse hasta 64 LEDs. Pueden enviarse 8 bytes de datos x al mismo tiempo.
R..	$Rn, Rn = x$	$n = 0 \text{ a } 6$ $x = 0 \text{ a } 255$	Lectura o escritura del registro de control: R0: dirección de datos de P0, 0=entrada, 1=salida R1: dirección de datos de P1, 0=entrada, 1=salida R2: dirección de datos de P2, 0=entrada, 1=salida R3: reservado R4: frecuencia del PWM, 0=1 kHz, 1=2 kHz, 2=4 kHz R5: retardo del PWM, 0=sin retardo, hasta 99=más lento R6: brillo del primer MAX7219, 0=oscurito, hasta 15=claro R7: brillo del segundo MAX7219, 0=oscurito, hasta 15=claro
T..	Tn	$n = 0 \text{ a } 6500$	Retardo temporal de $n * 10 \mu s$.
X..	Xn	$n = 0 \text{ a } 65535$	Todos los comandos de la línea que van después de la X, se repiten n veces.
?	?		El display muestra una guía de referencia de todos los comandos implementados.

tanto mayúsculas como minúsculas. Para cada comando hay una función especial en el código que interpreta las posibles sintaxis. En el **listado 2** pueden distinguirse las funciones para el comando del puerto.

En el más simple de los casos, se chequea el estado actual de un puerto con $P0, P1$ o $P2$. Como resultado, el intérprete envía de vuelta un número de 0 a 255. Para cambiar el estado de un puerto, éste también puede recibir el valor directamente ($P0 = 0 \text{ a } 255$). La función `PortCmd()` identifica automáticamente si se trata de una lectura o una asignación, pues es capaz de detectar el "=".

Aparte, también es posible direccionar bits indi-

viduales de un puerto. Para ello nos servimos de los números de pin dentro de cada puerto, $P0.0, P0.1, \text{ hasta } P0.7$. La función también detecta aquí el ".", para diferenciar entre puerto o pin. Finalmente, puede hacerse 'toggle' (cambiar al estado opuesto) en un pin de puerto, especificando el símbolo "~" (por ejemplo, $P0.0\sim$).

Set de instrucciones

Con los comandos de puerto descritos anteriormente ya podemos controlar muy bien una tarjeta de entradas/salidas digitales con relés y optoacopladores. Siempre hemos de configurar la dirección de los datos para cada uno de los pines del

cable USB-IO24. Tras cada reset, los pines se configuran por defecto como entradas. Esto ha de tenerse en cuenta a la hora de desarrollar nuestro propio hardware: las etapas de potencia han de disponer de un estado de seguridad. Con el comando *R* se describen distintos registros de control. R0, R1 y R2 almacenan la dirección de los datos de los tres puertos. El bit 0 de R0 establece la dirección de datos de P0.0, el bit 1 la de P0.1 y así sucesivamente. Un 1 significa salida y un 0 entrada. El comando *RO* = 15<CR> cambia la dirección de datos de los cuatros bits menos significativos de Port0 a salidas. Alternativamente también podemos introducir *RO* = \$0F<CR> o *RO* = %00001111<CR>. Con el símbolo del dólar se especifican cifras hexadecimales y con el “%” números binarios.

En la **tabla 1** se muestran todos los comandos soportados por el cable USB-IO24. Junto con los comandos mencionados anteriormente, existen otras instrucciones básicas.

Para aplicaciones con sensores resulta especialmente importante el comando *A* para la lectura de entradas analógicas. Pueden digitalizarse hasta ocho entradas analógicas. Los pines correspondientes A10 a A17 no necesitan ser configurados adicionalmente, con el primer comando *An* se programan como entradas analógicas de forma automática. Ya que en principio es posible enviar varios comandos a la vez, también lo es tomar lecturas de distintas entradas analógicas simultáneamente. La serie de comandos: *A0 A1 A2*<CR> lee las tres primeras entradas analógicas y devuelve sus valores como respuesta. Los distintos valores en una misma respuesta están separados por espacios.

El opuesto de la entrada analógica es la salida analógica. El microcontrolador utilizado no soporta salidas analógicas directamente, pero mediante el PWM podemos generar señales de hasta ocho valores analógicos. Las señales PWM se activan con el comando *Wn*. Al igual que con las entradas analógicas, no se necesita ninguna configuración extra; la frecuencia del PWM puede ajustarse con el registro de control R4. Para obtener un valor analógico a partir de una señal PWM necesitaremos un filtro paso bajo. En el más simple de los casos basta con una red RC; aunque lo mejor es optar por un amplificador operacional y diseñar un filtro de mayor orden, por ejemplo, un Butterworth.

Mediante los comandos presentados anteriormente podremos realizar multitud de aplicaciones estándar, con entradas y salidas tanto analógicas como digitales. Si lo pensamos bien, en teoría pueden conectarse hasta 127 cables USB-IO24 a un ordenador, lo cual supone $127 * 24 = 3048$ entradas/salidas digitales! Estaremos bien cubiertos si necesitamos una solución especial en el área de la electrónica para hobbies, lo cual suele pasar a veces. Se han añadido algunos comandos adicionales más potentes, descritos en las aplicaciones de ejemplo más adelante.

Comandos para servos de radiocontrol (RC)

Los servos de RC utilizados en el modelismo y de distintas potencias son también muy útiles en proyectos de robótica y otros dispositivos mecánicos. Éstos regulan su posición de acuerdo a un valor fijado anteriormente. Dicho valor se codifica a modo de pulso con una duración de entre 1 y 2 ms, repetido cada 20 ms (figura 4).

Generar esta serie de pulsos no es una tarea que le podamos dejar completamente al PC. La temporización aquí es crítica, especialmente en cuanto al ancho de los pulsos. Una desviación leve ya provoca un zumbido audible (lo cual viene acompañado de un aumento del consumo). Por ello está el comando *Sn*, con el cual se pueden generar señales PWM o alternativamente hasta ocho señales para servos. También es posible generar señales de ambos tipos a la vez. Dada la estructura de timers en el R8C25 al menos cuatro pines de Port2 han tener siempre la misma temporización; por ello, los pines P2.0 a P2.3 generan señales PWM, mientras que los pines P2.4 a P2.7 se encargan de las señales para servos. Los nuevos servos de RC se caracterizan por su alta velocidad de funcionamiento. En principio esto es una ventaja, pero para llevar a cabo movimientos intencionadamente lentos, tendremos que especificar muchas posiciones intermedias dentro del recorrido completo. No obstante, esto

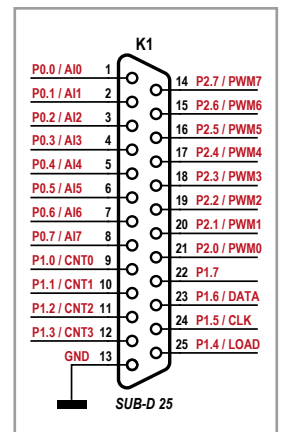


Figura 3. Asignación de pines del conector sub-D.

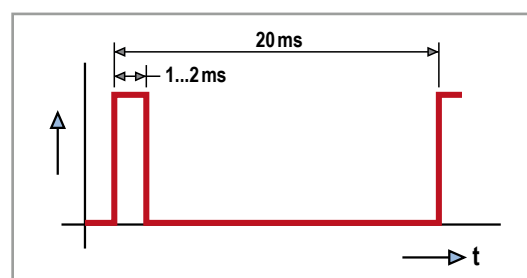


Figura 4. Serie de pulsos para servos RC.

a veces conduce a una especie de movimiento tembloroso en el servo, y en un brazo robótico puede hacer que vibre. Por ello, mediante el registro de control R5 podemos reducir la velocidad. El retardo se aplica directamente a la hora de generar los pulsos, lo cual reduce las vibraciones del servo considerablemente. El valor de R5 también afecta simultáneamente a la hora de generar las señales PWM.

Posición o velocidad de giro

Una de las áreas más extensas es el control de motores de corriente continua con baja potencia,

cuentan los pulsos (flancos) resultantes. El total hace de acumulador para la posición, o según el intervalo temporal es capaz de dar la velocidad de giro. Ambos se determinan mediante el comando *Cn*. El cable USB-IO24 dispone de cuatro entradas, CNT0 a CNT3, que pueden conectarse internamente en el controlador a un contador de 16 bits. El comando *C0+<CR>* cuenta ascendente según CNT0, mientras que con *C0-<CR>*, descendente. El estado del contador puede leerse con *C0<CR>*.

El sentido de la cuenta puede invertirse en cualquier momento, sin cambiar el estado del propio contador. De esta manera es posible seguir la posición actual en un eje siempre. Sólo tenemos que cambiar el sentido de la cuenta cada vez que cambie la dirección de giro. Por otro lado, para determinar la velocidad de giro, primero ponemos el contador a 0 mediante *C0=0<CR>* y tomamos la lectura posteriormente, al transcurrir cierto tiempo. Entonces, el valor del contador servirá como medida de la velocidad del motor.

Displays LED

El MAX7219 [4] de Maxim es un circuito universal capaz de controlar hasta 128 LEDs. Como componentes externos no se necesita nada más que una resistencia. Elektor ya presentó un display de 7 segmentos con el cual utilizar este chip [5], también pueden realizarse displays matriciales. En nuestro cable USB-IO24 pueden conectarse dos de estos chips (en cascada). Esto permite controlar hasta 128 LEDs individualmente (o bien 2 displays de 8 caracteres, así como combinaciones de ambos). Los datos se transfieren en serie a los displays. Sólo han de conectarse las tres líneas de DATA, CLK y LOAD a los pines del mismo nombre del MAX7219. Con el comando *Dn* se transmiten 8 bits de datos a la vez, por ejemplo *D0=48,109,121,51,91,95,38,105<CR>*. Los bytes del MAX7219 se almacenan en los registros Digit 0 a Digit 7; cada bit corresponde a un LED conectado. Para comprobar el orden exacto, hemos de echar un vistazo a la hoja de datos del chip.

PIOCable-Tool

Pero, ¿cómo es de complicado el funcionamiento? ¿Dónde se encuentran los drivers necesarios? ¿Hemos de escribir un programa por nuestra cuenta para probar las funciones? Esta son algunas de las preguntas típicas cuando queremos añadir nuevo hardware a nuestro PC. Pero por

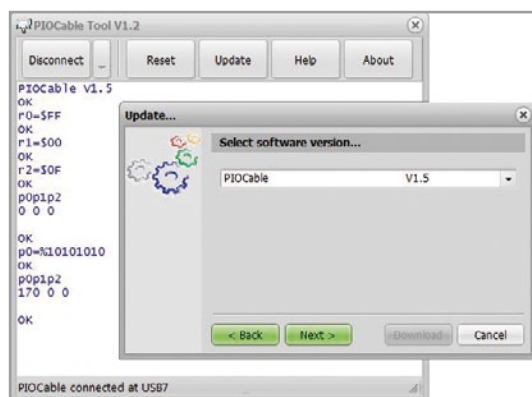


Figura 5.
Software para PC
"PIOCable-Tool".

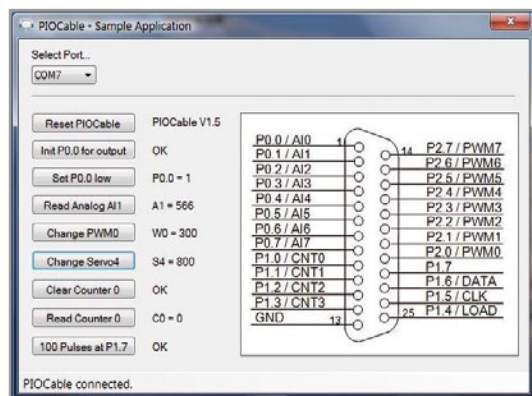


Figura 6.
Aplicación de demo en C#.

utilizados en muchos proyectos. En el caso más sencillo basta con un transistor bipolar, y para corrientes mayores, con uno de efecto de campo; y cuando se requiere cambio de sentido, un relé o un puente completo. La velocidad de giro se varía utilizando una salida PWM. Lo difícil viene cuando queremos regularla o avanzar hasta una posición determinada. Por ello, hay que utilizar un tacómetro de pulsos en el eje, por ejemplo un disco graduado que corte un haz de luz. Se

suerte, el funcionamiento del cable USB-IO24 que aquí presentamos resulta bastante simple. Tras conectarlo a un puerto USB del PC, Windows buscará el driver correspondiente. Los drivers de FTDI se instalan automáticamente, sin necesidad de hacer nada más (si tenemos conexión a Internet). Si surgen problemas, en la página web de FTDI encontraremos una guía para la instalación del driver VCP necesario (VCP = Virtual Com Port).

Finalmente iniciamos el programa para PC PIOCable-Tool, creado especialmente para el funcionamiento y prueba de todas las funciones (**figura 5**), y que puede descargarse en la página web de este artículo [6]. Así mismo, aquí también puede encontrarse el propio código fuente de PIOCable-Tool. Esta herramienta se ha generado con Delphi XE2; dicho código también puede compilarse utilizando versiones anteriores de Delphi.

Una vez iniciado el software, tras pulsar el botón "Connect", se buscará automáticamente el cable. Si la búsqueda no tiene éxito, tendremos que mirar en el administrador de dispositivos en qué puerto COM está direccionado el cable, e introducirlo manualmente en PIOCable-Tool.

PIOCable-Tool es básicamente uno de los llamados programas terminal. Todas las entradas se envían directamente al microcontrolador del cable USB-IO24, que por su parte responde directamente. De esta manera pueden introducirse todos los comandos manualmente, y comprobar las respuestas en cada uno de los pines (sirviéndonos, por ejemplo, de un osciloscopio).

Para que pueda continuarse el desarrollo del firmware del cable, con PIOCable-Tool es posible cargar nuevas versiones de software en el microcontrolador. El controlador de la placa montada incorpora un bootloader por defecto. Las actualizaciones de software posteriores pueden encontrarse en Elektor [6] o bien en la web TinyBasic del autor [7].

Quien quiera utilizar su propio firmware, naturalmente puede cargarlo mediante el bootloader. El entorno de desarrollo "High-performance Embedded Workshop" así como el compilador de C correspondiente pueden descargarse de la página web de Renesas [8]. Si disponemos de un programador adecuado, el controlador también puede programarse sin necesidad del bootloader. En tal caso, hemos de hacer un pequeño cable adaptador, equipado con un conector MicroMatch

(véase en la lista de materiales, bajo la etiqueta "Opcional"). Al programar de forma convencional, tanto el bootloader como el firmware original se sobrescribirán.

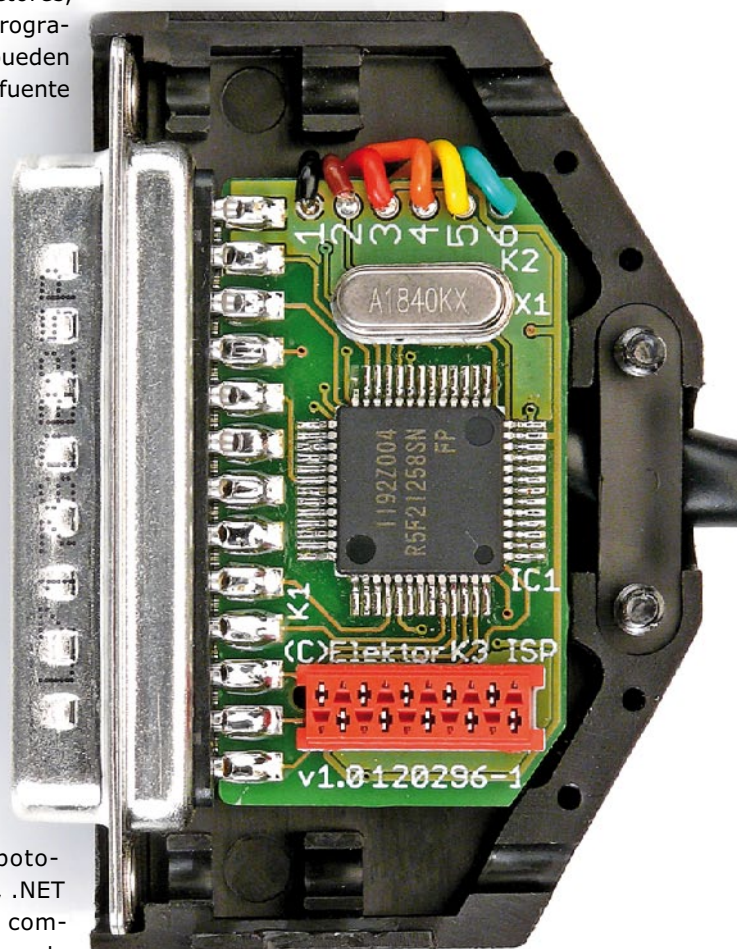
Aplicación de demo

Para aplicaciones simples (quizás un comprobador para servos) basta con introducir comandos con ayuda de PIOCable-Tool. Para el resto de aplicaciones, tiene más sentido diseñar nuestro propio software para PC, que se encargue de enviar los correspondientes comandos al cable USB-IO24.

De entre los lectores, los que sepan programar en Delphi pueden tomar el código fuente de PIOCable-Tool como base.

Resulta especialmente sencillo codificar nuestras propias aplicaciones también con el lenguaje de programación de Microsoft C# para .NET (existe una versión "Express" gratuita del entorno de desarrollo). Junto a los elementos de control típicos como menús, botones o etiquetas, .NET también ofrece componentes para puerto serie, con los cuales podremos acceder a nuestro cable con gran facilidad.

Primero creemos un componente del tipo *Serial-Port* en un formulario. Dicho componente ha de tener un nombre, por ejemplo "PIOPort". El resto de ajustes se hacen posteriormente en el código fuente. Ahora, añadimos un botón en el formulario y escribimos en esta rutina `OnClick` las siguientes líneas:




```
PIOPort.ReadTimeout = 100;
PIOPort.BaudRate = 115000;
PIOPort.NewLine = "\r\n";
PIOPort.PortName = "COM5";
PIOPort.Open();
```

En PortName hemos de introducir el puerto por el cual se accede al cable actualmente. Si todo va correctamente, se establecerá conexión con el hardware.

El autor ha creado una aplicación de demo en C# [6], en la **figura 6** puede verse una captura de pantalla. En el código se hace una llamada al fragmento anterior cuando el usuario selecciona uno de los puertos disponibles en la 'combo box'. La línea de configuración `PIOPort.NewLine = "\r\n"` notifica al componente `PIOPort`, que todas las llamadas de `PIOPort.WriteLine()` y `PIOPort.ReadLine()` siempre han de terminar con `<CR><LF>`. Al enviar un comando no debemos dar esos caracteres adicionalmente; y al leer una respuesta, dicha lectura tiene que continuar hasta el final de la línea.

Esto hace que resulte fácil escribir, por ejemplo, una sencilla función que ponga el pin de puerto P0.0 a 1. Añadimos otro botón y en el formulario escribimos la siguiente línea en la rutina `OnClick`:

```
PIOPort.WriteLine("r0=lp0.0=1");
```

El comando configura primero el pin como salida, después inmediatamente lo pone a nivel alto... ¡y listo! Resulta algo más complicado, por ejemplo, si queremos tomar la lectura de un valor analógico. En este caso primero hemos de esperar a

la respuesta, y después realizar la lectura `PIOPort.ReadLine()`. En el código de la aplicación de demo puede verse cómo se implementa esto. Y ahora, ¡ya nada se interpondrá en tu camino para programar tu propio software de medición y control!

(120296)

Enlaces

- [1] www.ftdichip.com/Products/Cables/USBTTL-Serial.htm
- [2] www.elektor.es/080213
- [3] www.renesas.com/products/mpumcu/r8c/r8c2x/r8c25/index.jsp
- [4] www.maximintegrated.com/datasheet/index.mvp/id/1339
- [5] www.elektor.es/081154
- [6] www.elektor.es/120296
- [7] www.tinybasic.de
- [8] www.renesas.eu/products/tools/ide/ide_hew/index.jsp



SPECIAL: SAVE 50% SPECIAL: SAVE 50% SPECIAL

Celebrate Circuit Cellar's **25th Anniversary**



\$25 Print or Digital :: **\$50** Combo

Celebrate Circuit Cellar's 25th year of bringing readers insightful analysis of embedded electronics technology.

Visit **www.circuitcellar.com/el912** to take advantage of these great deals.

BONUS OFFER! BONUS OFFER! BONUS OFFER! BONUS OFFER!

Sign up today and you'll also receive the **Special 25th Anniversary Edition** with your subscription!



¿Qué se está cocinando?

¡Esperamos que no sea la nueva SMPSU de Elektor!

Transformador de aislamiento válido para 115 y 230 VAC; esta serie usa una huella idéntica para diferentes rangos de potencia de salida. Muy conveniente.

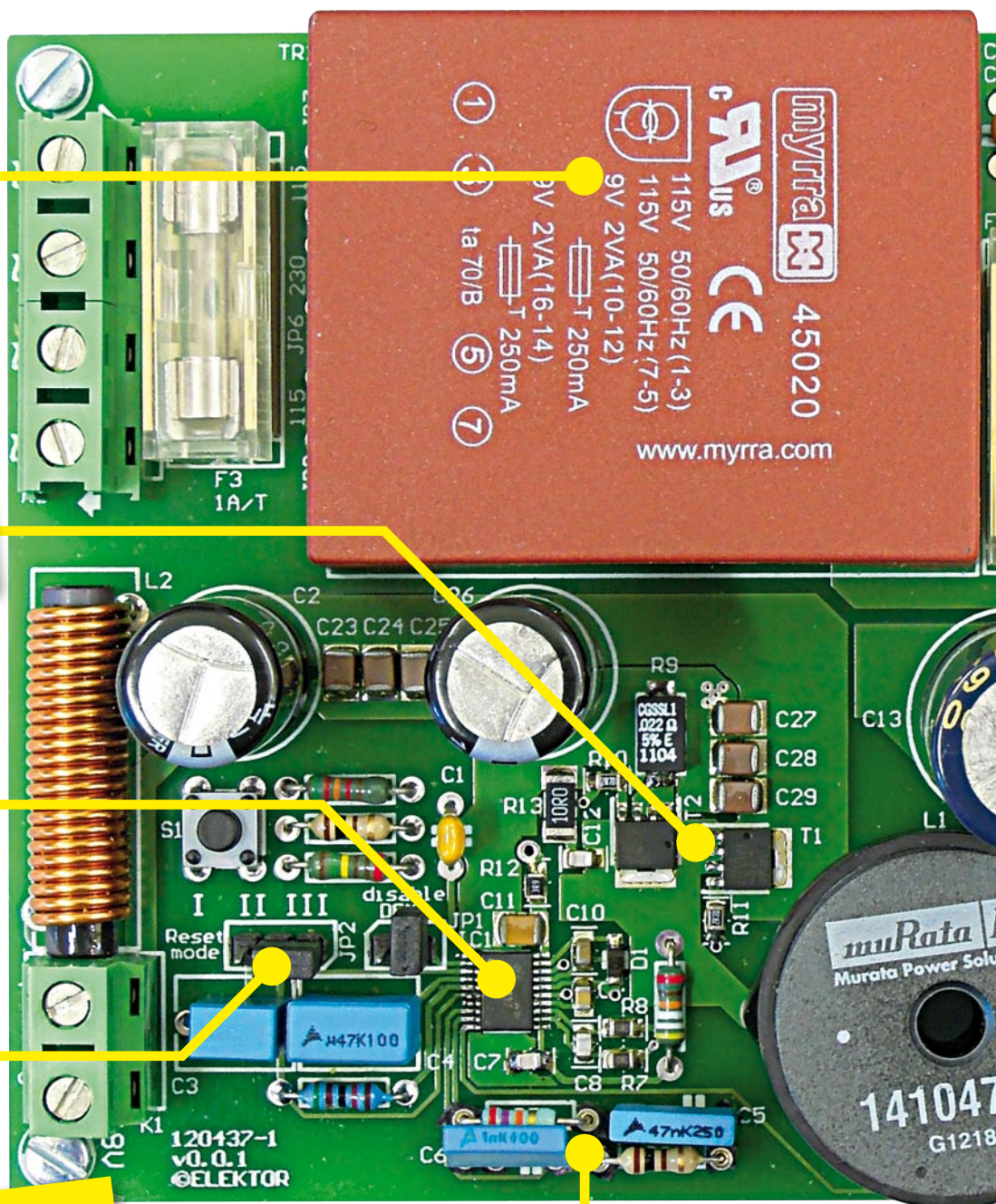
T1, T2: FETs especiales de alta tensión, de la casa NXP, con $R_{DS(on)}$ extremadamente baja.

IC1 e IC3 son difíciles de soldar a mano. Aunque manejables, se recomienda el uso de una estación de refundido de aire caliente. Un horno de reflujo sería ideal para este trabajo.

El modo "reinicio" se selecciona con JP2 (pequeños problemas, elemento de apagado, ciclo por ciclo)

Señalar la 'v0.0.1' en la placa.

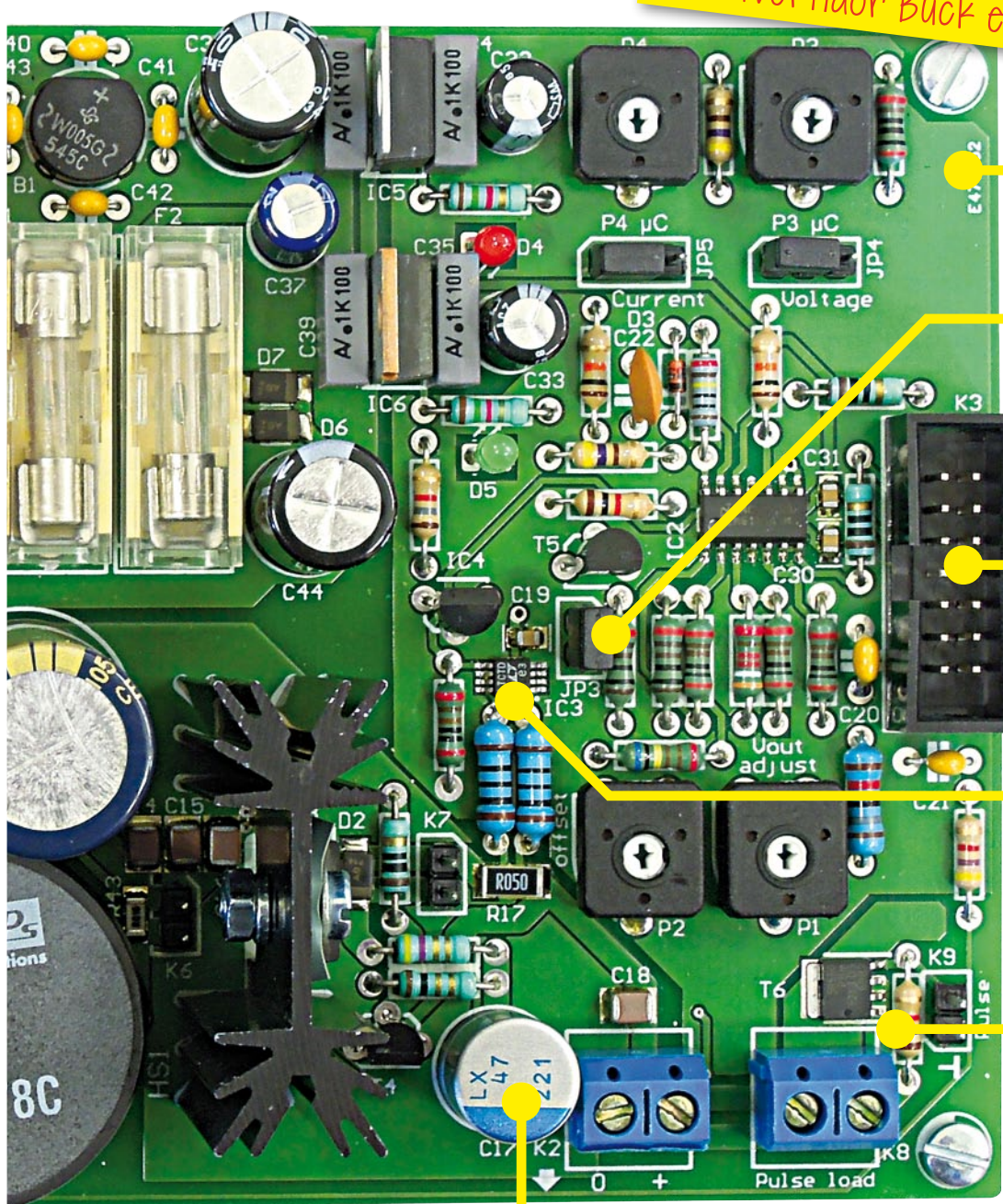
El ajuste fino del filtro de realimentación (C5, C6, R5, R28) se determina experimentalmente.



Un nuevo proyecto toma forma en nuestros Laboratorios de Elektor. En las capaces manos del diseñador Ton Giesberts, se está ideando ahora una fuente de alimentación conmutada (SMPSU) y ya está en sus primeras pruebas de las etapas de diseño. Vamos a dar una primera ojeada furtiva...

Thijs Beckers (Editorial & Laboratorios de Elektor)

En el fondo, la fuente de alimentación es un convertidor Buck en modo corriente.



PCB de doble capa.

JP3 proporciona la opción de elegir entre limitación de corriente hardware y/o software.

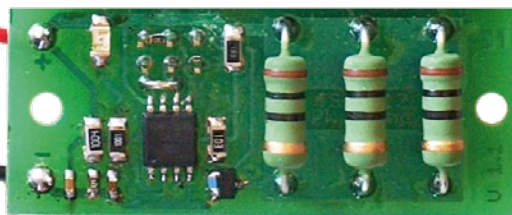
Interfaz con C; software y diseño a ser implementado.

¿IC3 tiene los terminales NC (No Conectado) cortocircuitados por pistas? Se investigará.

T6: sección implementada temporalmente con objetivos de pruebas de carga conmutada.

C17 tiene especificaciones de diseño, que sólo se pueden cumplir con un condensador Nichicon especial. ¿Disponibilidad?! Los resultados de las medidas de ESR en C17 no han sido satisfactorios (0,4xx ohmios). Las medidas realizadas 'del modo tradicional' no fueron concluyentes al principio. El tema resultó ser un lazo de masa entre los equipos de medida que distorsionaron las medidas.

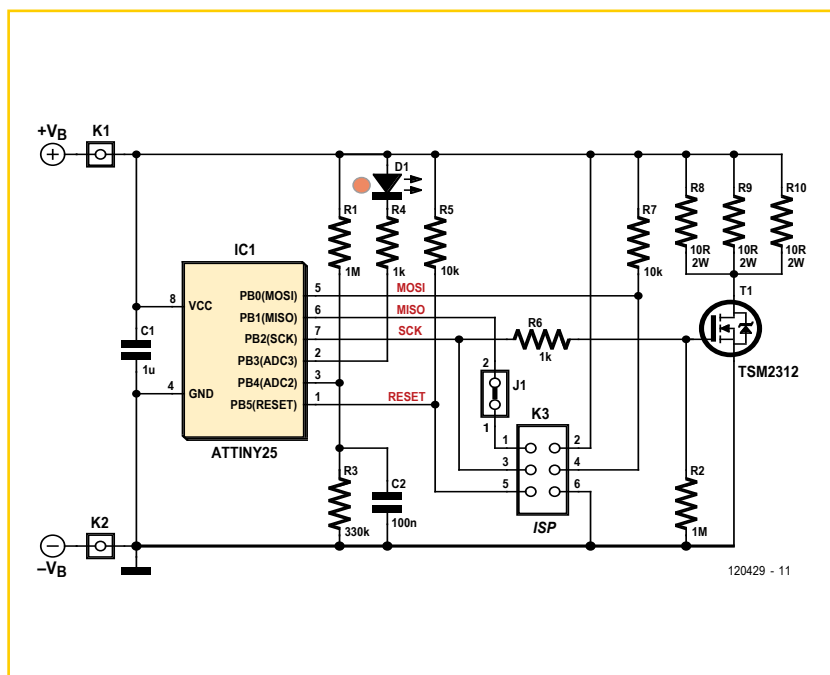
¡Pequeño pero matón!



Dr. Thomas Scherer (Alemania)

La pregunta del millón: ¿por qué en este circuito no se graba ningún firmware con un programador AVRISP-MKII por ISP? (iverídico!)

Llevaba tiempo dándole vueltas, probando esto y aquello, hasta que finalmente conseguí arrojar algo de luz al asunto. Primero, expliquemos las bases: quería diseñar un circuito lo más sencillo posible, para poder llevar a cabo el balanceo de mis nuevas baterías LiFePo. Debe haber alguna manera de hacerlo correctamente utilizando integrados especiales, pero yo estoy habituado a resolver los desafíos mediante sencillos dispositivos electrónicos con controladores AVR y algo de código. Aquí nos servimos de un ATtiny25 de ocho pines, que ya habíamos utilizado de distintas maneras (véase [1] y [2]), y difícilmente podía



saber yo que esta vez me iba a dar problemas. Pero por un lado está "lo que es realmente", y por otro "lo que creemos que es".

Tras medir por enésima vez las pistas y comprobar el montaje, mi primera suposición fue: es posible que el fabricante no haya configurado el fusible "SPIEN", y por lo tanto no podamos grabar nada mediante ISP/SPI. Ya que se trataba de la variante SMD, tampoco podíamos programar el controlador mediante una placa STK500 utilizando "High Voltage". Mediante cables, la soldamos directamente a la placa, sirviéndonos de los pines HV de la STK500, lo cual no dio resultado, a pesar de que la programación mediante HV debería funcionar siempre. Posteriormente, le soldé a un Tiny25 SMD unos cablecitos, de modo que el controlador ajustase en uno de los zócalos DIL de la STK500. Funcionó, y el SPIEN se configuró correctamente. Así que indudablemente, el error estaba en el circuito. ¿Pero dónde? Tras investigar un poco, se me ocurrió la idea de que la resistencia de 1 k Ω R4 en serie con el LED D1 podía atenuar la señal demasiado a la hora de programar. Pero no, otra vez estaba equivocado, ya que el pin 2 (PortB.3) en ningún caso está conectado con K3. Llegados a este punto, deduje que aunque T1 era físicamente muy pequeño, al tratarse de un MOSFET de cierta potencia, podría tener una capacitancia de entrada del mismo orden. Un vistazo a la hoja de datos confirmó la hipótesis: $C_{iss} = 900$ pF. ¡Ajá!

De modo que aumenté la resistencia de puerta R6 a 10 kΩ, conecté el AVRISP en K3, hice click en "Program" en AVR Studio, y listo, ya estaba todo en orden dentro del pequeño microcontrolador. Pero, ¿y mi premio por resolver la cuestión? Bueno, si has tenido el mismo problema, y lo has solucionado rápidamente consultando las especificaciones, entonces puedes reírte (justificadamente) un poco de mí ;-).

(120429)

[1] www.elektor.es/100691

[2] www.elektor.es/070797

¿¡Planos de masa perdidos!?

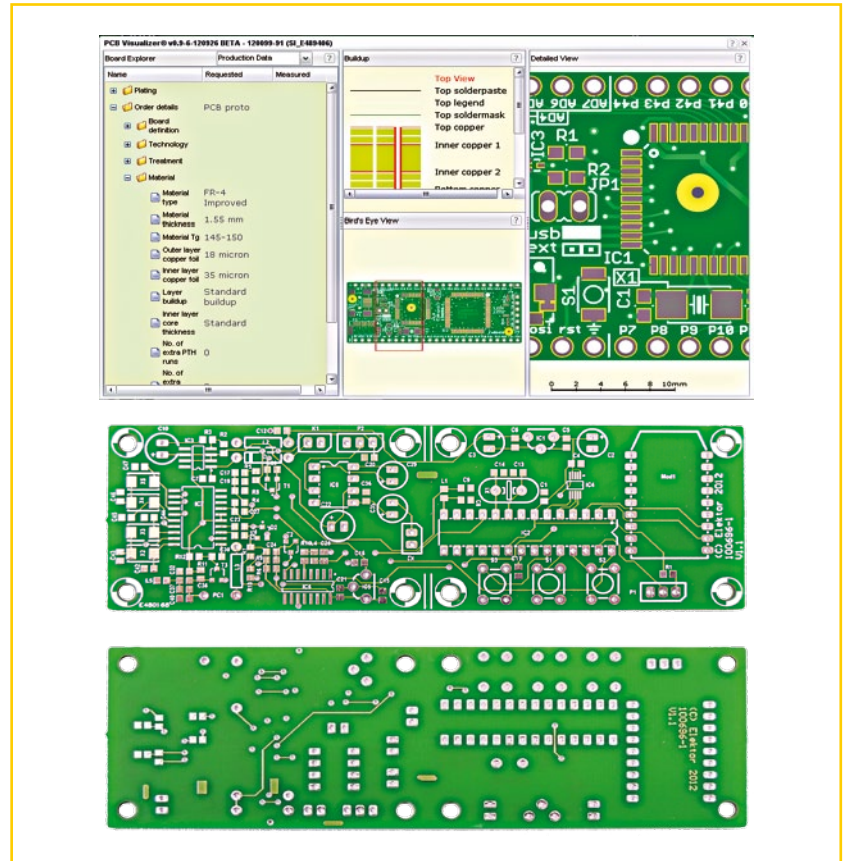
Verifica tu placa con PCB Visualizer!

Thijs Beckers (Editorial & Elektor.Labs)

No todo fue tranquilo cuando Luc, un compañero del laboratorio, estuvo trabajando en un proyecto de escáner de radio para aviación. Varios componentes usados por el autor ya no estaban en producción, aunque probablemente si estuvieran aún disponibles en distribuidores de pequeñas tiendas y en tiendas de stocks antiguos. Por lo tanto, para asegurar que el proyecto puede ser reproducido por nuestros lectores durante cierto tiempo en el futuro, varios componentes tuvieron que ser sustituidos por otros más modernos. Sin embargo, la nueva distribución de la PCB (con los nuevos componentes) y el pedido de todos los componentes necesarios para el montaje del prototipo, se hizo sin ningún problema. Hasta que la PCB prototipo llegó por correo.

En algún sitio durante el proceso de diseño, usando nuestra suite de trabajo CAD Altium Designer, Luc 'retiró' (ocultó) los planos de masa superior e inferior para poder tener una mejor perspectiva de las señales 'rutadas'. Cuando acabó, Luc generó los ficheros Gerber para la producción de la PCI prototipo usando nuestro fichero estándar .outjob y los envió a nuestro proveedor de prototipos Eurocircuits [1]. No se detectó ningún error durante el proceso y no fue hasta que Luc recibió las PCBs en el correo, que pudo ver que los planos de masa retirados no estaban en la placa física (ver las imágenes). Esto hizo que las PCBs fuesen inservibles ya que la mayoría de las conexiones a masa, si no todas, se realizaban usando los planos de masa. Deshaciendo sus acciones, Luc se acordó de que había ocultado los planos que faltaban. Como consecuencia de ello, todo eso que se había ocultado quedó omitido de los archivos Gerber. Por supuesto, esto podría haberse evitado echando un vistazo a los archivos con un programa visualizador de ficheros Gerber antes enviarlos a producción, pero la premura de tiempo y la distracción por otras obligaciones llevo de alguna forma a saltarse este paso de verificación adicional.

Una vez descubierto el error, el nuevo servicio 'PCB Visualizer' de Eurocircuits (ver capturas de pantalla) fue mencionado como, posiblemente, la mejor opción de comprobación de programas de visualizadores Gerber que nuestros laboratorios



hayan usando. Esta herramienta 'en línea' acepta nuestros archivos volcados (Extended Gerber o CadSoft Eagle V6) y, cómodamente, muestra una vista previa de la apariencia que tendrá la PCB tras el proceso de producción. Cualquier error de diseño o de pedido puede ser cortado de raíz en este momento ya que, mientras se muestra una descripción ampliada de los parámetros a la izquierda de la pantalla, podemos ver la placa al detalle a la derecha de la misma.

Una vez que hemos confirmado que todo está como se ha diseñado, podemos enviar la placa a producción y esperar a que llegue por correo, fabricada según nuestras especificaciones exactas, con la esperanza de no tener ninguna pista ausente, pobre de cobre o con polígonos.

(120568)

Enlaces en Internet

[1] www.eurocircuits.com

Simplificando Linux Embebido (6)

Servicios de red y servidor

Benedikt Sauter [1]

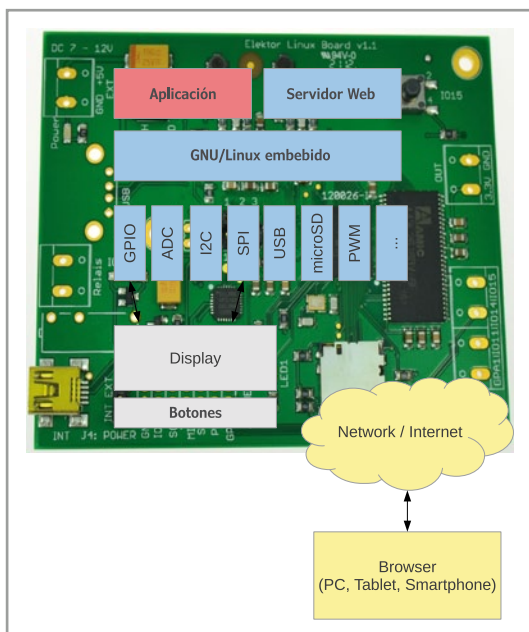


Figura 1.
Vistazo de la aplicación.

El sistema modular de Linux es perfecto para utilizarlo como plataforma completa de medida, control y regulación. Puede programarse un control a medida en el lenguaje que deseemos, mientras que mediante el entorno de usuario puede supervisarse mediante un PC, tablet o smartphone. En este capítulo convertimos la placa Linux de Elektor en una red domótica. Aparte, cubrimos otros temas como nuestra propia aplicación para servidor y el lenguaje basado en scripts Lua.

En esta serie ya hemos cubierto los temas esenciales relacionados con Linux, desde el boot loader en el kernel hasta la instalación de drivers de dispositivos. No obstante, hasta ahora no habíamos tratado uno de los conceptos básicos como es desarrollar una aplicación grande; por ejemplo, con un sistema “lógico” en segundo plano, un entorno de usuario web para la configuración y funcionamiento y una sencilla unidad de control con display y botones.

Mensajes entre aplicaciones

En la **figura 1** se ha representado un caso típico. En el centro tenemos un sistema Linux con su propia aplicación y un servidor web, para representar los distintos dispositivos en el navegador. Opcionalmente tenemos un display y algunos botones. Para controlar el display podemos hacer uso de una aplicación dedicada. La verdadera aplicación principal se trata de un único proceso, que se ejecuta en segundo plano junto con muchos otros. En Linux puede utilizarse una

aplicación de este tipo llamada *Daemon* o servicio (o servidor). Una vez iniciada, ésta funciona sin requerir la intervención del usuario. La comunicación con otras aplicaciones (por ejemplo el entorno web) puede implementarse de distintas maneras (en informática, este intercambio entre programas o procesos individuales se trata en el denominado Inter-Process-Communication/IPC). Podemos elegir entre pipes, fifos, memoria compartida, sockets, entradas y salidas estándar, archivos, bases de datos y otros programas de soporte. Lo que se utilice depende de distintos factores. ¿Están las aplicaciones programadas en el mismo lenguaje? ¿Ha de poder ejecutarse el software en otros sistemas operativos? ¿Qué tasa de transferencia requiere la comunicación entre aplicaciones?

Uno de los métodos más extendidos son los clásicos sockets de red. Estos sockets tienen la gran ventaja de que las aplicaciones pueden intercambiar datos y comandos a través de la red. Sirviéndonos de un socket, podemos enviar y

recibir mensajes fácilmente, disponiendo de las funciones y clases en los distintos lenguajes de programación.

Por este motivo, hemos optado por los sockets. Ya que el navegar por Internet se basa en sockets estándar, también aprenderemos cómo acceder a datos directamente desde la red. Por esta razón, el siguiente paso consiste en conectar nuestra placa Linux a Internet.

Conexión a Internet

El requisito es disponer de un router en la red con conexión a Internet, y que éste sirva de puerta de enlace entre los distintos dispositivos. Lógicamente, también hace falta un adaptador USB/LAN en el puerto USB de la placa y cargar previamente los drivers para el adaptador de red [2].

Para incorporar un dispositivo a la red, hemos de configurar la dirección IP y otros parámetros como la puerta de enlace y el servidor DNS. Lo podemos hacer manualmente o mediante DHCP ("Dynamic Host Configuration Protocol"). Si utilizamos el protocolo DHCP, el router configurará automáticamente la placa Linux de Elektor dentro de la red local (véase la **figura 2**). Si la dirección IP se asigna por DHCP, podemos por ejemplo probar el servidor web con la dirección www.elektor.es, utilizando el comando `ping` (ver la **figura 3**).

Si no disponemos de un servidor DHCP, tendremos que introducir los parámetros de la conexión manualmente.

Dirección IP

Para ello, introducimos:

```
ifconfig eth0 <dirección-ip>
```

La etiqueta `<dirección-ip>` ha de sustituirse por una dirección IP libre en la red. Para ver si ésta se ha asignado correctamente, podemos remitirnos al router mediante el comando `ping`.

```
ping <dirección-del-router>
```

La etiqueta `<dirección-del-router>` ha de reemplazarse por la dirección IP correspondiente al router.

Dirección del router

Para enviar paquetes de red a través del router, ha de establecerse la ruta por defecto de la red.

```
root@gnublin:~# udhcpd
udhcpd (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.190...
Lease of 192.168.0.190 obtained, lease time 864000
Setting IP address 192.168.0.190 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Figura 2.
Configuración DHCP de la dirección IP.

```
root@gnublin:~# ping www.elektor.de
PING www.elektor.de (94.236.12.177): 56 data bytes
64 bytes from 94.236.12.177: seq=0 ttl=108 time=89.798 ms
64 bytes from 94.236.12.177: seq=1 ttl=108 time=76.585 ms
```

Figura 3.
Respuesta de un servidor tras hacer ping.

En GNU/Linux puede introducirse en la línea de comandos:

```
route add default gw
<dirección-del-router>
```

La etiqueta `<dirección-del-router>` ha de reemplazarse por la dirección IP del router. Si queremos comprobar este paso por separado, podemos comunicarnos con cualquier ordenador de Internet haciendo `ping`.

```
ping 94.236.12.177
```

Aquí nos comunicamos de nuevo con el servidor web de Elektor. Si obtenemos una reacción positiva (sin pérdida de paquetes), podemos pasar al siguiente punto.

Dirección del servidor DNS

Para navegar por Internet con nombres de dominios, necesitaremos un servidor DNS para cada dispositivo de la red. El servidor DNS puede ser un dominio como `"www.elektor.es"` que determina la IP correspondiente.

Para establecer un servidor DNS para la placa Linux de Elektor, editamos el archivo `"/etc/resolv.conf"` (**figura 4**). Tras introducir...

```
nano /etc/resolv.conf
```

...se abrirá el editor. En el archivo está la entrada `"nameserver"`. Aquí establecemos el router local, que normalmente dispone de un servicio DNS. Si no dispone de servidor DNS local, podemos utilizar alguno de Internet. Los servidores DNS de Google son fáciles de recordar y resultan ideales para realizar pruebas, con la dirección IP 8.8.8.8.



Figura 4.
Configuración del servidor
DNS.

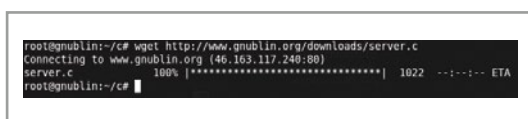


Figura 5.
Descarga del archivo
"server.c".

El archivo se guarda con "Ctrl+o". Podemos cerrar el editor mediante "Ctrl+x". Ahora, todo lector debería poder acceder a cualquier servidor en Internet utilizando su URL y el comando ping.

Escribiendo nuestros propios servicios para el servidor

Ahora que la placa está conectada a Internet, podemos descargar el archivo de ejemplo para nuestro propio servicio de servidor directamente. Con el programa "wget", que ya conocemos del Linux para PC, podemos obtener los archivos que queramos directamente de la red. La descarga del servicio de ejemplo para el servidor se muestra en la **figura 5**.

El servicio está programado en C. Para las primeras pruebas utilizamos el compilador de la placa. Si queremos continuar posteriormente con el desarrollo por nuestra cuenta, definitivamente habrá que trabajar con el toolchain del entorno de desarrollo en el PC.

En el **listado 1** puede verse el código fuente de la aplicación. Esencialmente, éste inicia un sencillo servidor que permanece a la escucha del puerto 5000. Cuando accede un dispositivo de la red, automáticamente envía una respuesta al sistema con la hora actual. Posteriormente podemos iniciar aquí un intérprete de comandos, que dependiendo de las instrucciones del cliente puede dar distintas respuestas o llevar a cabo acciones.

Para probar el servidor hemos de compilar el archivo en C:

```
gcc server.c
```

Una vez haya terminado el compilador, podemos iniciar el servidor:

```
./a.out
```

Ahora puede parecer que la aplicación actual bloquea la consola, y no podemos hacer nada más (excepto detener la aplicación con "Ctrl+c"). Pero no es así. Podemos "poner a dormir" a la aplicación del servidor mediante "Ctrl+z", de modo que sea posible llamar a otros programas o instrucciones desde la consola.

En ésta aparecerá:

```
[1]+  Stopped ./a.out
```

Si introducimos...

```
fg
```

...(que significa "Foreground"), la consola se bloqueará de nuevo. No obstante, puede liberarse otra vez pulsando "Ctrl+z". Entonces introducimos...

```
bg
```

...y el proceso del servidor volverá a estar activo en segundo plano.

Como ya se ha dicho anteriormente, nuestro servidor por sockets de prueba envía la hora actual tras ser consultado por un cliente. En la consola podemos probar esto con el programa "telnet", como parámetros se dan la dirección IP del servidor y el puerto:

```
root@gnublin:~/c# telnet 127.0.0.1 5000
```

```
Tue Sep 27 21:34:49 2011
```

```
Connection closed by foreign host
```

Si queremos mantener el servidor otra vez en segundo plano (por ejemplo, para detenerlo) simplemente tecleamos en la consola:

```
fg
```

Naturalmente, ahora podemos dar acceso a un entorno de usuario web cuando se le pregunta al

Listado 1: sencilla aplicación de servidor.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

int main(int argc, char *argv[])
{
    int listenfd = 0, connfd = 0;
    struct sockaddr_in serv_addr;

    char sendBuff[1025];
    time_t ticks;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    memset(&serv_addr, '0', sizeof(serv_addr));
    memset(sendBuff, '0', sizeof(sendBuff));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5000);

    bind(listenfd, (struct sockaddr*)&serv_addr,
        sizeof(serv_addr));

    listen(listenfd, 10);

    while(1)
    {
        connfd = accept(listenfd, (struct sockaddr*)
            NULL, NULL);

        ticks = time(NULL);
        snprintf(sendBuff, sizeof(sendBuff),
            "%.24s\r\n", ctime(&ticks));
        write(connfd, sendBuff, strlen(sendBuff));

        close(connfd);
        sleep(1);
    }
}
```

servidor. Y en lugar de devolver la hora, el servidor podrá controlar otra aplicación (como por ejemplo encender y apagar un LED). En la última entrega ya mostrábamos cómo [2].

Controlando a distancia por la red

Actualmente siempre accedemos a la consola de la placa desde “picocom” u otro programa terminal. Pero ya que ahora disponemos de una conexión de red, podemos abrir tantas consolas adicionales como queramos a través de ésta. Durante el desarrollo de aplicaciones esto supone una ventaja a tener en cuenta. En una de las consolas podemos, por ejemplo, abrir el editor con el programa, mientras que en otra podemos ejecutarlo directamente y ver cómo funciona en tiempo real.

Para establecer una conexión de red con la placa Linux de Elektor utilizaremos SSH (“Secure Shell”). Esta consola ofrece un canal con encriptación automática. La gran ventaja está en que ningún otro dispositivo de la red podrá leer lo que se envía a la placa.

SSH en Linux

En Linux el “ssh” forma parte de las herramientas estándar. En la consola del ordenador host introducimos la siguiente línea

```
ssh root@192.168.0.190
```

Aquí, la dirección IP ha de reemplazarse por la que se le asignó a la placa. Una vez conectada, se nos pedirá que introduzcamos la palabra “yes” (**figura 6**). Después veremos la consola estándar, y nos sentiremos como en casa.

Listado 2: script de inicio.

```
#!/bin/sh

if [ ! -d /var/log/lighttpd ]
then
    mkdir /var/log/lighttpd
    chown -R lighttpd:lighttpd /var/log/lighttpd
    chmod 777 /var/log/lighttpd/
    touch /var/log/lighttpd/error.log
fi

modprobe asix
udhcpc

echo 3 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio3/direction
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value

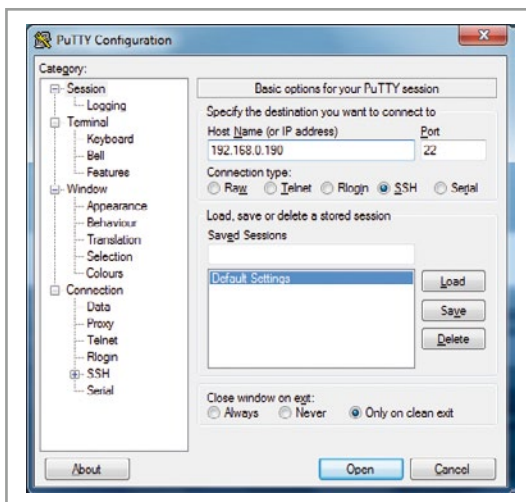
chown lighttpd:lighttpd /dev/lpc313x_adc
echo 1 > /dev/lpc313x_adc

chmod 666 /var/log/lighttpd/error.log
/etc/init.d/lighttpd start
```

Figura 6.
Acceso por SSH en Linux.

```
sauter@sauter-Inspiron-530:~$ ssh root@192.168.0.190
The authenticity of host '192.168.0.190 (192.168.0.190)' can't be established.
RSA key fingerprint is bf:7f:52:57:06:59:1f:cd:a6:a9:e3:cb:97:75:36:01.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.190' (RSA) to the list of known hosts.
root@gnublin:~#
```

Figura 7.
Acceso por SSH en Windows.



SSH en Windows

En Windows podemos utilizar el programa PuTTY [3]. En la **figura 7** se muestra la ventana principal de este software; aquí también tendremos que introducir la dirección IP de la placa. Tras hacer click en “Open” se preguntará si el *Fingerprint* del dispositivo es correcto (parte de la clave SSH utilizada como prueba de identidad del mismo). Hacemos click en “Yes”. En el Prompt hacemos “login as” como “root”. Tras confirmar con la tecla intro, nos habremos registrado en el sistema.

Lenguaje de programación Lua para controles sencillos

En el ejemplo de arriba hemos utilizado el lenguaje C, muy extendido, rápido y famoso. En un

```
require "elektor"
-- initialize hardware
initLED()
-- start main loop
print("blinking ...")
while 1 do
    setLED()
    wait(1)
    clearLED()
    wait(1)
end
```

Figura 8.
Programa de ejemplo
“blink.lua”.

Figura 9.
Programa de ejemplo
“adc_relay.lua”.

```
require "elektor"
-- initialize hardware
initButton()
initLED()
initRelay()
initADC(1)
-- start main loop
print("Please tune the resistor")
while 1 do
    value = getADC()
    print(value)
    if value > 500 then setRelay() else clearRelay() end
    wait(1)
end
```

sistema con Linux, aparte de C podemos usar otros muchos lenguajes más. Cada uno tiene sus inconvenientes y sus ventajas, y son aptos para distintas aplicaciones.

Uno de los más flexibles, rápidos y sencillos de utilizar es el lenguaje Lua. Se trata de un lenguaje interpretado, conocido principalmente por ser uno de los auxiliares a la hora de desarrollar juegos. El intérprete Lua ocupa tan sólo 120 KB y ofrece estructuras de programación bastante modernas. Para acceder a los archivos –y como ocurre en Linux, también a cualquier otro dispositivo o funciones de hardware– hemos de utilizar los comandos `io.open`, `io.read`, `io.write` e `io.close`. El autor ha desarrollado una librería en Lua para la placa Linux de Elektor, para que puedan utilizarse las funciones de hardware más importantes de forma fácil (GPIO y el conversor A/D).

Si queremos realizar un pequeño control de forma rápida, las ventajas de un lenguaje interpretado como este sobre C son enormes. En la placa podemos escribir directamente en el editor integrado (nano o vi) y ejecutar los programas inmediatamente.

Transferimos los ejemplos en Lua a la tarjeta SD haciendo uso de la conexión a Internet establecida recientemente:

```
mkdir lua
cd lua
wget http://www.gnublin.org/downloads/elektor.lua
wget http://www.gnublin.org/downloads/blink.lua
wget http://www.gnublin.org/downloads/button.lua
wget http://www.gnublin.org/downloads/adc_relay.lua
```

El listado en Lua de una simple luz parpadeando se muestra en la **figura 8**. Para empezar hemos de cargar las librerías de Elektor. El LED se inicia mediante la llamada `initLED()` y después se enciende y se apaga en un bucle infinito.

En la consola, iniciamos el programa:

```
lua blink.lua
```

La aplicación “button.lua” implementa un sencillo control del relé mediante el botón IO15 de la tarjeta. Y en la **figura 9** puede verse el código del ejemplo “adc_relay.lua”. En el bucle infinito, primero se lee el valor de tensión del conversor A/D. Si el resultado es mayor de 500, entonces el relé se cerrará, en caso contrario, se abrirá. Para probarlo, conectamos un potenciómetro en

Simplificando Linux Embebido

la entrada analógica, como hicimos en la anterior entrega [2].

Podemos encontrar suficiente documentación sobre Lua en Internet [4], y también algunos libros buenos [5].

Iniciar el servidor web y la aplicación automáticamente

Es un poco molesto tener que iniciar la aplicación del servidor a mano cada vez que iniciamos la placa (y a veces también configurar el hardware). Por ello, hemos reunido los comandos correspondientes en un script de inicio. El **listado 2** muestra dicho script para iniciar la aplicación del servidor web de la anterior entrega [2]. El archivo “start.sh” lo podemos descargar de Internet (ver la **figura 10**).

Si queremos que el script se inicie automáticamente al arrancar el sistema Linux, hemos de colocar una entrada en el propio archivo de script “/etc/rcS.d/S55bootmisc.sh” (los scripts de inicio se sitúan en el directorio “/etc/rcS.d”). Primero abrimos el archivo con el editor:

```
nano /etc/rcS.d/S55bootmisc.sh
```

Al final del archivo encontramos la entrada:

```
/bin/mkdir /var/run/sshd  
/usr/sbin/sshd
```

Ahora, añadimos ahí:

```
/home/root/start.sh
```

Hemos de guardar el archivo con “Ctrl+o” y cerrar el editor con “Ctrl+x”. Para que sea capaz de encontrar el archivo “start.sh”, éste ha de estar en el directorio “/root”. También hemos de activar el flag de ejecutable “x” para el archivo en concreto:

```
chmod +x /home/root/start.sh
```

A partir de ahora, la placa se conectará automáticamente a la red en el arranque, e iniciará el servidor web.

Acceso desde smartphones y tablets

En la última entrega mostramos cómo acceder mediante un navegador en el PC a una aplicación web demo. Lógicamente, esto también puede hacerse desde un smartphone o tablet. En la



Figura 10.
Descarga del script de inicio para el servidor web.



Figura 11.
Acceso desde un iPhone.

red en la que se encuentra la placa Linux de Elektor, hemos de disponer de un acceso WLAN para dispositivos móviles. La **figura 11** muestra el entorno en HTML en un iPhone. Con el botón en HTML puede encenderse y apagarse el LED en la placa.

Un consejo para los lectores que estén interesados: en la wiki de la comunidad [6] del proyecto Gnublin se tratan distintos temas de hardware y software. Entre otros, encontraremos cómo enviar y recibir datos mediante puertos I2C, SPI y UART (y transferirlos al espacio de usuario).

En el siguiente capítulo trataremos el tema que los lectores deseen (en el momento en el que se prepara este artículo, se está llevando a cabo una encuesta). ¡Y no aguantamos más! Por ahora podemos decir que estamos desarrollando una placa de expansión que ofrecerá puertos adicionales. ¡Más detalles en la próxima edición!

(120578)

Enlaces

- [1] sauter@embedded-projects.net
- [2] <http://www.elektor.es/120182>
- [3] www.putty.org
- [4] www.lua.org
- [5] www.lua.org/pil
- [6] <http://wiki.gnublin.org>

Arduino en marcha (3b)

Part 3b: Los sensores hablan a los teléfonos Android

David Cuartielles
(España)
Andreas Goransson
(Suecia)

El último mes hicimos una introducción a la serie de herramientas disponibles para trabajar con el Mega ADK de Arduino. Este mes vamos a dar un paso más en este proceso, mostrando a nuestros lectores cómo leer los sensores conectados a Arduino sobre el teléfono móvil. Con todas las herramientas de desarrollo listas para usar, ejecutar el código en nuestro teléfono y hacer que “hable” con nuestro Arduino será muy sencillo. ¡Podemos comenzar siguiendo los ejemplos que vienen con las librerías ADK!



Los dispositivos Android están por todas partes. Hay muchas maneras de acercarnos a la creación de aplicaciones

para dichos equipos. En nuestro caso, continuaremos explorando el uso de Processing [1] como una manera de escribir, compilar y volcar código desde/hacia los teléfonos Android.

Materiales

Continuando con los experimentos realizados el último mes, los materiales para realizar los experimentos que vamos a hacer son los siguientes:

- Placa Mega ADK Arduino o Placa Arduino Uno con extensión Host USB.
- Teléfono Android con SO Android 2.3.4 o superior (los nuevos SO son mejores)
- Cable USB y cable microUSB.
- Shield TinkerKit (ésta es opcional, podemos usar una placa de experimentación e hilos para puentes).
- Módulos TinkerKit: Potenciómetro y Pulsadores (también opcionales).

Nota: en este artículo vamos a necesitar tener un teléfono Android para probar cómo funciona todo. Hemos hecho un esfuerzo extra para asegurarnos que el código es compatible con las últimas

versiones del SO Android que, en el momento de escribir este artículo, era la versión 4.1, código de Jelly Bean.

Consejo: descargar el código mostrado en este artículo desde la página web de Elektor, ya que en este artículo sólo mostramos extractos del código.

¿Dónde estamos ahora?

En este punto ya debemos tener unos ciertos conocimientos del modo “Accesorios” de Google y de cómo los accesorios le dicen a los dispositivos Android que tienen que arrancar una cierta app (aplicación) cuando se conectan. Ya estamos familiarizados con Processing, la herramienta de aprendizaje de Java y ya hemos creado y volcado pequeñas apps a nuestro teléfono móvil o nuestra tableta.

El siguiente paso es el de crear nuevas apps que harán uso de la conectividad entre Arduino (como un accesorio) y nuestro dispositivo Android.

La librería host USB

El Protocolo Abierto de Accesorios Android (es decir, Android Open Accessory Protocol o AOAP) es el que determina cómo pueden comunicarse entre sí nuestro dispositivo Android y nuestra placa Arduino. La placa Arduino es técnicamente un Host USB, mientras que el teléfono se convierte en un cliente USB. Los dispositivos Android, al igual que la mayoría de los dispositivos punteros actuales, van equipados con un periférico que realiza la funcionalidad de llamada ‘USB On

The Go'. De acuerdo con esto, el mismo conector USB (actualmente un conector micro-USB) puede convertirse en un cliente USB (como un ratón o un teclado) o un Host USB (como el de nuestro ordenador personal).

El AOAP establece qué accesorios actúan como “huéspedes” (‘Host’) mientras que los dispositivos Android actúan como clientes. Esto permite que el host alimente al cliente con electricidad (nuestra placa Arduino recargará nuestro teléfono móvil cuando se conecte), entre otras prestaciones.

Para establecer la comunicación entre ambas partes, ya vimos en el artículo previo que necesitamos ejecutar la librería Host USB como parte de nuestro código Arduino. Podemos comprobar uno de los ejemplos de la librería para confirmar que la tenemos instalada.

Nota: usaremos el menú para navegar a través de: “File / Examples / USB Host”, si no encontramos estos ejemplos aquí quiere decir que no tenemos instalada la librería host USB en el entorno Arduino. Comprueba la parte de descargas de este artículo [3] e instálala; la vamos a necesitar. La **Figura 1** muestra la circuitería involucrada.

Configuración experimental: la parte de procesamiento

Primero vamos a dar un vistazo al software que necesitamos para programar nuestras aplicaciones Android:

- SDK Android con todas las actualizaciones a la última versión de las APIs.
- Processing 2.0b3 o superior (si leemos el primer artículo podremos ver que ya hemos actualizado a la última versión. Debemos hacer lo mismo para este punto).
- La herramienta ADK Arduino para Processing.
- La Librería USB ADK Arduino para Processing.

El último elemento de esta lista es lo que no estaba si hemos experimentado con el artículo previo de esta serie. Fue omitido intencionadamente en dicho artículo.

Necesitamos la librería complementaria para el Host USB en la parte de Processing, para asegurarnos que podemos conseguir que nuestra placa hable a nuestro teléfono. Llamamos a esta librería “USB ADK Arduino”. La siguiente sección explica cómo instalarla. La **Figura 2** muestra la parte hardware de las cosas.

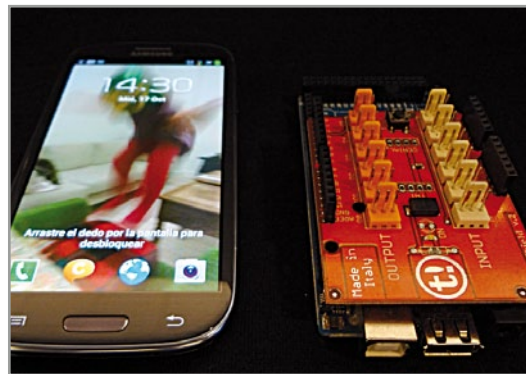


Figura 1.
ADK Mega Arduino con
Shield TinkerKit y teléfono.



Figura 2.
ADK Mega Arduino
conectado a un PC y a un
teléfono.

Instalando la Librería USB ADK Arduino para Android

Dentro del fichero descargado para este artículo encontraremos un directorio llamado “Processing” en el que tendremos un subdirectorio llamado ‘libraries’. Debemos copiar el contenido de este directorio dentro de nuestro directorio de librerías Processing. Éste último está localizado dentro de nuestro directorio “sketchbook” de Processing, normalmente almacenado en el directorio “Mis documentos/ Processing”, en ordenadores con Windows, o en el directorio “Documents/Processing”, para ordenadores con Mac o Linux.

Esta librería viene con cuatro ejemplos, recomendamos arrancar nuestros accesorios con uno de estos ejemplos, ya que eso hará nuestra vida mucho más fácil. Vamos a comenzar intentando ejecutar el ejemplo llamado ‘digitalWrite’. En este caso tendremos que conectar nuestra placa Arduino a nuestro teléfono y, al tocar la pantalla, se encenderá un diodo LED conectado a nuestra placa. Si dejamos de tocar la pantalla el LED se apagará.

DigitalWrite

Cada ejemplo tiene dos partes. La primera se ejecuta en la placa Arduino, controla las comunicaciones hacia y desde el teléfono a la vez que

Listado 1. Ejemplo DigitalWrite para la Librería Host USB de Arduino.

```
#include <AndroidAccessory.h>

// descriptor de accesorios. Así es como Arduino se identifica con Android
char accessoryName[] = "DigitalWrite"; // nuestra placa Arduino
char companyName[] = "Arduino SA";

// variables led
int ledPin = 10;

// contadores
long timer = millis();

// inicializa el accesorio:
AndroidAccessory usb(companyName, accessoryName);

void setup() {
    // comienza la conexión al dispositivo por el host USB:
    usb.begin();
    // configura el terminal LED como salida
```

realiza acciones sobre los otros terminales de la placa. La segunda parte es el código que tiene que ser compilado sobre Processing y que se ejecutará dentro del dispositivo Android. Esta app leerá y escribirá datos a través de la conexión por cable del teléfono, a la vez que realizar algunas acciones en la pantalla o leerá cualquiera de los sensores del teléfono como la cámara, el acelerómetro, etc.

Primero vamos a analizar el código Arduino, abriremos nuestro IDE Arduino y escribiremos el código del **Listado 1**. ¡Debemos recordar vol-

car dicho código sobre la placa una vez que lo hayamos hecho!

Este código de ejemplo será “escuchado” por el puerto USB. Una vez que el teléfono esté conectado a la placa Arduino, la función `usb.isConnected()` devolverá una respuesta afirmativa. Cuando el teléfono comience a enviar datos, la función `usb.available()` devolverá un valor diferente de cero que Arduino leerá y almacenará en la variable `val`. Algo muy similar a lo que se muestra en la **Figura 3** debe ser lo que tengamos también nosotros.

Listado 2. Procesando el ‘sketch’ que detectará los toques de pantalla.

```
import cc.arduino.*;

// crea el objeto ADK
ArduinoAdkUsb arduino;

void setup() {
    // Establece la vista PORTRAIT
    orientation( PORTRAIT );
    // inicializa el objeto ADK
    arduino = new ArduinoAdkUsb( this );
    if ( arduino.list() != null )
        arduino.connect( arduino.list()[0] );
}

void draw() {
    // Dibuja una recta rellena basada en el estado de la conexión Arduino
    connected( arduino.isConnected() );
}
```



```

    pinMode(ledPin, OUTPUT);
}

void loop() {
    // saca al USB 10 veces por segundo
    if(millis()-timer>100) {
        if (usb.isConnected()) { // ¿está abierta la comunicación USB?
            if (usb.available() > 0) { // ¿está el dato?
                char val = usb.read();
                // 'a' pone el LED on, 'b' off
                if( val == 'a' )
                    digitalWrite( ledPin, HIGH );
                else if( val == 'b' )
                    digitalWrite( ledPin, LOW );
            }
            timer = millis();
        }
    }
}

```

En el lado de Processing necesitaremos escribir un programa que compruebe el evento de alguien tocando la pantalla. Si hay una conexión por cable, el teléfono enviará 'a' para indicar que alguien acaba de comenzar a tocar la pantalla. Enviaré 'b' para indicar que el dedo ya no la toca. Estamos detectando los eventos de 'comenzar a tocar' y 'para tocar', ya que no queremos inundar el puerto de comunicaciones. Abriremos nuestro IDE Processing y buscaremos el ejemplo `adk_digitalWrite`. Lo encontraremos en el menú 'Examples / Contributed Libraries /

ArduinoADKUsb', si hemos instalado la librería adecuadamente. Ver **Listado 2**.

El código del **Listado 2** es bastante auto-explicatorio. Una parte que podemos encontrar un poco y esa es la de definición del método `surfaceTouchEvent(MotionEvent event)`. Java es un lenguaje de programación muy potente y, en este caso, permite suplantar algunos de los métodos de llamada existentes. En el momento en que tocamos la pantalla del dispositivo Android, el sistema está configurado para llamar a la función específica (`surfaceTouchEvent(MotionEvent`

```

public boolean surfaceTouchEvent(MotionEvent event) {
    if ( arduino.isConnected() ) {
        if ( event.getAction() == MotionEvent.ACTION_DOWN )
            arduino.write('a');
        else if ( event.getAction() == MotionEvent.ACTION_UP )
            arduino.write('b');
    }

    // si queremos las variables para motionX/motionY,
    // mouseX/mouseY etc.
    // para trabajar adecuadamente necesitaremos llamar a
    // super.surfaceTouchEvent().
    return super.surfaceTouchEvent(event);
}

[...]
```

event)). Si suplantamos esta función en nuestro programa (básicamente, rescribiéndola) podemos conseguir que haga lo que queramos.

En este caso, nosotros detectamos dos de los muchos eventos del sistema: `MotionEvent.ACTION_DOWN` (el equivalente a comenzar a pulsar la pantalla), que envía una 'a' a Arduino, `MotionEvent.ACTION_UP` (el mismo para detener el pulsar la pantalla), que envía una 'b'.

Algo más divertido

Una vez que hemos conseguido ejecutar el primer ejemplo... ¿por qué no construir algo más divertido? Internet está repleta de gatos, por lo que el último mes hemos decidido crear la aplicación `Elektor_Miau`. Queríamos hacer un comentario divertido y, al mismo tiempo, mostrar a nuestros lectores que es posible comenzar a hacer cosas más llamativas que el clásico ejemplo "Hola mundo" cuando aprendemos a programar.

Este mes queremos crear las bases y animar a nuestros lectores a aprender cómo hacer pequeñas animaciones usando Processing. Podemos buscar cualquier GIF animado en Internet y usarlo para hacer una réplica de lo que queremos ver en este ejemplo. La única condición es que necesitamos que en el GIF animado sea exportado como tramas. Tendremos que nombrar cada trama de la siguiente manera: `s00.gif`, `s01.gif`, `s02.gif`...

Por supuesto, también podemos utilizar la misma imagen que estamos usando para nuestro artículo. Hemos conseguido que un dibujante nos dibuje un pequeño conejo que sale y se deja ver. El objetivo de este ejemplo es hacer un pequeño juego interactivo donde utilizaremos sensores (un botón y un potenciómetro) para interactuar con el pequeño conejo en la pantalla del teléfono, como en la **Figura 4**.

Esta vez es mejor comenzar mirando el código que se va a ejecutar dentro del teléfono. El primer paso es construir una animación dentro de Processing. Si somos conocedores de Android, podemos intentar exportar la librería `GIFdecoder`, que hemos utilizado el mes pasado, para reproducir GIFs animados dentro de nuestra app Android. Como queremos mantener la programación sencilla y fácil de entender, la manera en que recomendamos a nuestros lectores que animen su conejo es muy simple: debemos construir un 'array' de imágenes diseñado para contener las tramas que queremos que formen parte de la animación. Este concepto está basado en el ejemplo `Animated Sprite` de Processing de James Paterson [2], ver el **Listado 3** que podrás encontrar en [3].

Nota: El Listado 3 no nos muestra la clase `Animation` que abrirá todas las imágenes llamadas `s00.gif`, `s01.gif`, etc. Recomendamos que verifique y el listado completo del código que viene con el ejemplo `Elektor_Simple_Animation`. También veremos un directorio llamado 'data', dentro del directorio 'sketch' de Processing, que contiene todas las imágenes que forman parte de la animación (54 en total).

La configuración final

Lo siguiente será modificar el ejemplo para incluir la librería que controla las comunicaciones por USB. Recomendamos a nuestros lectores que agreguen el código previo con el ejemplo `adk_analogRead`. El resultado de ello debe ser similar al código del **Listado 4** en [3].

Hay un par de cosas en el ejemplo que han que ser remarcadas. En primer lugar, hay un array llamado `delays[i]`. Este array es responsable de controlar el tiempo durante el que cada trama será mostrada en la animación. Cada vez que cargamos una nueva trama, también cambiamos la longitud del temporizador en la condición `millis() - timer > delays[index]`. Animamos a nuestros lectores a experimentar con las cifras de este array para ver como afectan a la animación. Debemos recordar que tiene que haber tantas cifras como tramas en la animación.

También está el array `keyFrames[i]`, que aloja una lista de tramas en la animación que esperarán una acción del usuario para que continúe la película. En este caso, vamos a detectar la pulsación de un botón. La animación se detendrá en aquellas tramas que se marquen hasta que el usuario presione el botón conectado al módulo expansor `TinkerKit`, como podemos ver en la **Figura 5**.

Si nos fijamos en la línea que verifica cuantos bytes llegan desde la placa Arduino, verificamos `arduino.available() >= 3` para ver si hemos recibido, o no, 3 bytes. La información se codifica de la siguiente manera:

- Byte 0: tiene que ser 255, es un marcador que usamos para distinguir cuando comienza el bloque;
- Byte 1: es el valor del potenciómetro; rango de 0 a 254;
- Byte 2: viene desde el botón; su valor será, bien 0 o bien 1.

El valor del potenciómetro será usado para cambiar el color de fondo de la pantalla, mientras que el botón disparará la siguiente secuencia de la animación.

Nota: usaremos la herramienta USB ADK Arduino (menú 'Tools / Arduino ADK' en el IDE de nuestro Processing) para volcar este programa a nuestro teléfono. Aparecerá una ventana de diálogo. En el campo de texto denominado "model" introduciremos el nombre de la app a la que llamamos dentro del IDE de Arduino. El último parámetro se emplea para seleccionar la versión del SDK para nuestro modelo de accesorio. Si nuestro teléfono o tableta trabaja con Android 3.0 o posterior, debemos tener un SDK v12. El SDK v10 es sólo para aquellos dispositivos que trabajan con Android 2.3.4. En el caso desafortunado de que nuestro dispositivo Android esté trabajando con la versión 2.3.3 de SO o anterior, no seremos capaces de usar el modo 'Accesorio'.

Por último, el código Arduino es prácticamente una repetición del ejemplo `adk_analogRead`, añadiendo el byte marcador adicional, limitando la lectura analógica a 254 y añadiendo el botón. El bucle para el este programa se muestra en el **Listado 5** en [3]. Por favor, verificad el programa completo dentro de la documentación que podéis conseguir de la página web de Elektor [1], en el directorio "Arduino / code".

Palabras finales

Con este artículo hemos acabado nuestra introducción a Android y cómo conectarlo a Arduino. Ahora ya deberíamos ser capaces de crear un nuevo proyecto, crear robots donde nuestro teléfono móvil pueda ser la inteligencia del mismo, o integrar sensores que no vienen con nuestro dispositivo Android. Si hacéis cualquier proyecto interesante usando cualquier parte de nuestro código, no os importe dejar una línea en blog@arduino.cc, estaremos encantados de hacer saber al resto de la comunidad lo que estáis haciendo.

(120573)

Enlaces en Internet y Referencias

- [1] Processing Project: <http://processing.org>
- [2] Animaciones usando múltiples imágenes con Processing: <http://processing.org/learning/topics/sequential.html>
- [3] www.elektor.es/120573

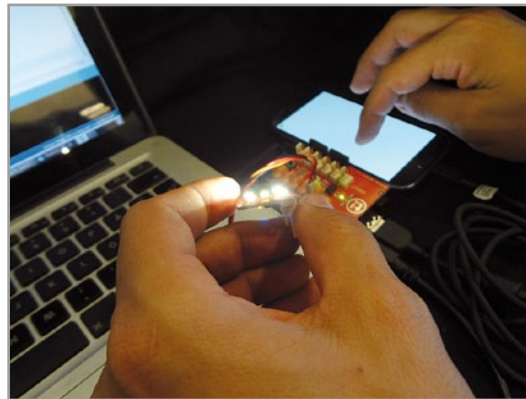


Figura 3.
El LED se enciende cuando tocamos la pantalla.

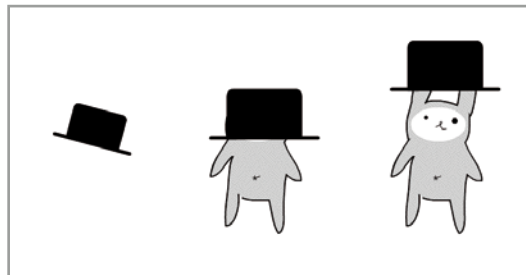


Figura 4.
Un par de tramas de nuestra película de conejo.

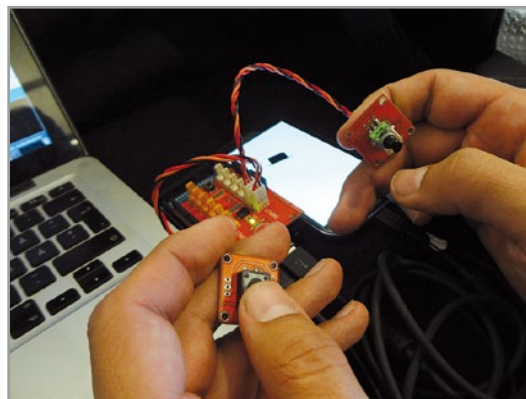


Figura 5.
Botón y potenciómetro en la parte superior del Shield TinkerKit.

Agradecimientos

El equipo de diseño Arduino desea expresar su agradecimiento al equipo de Circuits@Home por sus primeros trabajos sobre la librería Host USB para Arduino y a Philip Lindsay por su trabajo en la depuración de la colección inicial de las librerías Arduino para Android. También a Rodrigo Calvo, que ayudó en el trabajo de mejorar la librería Host USB para que funcionase con dispositivos Android 4.1. Y a Laura Balboa, por su trabajo de dibujo e ilustraciones.

Consejos sobre componentes

Raymond Vermeulen (Elektor.Labs)

Controlador de ventilador dependiente de la temperatura

Para este mes he seleccionado dos integrados que se encargan de una regulación de ventilación independiente. Puede ser útil si dispones de un sistema que consuma mucha potencia y se encuentre en una caja, de modo que requiera una refrigeración activa mediante un ventilador. Sería muy útil utilizar un integrado separado de control si quieres que la regulación del ventilador dependa de la temperatura, pero no quieres que el microcontrolador presente se encargue de eso. Sólo necesitas conectar un sensor y un ventilador. Un detalle llamativo de los integrados que presentamos aquí es que no siguen el estándar de control de ventilación de 25 kHz PWM y de un conector de 4 terminales, sino que conmutan directamente la tensión de alimentación a una frecuencia mucho más baja.

(120569)

MIC502

EL MIC502 de Micrel utiliza un NTC o un PTC como sensor de temperatura, con la opción de un segundo sensor. Los sensores se conectan a las entradas VT1 y VT2. Una tensión desde el 30% hasta el 70% de la tensión V_{dd} produce un ciclo de trabajo desde 0% hasta el 100%. Tiene prioridad la más alta de las dos entradas, una característica perfecta si mides con un sensor la temperatura ambiental y con el otro el componente que está disipando fuertemente. En el terminal V_{slp} se coloca una tensión que haga que el chip entre en modo reposo cuando las entradas VT1 y VT2 caigan por debajo de ese valor. Cuando VT1 o VT2 supere dicho valor, el integrado se reactivará de nuevo. Esto se hace para evitar que el ventilador funcione con un ciclo de trabajo demasiado bajo. Sobre el terminal CF se conecta un condensador de temporización cuyo valor recomendado es 100 nF para una frecuencia de 30 Hz. Sin embargo, también se puede configurar una frecuencia más alta, según la hoja de datos el alcance va desde una frecuencia de 15 hasta 90 Hz.

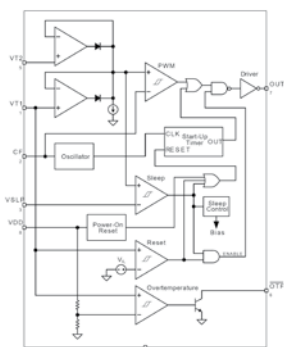


Figura 1. Diagrama de bloques del MIC502.

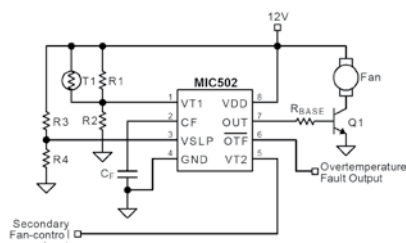


Figura 2. Ejemplo de aplicación con el MIC502.

Hoja de características MIC502:
www.micrel.com/_PDF/mic502.pdf

MAX6643

El MAX6643 utiliza un transistor configurado como diodo para medir la temperatura. Este integrado tiene tres opciones diferentes para la configuración de la temperatura. Con el 'Overtemperature Threshold' (terminales OT1 y OT2) se configura la temperatura (entre 60 y 100°C) que hace que la salida OT conmute en el momento en que se sobrepase dicha temperatura. El 'High temperature Threshold' (terminales TH1 y TH2) determina a qué temperatura hay que subir un peldaño el ciclo de trabajo de la señal PWM. El 'Low temperature Threshold' (terminales TL1 y TL2) determina a qué temperatura hay que bajar un peldaño el ciclo de trabajo de la señal PWM. Para todos estos umbrales (thresholds) de la temperatura se pueden configurar unos valores determinados conectando los terminales correspondientes a V_{dd} o GND o dejándolos abiertos (ver hoja de características).

También dispone de un terminal 'FULLSPD' para forzar un ciclo de trabajo de 100%. Esto puede ser útil cuando se detecta un estado con una temperatura demasiado alta. Además dispone de una entrada para recibir la señal de tachómetro del ventilador, con el que se indica que el ventilador ha dejado de funcionar a través del terminal 'FANFAIL'.

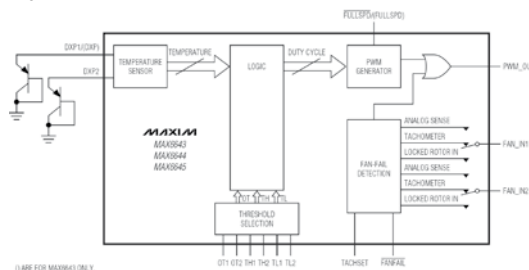


Figura 3. Diagrama de bloques del MAX6643.

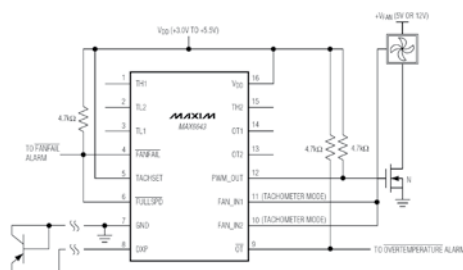


Figura 4. Ejemplo de aplicación con el MAX6643.

Hoja de características MAX6643:
<http://datasheets.maximintegrated.com/en/ds/MAX6643-MAX6645.pdf>

● **Subscribe** to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words "**audio,**" "**vacuum tubes,**" and "**speaker technology**" music to your ears?

Then you should be **reading audioXpress!**

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

Choose from print delivery, digital, or a combination of both for maximum accessibility.

Subscribe to *audioXpress* at
www.audioamateur.com
today!



audioXpress

Electrónica para Principiantes (10)

Alta frecuencia (HF)

Burkhard Kainka
(Alemania)

Con la alta frecuencia, las comunicaciones por radio han pasado a ser el motor de la electrónica. Todo empezó con las válvulas y la radio AM, más tarde llegaron el VHF y el FM. Entre tanto, al trabajar con semiconductores las frecuencias son cada vez más altas. Sin embargo, los experimentos en HF con onda media y corta pueden resultar muy interesantes para los principiantes.

Construir nuestra propia radio es una buena manera de iniciarse en la electrónica. Pero debería ser posible sin utilizar componentes especiales, y que ésta tuviera cierta fiabilidad. Echemos un vistazo al audión de banda ancha de la **figura 1**.

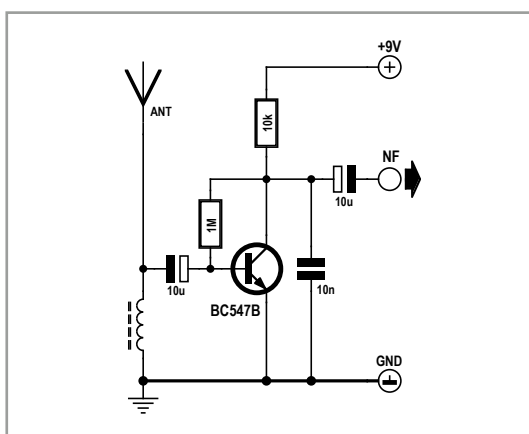


Figura 1.
Audió de banda ancha.

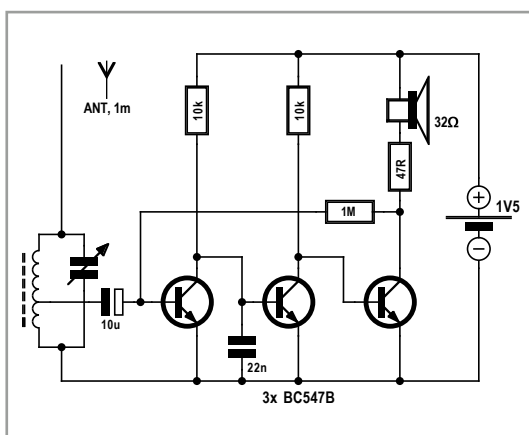


Figura 2.
Mayor selectividad y
ganancia para el audión.

Como audión nos referimos a un circuito capaz de realizar ambas funciones, tanto demodular señales en HF, como amplificarlas. Este sencillo receptor incluye todo lo necesario para escuchar todo lo que surca el aire, que puede proceder de emisores de onda media cercanos, pero también de onda corta que se encuentren en el rango, y naturalmente, también todos los ruidos de alta frecuencia generados en las inmediaciones. Lo único que necesitamos es una antena de cable lo suficientemente larga. Lo más óptimo sería disponer de un cable colgado de unos 10 metros de longitud, pero con antenas más cortas también podemos obtener buenos resultados. También necesitamos una conexión a tierra. En la salida de audio en baja frecuencia podemos conectar unos auriculares con cierta sensibilidad, y mucho mejor si se trata de un amplificador para altavoces. El audión de banda ancha recibe todo simultáneamente. Normalmente puede escucharse más de una emisora, entre las cuales destaca una especialmente intensa. Ajustando adecuadamente la bobina, influiremos en cierta medida en el rango de recepción. También puede utilizarse una inductancia fija. Para la recepción en onda media, necesitaremos unos 300 μH , para la onda corta unos 10 μH , o tan sólo 3 μH , si queremos centrarnos en las bandas más altas, de 15 a 17 MHz. Una bobina diseñada por nosotros mismos para onda media podría tener por ejemplo 100 espiras alrededor de una varilla de ferrita; para onda corta bastan entre 10 y 30 espiras con un diámetro de 10 mm, sin necesidad de ferrita. El resultado también depende de la propia antena. Podemos probar con distintas bobinas. Depen-

diendo de dónde nos encontremos, el momento del día y la bobina utilizada, pueden escucharse distintas emisoras. Puede pasar, por ejemplo, que capturemos una emisora de China, ya que en la bandas más altas del HF el rango de alcance es bastante grande.

Si echamos un vistazo con más detenimiento al circuito, veremos que en realidad se trata de un amplificador. La señal en alta frecuencia de la antena es amplificada. Si se tratara de un amplificador absolutamente perfecto, no escucharíamos nada, ya que las frecuencias utilizadas no pertenecen al rango audible. Sin embargo, funciona debido a la "irregular" curva característica de entrada del transistor. La semionda positiva de la señal en HF amplifica la corriente de colector en mayor medida que la semionda negativa la atenúa. Por ello, la corriente media depende de la tensión de entrada en HF. Debido a que por parte del emisor dicha amplitud de la señal HF es modulada en función del reloj de la señal de audio, en la salida aparece la señal de baja frecuencia.

En el circuito también son importantes los condensadores. El condensador de la entrada, relativamente grande, impide el paso de señales de baja frecuencia a la base, mediante la bobina de entrada. Por lo tanto, en la base sólo tendremos tensiones en HF. En el circuito del colector circulan corrientes en HF amplificadas y al mismo tiempo otras en baja frecuencia, ya demoduladas. No obstante, un condensador de 10 nF en la salida cortocircuita las señales en HF. Esto conserva únicamente las señales audibles, es decir, en baja frecuencia. De modo que, con muy poco esfuerzo, tendremos un receptor con una buena sensibilidad!

La **figura 2** muestra una sencilla radio con ambos principios de recepción. Salvo que esta vez, cuenta con mayor ganancia y una mejor selectividad. En la entrada hay un circuito resonante ajustable. Dependiendo de las especificaciones de las bobinas, la radio puede recibir desde en onda larga a corta. La bobina receptora es capaz de empezar a funcionar con aproximadamente el 10 % de las espiras. En este caso, el circuito resonante atenúa sólo ligeramente, y mantiene una buena calidad de la señal (véase el cuadro "Circuitos Resonantes"). Esto permite filtrar emisoras individualmente. Con tres transistores en total, y una tensión de alimentación de sólo 1,5 V, dispondremos ya de una radio funcional, para

El circuito resonante

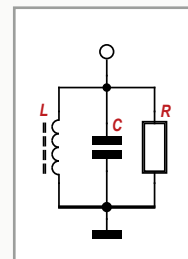
Una bobina L y un condensador C forman un circuito resonante. La energía eléctrica puede interpretarse de la misma manera que la mecánica en un péndulo, en el cual bobina y condensador se balancean hacia delante y hacia atrás, siendo f una frecuencia de resonancia definida. El circuito resonante entra en oscilación de forma natural una vez se le aplica una pequeña corriente. El resonador ideal no tendría pérdidas de potencia, y podría estar oscilando infinitamente. En la práctica, sin embargo, tenemos pérdidas dada la resistencia del cable de la bobina, sumadas a las magnéticas en el núcleo de ésta debidas al flujo, las cuales atenúan las oscilaciones. Sin embargo, podemos simplificar las pérdidas a una resistencia R en paralelo con el circuito. Para obtener la frecuencia de resonancia f tenemos:

$$f = 1 / (2 \pi \sqrt{LC})$$

Cada circuito resonante tiene un factor de calidad (o selectividad) Q. Éste se puede determinar mediante la relación entre la resistencia de atenuación en paralelo R, con la impedancia inductiva $R_L = 2 \sqrt{f L}$, o la capacitiva $R_C = 1/(2\sqrt{f C})$, dadas a la frecuencia de resonancia.

$$Q = R/R_L \text{ o } Q = R/R_C$$

Si excitamos un circuito resonante a frecuencia variable mediante una fuente de corriente alterna (con alta resistencia interna), la tensión del circuito resonante en resonancia será la máxima posible. Cuanto menor sea la atenuación de las oscilaciones debido a las pérdidas de cualquier tipo, mayor será la calidad del resonador, y mayor la tensión en resonancia. Tanto por encima como por debajo de la frecuencia de resonancia, podemos determinar los puntos de la curva en los que la frecuencia cae según el factor $1/\sqrt{2} = 0,707 = -3 \text{ dB}$. El espacio comprendido entre dichos puntos se conoce como ancho de banda (b) del circuito resonante. La frecuencia de resonancia f, el ancho de banda b y el factor de calidad Q están relacionados según la ecuación $b = f/Q$. Con una bobina de una calidad razonable, puede alcanzarse hasta $Q = 100$, aproximadamente. No obstante, los circuitos resonantes se ven atenuados por los circuitos o antenas conectados. Esta atenuación puede contrarrestarse en el circuito resonante mediante una pequeña bobina adicional en el acoplamiento, el llamado "coil tap" o con el condensador adecuado. Si lo conectamos directamente a un amplificador, su resistencia interna ha de ser lo más alta posible, manteniendo la atenuación al mínimo.



utilizar con auriculares. El circuito guarda cierto parecido con el detector de HF Tapir de la edición de Elektor de Julio/Agosto de 2012 [1], que también incorpora tres transistores, pero funciona a modo de audión de banda ancha.

Figura 3.
Emisor de pulsos
sincronizado.

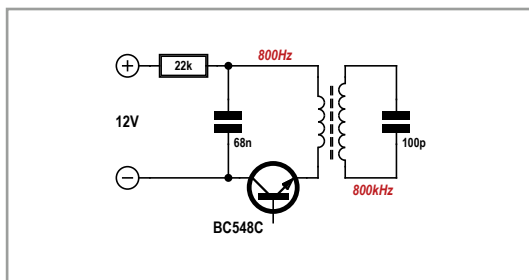


Figura 4.
Oscilador sinusoidal
con acoplamiento por
transformador.

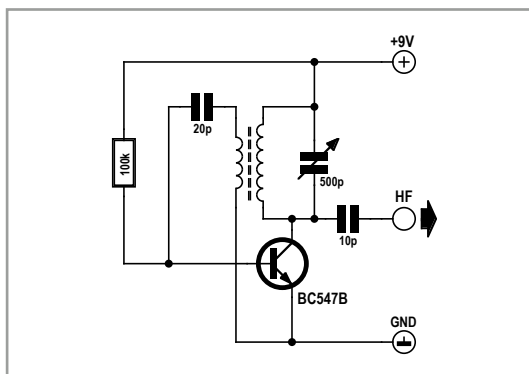


Figura 5.
Oscilador para tensiones de
alimentación reducidas.

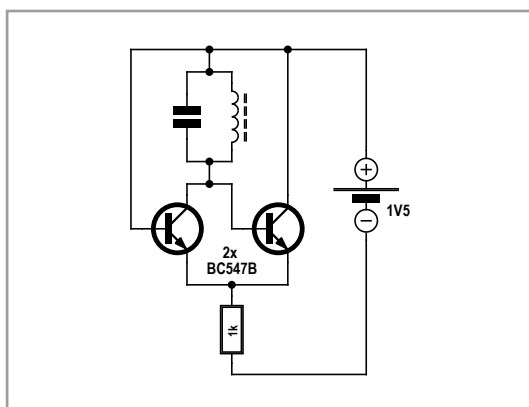
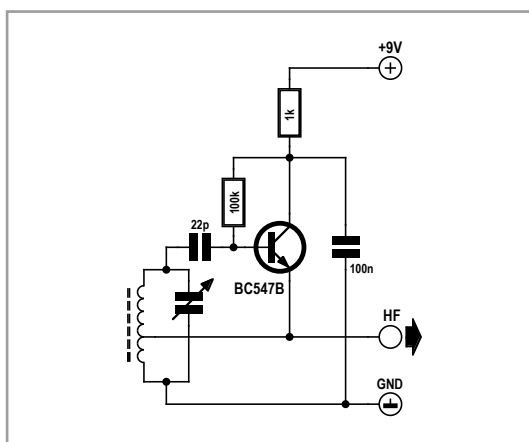


Figura 6.
Oscilador con bobina
(mediante "coil tap").



Generación de señales en HF

Los primeros emisores operativos fueron los canales de radio. Pero en realidad, basta un interruptor de luz normal para comprender el principio. Al pulsar el botón, en la radio podremos oír un chasquido, generalmente de onda media, en caso de que no haya ninguna otra emisora de OM con mayor intensidad. Una señal generada por un interruptor, con un flanco pronunciado, siempre incluye componentes en alta frecuencia. De hecho, la banda de onda media en muchos lugares se ve muy afectada por las fuentes de alimentación de otros equipos. De modo que el problema en sí no está en generar señales de HF, sino en evitar que estas se propaguen indeseablemente.

No tiene que ser necesariamente un emisor que no cumpla con la normativa, sino cualquier aparato capaz de generar señales de HF que sirva para probar nuestro propio receptor. La **figura 3** muestra una pequeña "emisora", que funciona mediante el viejo principio de transmisión por radio. Las "ondas" periódicas tienen una frecuencia de unos 800 Hz, utilizando un transistor NPN para generar la señal de diente de sierra (véase el capítulo 7, en la Elektor de Septiembre de 2012 [2]). Cada uno de estos flancos afilados hace oscilar al circuito resonante, lo cual se compensa rápidamente (atenuación de las oscilaciones). La frecuencia de emisión está fijada por el circuito resonante. Si utilizamos por ejemplo una varilla de ferrita de una vieja radio, la señal estará en el rango de onda media, por ejemplo alrededor de los 800 kHz. Si colocamos la radio al lado de la ferrita, podremos oír un zumbido. Sólo nos falta un interruptor para el código Morse, y listo. Las emisoras de radio de los barcos funcionaban de una manera similar, salvo que con mucha mayor potencia, y en frecuencias bastante más bajas.

Osciladores LC

Mientras que las emisoras de radio sólo eran capaces de generar oscilaciones atenuadas, el verdadero avance llegó con los tubos de electrones (o válvulas), los cuales permitían oscilaciones sin atenuar. La última entrega del curso trataba sobre los osciladores sinusoidales [3] mostrábamos lo necesario: suficiente ganancia y una realimentación adecuada.

La forma primigenia de los osciladores HF es el oscilador Meißner (también conocido como oscilador Armstrong). Incorpora una bobina de aco-

plamiento para asegurar la fase correcta (**figura 4**). Así estaba construido el primer emisor de radio: una válvula, un circuito resonante con realimentación, y finalmente un oscilador de potencia de HF. En la antena se integraba un micrófono de carbón, con el propósito de solucionar la modulación.

El circuito de la figura 4 se trata de un oscilador Meißner con un transistor. No resulta difícil hacer oscilar a este circuito. Si no funciona, es posible que la fase sea incorrecta, y en tal caso basta con invertir la polaridad de la bobina de acoplamiento.

En este sencillo oscilador, la frecuencia puede ajustarse mediante el condensador variable. La amplitud depende de muchas cosas, por ejemplo de la atenuación de la bobina, del lazo de realimentación y del condensador de acoplamiento. A veces hemos de experimentar primero un poco hasta que el oscilador sea estable. En este circuito puede darse un problema especial: en amplitudes muy grandes podrían aparecer oscilaciones sinusoidales. El diodo base-emisor rectifica la tensión en HF y pone la base en negativo, hasta que el transistor entra en corte y las oscilaciones desaparecen. Después transcurren algunos microsegundos hasta que la resistencia de la base alimenta de nuevo el transistor, hasta alcanzar otra vez el punto de trabajo correcto, y el proceso comienza de nuevo.

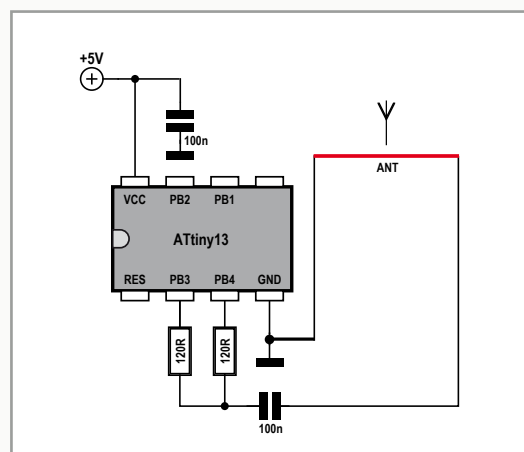
Estos problemas no ocurren en el oscilador de la **figura 5**, ya que la amplitud se estabiliza automáticamente. Aparte, sólo se utiliza una sencilla bobina sin acoplamiento, y ni siquiera hace falta que sea de muy buena calidad, ya que la ganancia es bastante alta. El circuito puede construirse con componentes que tenemos a mano, y siempre oscila sin problemas. La realimentación se hace mediante dos transistores. Ambos funcionan a una tensión muy baja (unos 0,6 V). El circuito en sí dispone de una tensión de alimentación de 1 V, y puede seguir activo a corrientes muy bajas. Esto supone una ventaja en algunas aplicaciones. En principio, como todo resonador, este circuito funciona en un amplio rango de frecuencias, de baja a VHF, con lo que pueden construirse, por ejemplo, generadores de prueba a coste muy reducido. Pero tienen un inconveniente: las capacidades internas de los transistores influyen en la frecuencia y dependen en gran medida de la tensión de alimentación. El circuito es muy simple, pero no se trata precisamente

Generador de AM con un Tiny13

Quien diseñe y construya receptores de radio, también puede utilizar un pequeño generador AM de prueba. Con la forma de señal adecuada no será necesario ni ajustar la frecuencia. Esto sucederá si el generador funciona a una frecuencia lo suficientemente baja y se producen bastantes armónicos.

Nuestro generador de AM con un ATtiny13 produce pulsos cortos a una frecuencia de 70 kHz. Aparecen potentes armónicos en las bandas de onda larga y media. De esta manera, logramos una modulación en amplitud con un tono de 750 Hz. La radio recibirá ahora sólo a 70 kHz, 140 kHz, 210 kHz, y según este orden otras muchas emisoras de AM. Como antena basta un lazo de cable de unos 10 cm de diámetro, por ejemplo. Éste genera un campo magnético oscilante que se acopla directamente con la varilla de ferrita del receptor. El generador también sirve para tomar medidas comparativas de la sensibilidad. De esta manera podemos probar si aún se recibe la señal a cierta distancia. Un buen receptor es capaz de captar una señal perfectamente clara a partir de un metro de distancia.

El código fuente puede descargarse en la página de este artículo [4].



```
'ATTiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4

Config Portb = Output
Dim N As Byte

Do
  For N = 1 To 50    '70 kHz
    Portb = 255
    Portb = 0
  Next N

  For N = 1 To 50    'AM 750 Hz
    nop
    nop
  Next N
Loop

End
```


Figura 7.
Oscilador con divisor de
tensión capacitivo.

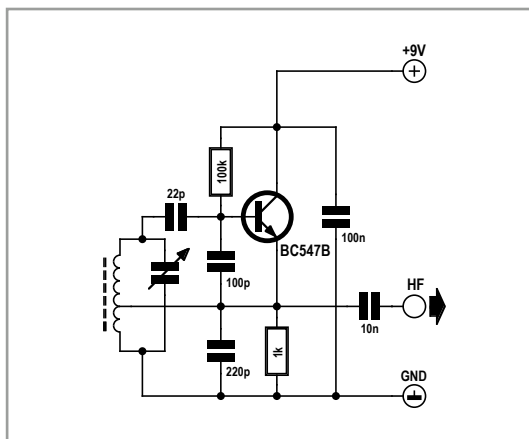


Figura 8.
Oscilador con acoplamiento
en corriente continua en
colector común.

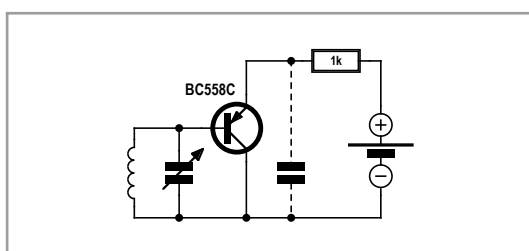
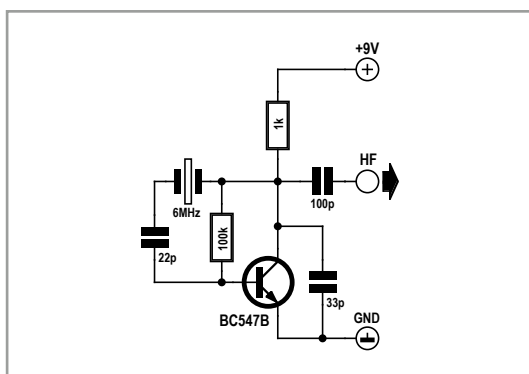


Figura 9.
Oscilador de cuarzo.



de una maravilla de la estabilidad en lo que a frecuencia se refiere. Pero a lo mejor podemos darle un uso, ¿quizá como supervisor de baterías inalámbrico?

La fase correcta en un oscilador también puede obtenerse utilizando una bobina (figura 6). Si no, también podemos hacerlo mediante un divisor de tensión capacitivo (figura 7). Si se utiliza un transistor PNP como seguidor de emisor (colector común) con acoplamiento en corriente continua en colector común (figura 8), el circuito se simplificará notablemente. El divisor de tensión capacitivo que se muestra en la figura 7 puede eliminarse completamente si la atenuación de la bobina es lo suficientemente reducida, ya que las capacida-

des internas de los transistores ocupan su lugar. Aparte, funcionará con una tensión de alimentación muy reducida, de aproximadamente 1,5 V, por ejemplo. Sin embargo, el acoplamiento directo con el circuito resonante vuelve a tener influencia sobre la frecuencia, ya que las capacidades de la unión cambian según la tensión de alimentación. Este sencillo oscilador también sirve para comprender posibles problemas con la estabilidad en muchas etapas amplificadoras en seguidores de emisor. Quizá sea así porque en un principio se buscaba un seguidor de emisor como convertidor de impedancias, que terminó actuando de oscilador. En cualquier caso, ya sea como amplificador o como oscilador, todo depende del punto de trabajo y la atenuación del circuito resonante.

Osciladores de cuarzo

Cuando queremos obtener una frecuencia estable de verdad, lo mejor es utilizar un oscilador de cuarzo. Un cuarzo se corresponde en realidad con un circuito resonante extremadamente preciso. Gracias a ellos, pueden realizarse osciladores con una buena estabilidad, a coste muy bajo y sin necesidad de inductancias. La figura 9 muestra un circuito típico.

A quien le guste experimentar con radios de onda media caseras, quizá quiera utilizar un oscilador modulado para hacer algunas pruebas. Un transmisor de prueba como este no incorpora antena, con lo que no creará interferencias molestas para los vecinos, ya que la transferencia de energía en HF sucede directamente de bobina a bobina.

Para que resulte entretenido, el modulador de onda media debería ser estable. Pero los cuarzos para esas frecuencias tan bajas siempre son muy caros. Por ello, en estos casos utilizamos resonadores cerámicos. Con éstos resulta fácil generar frecuencias entre 500 kHz y 2 MHz, ambas fuera del rango de la onda media. Existen otras frecuencias que ajustan mejor. Por ejemplo, descubrimos un resonador cerámico de 976 kHz en el antiguo mando a distancia de un televisor. La figura 10 muestra un oscilador AM de prueba completo. Mediante el trimmer puede realizarse un ajuste fino. Si en el fondo puede escucharse una emisora (probablemente muy atenuada), sencillamente realizamos el ajuste de cero, por ejemplo a 981 kHz. La pequeña bobina de ferrita emisora se acopla directamente con

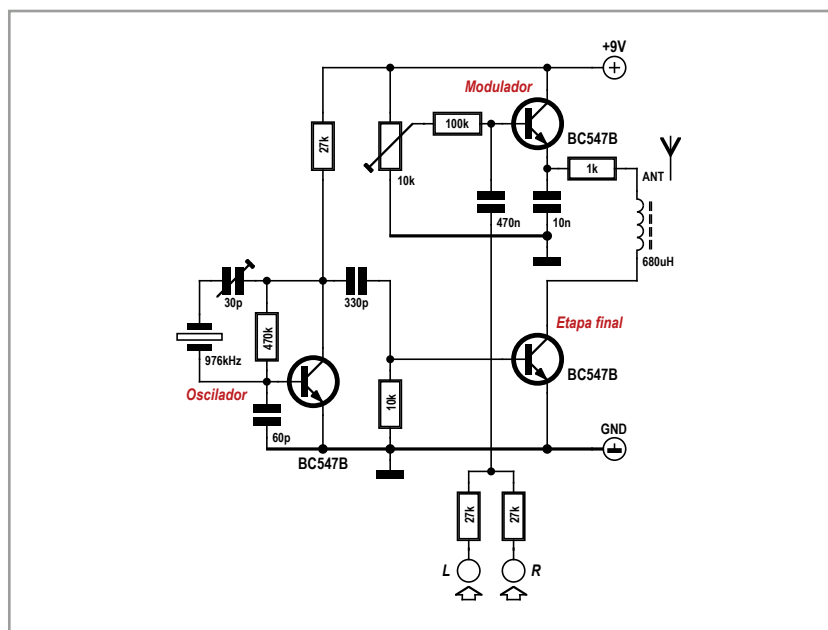
la varilla de la radio. También podemos utilizar el circuito con un cuarzo, sin realizar mayores cambios, por ejemplo para aplicaciones experimentales, en una frecuencia permitida como los son los 13,56 MHz.

La verdadera etapa de modulación se ha realizado a modo de seguidor de emisor y regula la tensión de alimentación del amplificador de salida. Ya que en onda media y corta aún se trabaja en mono, se combinan ambos canales de entrada de la señal en estéreo. Ajustamos el potenciómetro hasta lograr la menor distorsión posible, y el mejor sonido. La etapa amplificadora de HF se ha dimensionado de forma bastante modesta, ya que aquí no se trata de conseguir mucha potencia.

Audió de onda corta con realimentación

El receptor de onda corta de tres etapas de la **figura 11** dispone de controles adicionales para la realimentación. En principio, la primera etapa de un oscilador es como la de la figura 8. Pero en este caso, puede ajustarse el punto de trabajo y por lo tanto, reducirse la ganancia. El truco está en ajustar la ganancia de modo que se compensen las pérdidas del circuito resonante, y no aparezcan oscilaciones naturales. En este punto, el receptor tendrá la máxima ganancia y la mejor selectividad posible. La etapa de oscilación PNP en el seguidor de emisor actúa al mismo tiempo de etapa de audió y como demodulador de la señal en HF. Dos etapas de baja frecuencia consecutivas se encargan de suministrar potencia necesaria, con lo que puede utilizarse incluso un pequeño altavoz.

El hecho de que el audió sirva o no para generar oscilaciones depende de la atenuación del circuito de entrada en el cual se conecta la antena. Disponemos de dos conectores para la antena. Conectando un pequeño condensador solucionamos el acoplamiento, y limitamos la atenuación. La conexión directa es apta incluso para antenas muy cortas (una antena más larga emite a su vez energía en HF, atenuando de esta manera al circuito resonante). Con un ajuste meticuloso, en el audió puede lograrse una sensibilidad muy alta. En los inicios de la radiofrecuencia, estos receptores con audió formaban parte de todo diseño estándar. Su cobertura es de varios miles de kilómetros, incluso para las emisoras más débiles. La fascinación por este circuito aún puede verse hoy en día, ya que partiendo de unos requisitos mínimos puede construirse un excelente receptor.



Con esta entrega hemos llegado al final de nuestra serie. Pero en las siguientes ediciones trataremos otros temas sobre las bases de la electrónica, aunque sin un orden concreto. En nuestra agenda están, por supuesto, los amplificadores operacionales y los circuitos integrados digitales más importantes. ¡Estate atento!

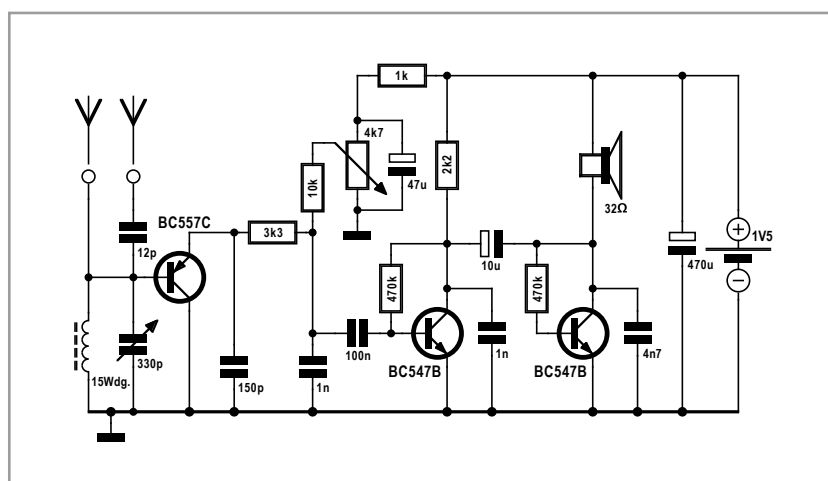
120010

Figura 10.
Modulador de onda media.

Enlaces

- [1] www.elektor.es/120354
- [2] www.elektor.es/120007
- [3] www.elektor.es/120009
- [4] www.elektor.es/120010

Figura 11.
Audió con realimentación.



El Internet de las Cosas



Tessel Renzenbrink (Editor de TTF Elektor)

Internet de las Cosas (Internet of the Things o IoT) es la fusión del mundo físico y el mundo digital. A los objetos comunes que contendrán ordenadores diminutos se les adjudicará una dirección IPv6 y tendrán una representación digital en Internet. Seremos capaces de tener acceso a la información de los objetos presentes en nuestras vidas cotidianas a través de Internet e incluso controlarlos remotamente.

Incluso muchos casos del IoT no requieren tener presencia humana en su bucle de ejecución. El IoT es como una red máquina-a-máquina donde los sensores y los actuadores se comunican unos con otros para cambiar el entorno al estado deseado. En septiembre se celebró el taller ROBIOTS: donde tuvo lugar un encuentro de los Robots con el Internet de las Cosas, como parte del festival PICNIC, de dos días duración, para medios e innovación, en Amsterdam. Después traté del futuro de IoT, sus desafíos tecnológicos y el potencia de su lado oscuro, con tres de los ponentes: Florian Michahelles, Heico Sandee y Stefan Gessler.

Los Expertos en IoT

El doctor Florian Michahelles es el director adjunto de los Laboratorios Auto-ID Zurich/St. Gallen. Los Laboratorios Auto-ID son la red global principal de siete laboratorios académicos de investigación científica en el campo de las redes RFID. Su interés científico se centra alrededor del Internet de las Cosas. El doctor Heico Sandee, director del proyecto en el Departamento de Ingeniería Mecánica de la Universidad Técnica de Eindhoven, está especializado en sistemas de control. Participa en el proyecto RoboEarth, una sociedad multidisciplinaria de investigadores de robótica de las academias y la industria, que pretende crear un 'world wide web' para robots. Stefan Gessler es director de proyecto en La División de Software y el Servicio de Investigación de los Laboratorios NEC en Europa. Participa en el proyecto Florencia, una actividad para crear robots de bajo coste que interactúan con un entorno doméstico inteligente para apoyar a la

gente mayor en su vida diaria en casa. También está involucrado en la formación del Foro del IoT.

El propósito de IoT

Tessel: ¿Por qué necesita el mundo el IoT?

Florian: Una respuesta sería: tiene que pasar. Es sólo la evolución natural de la tecnología.

Heico: Es la ampliación del control. Estoy tan acostumbrado a tener el poder en la yema de mis dedos para comunicarme con todo el mundo, que me siento casi estúpido de no poder comunicarme con las cosas que me rodean. Quiero ser capaz de verificar en mi teléfono si olvidé de cerrar con llave mi puerta. Quiero tener acceso a esta información e incluso ser capaz de cerrar la puerta remotamente.

Stefan: El IoT lo es todo sobre cooperación y comunicación. La comunicación es un requisito previo para la cooperación. Podemos pensar en miles de escenarios en los que es una ventaja el que las cosas trabajen juntas. Por ejemplo, un sensor que detecta agua en el suelo del baño puede indicar que la llave del agua se cierre por sí misma. O un coche que detecta un peligro para la seguridad delante de nosotros y que informa a los coches de detrás de ese hecho para que reduzcan la velocidad.

Heico: Y esto abre completamente nuevas posibilidades. En el proyecto RoboEarth estamos creando un sistema para intercambiar información entre objetos, robots y personas. Digamos que conectamos la cafetera con el despertador. Tan pronto como se apague la alarma, la cafetera empieza a hacer el café. Esto ya es posible si disponemos de una cafetera inteligente. Pero, ¿qué pasa si la máquina se queda sin café? Aquí es donde entra

el robot. Como tiene la capacidad de moverse él puede rellenar el contenedor del café en la cafetera. Los robots darán brazos y piernas a Internet. Y cuando los robots sean parte de una red, ellos podrán aprender unos de otros. RoboEarth está construyendo una base de datos basada en 'la nube' a la que pueden acceder diferentes tipos de robots. Cuando un robot ha solucionado un problema puede añadir esta información a la nube, de modo que el siguiente robot que encuentre el mismo problema pueda descargar las instrucciones para solucionarlo. De este modo no tenemos que planificar cada problema posible en la fase de diseño del robot, o equiparlos a todos con una serie grande de sensores para darles capacidad de aprendizaje.

Retos Tecnológicos

Tessel: *¿Cuáles son los desafíos tecnológicos de realizar IoT?*

Stefan: El desafío más grande es el modelado de la información. Permitir que cada dispositivo se comunique y coopere con el resto de dispositivos. Lo que pasa mucho en este momento es el denominado "enfoque parcelado". Tenemos un cierto campo de aplicación y dispositivos diseñados para ejecutar una tarea en particular. Este sistema es una caja cerrada, que consiste en sensores, actuadores y aplicaciones. Si queremos añadir una segunda aplicación tendremos que comprar todo esto de nuevo, los sensores, los actuadores, pero esta vez con diferentes programas y protocolos. El siguiente paso que tenemos que hacer es reutilizar los datos, compartiéndolos entre los dispositivos. Pero esto significa que tenemos que tener un interfaz interoperable. Eso es en lo que estamos trabajando en este momento.

Florian: En los laboratorios Auto-ID nos concentramos en el lado del producto: como hacer que el flujo de artículos y la cadena de suministro sea más eficiente. La idea es tener un código de barras o número RFID identificable de forma única que está asociado con los datos que describen el tipo de producto. Y ahora reconocemos que el código de barras (una tecnología de cuarenta años) también puede ser útil para los consumidores con 'smartphones'. Pero el consumidor está interesado en datos diferentes. Él no está interesado en el tamaño o el peso del producto, sino en los ingredientes o en su aplicación. A continuación tenemos que sacar la semántica de los datos que puede ser interpretada por robots. En el IoT sistemas diferentes, que surgen por motivos diferentes, tienen que trabajar juntos. Pero la estandarización no es tanto un desafío técnico, sino más bien organizativo y de motivación. Ya

que no se trata de una compañía que proporciona toda la infraestructura, sino elementos separados. ¿Cómo conseguir que todos los elementos trabajen juntos? En efecto, tenemos los datos generados por una compañía, pero ¿cuál es el incentivo a compartir? Así pues, realmente la pregunta es: ¿cuál es el ecosistema? ¿Cuál es el tipo de negocio que podemos establecer que haga que elementos diferentes quieran compartir porque todos se benefician mutuamente de ello?

Heico: estoy de acuerdo en que la interoperabilidad es el desafío más grande, pero ¿realmente es un problema tan grande? ¿No es más bien un tema de tomar una opción y atenerse a ello?

Stefan: es sobre la toma de una opción pero en esta opción tenemos que considerar todos los casos que posiblemente podrían aparecer.

Florian: Exactamente, ese es el punto. Tenemos que entender para qué lo necesitamos. Tecnológicamente es difícil construir una solución que encaje con cada caso. Así pues, es más fácil llegar a un acuerdo con, digamos, cinco casos y conseguir un estándar de ello.

Stefan: El IoT está desarrollándose desde abajo. No construiremos un sistema del IoT y lo encenderemos sin más como hicimos una vez con la tecnología de TV.

Infraestructura de Internet

Tessel: *Internet es la columna vertebral de IoT. ¿Es suficiente el estado actual de la infraestructura de Internet?*

Stefan: necesitaremos cada vez más un mayor ancho de banda. Todo está yendo muy rápido pero tenemos que estar preparados para los "Grandes Datos". El crecimiento de los datos es exponencial. Como uno de los portavoces indicó en el taller: se han creado 5 mil millones de gigabytes de datos desde que se tiene registro de ello. En 2011, el mismo volumen de datos fue creado en dos días. En 2013 lo crearemos en 10 minutos.

La capacidad del espectro de radio está intrínsecamente limitada. No podemos añadir una cantidad ilimitada de dispositivos a la red inalámbrica.

Florian: Tenemos que gestionar la escasez. Hoy día, si Internet se satura, nos puede llevar algo más de tiempo el descargar una imagen. La mayor parte del tráfico de datos no es urgente. Pero con el IoT podría ser diferente. Podría haber aplicaciones para la salud implicadas que exigen un acceso permanente. Para gestionar la escasez de ancho de banda tendríamos que dar prioridad las aplicaciones de salud o hacer



Figura 1. Florian Michahelles.



Figura 2. Heico Sandee.



Figura 3. Stefan Gessler.



que os usuarios paguen unos pequeños honorarios por el acceso con prioridad. Creo que siempre estaremos en los límites del ancho de banda pero no creo que sea un elemento de bloqueo para el IoT. Stefan: Al contrario, el IoT debería contribuir a una organización más eficiente del tráfico de datos. Se trata de tomar decisiones lo más pronto posible. Los datos que son relevantes para una red local no tienen por qué ser llevados a un centro de datos remoto. Lo que vemos ahora son dos distinciones: los datos locales y la nube. En el futuro tendremos diferentes capas y niveles diferentes de la nube. Los datos estarán organizados jerárquicamente: pueden quedarse de forma local, o hacer que estén disponibles regional o globalmente.

Cara oculta

Tessel: ¿Cuales son los posibles peligros de esta tecnología?

Heico: Una de las preocupaciones es la seguridad. Ya hemos visto el daño que un hacker malicioso puede hacer a Internet. Cuando todo tipo de dispositivo este conectado a Internet el impacto de los sistemas que se verán en peligro será cada vez más grande.

Stefan: Siempre que introducimos algo nuevo también puede ser usado para algo negativo. Otra cosa es que una nueva tecnología siempre influye en el modo en que vivimos nuestras vidas. Muchas personas consideran los cambios como una amenaza, por lo que también tenemos que tener cuidado con la aceptación del IoT en el futuro. La seguridad, pero también la intimidad, son seguramente factores claves.

Florian: Un peligro posible es la transparencia. Como humanos, estamos acostumbrados a entender por qué pasan las cosas. Con este componente digital añadido a nuestro mundo, puede que ya no percibamos la causa de los acontecimientos con nuestros sentidos. Por eso tenemos que estar pensando continuamente en el diseño de la tecnología, de manera que siempre tengamos un modelo mental para entender lo que está sucediendo. Da miedo si no puedo entender lo que pasa. Eso significaría que estoy siendo controlado por algo. Este es el escenario oscuro. Pero, en mi opinión, el peor de los escenarios para el Internet de Cosas es si esto no pasa.

Elektor y el IoT

Como indica correctamente Stefan Gessler, el IoT no es una tecnología con un único momento de puesta en funcionamiento. Más bien es un conglomerado ilimitado de tecnologías y aplicaciones,

muchas de las cuales ya están en funcionamiento. Probablemente de forma inconsciente, Elektor ya ha cubierto muchas de ellas. Sólo en la edición de noviembre he contado cinco proyectos del IoT, demostrando que los nuevos departamentos electrónicos de Elektor, .LABs y .MAGAZINE, cubren el IoT sin una identificación expresa del mismo. ¡Tenemos que trabajar en ello!

Una de las mejores aplicaciones conocidas del IoT es el medidor inteligente. Sirviendo de terminal nervioso de la red (eléctrica) inteligente, los medidores inteligentes proporcionan la información del consumo de energía a nivel local, permitiendo que, tanto los proveedores de energía como los consumidores, tomen decisiones con datos sobre el ahorro de la energía. El mes pasado Søren Mikkelsen nos mostró como instalar un medidor de electricidad con S0 compatible, que envía datos detallados del consumo de energía a una base de datos localizada de forma local o en la nube. En la edición de noviembre, Benedikt Sauter, en su serie de artículos "Linux Embebido de Manera Fácil", describía cómo liberar nuestra placa Linux de Elektor de una existencia solitaria, estableciendo una conexión de red. Y Jens Nickel introdujo la librería C para el ElektorBus, para que les sea más sencillo a los desarrolladores añadir nodos con microcontroladores. Tanto la placa Linux de Elektor como el ElektorBus están pidiendo a gritos una conexión a Internet.

También en la edición de noviembre de 2012 el doctor Joost Mertens llegó a los fundamentos básicos de la tecnología de Internet explicando como construir una tarjeta interfaz de red (NIC) usando de un microcontrolador AVR de 8 bits. Cuando lo emparejamos con una vieja cámara con puerto paralelo, el microcontrolador permite que envíemos vídeo a múltiples receptores de una red.

En 3ª entrega del "Curso de Arduino" titulado 'Conectando el mundo físico a nuestro teléfono Android', David Cuartielles y Andreas Goransson han intentado enseñarnos cómo comenzar a construir nuestro propio IoT. ¡Que continuará alegremente este mes! ¿No es importante descubrir una tendencia diferente cuándo parece que ya tenemos en nuestras manos todo lo que lo define? ¿Qué proyecto de la edición de diciembre de 2012 creéis que está preparado para el IoT?

(120632)

Referencias en Internet

- [1] www.autoidlabs.org
- [2] www.roboearth.org
- [3] <http://iot-forum.eu>



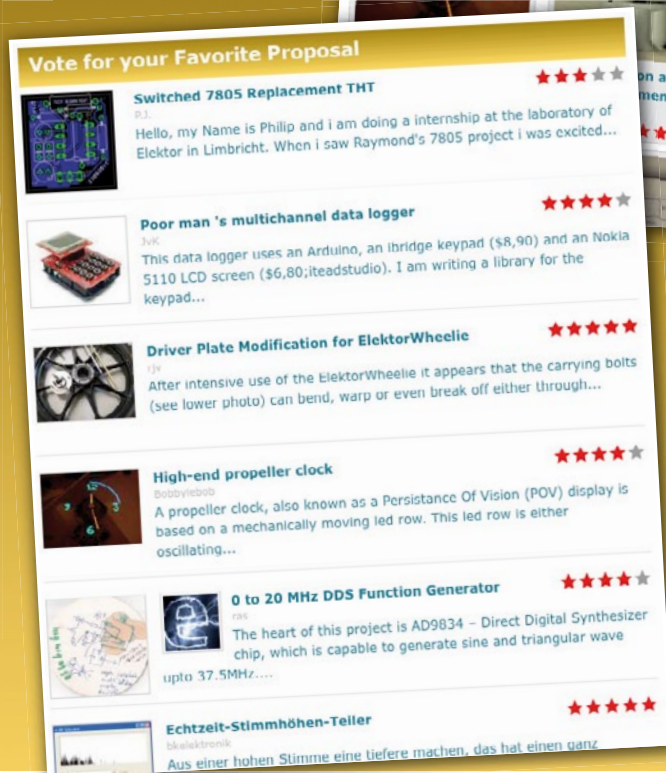
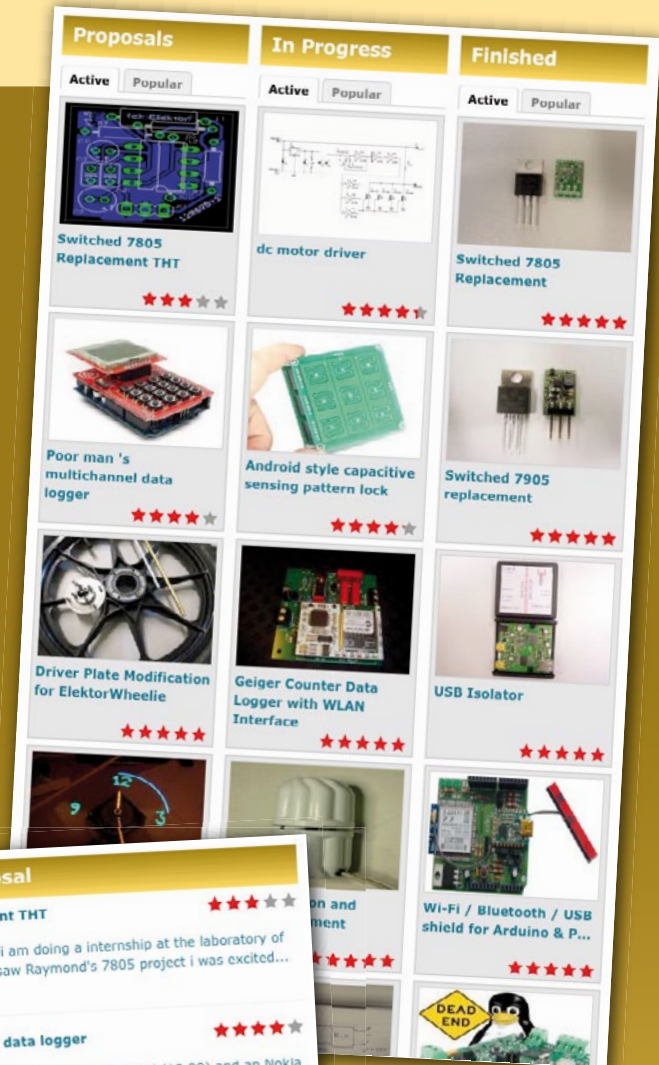
elektor@labs

Sharing Electronics Projects

Elektor.LABS es una comunidad online para apasionados de la electrónica. Un lugar donde compartir tus proyectos y tomar partido en aquellos que otros hayan creado. Se trata de un sitio en el cual discutir sobre desarrollos y electrónica en general.

El equipo de editores e ingenieros de Elektor te ofrecerá su apoyo hasta finalizar satisfactoriamente tus proyectos. Así mismo, te ayudarán a redactar un artículo para ser publicado en Elektor.MAGAZINE, ie incluso desarrollar un producto completo que puedes vender en Elektor.STORE!

Únete
ya



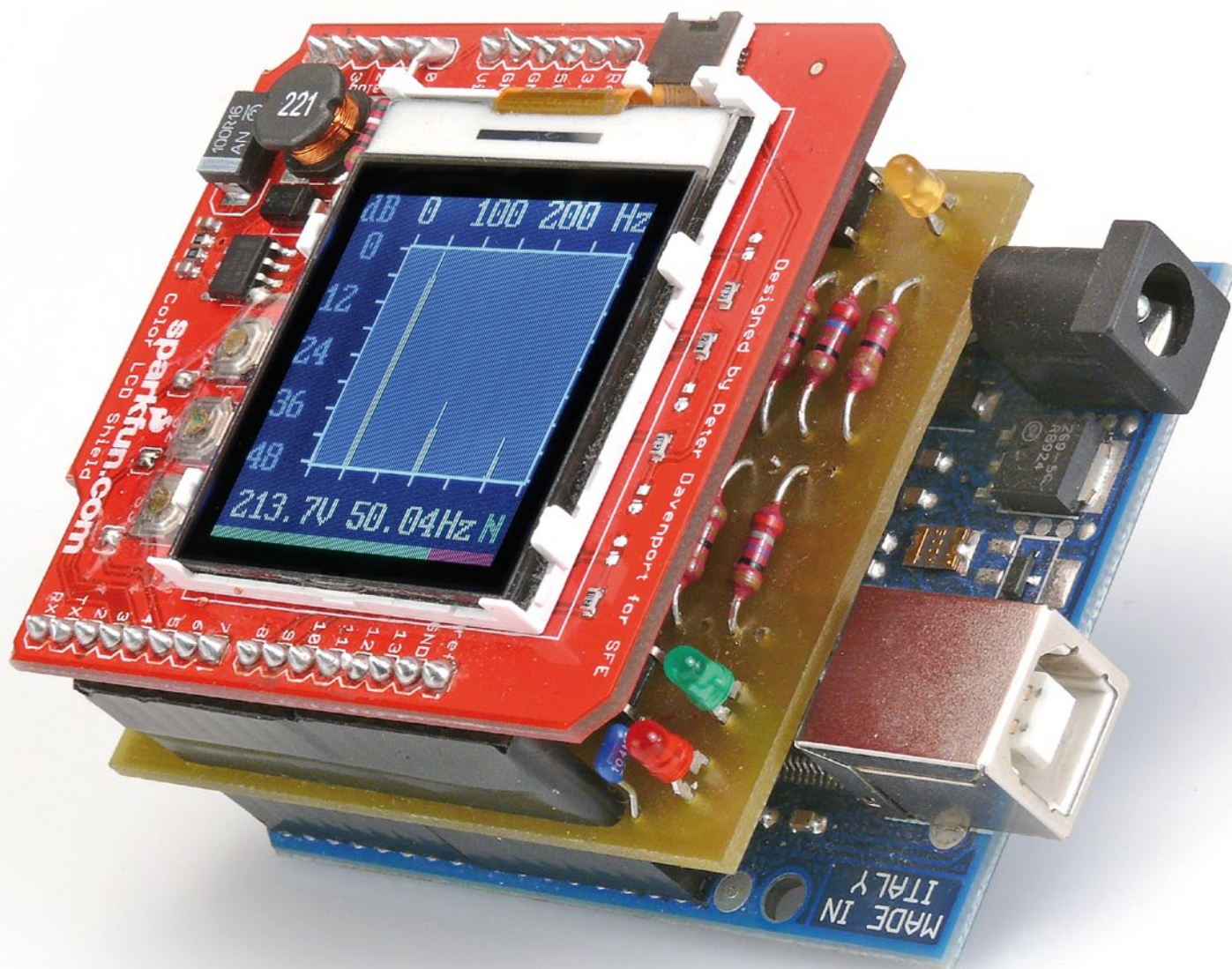
¡Elektorízate! Visita www.elektor-labs.com

Analizador de tensión de red con Arduino

Análisis de frecuencia en un mini display de color

Fidelis Theinert
(Holanda)

¿Un analizador de espectro completo con una sencilla placa de Arduino? En este artículo demostraremos que se puede hacer esto si se adapta el software a la potencia de cálculo disponible. Con la ayuda de un shield de filtrado y un shield de display en un Arduino estándar se puede realizar un analizador de espectro de la tensión de red, que muestre mediante un análisis rápido de Fourier la 'polución' de la tensión de red de la casa o de la oficina.



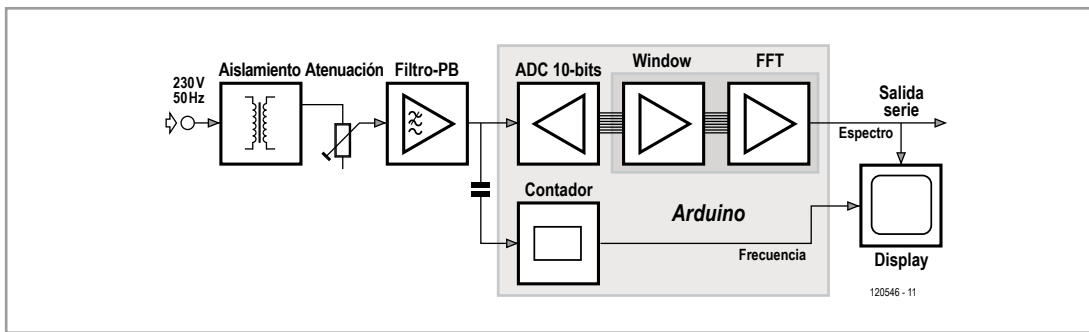


Figura 1.
Concepto de hardware del
analizador de espectro.

Una sencilla placa Arduino es mucho más potente y polifacética de lo que la mayoría de los lectores creen. En este artículo demostramos, con la ayuda de un ejemplo práctico, no sólo cuáles son las posibilidades de un microcontrolador de 8 bits, sino también dónde están sus límites. Luego se explicará cómo utilizar los componentes de un microcontrolador de manera eficiente y qué herramientas gratuitas de desarrollo se necesitan para crear un proyecto parecido. El software y hardware están contruidos de forma modular por lo que los módulos también se pueden utilizar en otros proyectos. La aplicación se probó con éxito en un Arduino Duemillennova, UNO R1, R2 y R3. El circuito analiza la tensión de red mediante una transformada rápida de Fourier (FFT del inglés Fast Fourier Transform) y muestra en un display los armónicos presentes. A partir de ellos se puede evaluar la calidad de la señal suministrada (de energía). Estos armónicos más altos provocan pérdidas en máquinas y motores que se convierten en calor, pero por las que tiene que pagar el usuario. Además, una señal de alimentación no ideal senoidal influye de forma negativa en el rendimiento de las fuentes de alimentación conmutadas en PCs u otros aparatos.

Análisis de Fourier

La transformación rápida de Fourier es un algoritmo matemático con el que se puede convertir una señal discreta del dominio del tiempo al dominio de la frecuencia. La ventaja en tiempo de la FFT en comparación con la transformación discreta de Fourier (DFT) puede llegar a ser bastante elevada para un gran número de muestras (N). En la FFT se divide una transformación de Fourier con una longitud de N en dos transformaciones diferentes con una longitud de N/2. Esto proporciona un gran ahorro en tiempo porque se limita el número necesario de operaciones de cálculo, pero tiene como consecuencia que

el número de muestras (N) tiene que ser una potencia de 2 (64, 128, 256, ... 4096 ...). Sin embargo en la práctica esto no supone ninguna limitación, porque se puede elegir libremente la longitud N del array utilizado.

Según el teorema de muestreo Nyquist-Shannon, el ancho de banda de la señal a analizar no puede ser mayor que la mitad de la frecuencia de muestreo. Esto significa que la frecuencia de muestreo tiene que ser, por lo menos, dos veces la frecuencia más alta a analizar. Así que con una frecuencia de muestreo de 800 Hz se puede analizar una señal con un ancho de banda de 400 Hz y mostrar el espectro de una señal de tensión de red hasta el séptimo armónico (50 Hz x 7 = 350 Hz).

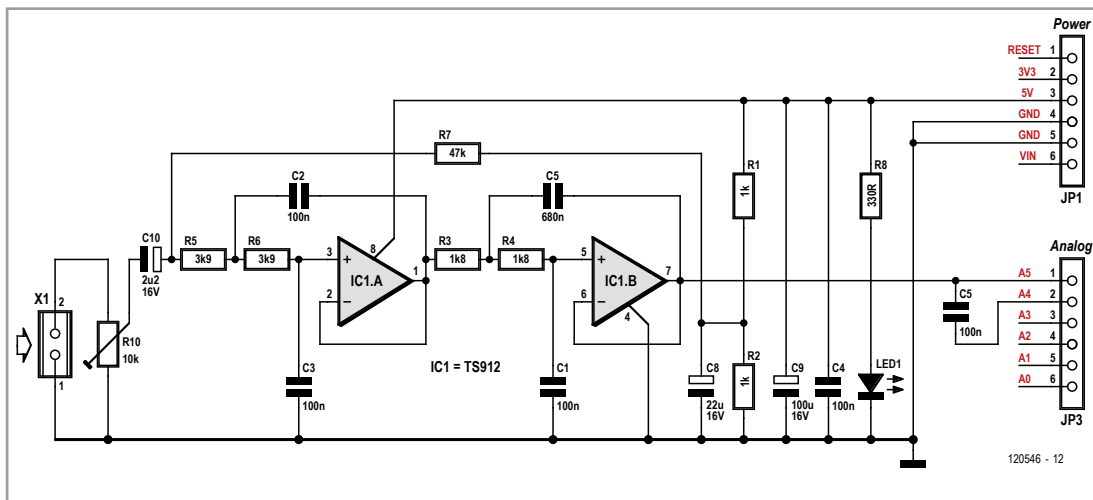
Hardware

El hardware consta de 3 placas: la placa Arduino [1], un shield de filtrado desarrollado por el propio autor y un shield de display [2]. Los dos shields, se pueden colocar en la placa Arduino (por supuesto el shield LCD colocado encima de todo), de modo que surja un módulo compacto y robusto sin hilos sueltos. La **figura 1** muestra el esquema de bloques de todo el diseño.

También es posible enviar al PC el espectro calculado por el Arduino a través del puerto serie. La gráfica se puede mostrar con la ayuda de un programa de terminal (por ejemplo Putty). El envío de los datos tarda un poco más de tiempo porque el UART envía el dibujo completo a sólo 57.600 baudios al PC. Esta solución es interesante para, por ejemplo, los usuarios que no disponen de un shield LCD.

Para evitar que la señal de entrada contenga componentes espectrales por encima de los 400 Hz, se ha colocado delante de la entrada un filtro de paso bajo Sallen Key de cuarto orden (o sea un filtro antialiasing), con un ancho de banda de 350 Hz (**figura 2**). Las dos secciones Butterworth

Figura 2.



están construidas alrededor de un operacional dual del tipo TS912. Este operacional tiene una salida 'rail-to-rail' y un amplio alcance de tensión de alimentación desde 2,7 hasta 16 V, de modo que se puede alimentar directamente con

la tensión de alimentación de la placa Arduino. La señal de salida del filtro pasa del terminal 7 del IC1 al canal 5 del ADC dentro del microcontrolador (un ATmega328P). Además la salida del operacional está acoplada capacitivamente

Trucos FFT

El algoritmo ‘butterfly’ de Cooley-Tukey implementado es una ‘transformación in situ’; esto significa que el resultado de la transformación al final del cálculo se encuentra en el mismo buffer en que originalmente se encontraban las muestras de la señal de entrada. Por lo tanto ya no se dispone de la señal de entrada después de la transformación. La ventaja de esto es que la memoria de buffer requerida se reduce a la mitad, una característica muy útil porque el microcontrolador elegido (ATmega328) sólo dispone de 2048 bytes de memoria de trabajo.

Como hemos comentado anteriormente, el número de muestras tiene que ser una potencia de 2 y con 256 (28) muestras se cumple este requisito. El ADC tiene una resolución de 1024 pasos, de modo que se necesitan 2 bytes por muestra para su almacenamiento o sea 512 bytes para toda la serie de muestras. Luego se necesita un buffer para el cálculo de la parte imaginaria de los números; después de todo la transformación de Fourier es una conversión con números complejos. Por supuesto que este buffer tiene el mismo tamaño que el buffer de la parte real y ocupa también 512 bytes. Así que, en total se necesitan 1024 bytes para guardar los valores de la FFT. Esto es la mitad de la memoria disponible.

Para poder utilizar la librería matemática del compilador GCC (en una FFT no sólo se trata de operaciones matemáticas sencillas como son sumar y multiplicar, sino también de las operaciones con senos, cosenos, raíces y potencias), sería

muy útil guardar las muestras no como variables enteros sino como variables en coma flotante de simple precisión. Sin embargo, una variable en coma flotante de simple precisión necesita 4 bytes, lo que requiere un buffer de 2048 bytes, exactamente la misma cantidad que la memoria total del microcontrolador, por lo tanto demasiado para la plataforma elegida. Una solución a este dilema podría ser la reducción del tamaño de la FFT a 128 muestras, pero esto produce una resolución espectral más baja y por lo tanto no es deseable. Para poder trabajar aún con 256 muestras, tanto la parte real como la imaginaria de los números complejos se graba como un valor entero (2 bytes/muestra), antes de cada operación matemática se convierte a una variable en coma flotante y después de las operaciones se vuelve a convertir a una variable entera. Este incremento adicional en tiempo de procesamiento antes de estas conversiones genera un retardo, pero es aceptable. El error de redondeo ($1/65536$) introducido por esta conversión es un factor 64 veces más pequeño que el error de cuantificación del ADC ($1/1024$) y es por lo tanto inapreciable (ver las funciones 'FFT_Int2Float()' y 'FFT_Float2Int()' en el módulo `fft lib.c`).

Para hacer un uso lo más eficiente posible de la memoria, se colocan primero las muestras leídas por el ADC en el buffer destinado a los números imaginarios. En ese momento la FFT aún no utiliza este buffer. A continuación las muestras se normalizan, se adaptan a la escala, y se colocan en el

para la medir la frecuencia con la conexión A4 del conector de la entrada analógica de la placa del Arduino.

En la **figura 3** se puede ver el placa-shield diseñada para el filtro. Ahí se encuentran algunos componentes extras que no se utilizan en esta aplicación. El diseño de la placa está disponible en la página web correspondiente a este artículo [3].

La tensión de red se separa galvánicamente mediante un adaptador de red (un tipo 'antiguo' con un transformador), la salida del adaptador se conecta con la regleta X1. Como el filtro completo amplifica una sola vez, una señal con una amplitud de $2 V_{eff}$ ($5 V_{tt}$) es suficiente para cubrir el alcance dinámico del ADC dentro del ATmega. El transformador casi no deforma la señal de salida porque casi no lleva carga, como muestra la **figura 5** (la alimentación del analizador se realiza a través de otro adaptador o una conexión USB). Así se pueden medir todos los armónicos en la señal de red con bastante

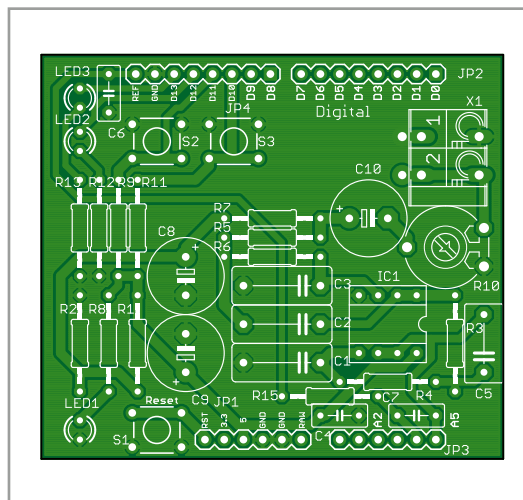
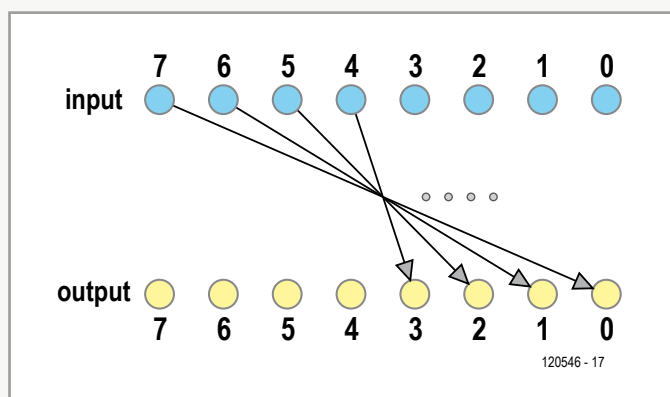


Figura 3.
El diseño de la placa del shield de filtrado.

precisión. Con la ayuda del potenciómetro R10 se puede atenuar la señal de salida del adaptador en C10 hasta unos $1,7 V_{eff}$ con una tensión de red de 230 V. Con un polímetro de referencia se puede verificar la tensión de red y ajustar

buffer de la parte real de las muestras. Ahora se puede borrar el buffer imaginario y aplicar la función de ventana sobre la parte real de las muestras (la función de ventana es necesaria para cumplir con las condiciones esenciales de la FFT).

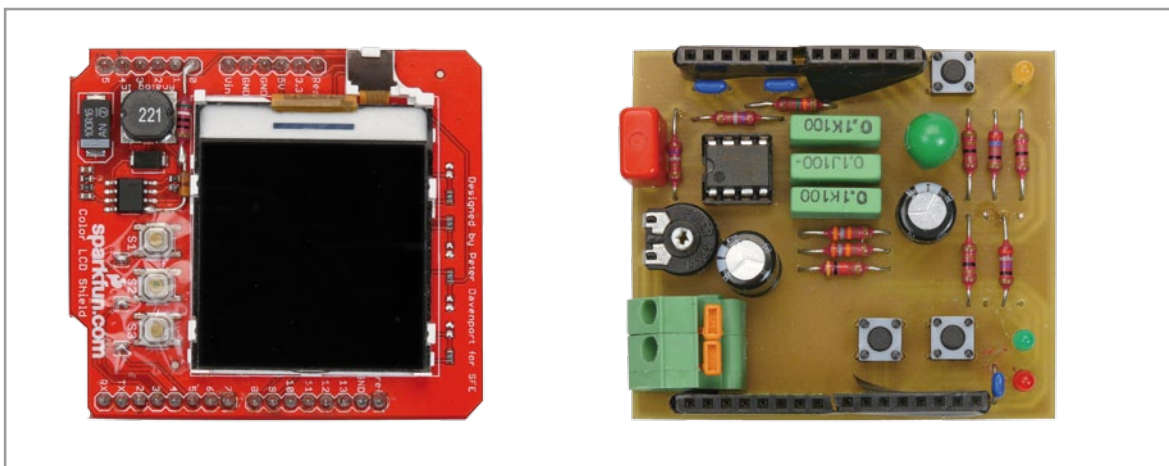


El algoritmo de bitreversal

Un componente importante en la implementación de la FFT es el denominado algoritmo de 'bitreversal'. Se utiliza para el direccionamiento de las muestras de los buffers. En sí es una operación muy sencilla (poner al revés los bits de un

valor en un registro o intercambiar el orden de principio a fin, ver dibujo), pero no es tan fácil de implementar. Las implementaciones rápidas utilizan para esto unos registros hardware adicionales. A menudo se utiliza también una tabla, pero para eso se requiere mucha memoria y eso es algo de lo que los pequeños microcontroladores disponen de forma limitada. La variante más lenta es la de desplazar todos los bits hacia fuera del registro y a continuación desplazarlos hacia dentro de otro registro en orden inverso. Para eso se necesita un contador dentro de un bucle, que hay que controlar, y a continuación 16 operaciones de desplazamiento. Para una buena consideración entre el uso de la memoria y el tiempo de cálculo, se utiliza una combinación de ambos métodos. El byte se divide en 2 tramos de 4 bits. Estos valores de cuatro bits se convierten mediante una tabla por separado para luego juntar los dos resultados en el orden correcto (ver la función 'fft_BitRev()' en el módulo fft_lib.c). El ahorro de tiempo resultante asciende a $3 \mu s$ ($2 \mu s$ en comparación de $5 \mu s$) por cada llamada a esta función, pero cuesta 75 bytes más de código.

Figura 4.
Las tres placas impresas
utilizadas juntas una al lado
de otra.



R10 de tal manera que la tensión mostrada en el display coincida con la tensión de red actual. Se aconseja utilizar un adaptador con una tensión de salida de menos de 20 V.

Entorno de desarrollo

Para el desarrollo del software no se ha utilizado el entorno estándar de Arduino, sino el 'GCC cross-compiler' de la serie microcontroladores AVR. El compilador 'cross-avr' se puede instalar bajo Linux junto con los 'bin-utils' correspon-

dientes desde los repositorios de la distribución, el paquete 'WinAVR' [4] se puede instalar bajo Windows. Para ambos sistemas operativos se ha utilizado la versión 4.3.3 del compilador GCC [5] y el programa 'avrdude' para subir el fichero hex compilado. Este programa es parte del paquete WinAVR de Windows, pero también existe una versión para Linux. En ambas versiones se incluyen ficheros scripts para 'subir' el ejecutable al Arduino. Sólo hay que configurar el puerto serie correcto utilizado en estos ficheros para la comu-

Tabla 1. Módulos de software

Nombre del fichero código fuente	Nombre del fichero header correspondiente	Descripción de la funcionalidad
avrfft.c	(no tiene)	Fichero código fuente principal
drw_fft.c	drw_fft.h	Funciones de dibujo de alto nivel para la presentación de la gráfica en el LCD
fft_lib.c	fft_lib.h	Definiciones y funciones para el cálculo FFT
freq_cnt.c	freq_cnt.h	Definiciones y funciones para la adquisición de datos y medición de la frecuencia
lcd_driver.c	lcd_driver.h	Funciones de bajo nivel y definiciones para controlar el display LCD
menu.c	menu.h	Funciones para la lectura del estado de los botones en el shield LCD
plt_fft.c	plt_fft.h	Funciones de alto nivel para generar la gráfica a través del puerto serial
ser_avr.c	ser_avr.h	Funciones de bajo nivel para el envío y recepción de datos
std_avr.c	std_avr.h	Definiciones y funciones estándar frecuentemente utilizados
timer_0.c	timer.h	Funciones para los temporizadores y diseño de la temporización del sistema

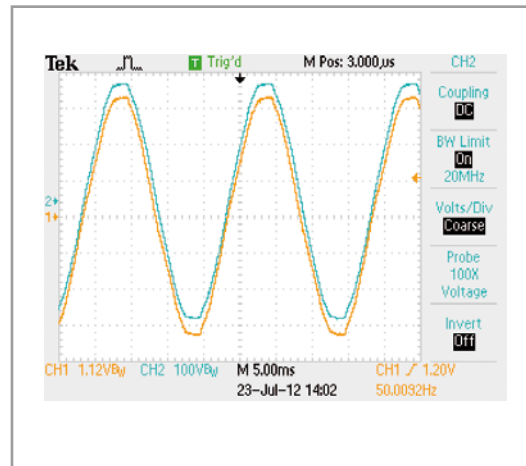
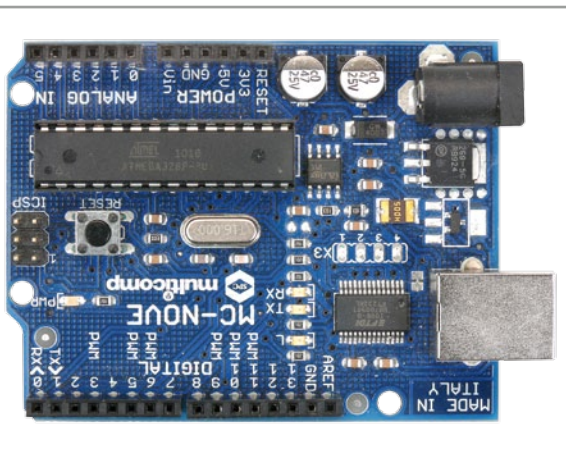


Figura 5. La tensión de red en el lado primario del transformador (azul) y en el lado secundario (amarillo).

nicación con el Arduino y la ruta de la ubicación del ejecutable en el PC.

Más adelante se necesitará el otro programa 'make' para compilar y linkar el código fuente de todos los módulos con la ayuda del makefile; Este programa también es parte de WinAVR y también existe para Linux.

Se puede elegir cualquier editor para modificar el código fuente. Eclipse con la extensión CDT es un editor práctico, pero Notepad++, Scite o cualquier otro editor es suficiente.

Software

El software está creado de forma modular y dividido en varios ficheros con el código fuente (ver **tabla 1**).

Cualquier programa en C empieza con la función 'main()'. Esta se puede encontrar en el fichero avrfft.c. Después de arrancar el programa se inicializan primero varias variables globales del sistema y del hardware con la ayuda de las funciones init correspondientes. Luego se borra el display del shield LCD y se prepara el temporizador del sistema. Después el programa entra en un bucle interminable, donde se configura el temporizador del sistema en un tiempo de 2,5 s. Esto significa que cada 2,5 segundos se ejecuta un nuevo ciclo.

Los distintos pasos del programa se ejecutan siempre en el mismo orden por cada ciclo. En la **figura 6** se puede ver un diagrama de flujo simplificado.

Después de que el temporizador (timID) haya liberado el siguiente ciclo, el ADC lee primero todas las 256 muestras necesarias para la FFT con una frecuencia de muestreo de 880 Hz. Luego se calcula el valor efectivo real (TRMS) de la tensión.

En el siguiente paso se mide la frecuencia de la señal del terminal A4 del conector 'Analog In' con la ayuda del comparador incorporado y del Timer1 del microcontrolador.

Los pasos posteriores (la adaptación de la escala de los valores medidos y la aplicación de la función de ventana seleccionada) sirven de preparación para el cálculo real de la FFT. Al final se envían los cálculos al display del shield LCD para mostrar el espectro.

Presentación y utilización

El display utilizado en el shield LCD tiene una resolución de 131 x 131 píxeles, que se utilizan para dibujar el espectro en una ventana de 80 vertical y 96 horizontal. Esto resulta en una resolución del dibujo de 0 Hz...300 Hz (96 líneas x 3,125 Hz = 300 Hz) y 10 píxeles por 6 dB vertical. La FFT calcula el espectro de 0...400 Hz, pero las frecuencias por encima de 300 Hz ya no se muestran; estas sí se incluyen en el dibujo enviado por la salida serie. Debajo de la ventana se muestran la tensión y la frecuencia de red medidas, la tensión también se muestra en una barra analógica.

En el shield LCD hay 3 botones a la izquierda del display que sirven para ajustar el analizador de espectro. La funcionalidad se ha programado de la siguiente manera. Después del arranque, el analizador empieza a medir directamente y muestra en el display la figura logarítmica del espectro medido. Pulsando el botón superior se puede conmutar a una figura de frecuencia con un eje vertical lineal. La siguiente pulsación sobre este botón hace que conmute al dominio de tiempo (función osciloscopio) y el display muestra las primeras 96 muestras adaptadas a la escala del

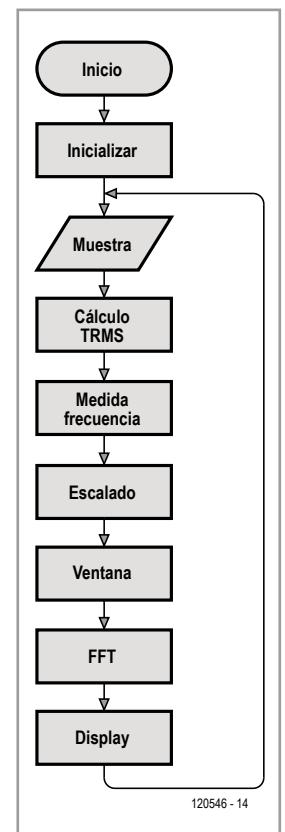
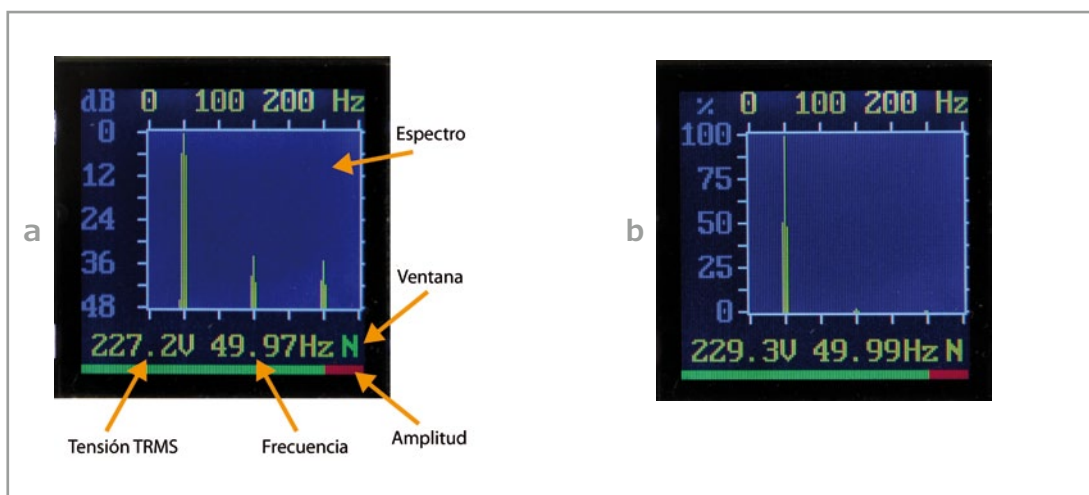


Figura 6. Diagrama de flujo del programa.

Figura 7.
Representación del dominio de frecuencia en el display (gráfica logarítmica (a) y lineal (b)).



buffer de entrada (ver las **figuras 7 y 8** para las configuraciones posibles). Otra pulsación sobre el primer botón hace que se muestre el contenido completo del buffer de entrada con la función de ventana aplicada. La cuarta pulsación sobre el botón superior hace que vuelva a conmutar al dominio de frecuencia con una división logarítmica, igual que la situación después del arranque. Con el botón inferior se puede seleccionar la función de ventana FFT. Después de arrancar se utiliza por defecto una ventana Hanning. Una pulsación sobre el botón inferior conmuta a una ventana Blackman y la siguiente pulsación a una ventana Hamming. En la parte inferior derecha del display se muestra la función de ventana: 'N' para Hanning, 'B' para Blackman y 'M' para Hamming.

Con una pulsación sobre el botón del medio se puede conectar o desconectar la comunicación serie con un terminal. Si no se envían datos al ordenador, el indicador de la función de ventana conmuta de color amarillo a verde y viceversa, después de cada espectro calculado. Si la comunicación serie está conectada, el indicador cambia de color amarillo a rojo.

Prestaciones

Dependiendo de las configuraciones del compilador un cálculo de la FFT tarda en el peor de los casos sin optimización (-O0) 1 segundo y el tamaño del programa compilado es de unos 26 KB. Con una optimización (-O2) se reduce el tamaño del ejecutable a unos 18 KB y el cálculo de la FFT a 700 ms. La ejecución del programa completo (medición de la frecuencia, adquisición, cálculo de TRMS, adaptación a la escala, función

de ventana, cálculo FFT, adaptación a la escala, dibujar en el display) dura unos 2100 ms con una optimización de -O2.

Se ha optado por un ciclo de tiempo de 2500 ms para garantizar un tiempo de reacción previsible y constante del sistema. Por supuesto que esto es sin el envío de los valores al puerto serie. Para una descripción detallada de los flags '-O', mira la documentación del compilador GCC.

De la memoria de trabajo (2048 bytes RAM) disponible se utilizan 1372 bytes (de los cuales 1024 son para los buffers de la FFT). Esto es 2/3 de la memoria total disponible. De la memoria de programa (32 KB ROM) se utilizan sólo 18 KB. Teniendo en cuenta el bootloader del Arduino (unos 2 KB) queda más de 10 KB de ROM disponible para ampliaciones.

Conclusión

El sistema aquí descrito es un analizador completamente funcional de FFT en tiempo real para la tensión de red, que con una FFT en coma flotante de 256 puntos calcula cada 2,5 s un nuevo espectro y lo muestra en una pantalla gráfica. Todo esto está implementado en una plataforma Arduino de la que sólo se utiliza un tercio de la memoria. Esta aplicación muestra lo potente y eficiente que puede ser un sencillo microcontrolador de 8 bits utilizado para realizar avanzados cálculos matemáticos y mostrar los resultados gráficamente, incluyendo un interfaz de usuario y la comunicación con un PC. Pero también hay limitaciones, como son la memoria de trabajo disponible (por lo que hay que limitar la FFT a 256 puntos) y la velocidad de los cálculos. Pero aquí todavía queda espacio para una mayor optimización.

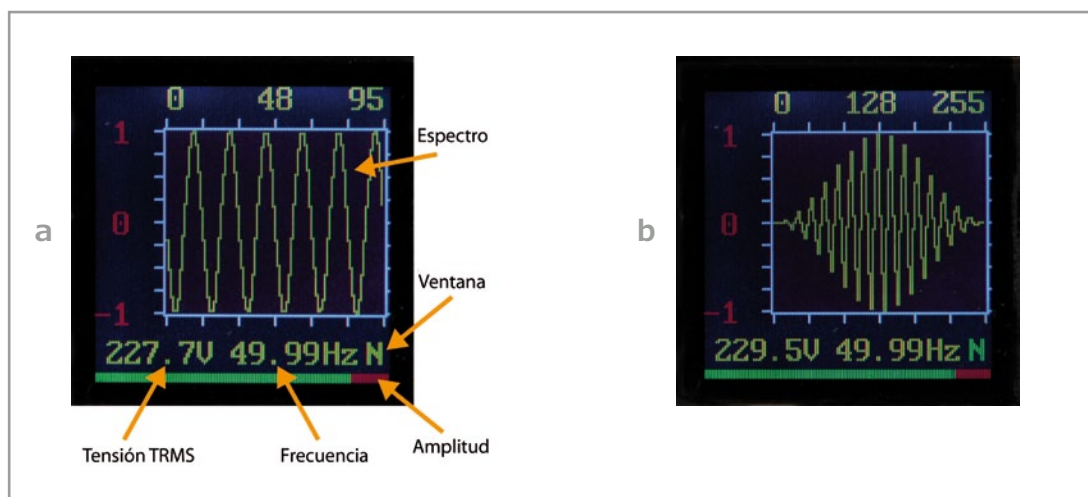


Figura 8. Representación del dominio del tiempo en el display (96 muestras sin función de ventana (a) y 256 muestras con una ventana Hanning (b)).

Durante el desarrollo sólo se han utilizado herramientas de código abierto, y la plataforma Arduino que es igualmente abierta y está disponible gratuitamente.

El software está hecho de forma modular y los módulos son fáciles de utilizar en otros proyectos. Además el software se ha publicado bajo la licencia GPL V3, de modo que se puede integrar también en proyectos propios. Las condiciones de GPL están publicadas en el sitio web de 'GNU Operating System' [6]. El software completo y el diseño de la placa se pueden descargar gratuitamente de la página correspondiente al sitio web de Elektor [3].

Posibles ampliaciones

Pese a la amplia funcionalidad ya implementada, se podría ampliar con un cálculo de la THD (total harmonic distortion).

Otra ampliación podría ser un analizador del habla, que calcula el espectro de la señal del habla con la ayuda de FFT y lo muestra en el display. Para eso hay que adaptar el ancho de banda del filtro de paso bajo. El ADC del ATmega328 se puede leer con una resolución máxima de 10 bits y una frecuencia de muestreo de 15 kHz, más que suficiente para una señal del habla con un ancho de banda de unos 5 kHz.

Para mejorar la velocidad del sistema, se puede ejecutar la adquisición de muestras nuevas en un segundo plano a base de interrupciones, mientras el programa está ocupado en el envío de datos al display. Esto puede suponer una mejora en tiempo de 0,3 s (256 muestras/800 Hz frecuencia de muestreo). Luego se puede controlar el

display más rápidamente a través del interfaz SPI del ATmega328.

Por supuesto que se puede programar un amplio backend en el PC para la representación de la gráfica, pero esto no tiene que ver con la implementación de una FFT en una placa Arduino.

(120546)

Enlaces Web

- [1] <http://arduino.cc/en/>
- [2] www.sparkfun.com/products/9363
- [3] www.elektor.es/120546
- [4] <http://sourceforge.net/projects/winavr/>
- [5] <http://gcc.gnu.org/onlinedocs/gccint/index.html>
- [6] <http://www.gnu.org/copyleft/gpl.html>

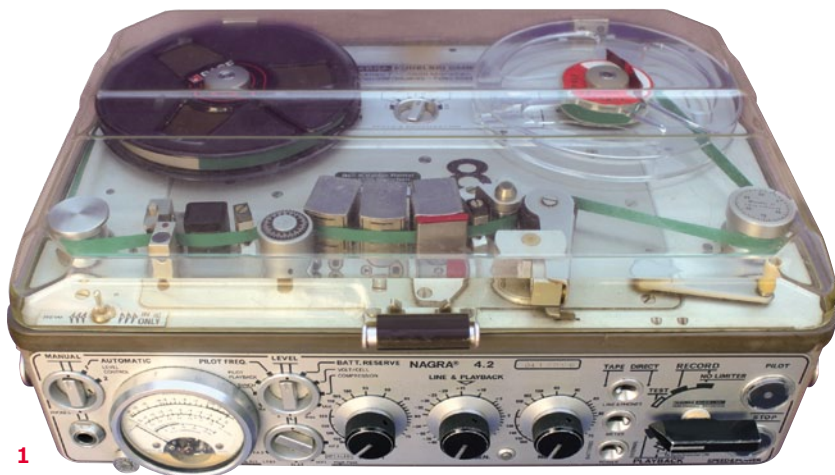
El autor

Después de sus estudios en electrotécnica en la Technische Universität Berlin, Fidelis Theinert trabajó durante más de 20 años en varias empresas en Alemania y Holanda como desarrollador de hardware y software y como profesor en la universidad. Actualmente trabaja como profesor de electrotécnica en la Haagse Hogeschool en Delft. Sus conocimientos se centran en el área de sistemas embebidos enfocados en el tratamiento de señales tanto de audio como de video.



El Grabador de Cinta Nagra IV

Una delicia de un reportero de 1968



Peter Beil (Alemania)

Erase una vez el grabador de cinta magnética (magnetófono) portátil más famoso del mundo (y del que muy pocos lectores de Elektor habrán oído hablar). El Nagra (ver **Figura 1**) era un dispositivo profesional, desarrollado para aplicaciones transportables profesionales, bien para su uso en transmisiones o bien para la grabación de audio sincronizado con el vídeo, para el cine o la televisión.

Stefan Kudelski, un Polaco de nacimiento y su padre espiritual, especificaron para este equipo los estándares más altos técnicamente mensurables en aquel momento. Sus transistores internos de América, los potenciómetros de Inglaterra, los interruptores de Suiza, los condensadores de Holanda y los conectores de Alemania. El nombre Nagra deriva de la palabra polaca para la grabación. Su calidad y fiabilidad han dado el estatus legendario Nagra y uno podía encontrar

los equipos desde la fría y helada Alaska hasta el calor insoportable del Sáhara.

La era del magnetófono portátil comenzó en los años 50 usando modelos controlados por mecanismos de relojería, con el lanzamiento del Nagra III, controlado eléctricamente, que se produjo a principios de los años 60. Al final de esa década llegó el modelo más extensamente usado, el Nagra IV, que es el sujeto de este artículo.

El Nagra toma su energía de 12 pilas de formato 'D' (IEC R20) que, juntas, suman la tercera parte del volumen del equipo (ver **Figura 2**). La tensión no regulada de estas pilas estaba disponible para el bobinado de la cinta o para el amplificador monitor de reproducción. Éste último entregaba una potencia clara de 1,5 vatios con intensidad y calidad de sonido remarcables. Aun siendo ahorradores en el uso del avanzado y el rebobinado rápido de la cinta y en la reproducción en el altavoz, un juego de baterías se gastaba sin mucha dificultad. Más tarde se pudo disponer de unas baterías recargables con este factor de forma, lo que significaba que tenías que arrastrar 20 libras (unos 10 kilos) contigo.

Internamente una tensión de referencia de -10 voltios gobernaba todas las secciones de la circuitería que controlaba la modulación requerida. Si nos estamos preguntando por los -10 voltios (volveremos a este dato más tarde), déjenme decirles que la circuitería completa está basada en una tensión de masa positiva extrañamente poco usual. Esto puede estar relacionado en parte con los transistores de germanio que se usaban originalmente. Por compatibilidad con versiones anteriores esto ya no se pudo modificar posteriormente, pero causó dolores de cabeza significati-

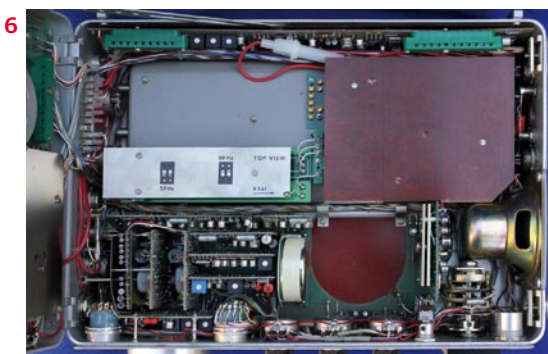


Retrónica es una sección mensual que trata sobre electrónica antigua, incluyendo diseños legendarios de Elektor. Contribuciones, sugerencias y peticiones son bienvenidas; por favor, enviad un "e-mail" a editor@elektor.com.



vos para la floreciente industria de accesorios. Un simple contacto con otro dispositivo podía provocar también un cortocircuito, ya que la mayor parte de estos dispositivos estaban conectados con masa negativa. Esto podía pasar cuando se enchufaba un cable de audio con su terminal apantallado conectado masa en cada extremo... Electromecánicamente no hay mayores sorpresas. El motor del eje rotatorio, totalmente encapsulado, tenía tres velocidades reguladas electrónicamente que proporcionaba velocidades de 3,75, 7,5 y 15 pulgadas/s (9,5, 19 y 38 cm/s). Un sofisticado mecanismo de fricción y control hacía funcionar los rodillos de entrada y salida usando, como ya se ha mencionado, un único motor (ver **Figura 3**). A máxima tensión el equipo podía rebobinar bastante rápidamente, así como el avance rápido a doble velocidad. Aceites y grasas especiales, que podían ser aplicadas por solicitud al servicio de distribuidores, hacían posible un funcionamiento a temperaturas ambiente comprendidas entre -20 y +50 grados C (!) (-4 a +122 grados F). El dispositivo de la figura, un Nagra 4.2 de los años 70, idéntico a simple vista con las primeras máquinas, representa la cima del desarrollo técnico de la serie. En el panel delantero tenemos, además de los elementos de control que se explican por sí mismos en el lado derecho, dos indicadores denominados rotatorios, normal-

mente conocidos como indicadores de estrella (ver **Figura 4**). Estos indicadores proporcionaban un control fiable de los parámetros de funcionamiento y sólo consumían una fracción del consumo de una bombilla indicadora. Toda la electrónica fue construida usando componentes discretos. Por aquel entonces los buenos integrados de audio estaban todavía en desarrollo y tenían una reputación pobre de fiabilidad, distorsión y ruido. Sólo unos años más tarde se construyó un modelo usando integrados. Las entradas y salidas usaban transformadores balanceados, mientras que las etapas amplificadoras internas eran asimétricas. Una característica peculiar era la salida balanceada flotante de una amplitud de 4,4 voltios, un orden de magnitud requerido por las autoridades de telecomunicaciones postales de aquellos días, para líneas de transmisión (ver **Figura 5**). El dispositivo fue construido de modo modular, algo que era pionero en ese tiempo, con una placa madre en la que se conectaban todos los módulos. Todo se conectaba usando contactos de oro con bloqueo (ver **Figura 6**). Por cierto, si el fusible blanco de coche, en el centro superior, molesta al ojo entrenado del ingeniero electrónico, es que, realmente, no es un elemento original. Está ahí porque el Sr. Kudelski, a pesar de todo su ingenio, olvidó una cosa: el fusible...





El equipo puso una atención especial en las indicaciones del instrumento. Aquí el equipo tuvo que tener en cuenta las normas cambiantes o las convenciones para los indicadores. El 'Modulometer' cumplía estos requisitos perfectamente, en el que los criterios para acelerar la aguja del puntero y controlar la inercia de los indicadores estaban combinados de forma brillante (ver **Figura 7**). Hablando de forma metrológica, las tres velocidades de grabación eran ajustables de modo individual según la respuesta en frecuencia, y eran ecualizadas por separado para una corrección de distorsión estándar y de bajo ruido. Incluso se podían optimizar la simetría de HF y los factores de distorsión H3 y H4. Según el dispositivo, hemos conseguido rangos de frecuencia desde los 20 Hz en parte, hasta 18 kHz -2 dB. Con una relación señal/ruido no ponderada de -60 dB. Estos valores podían rivalizar con cualquier equipo de estudio.

El grabador proporcionaba dos entradas de micrófono y entradas adicionales modificables por el usuario (ver **Figura 8**). Las entradas de micrófono fueron implementadas, bien con transformadores de entrada o bien como amplificadores de corriente balanceados electrónicamente. En los primeros modelos teníamos que conectar en el preamplificador específico del tipo de micrófono (impedancia alta y baja dinámica, micrófono de condensador, etc.) (ver **Figura 9**). Más tarde, el preamplificador se hizo conmutable. Las entradas eran ajustables a un grado muy alto, lo que significaba que la atenuación previa interna conmutable apenas necesitaba ser usada. Junto con un limitador conectable y un limitador de nivel de pico que tenía una función 'rodilla', apenas era posible sobrecargar la grabación. El usuario también disponía de la denominada 'saturación de cinta magnética', que era capaz de 'aguantar' los picos adicionales.

Apareció un problema con las alimentaciones fantasma de 48 voltios, bastante antiguas pero usadas universalmente. En aquellos tiempos todavía era técnicamente imposible generar esta tensión de funcionamiento de 48 V a partir de los, aproximadamente, 18 V disponibles. Naturalmente Nagra conocía los convertidores DC-DC, pero las formas de onda cuadradas o triangulares empleadas se expandían mucho más en el espectro que la señal deseada. (En cualquier caso el convertidor estaba localizado, por motivos de modularidad, en el punto más sensible en la cadena: la entrada micrófono.) Aquí la firma Sennheiser aportó su ayuda, trayendo al mercado los entonces revolucionarios micrófonos de condensador de HF para funcionamiento con baja tensión. Visto frente a las alimentaciones fantasma de 12 voltios, la tensión 'T-Power' de 10 V pasaba a lo largo de los conductores de audio (T

significa Tonaderspeisung [1]; Polonia está justo al lado de Alemania) y aparecía como la Luz del Mundo (probablemente la razón de la tensión de referencia interna de 10 V). Por motivos que no se pueden comprender, la versión fantasma de 12 V desapareció sin hacer ruido, mientras que el modelo 'T-Power' se mantuvo durante mucho tiempo. Lamentablemente esto no fue cosa de un capricho.

En este sistema, como su nombre alemán implica, la tensión de funcionamiento se lleva, de forma simétrica a lo largo de los dos hilos conductores de audio, algo que, a su vez, no causa interferencias en la señal buscada. A los denominados amplificadores de tensión regulada no les gusta para nada tener una tensión DC en su entrada. Las entradas de corriente especiales del Nagra eran bastante indiferentes a esto, pero las entradas de tensión tenían que proporcionar este bloqueo de tensión DC con dos condensadores. ¡Ya está!: ¡con la combinación de la resistencia de entrada y el condensador hemos creado un filtro! Por desgracia, en términos de rango de frecuencia, definitivamente esto sólo es una solución de compromiso.

Si por casualidad conectamos un micrófono dinámico a una entrada 'T-Power', todas las notas bajas desaparecen, porque la tensión DC magnetiza la bobina móvil... Pero, de alguna manera, podemos tolerar estos pequeños problemas. Volvamos de nuevo al equipo de grabación. En el lado superior vemos otro cabezal de audio entre los cabezales de reproducción y de grabación. Este era el elemento más vital para cualquier propósito de sincronización: el cabezal de tono piloto (ver **Figura 10**).

Para ampliar esto: a fin de excluir cualquier variación de sincronización en funcionamiento en estudio, por ejemplo, por desplazamiento, el material de la pista de sonido sep-mag (perfo) tenía que tener exactamente el mismo formato y perforaciones que la cinta almacenada [2] (ver **Figura 11**). La cinta y las máquinas seguidoras de sonido están controladas por los denominados 'motores síncronos', cuya velocidad rotacional no está gobernada por una tensión sino por una frecuencia. Por consiguiente, la velocidad rotacional de estos motores es idéntica y sincronizada, y está enganchada a la frecuencia de la tensión de red AC.

Las cámaras de cinta (alimentadas generalmente por baterías) contenían un pequeño generador de frecuencia que, para 24 cuadros por segundo

(más tarde 25 fps), entregaba una velocidad nominal de, exactamente, 50 Hz. Si la velocidad de cámara variaba (resultado de muchos y variados parámetros mecánicos), la frecuencia cambiaba de forma análoga. Este rango de frecuencia era grabado en medio de la pista de audio usando el cabezal de tono piloto mencionado anteriormente.

Para evitar cualquier zumbido de audio audible la ranura de este cabezal de audio se había rotado 90 grados, lo que eliminaba el tono en la señal requerida prácticamente en el 100%.

Cuando transferimos una grabación de una cinta a una máquina seguidora de sonido controlada por tensión de red AC, esta frecuencia era comparada con la frecuencia de línea y usada para reajustar electrónicamente, bien la máquina de reproducción (en nuestro caso, Nagra) o bien la máquina de grabación. Para este fin Nagra proporcionaba la capacidad de hacer correcciones de hasta $\pm 4\%$, usando el denominado Sincronizador. Sin embargo, para valores mayores debían usarse los accesorios adecuados, algo que sólo pasaba en casos extremos.

Una función adicional relacionada con esto era el llamado 'Procedimiento de Reproducción', que era empleado para números vocales, por ejemplo. Antes de comenzar, la frecuencia de línea AC era grabada sobre la cinta como un tono piloto. Así, ahora Nagra recibía un tono piloto de la cámara, comparaba este tono con el otro tono proveniente de la cinta y dejaba al Sincronizador que corrigiese la diferencia. Este sistema demostró su eficiencia durante décadas.

El interfaz, en la forma de varios conectores de enchufe, era realmente un 'sistema abierto' y la variedad de dispositivos accesorios era infinita, desde grandes carretes de cinta de 3.000 pies (unos 1000 m), hasta mesas de mezclas y dispositivos de control de cristal, y aplicaciones sin hilos. El precio de compra inicial era bastante elevado para la época en cuestión, la versión totalmente equipada (sin caja de transporte y fuente de alimentación de red) costaba el equivalente actual de 10.000 \$.

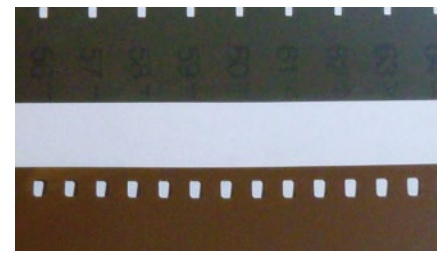
Hay tres accesorios que no deberían quedarse sin mención: el contador de longitud de cinta incorporado al rodillo de auto-alineamiento, el rodillo estroboscópico para el control de velocidad y un limpiador de cinta mecánico en la forma de una cuña metálica (ver Figuras 12 y 13).

Para una máquina de este calibre, naturalmente, había los correspondientes auriculares. En esa

época sólo la firma Beyer Dynamic podía proporcionar el modelo más lineal, el llamado 'Normal-Telefon' (micrófonos estándar de referencia) de 2×25 ohmios. ¡Con sus modificaciones, estos auriculares permanecieron en el uso durante más de 50 años!

Este grabador era un dispositivo mono (ver **el apartado correspondiente**). Por supuesto, más tarde le siguió la versión estéreo y, en los primeros días de la era del vídeo, apareció una versión con código de tiempo. ¡Al final de años 1980 la tecnología digital todavía estaba en su infancia, pero ya en 1992 apareció una versión digital que grababa cuatro pistas PCM en una cinta estándar y con unos increíbles 24 bits!

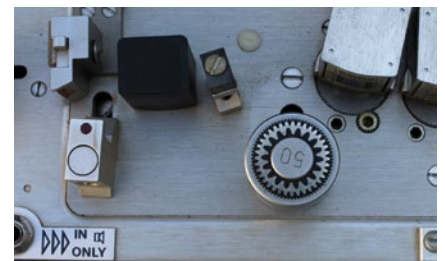
Naturalmente el desarrollo sigue adelante; ¡hoy, por ejemplo, hay un Nagra con memoria flash y todos los parámetros necesarios para la grabación del sonido digital moderno!



11



12



13

(120570)

Enlaces en Internet

[1] http://www.ips.org.uk/faq/index.php?title=Tonader_Power

[2] http://en.wikipedia.org/wiki/Sound_follower

¿Por qué mono?

Por si cualquiera se está preguntándose por qué podríamos desear usar un dispositivo mono, he aquí la explicación. Si deseamos grabar conversación auténtica con un micrófono estéreo o dos micros separados espacialmente, la fuente de sonido tiene que ocupar una posición definida en la sala de audio virtual. Si ahora, como es habitual en las películas, cambiamos el ángulo de captura o la distancia dentro de la escena, esto crea una nueva posición efectiva en la sala. El oído humano registra los cambios espaciales con gran exactitud, y este es el motivo por el que si se hace una edición después de este movimiento, se podría causar la irritación de oyentes/espectadores. Una atenuación o un fundido podrían no distinguir este cambio. Así pues, en la práctica, la conversación en mono no gana su posición final en la sala hasta la etapa de mezcla de sonido. El producto final es perfeccionado mezclando la conversación junto con los efectos de sonido 'multicanal' grabados de forma espacial.

Hexadoku

Un rompecabezas con un toque electrónico

En el extraño caso en el que no estés fuera de casa comprando regalos navideños o limpiando la nieve (o buscando la sombra en el hemisferio sur), por qué no relajarte un rato con este nuevo rompecabezas Hexadoku. Introduce los números o las letras de la A al F correctas en las casillas libres, encuentra la solución en las casillas grises, envíalos y entrarás automáticamente en el sorteo de uno de los cuatro bonos para la tienda de Elektor. ¡Diviértete!

El Hexadoku utiliza los números hexadecimales de 0 a F. Rellena el diagrama de 16 × 16 casillas de modo que todos los números hexadecimales de 0 a F (es decir, 0...9, A...F) aparezcan una sola vez en cada fila, en cada columna y en el recuadro de 4×4 (marcados con líneas en negrita). Ya hay

algunos números rellenos en el crucigrama, que determinan su situación inicial. Entre todas las soluciones correctas recibidas, sorteamos cada mes un gran premio y tres premios de consolación. Para ello, debes enviarnos los números de las casillas en gris.

¡Resuelve el Hexadoku y gana!

Las soluciones correctas recibidas de los lectores de Elektor participan automáticamente en el sorteo de cuatro bonos para la tienda de Elektor, uno de **100 €** y tres de **50 €**. ¡Esperamos que estos premios animen a todos nuestros lectores a participar!

¡Participa!

Por favor, envíanos tu solución (los números de las casillas grises) por correo electrónico a **hexadoku@elektor.es** – Asunto: hexadoku 12-2012 (por favor, cópialo exactamente). Incluye en el correo tu nombre completo y tu dirección. Envíalo antes del **1 de enero de 2013**.

Ganadores del sorteo

La solución del Hexadoku de Octubre de 2012 es: **75E2B**.

El **Bono de 100 € para la tienda de Elektor** es para: Steve Hasko (UK).

Y los **3 Bonos de 50 €** son para: Michael Franke (Alemania), Jean-Paul Lagaisse (Belgica) y Franz Maurer (Países Bajos).

¡Enhorabuena a todos!

		1		A		8	E		6				C		
2		8			1	F	6		3	C			E		
	C	F		3		5		B	A	D		7		0	2
	A		E		B	C		2							9
B			2	8			3	6	7						
F	1		A	D		7		3		8		0	6		C
3	8	5													
		C		F	5								D		4
A		0							E	2		B			
													7	E	1
5		E	4		6		B		D		A	C		9	F
						2	1	7			C	4			8
C						7		E	2		B		A		
D	4		F		8	B	5		9		6		C	1	
		2			9	A		C	1	7			5		D
	5				F		C	D			0		9		

B	9	E	2	8	5	F	6	C	A	0	3	7	D	1	4
5	7	6	F	1	9	2	E	B	4	D	8	3	A	C	0
A	C	3	D	0	7	B	4	F	1	9	6	E	5	2	8
0	1	8	4	C	D	3	A	7	5	E	2	B	6	9	F
4	E	9	6	2	B	5	8	D	F	3	0	C	1	7	A
C	F	5	A	6	3	7	D	8	9	1	B	4	E	0	2
7	B	D	0	E	A	4	1	2	6	C	5	F	3	8	9
8	2	1	3	9	F	C	0	4	E	7	A	D	B	5	6
D	3	7	9	F	1	8	2	5	B	A	C	0	4	6	E
6	0	4	E	D	C	9	B	3	2	8	F	1	7	A	5
F	5	B	C	A	E	6	7	9	0	4	1	2	8	D	3
1	A	2	8	3	4	0	5	E	7	6	D	9	C	F	B
E	8	F	7	B	6	1	9	A	D	2	4	5	0	3	C
9	D	C	B	4	2	A	3	0	8	5	7	6	F	E	1
2	4	A	1	5	0	D	C	6	3	F	E	8	9	B	7
3	6	0	5	7	8	E	F	1	C	B	9	A	2	4	D

Los empleados de Elektor International Media, sus empresas subsidiarias y/o editoriales asociadas no podrán participar en este concurso.

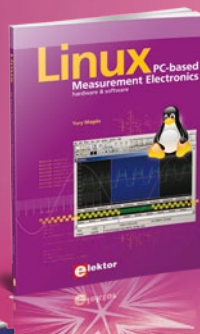
Teníamos más para ti, pero no nos quedaban páginas

Si eres miembro **GREEN**,
tu ejemplar digital incluye
también este artículo



OFERTA ESPECIAL DE NAVIDAD

3 PRODUCTOS ELEKTOR POR TAN SOLO 49,90 €



20
Productos
para
elegir

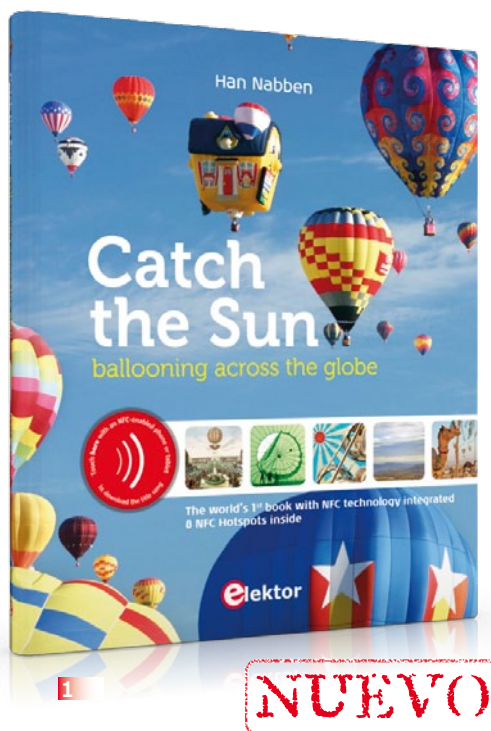


Descuentos de hasta el 62%

Limitada a las existencias de cada product

Oferta válida hasta el 8 de enero de 2013

HAZ TU ELECCIÓN EN WWW.ELEKTOR.ES/NAVIDAD



El primer libro del mundo con tecnología NFC integrada
1 Catch the Sun

La técnica de comunicación sin contacto más antigua conocida data de hace 2000 años en China, con la dinastía Han. Aquella época vio nacer a la linterna Kongming: un pequeño globo de aire caliente utilizado principalmente para transmitir señales militares. Hoy en día, los globos Kongming han sido sustituidos por chips. El "Near Field Communication", o NFC, ofrece conectividad inalámbrica a corta distancia, basándose en la tecnología de los semiconductores. Este libro enlaza ambas técnicas. "Catch the Sun" es el primer libro del mundo con tecnología NFC integrada, y curiosamente, el contenido de este libro tecnológicamente puntero trata sobre algo en cierta medida primitivo: la magia de los globos aerostáticos. El libro incluye numerosos chips NFC que permiten conectarlo a Internet, basta con tocar en los tags con nuestro smartphone o tablet compatibles con NFC.

128 páginas • ISBN 978-9-07545-861-9 • 45,00 €

Más de 75.000 componentes

2 CD Elektor's Components Database 7

Este CD-ROM te permite acceder fácilmente a los datos sobre más de 75.000 componentes. Consta de



ocho bases de datos que comprenden circuitos integrados, transistores, diodos y optacopla dores. Incluye once aplicaciones de cálculo de, por ejemplo, resistencia serie de diodos zener, reguladores de tensión y AMV's. ECD facilita el acceso a los datos unos de 11.100 circuitos integrados, 37.000 transistores, FET, tiristores y triacs, 25.100 diodos y 2.000 op to acopladores. Todas las aplicaciones son totalmente interactivas, permitiendo al usuario añadir, editar y completar los datos de los componentes.

ISBN 978-90-5381-298-3 • 29,50 €

LabWorX 2

3 Mastering Surface Mount Technology

LabWorX es una colección de libros (en inglés) cada uno de los cuales trata un tema particular de la electrónica. Este segundo volumen, Mastering Surface Mount Technology (Dominando la Tecnología de Montaje Superficial), te introduce en un curso intensivo de técnicas, consejos y conocimientos para introducir la Tecnología de Montaje Superficial en tu flujo de trabajo. Incluso si tienes presupuesto también puedes poner en marcha tus diseños con avanzados componentes de pequeño tamaño. Además de explicar la metodología y equipamiento, se presta atención a la tecnología de los componentes y a las técnicas de soldadura. Varios proyectos te introducen paso a paso en el manejo de com-



iPide también el Kit de Componentes asociado!
www.elektor.es/labworx

ponentes de montaje superficial y las técnicas necesarias para realizar montajes SMT satisfactoriamente. Se revelan muchos consejos y trucos prácticos para poner la tecnología de montaje superficial al alcance de todos sin arruinarse.

282 páginas • ISBN 978-1-907920-12-7 • 34,50 €

Nunca más problemas con interferencias o bucles de masa

4 Aislador de USB

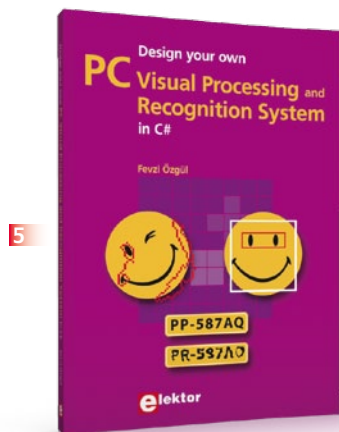
Este aislador de USB es una solución perfecta para un dispositivo USB con problemas de ruido causado por los bucles de masa o para proteger al PC contra tensiones externas. El circuito aquí descrito garantiza una óptima separación galvánica tanto de las líneas de datos como de las líneas de alimentación entre el PC y el dispositivo USB conectado.

Placa montada y comprobada
Art.# 120291-91 • 69,95 €

Tecnología de procesamiento de imagen sobre

Design your own PC
5 Visual Processing and Recognition System in C#

Este libro (en inglés) está dirigido a Ingenieros, Científicos y aficionados con habilidades de programación desarrolladas o con mucho interés en la



5



6

tecnología de procesamiento de imagen sobre PC. Este libro es una exhaustiva guía práctica escrita utilizando Microsoft C# y programación orientada a objetos. El código utilizado en los ejemplos está disponible – sin cargo alguno – en la página web de soporte de Elektor; puedes crear y desarrollar fácilmente tus propios ejemplos para probar los conceptos y técnicas explicadas.

307 páginas • ISBN 978-1-907920-09-7 • 39,95 €

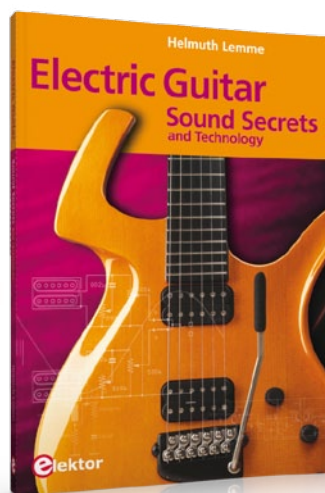
Placa Linux de Elektor

6 Simplificando Linux Embebido

A día de hoy se puede encontrar Linux corriendo en todo tipo de dispositivos, incluso en máquinas de café. Muchos entusiastas de la electrónica estarán ansiosos de usar Linux como base de un proyecto para micro-controlador, pero la aparente complejidad del sistema operativo y el alto precio de las placas de desarrollo han sido hasta ahora un obstáculo. Elektor resuelve ambos problemas con un curso para principiantes acompañado por una tarjeta compacta y barata. Esta tarjeta incluye todo lo necesario en un proyecto embebido actual: un interfaz USB, una conexión para tarjetas SD y varias opciones más de expansión!

Placa montada y comprobada

Art.# 120026-91 • 64,95 €



7

NUEVO

Los Secretos del Sonido y Tecnología

7 Electric Guitar

Este libro (en inglés) muestra, de una manera sencilla y bien fundamentada, lo que, hasta ahora, se han considerado como secretos del fabricante. El examen explora las profundidades de la guitarra, incluyendo las pastillas y el entorno eléctrico, de forma que la electrónica de la guitarra ya no es considerada como un alto secreto. Con unas pocas intervenciones hábiles, muchos instrumentos pueden hacerse más versátiles y que suenen mucho mejor – de forma asequible en la mayoría de los casos. El autor, Helmuth Lemme, es un profesional experimentado de la electrónica y músico activo. Realmente ha probado en la práctica todo lo que se describe aquí.

287 páginas • ISBN 978-1-907920-13-4 • 34,50 €

Detector de "Niebla Electrónica"

8 ¡Descubrimos a TAPIR!

¡Atención exploradores, profesionales y abuelos! Este sabueso de "niebla electrónica" nos ofrece dos sentidos adicionales para seguir el ruido que normalmente es inaudible. TAPIR es también un proyecto agradable para construir: el kit contiene todo lo que necesitamos, incluso la caja, constituida ingeniosamente con la propia PCB. El uso de TAPIR es muy fácil. Conecta unos auriculares y una antena y enciéndelo. Muévete



8



9

alrededor de algún equipo eléctrico y oírás distintos tipos de ruido con cada equipo, según el tipo y la frecuencia del campo emitido.

Kit de componentes

Art.# 120354-71 • 14,95 €

Contador de alfa, beta y gamma

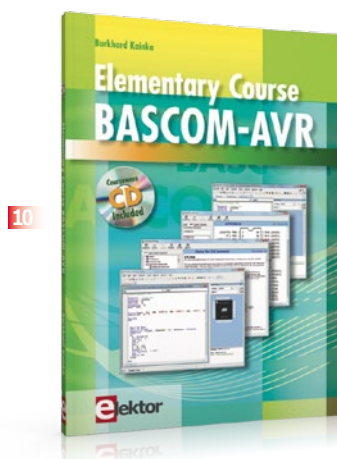
9 Medidor de radiación mejorado

El dispositivo puede utilizarse con distintos sensores para medir la radiación gamma y alfa. Es apto para medidas durante largo tiempo y el análisis de probetas de baja actividad. En comparación con un contador Geiger, un fotodiodo ofrece valores de cero más bajos, y la radiación de pequeñas muestras se aprecia con mayor facilidad. Podemos testear probetas de forma más precisa que con un contador Geiger. Mediante un software de PC opcional puede obtenerse el espectro energético y sacar conclusiones del objeto a examen.

Kit de componentes incluyendo display y controlador programado

Art.# 110538-71 • 39,95 €

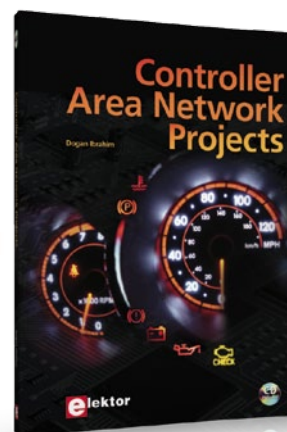
Más Información en la página web de Elektor: **www.elektor.es/store**



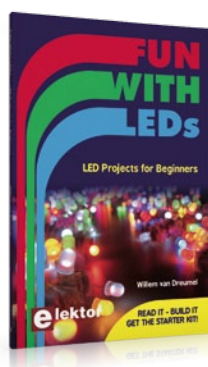
10



12



14



11



13

Curso Elemental

10 Elementary Course BASCOM-AVR

Los microcontroladores AVR son populares, fáciles de usar y extremadamente versátiles. En la revista Elektor ya hemos publicado muchas aplicaciones interesantes utilizando microcontroladores ATmega o ATTiny. La mayoría de estos proyectos desempeñan una función particular. En este libro (en inglés) nos centramos más en los aspectos relacionados con el software. Con muchos ejemplos prácticos mostramos cómo, utilizando BASCOM, puedes poner en marcha rápidamente tus ideas de diseño y ejecutarlas en silicio.

224 páginas • ISBN 978-1-907920-11-0 • 39,95 €

Proyectos con LED para Principiantes

11 Fun with LEDs

Este librito (en inglés) presenta más de veinte interesantes proyectos dirigidos a jóvenes y viejos. Desde un Escritor en el Aire, una Luz de Fiesta, Luces en Movimiento, un atenuador de LED hasta un Árbol de Navidad. Utiliza este libro para reproducir diversos proyectos y ponerlos después en la práctica. Para darte una ventaja inicial, cada proyecto está respaldado por una breve explicación, esquemas y fotos. Además, la página de soporte gratuito de la web de Elektor tiene

disponibles algunos enlaces a videos elaborados con los proyectos. Un par de proyectos utilizan la popular placa microcontroladora Arduino, agraciada con una galaxia de aplicaciones de código abierto. El Kit de Iniciación (60 componentes incluyendo una breadboard) opcional disponible con este libro es una buena manera de montar y comprobar los circuitos en una breadboard, es decir, sin necesidad de soldaduras.

96 páginas • ISBN 978-1-907920-05-9 • 19,95 €

Conoce a BOB

12 Convertidor USB/Serie BOB-FT232R

Este convertidor USB / Serie te sorprenderá primero por su tamaño: ¡no más grande que un conector moldeado de un cable USB! Nos impresionará también porque es práctico, rápido de poner en funcionamiento, reutilizable, multiplataforma y no por ello demasiado caro. Los diferentes módulos basados en el FT232R, disponibles en el mercado, no me han satisfecho. Demasiado caros, demasiado voluminosos, mal hechos... Por eso acepté el desafío de diseñar esta miniatura bajo la forma de una "placa de evasión" ("breakout board").

Placa montada y comprobada

Art.# 110553-91 • 15,00 €

El puente entre Android y tus proyectos electrónicos

13 AndroPod

Pantalla táctil de alta resolución, gran potencia de cálculo, características wireless y telefónicas: los smartphones y tablets con Android están predestinados a servir como centrales de control de nuestros propios proyectos. Hasta ahora ha sido un poco difícil conectar las tabletas y Smartphones basados en Android con circuitería externa que nos permita a los electrónicos acceder a las señales con propósitos de control. La tarjeta interfaz AndroPod de Elektor, que añade un puerto serie TTL y un puerto RS485, cambia esta situación.

Módulo AndroPod con extensión RS485

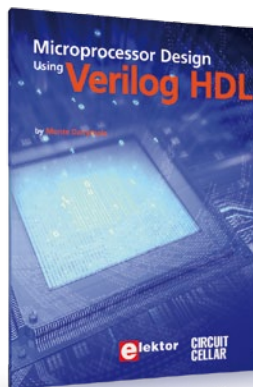
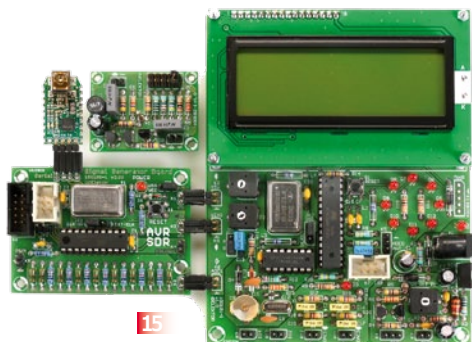
Art.# 110405-91 • 59,95 €

Incluye gratuitamente el compilador mikroC en CD-ROM

14 Controller Area Network Projects

El objetivo de este libro (en inglés) es enseñarte los principios básicos de las redes CAN además del desarrollo sistemas basados en microcontrolador utilizando el bus CAN. Aprenderás cómo diseñar nodos de bus CAN basados en microcontroladores, cómo construir un bus CAN, cómo desarrollar programas de alto nivel

Libros, CD-ROMs, DVDs, Kits y Módulos



y a intercambiar datos en tiempo real sobre el bus. Aprenderás también cómo construir hardware con microcontroladores y a conectarlo con LEDs, LCDs y convertidores A/D.

260 páginas • ISBN 978-1-907920-04-2 • 34,50 €

Oferta de conjunto: Ahora 12%

15 Radio Definida por Software con AVR

Este paquete consta de las tres placas asociadas con la serie de artículos de las revistas Elektor Radio Definida por Software con AVR. La primera placa, que incluye un ATTINY2313, un oscilador de 20 MHz y un CDA R2R, será utilizada para hacer un generador de señal. La segunda placa sacará señales de la nada. Contiene todo el hardware necesario para hacer una radio digital definida por software (SDR), con un interfaz RS-232, una pantalla LCD y un VXCO de 20 MHz (oscilador de cristal controlado por tensión), que se puede enganchar a una señal de referencia. La tercera placa proporciona una antena activa de ferrita.

Kit de componentes Generador de Señal

+ Receptor Universal

+ Antena Activa: PCBs y todos los componentes

+ Convertidor USB/Serie BOB-FT232R

Art.# 100182-72 • 119,95 €

120 revistas de Elektor en español

16 DVD Elektor desde 1998 a 2007

Este DVD-ROM contiene la colección completa de la revista de electrónica Elektor editada en España entre los años 1998-2007. Las ediciones de Elektor, que ascienden a un total de 120 revistas publicadas durante esos 10 años, están en formato PDF y ordenadas cronológicamente por fecha de publicación (año/mes). El DVD contiene más de 2100 artículos imprimibles con alta calidad. Puedes abrir un documento PDF seleccionando el año de publicación y luego la edición deseada. Entre otras, las siguientes categorías están bien representadas en el DVD: sistemas con microcontroladores, técnicas de audio y video, técnicas de alta frecuencia y software de diseño de circuitos.

ISBN 978-90-5381-241-9 • 59,00 €

Una guía práctica del diseño de procesadores

17 Microprocessor Design using Verilog HDL

Este libro (en inglés) es una guía práctica del diseño de procesadores en el mundo real. Presenta el Verilog HDL de manera fácilmente digerible y sirve como introducción detallada sobre cómo reducir una arquitectura de ordenador y el set de instrucciones en la práctica. Será guiado a lo largo del proceso de diseño

de un microprocesador de principio a fin, dejando al descubierto los temas esenciales que van de la escritura en Verilog a la depuración y las pruebas.

337 páginas • ISBN 978-0-9630133-5-4 • 34,50 €

165 min. de vídeo y más

18 DVD Masterclass Modern Valve Electronics

Este doble DVD publicado por Elektor trata las nuevas tecnologías para amplificadores de válvulas mejorados y está dirigido a cualquiera que busque información sobre el funcionamiento de estos amplificadores en profundidad. Esta clase magistral la presenta Menno van der Veen, MSc, un reconocido especialista a nivel mundial en el diseño de amplificadores de válvulas.

ISBN 978-1-907920-10-3 • 29,95 €

Más Información en la página web de Elektor: **www.elektor.es/store**

Elektor International Media Spain, S.L.

Apartado de Correos 62011

28042 Madrid – España

Tel.: +34 91 101 93 95

Fax: +34 91 101 93 96

Email: service@elektor.es



El próximo mes en Elektor

Por primera vez en la historia de Elektor, la revista de enero de 2013 será doble.
De gran tamaño, como en verano, con mucho más contenido.

Mercado elektor El mundo de la Electrónica

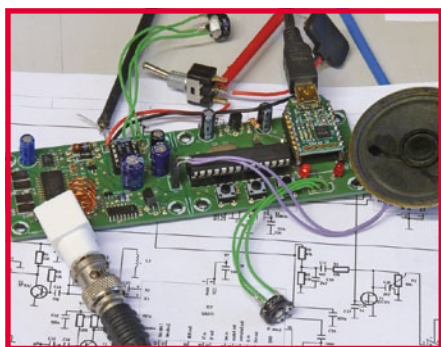
Contacto

Teléfono: +34 91 101 93 95

Fax: +34 91 101 93 96

E mail: info@elektor.es

Elektor tiene una sección para ayudar a sus clientes a promocionar sus negocios, Mercado – una sección permanente de la revista donde puede mostrar sus productos y servicios.



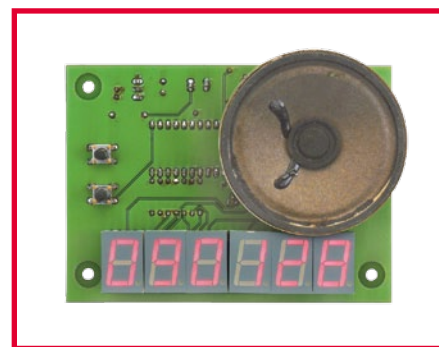
Escáner de VHF para la banda de aviación

Muchos aficionados a los aviones tienen gran interés en seguir las comunicaciones de voz entre los aviones y la torre de control. Con este receptor, podrás sintonizar todo el tráfico que haya en la banda de aviación civil, 108-137 MHz. Es fácil de construir, tiene solo un ajuste mecánico y puede operar incluso desde un PC con conexión USB. Se emplea un circuito integrado receptor de VHF-FM de conversión simple trucado para que funcione como receptor AM con un inteligente uso de la salida del detector de nivel de señal.



Mini Multitester

Estás trabajando en un proyecto con microcontrolador y con toda suerte de cosas que se deben medir, como tensiones de salida, niveles lógicos, cuenta de pulsos y cosas así. ¿Qué instrumento usas? Con este manejable multitester tienes un versátil dispositivo medida que combina un amplio número de funciones en un pequeña caja que puede manejar con una mano. El dispositivo actúa como tester lógico, voltímetro, contador de pulsos, receptor y transmisor TTL/UART, frecuencímetro y mucho más.



Cuco Digital

Los relojes electrónicos han tomado todas las formas posibles e imaginables, y sin embargo aparecen constantemente nuevas versiones más o menos originales. Aquí presentaremos uno que a la vez que nos distraerá, es singular y nos enseñará cosas nuevas. Como su propio nombre indica, pertenece a la famosa familia del 'cuculus canorus' pero con microcontrolador y cantará la melodía que se ha aprendido cada cuarto de hora.

Los títulos y los contenidos de la revista están sujetos a modificación; por favor, verifica el contenido en "revista" en www.elektor.es

ÚNETE A LA PLATAFORMA PARA INGENIEROS ELECTRÓNICOS LÍDER EN EL MUNDO

Hazte miembro GRATIS del Elektor Weekly



- Lo último en electrónica y tecnologías de la información
- Trucos, consejos y ofertas interesantes
- Cada viernes en tu bandeja de entrada

Regístrate hoy en www.elektor.es/boletin



Elektor pasa a ser Digital

!Hazte socio GREEN ahora!

Como miembro GREEN obtendrás:

- 8 ediciones estándar digitales de Elektor
- 2 ediciones especiales digitales (enero/febrero y julio/agosto)
- Un mínimo del 10% de descuento en todos los artículos de Elektor.STORE
- Acceso completo a Elektor.LABS
- Acceso completo a Elektor.MAGAZINE
- Recibir Elektor.POST en tu cuenta de email (incluyendo 25 proyectos extra cada año)
- Tu propia tarjeta GREEN CARD de socio ecológico



OFERTA ESPECIAL

Precio normal: 92 €

Ahora un año por solo **50€**

¡y si eres menor de 25 años* pagarás solo 25 €!

* Deberás facilitarnos algún documento que acredite tu edad.



elektor

Hazte socio GREEN ahora en www.elektor.es/miembro

Una Mejora en la Medida para la Enseñanza de Ingeniería

$$v_f = v_i + a \cdot t$$

$$Q = mc\Delta T$$

$$v_f^2 = v_i^2 + 2 \cdot a \cdot d$$

$$v_i \cdot t + \frac{1}{2} \cdot a \cdot t^2$$

$$d = v_i \cdot t + \frac{1}{2} \cdot a \cdot t^2$$

$$V_g = \frac{\frac{R_1}{R_2} - \frac{R_4}{R_3}}{\left(\frac{R_1}{R_2} + 1\right) \left(\frac{R_4}{R_3} + 1\right)}$$

No solo enseñe ingeniería. Haga ingeniería

Las medidas son bloques fundamentales en la enseñanza de la ingeniería, ofreciendo a los estudiantes la primera oportunidad de interactuar con datos adquiridos del mundo real. National Instruments proporciona a los estudiantes el hardware y el software que necesitan para experimentar, ir más allá de la teoría y de la simulación y saber lo que significa hacer ingeniería.



HERRAMIENTAS DOCENTES

NI LabVIEW
NI myDAQ
NI ELVIS

>> Aprenda cómo NI soporta la próxima generación de innovación en ni.com/academic/esa

91 640 0085
93 582 0251



National Instruments Spain S.L. • Europa Empresarial • c/Rozabella, 2 - edificio Berlin • 1ª planta • 28230 Las Rozas (Madrid) • España
Tel: +34 91 640 00 85 ó 93 582 0251 • Fax: +34 91 640 05 33 ó 93 582 4370 • CIF: B-80021462 Inscrita en el Registro
Mercantil de Madrid, Folio, 115, Tomo 1181, Hoja N°22335, Inscrp. 1ª • Sociedad Unipersonal S.L.

©2012 National Instruments. Todos los derechos reservados. LabVIEW, National Instruments, NI, and ni.com son marcas registradas de National Instruments.
Los nombres de los otros productos y las razones sociales mencionados son marcas comerciales o nombres comerciales de sus respectivas compañías. 04567