

DMX512
Regulador de 1 canal

+ Diseño profesional de paneles frontales gratuito

elektor

www.elektor.es

Robot Proton de Elektor

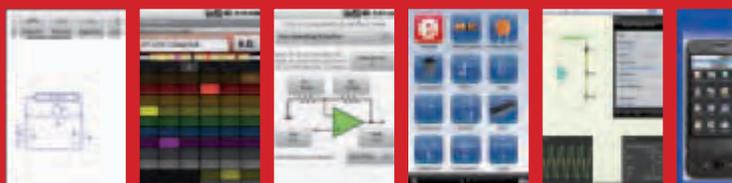
Una plataforma versátil para aprendizaje y experimentación



Curso de Audio DSP

Avance rápido en el procesamiento de señales de audio

¿Cómo son de útiles las aplicaciones de electrónica para tablets y smartphones?



+ Tarjeta gráfica VGA para micros

+ Sistema de micrófono para conferencias



Audio digital

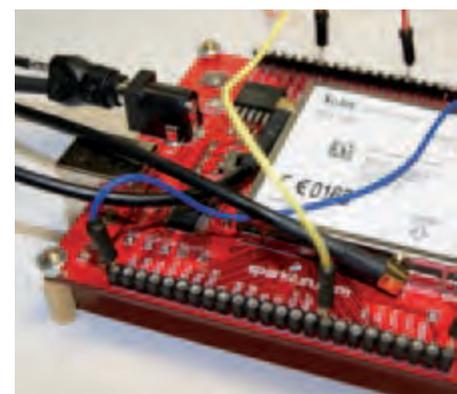
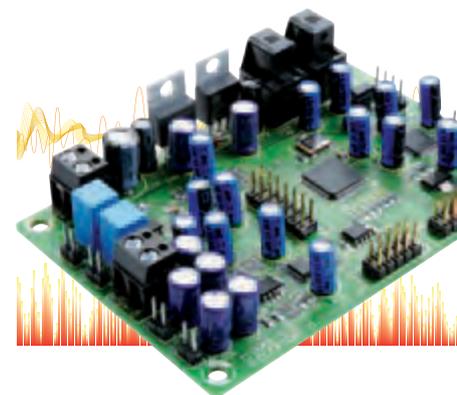
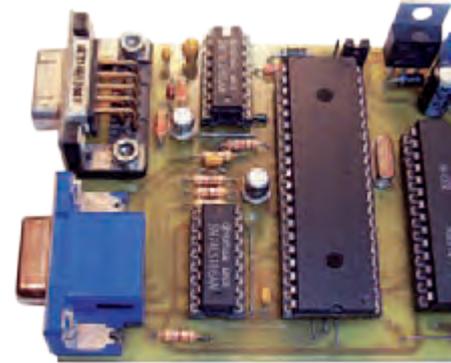
Antes de que el compact disc (CD) apareciera en el mercado, en la edición de sonido solo se utilizaban señales analógicas. En esencia, una pequeña tensión alterna procedente de un micrófono o de la aguja de un giradiscos se envía a través de una serie de etapas de amplificación antes de ser aplicada a una cabeza grabadora, un cortador de disco o un altavoz. En esos días, la edición y grabación de audio era considerablemente más compleja, aunque los consumidores no se dieran cuenta (¡excluidos los lectores de Elektor, claro está!). Hoy en día, incluso el reproductor de MP3 más barato contiene un microprocesador o un DSP que realiza un procesado digital avanzado que permite que los datos almacenados en su memoria lleguen a tus tímpanos en forma analógica.

Por supuesto, los DSP actuales son núcleos muy potentes válidos para toda clase de aplicaciones (de audio), pero al mismo tiempo son bestias altamente complejas para lidiar con ellas. Consecuentemente, no es del todo sencillo diseñar un circuito basado en DSP y el software asociado si no estás familiarizado con esta materia. Esperamos que el curso de DSP que lanzamos en esta edición represente el fin de los miedos y el desconocimiento que rodean a los DSP. Aunque intentamos que prevalezcan los aspectos prácticos, inevitablemente se requiere alguna interacción con la teoría considerando que todo en este curso se desarrolla en torno a circuitos reales que incorporan un DSP. A lo largo del curso discutiremos varios proyectos de ejemplo, cada uno de los cuales puede reproducirse en casa o en el laboratorio utilizando una placa DSP universal desarrollada por el autor en estrecha colaboración con el laboratorio de Elektor. Tras haber digerido el curso, en unos meses deberías estar preparado para comenzar a desarrollar tus propias aplicaciones con DSP. En esta edición iniciamos el curso con un poco de teoría sin la cual realmente no podrías empezar a programar y soldar. Así que haz de tripas corazón y súmate a al curso de DSP, seguro que te resultará entretenido y educativo.

Disfruta de esta edición
Eduardo Corral, Editor

elektor

- 6 Colofón**
Información Corporativa de la revista Elektor.
- 8 Noticias Locales**
Un paseo mensual por lo último en el mundo de la electrónica.
- 12 Convertidor RGB - YPbPr (o YUV)**
Cómo utilizar antiguos receptores o decodificadores de TV satélite con los televisores actuales.
- 16 Tarjeta gráfica VGA**
Muestra la información de un sistema con microcontrolador en una pantalla VGA recuperada del olvido.
- 22 El Robot Proton de Elektor**
Un diseño de robot para construir uno mismo con enorme versatilidad en lo que a sensores, actuadores e incluso CPU se refiere.
- 26 Regulador de Intensidad de Luz DMX512 de 1 canal**
E-Blocks y Flowcode 4 para PIC unidos para construir un potente regulador para iluminación DMX.
- 31 Generador de sonidos marinos**
Simulamos el sonido de las olas del mar para ayudar a relajarte y dormir profundamente.
- 34 ElektorBus 1105 (JN)**
El bus de Elektor se ha establecido fuera de las páginas dedicadas al Laboratorio y toma forma en términos de hardware y software.
- 39 ¡Ayúdame! No encuentro una solución...**
Sobre las consultas técnicas que nos enviáis por correo electrónico.
- 40 Medir la temperatura en un santiamén (2)**
Una introducción a los sistemas de imagen termográfica y a la interpretación de sus resultados.



SUMARIO

Volume 32
Mayo 2011
nº 371



16 Tarjeta gráfica VGA

Cuando los sistemas embebidos tienen que presentar un gran volumen de información en forma de texto o gráfico el tamaño del visualizador suele ser un problema. Lo que proponemos con este diseño es utilizar una pantalla VGA en desuso para resolver este problema. Esta tarjeta VGA es una buena solución para todos los micros que dispongan de un puerto serie.



43 Herramientas de desarrollo de bolsillo

Los programas de ayuda a la ingeniería como el CAD ya no están solo confinados en los PCs: los actuales smartphones y ordenadores tablet ya están llenos de aplicaciones de este tipo. Por eso, decidimos averiguar qué programas hay disponibles y cómo de útiles son.



48 Curso de audio DSP

En este curso, además de introducirte en las propiedades de los procesadores digitales de señal (DSPs) para señales de audio y los aspectos de programación asociados, presentamos varias aplicaciones basadas en un módulo procesador de señal barato pero de altas prestaciones. ¡Hardware y software incluido!



58 GSM, SMS, CALL Identificación

Con este proyecto para el módulo ATM18 de Elektor, todo lo que tienes que hacer para que tu placa favorita realice la acción adecuada es identificarte mediante la identificación de llamada (call id) en tu ATM18 o bien enviarle un mensaje de texto directamente.

42 Problemas con Skype

Cómo adaptar el sistema microfónico para reuniones online a Skype.

42 Problemas de ruido

Sobre ciertos tipos de transistor que no quieren producir suficiente ruido.

43 Herramientas de desarrollo de bolsillo

La última moda son las "apps" y seguro que también las hay para ingenieros electrónicos.

48 Curso de audio DSP (I)

Si el procesado digital de señales te suena a Chino, sigue este curso, que tratará tanto el hardware como el software.

54 Hacer reuniones online más efectivas

Sistema microfónico para que tus conferencias online sean más inteligibles que con el micro de tu portátil.

58 GSM, SMS, CALL Identificación

Este mes el módulo ATM18 se conecta a la red de telefonía móvil con excelentes resultados.

65 Diseño gratuito de cubiertas frontales profesionales

Un vistazo a la nueva herramienta gratuita de Schaeffer AG para diseñar paneles frontales.

68 Hexadoku

Nuestro rompecabezas mensual con un toque de electrónica.

70 Retronica: Osciloscopio de Almacenamiento Tektronix 564 (1963)

Las usuales características de la electrónica "extraña y antigua".

76 Próximo número

Un avance de los contenidos de la próxima edición.

Convertidor RGB - YPbPr (o YUV)

Christian Tavernier (Francia)

Incluso si la llegada de la alta definición está comenzando a ser el final de los diferentes intercambios de señales analógicas en beneficio de la conexión digital por excelencia, la toma HDMI, aún queda un cierto número de situaciones donde se hace obligatorio trabajar con estas señales.



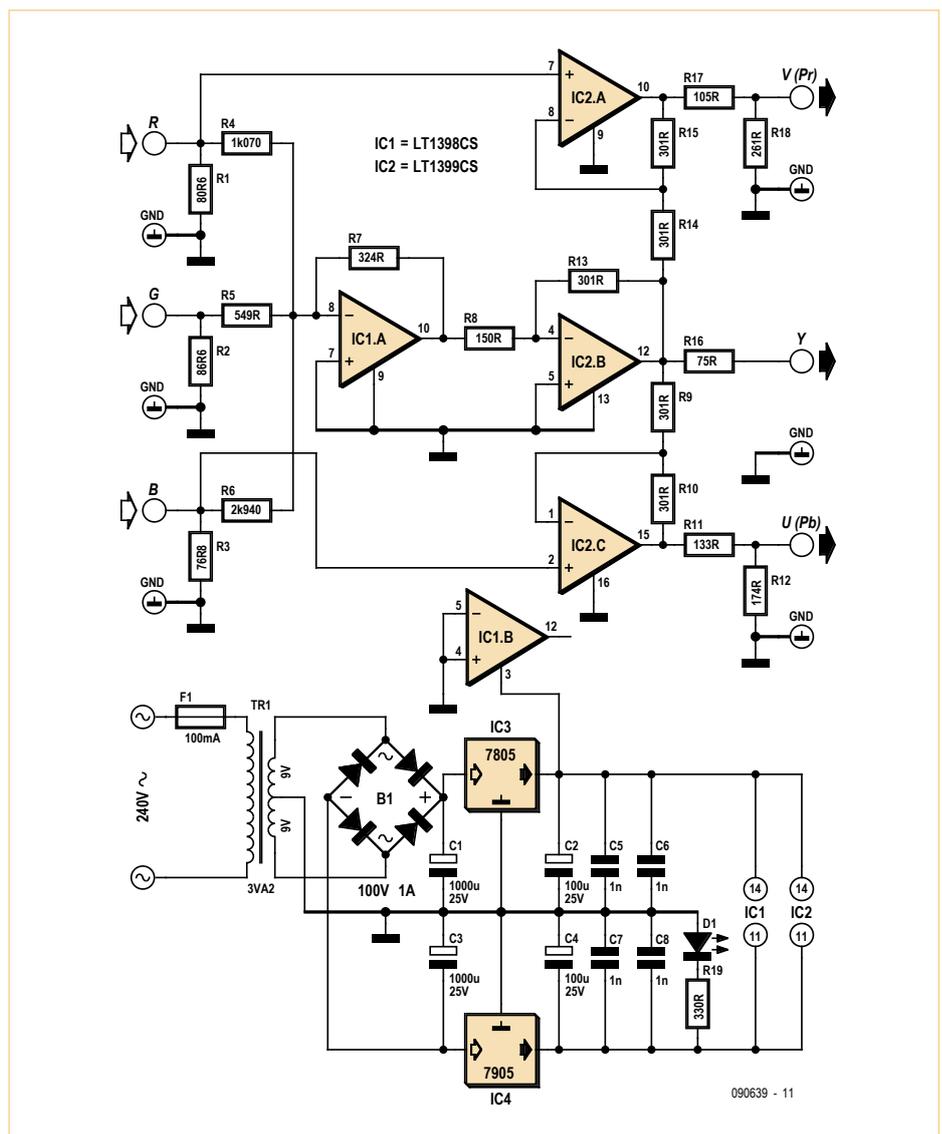
Así, numerosos receptores y/o decodificadores de televisión vía satélite o por ADSL aún no disponen de salidas HDMI sino que, como contrapartida, proponen un buen y viejo euroconector. Por otra parte, la mayoría de los televisores de alta definición de pantalla plana, así como los videoproyectores de calidad, están equipados con conectores de entradas llamadas “componentes” o, mejor aún, es más correcta la denominación YPbPr, bautizada a veces, impropiamente, como YUV.

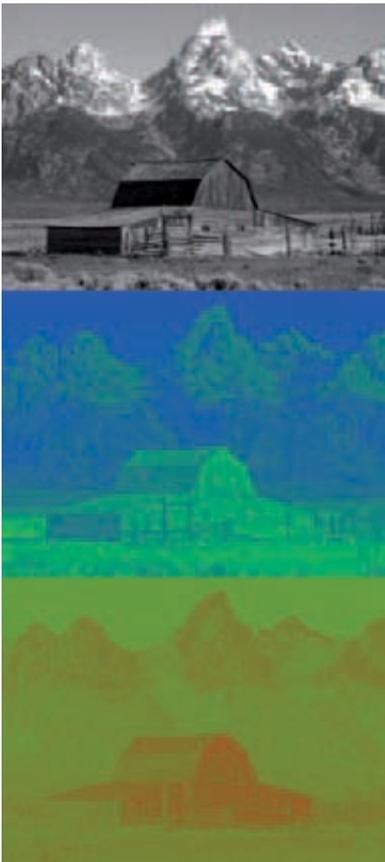
Por desgracia, hasta si el euroconector es, a menudo, capaz de proporcionar las señales vídeo de los tres colores primarios rojo, verde y azul (señales que, sin ser alta definición son, sin embargo, de calidad mucho mejor que las señales S-Vídeo y, por lo tanto, mejores que la señal vídeo compuesto), es imposible aplicarlas directamente sobre las entradas Y, Pb y Pr de los televisores y videoproyectores.

En el comercio podemos encontrar conversores disponibles con este fin, pero su precio los hace bastante disuasivos ya que, generalmente, es superior a 100 € para los modelos de buena calidad. Así pues, os proponemos realizar un convertidor de este tipo que, dando las mismas prestaciones (si no más) que sus homólogos comerciales, es, claramente, menos caro.

Un poco de teoría

Incluso si las tres tomas, Y, Pb y Pr, suelen ser de colores rojo, verde y azul, dichas conexiones no están preparadas para recibir las señales vídeo “en bruto” generadas en las salidas R, G (V) y B (A) del euroconector.





La toma Y espera, en efecto, una señal llamada de luminancia, que es una suma inteligente ponderada de las tres señales de base, mientras que las tomas Pb y Pr transportan las señales denominadas como diferencias de color y que también son combinaciones ponderadas de las tres señales R, G (V) y B (A) básicas.

En posesión de esta información y de los coeficientes de ponderación utilizados, es pues, relativamente fácil concebir tal conversor ya que no es otra cosa que una combinación de sumadores y restadores. Estos montajes son muy fáciles de realizar con amplificadores operacionales pero, si recurrimos a un bueno y viejo TL081 (o a alguna de sus familias), este circuito apenas tendrá la posibilidad de funcionar. En efecto, las señales de vídeo con las que hay que tratar, si son de buena calidad, que suele ser generalmente el caso cuando provienen de fuentes de alta definición, se extienden sobre un margen de frecuencia de varias decenas de MHz y, por lo tanto, necesitan amplificadores operacionales de una banda pasante muy ancha si no queremos degradar su calidad. Afortunadamente, Linear Technology propone en su catálogo a la familia de los LT1398 Y LT1399, que son unos amplificadores operacionales de 300 MHz de banda pasante, ofreciendo además una estabilidad de ga-

¡Ecuaciones como si llovieran del cielo!

Las conversiones entre señales RGB, YPbPr y YUV son sólo una simple cuestión de coeficientes. Pero es necesario elegir la correcta y, más aun, cuando reina una cierta confusión a veces sobre este tema, como podemos comprobar recorriendo los diferentes sitios de Internet que abordan este tema.

Recordemos en primer lugar que las señales YPbPr, incluso si son muy parecidas a las señales YUV, hasta el punto de que las denominaciones son a menudo confundidas, no son totalmente idénticas, ya que no han sido definidas para el mismo fin. Las señales YUV han sido definidas, en efecto, en un principio, con el fin de facilitar la codificación de las emisiones de televisión en color, ya sean en PAL, SECAM o NTSC, mientras que las señales YPbPr han sido definidas en el ámbito de las aplicaciones de televisión profesional analógica.

Así pues, he aquí cuales son las ecuaciones que rigen las relaciones entre estas diversas señales.

YUV hacia RGB y viceversa:

$$\begin{aligned} Y &= 0,299 R + 0,587 G + 0,114 B & R &= Y + 1,14 V \\ U &= -0,147 R - 0,289 G + 0,436 B & G &= Y - 0,395 U - 0,581 V \\ V &= 0,615 R - 0,515 G - 0,1 B & B &= Y + 2,032 U \end{aligned}$$

YPbPr hacia RGB y viceversa:

Contrariamente a las conversiones precedentes, que solo afectan a la televisión de definición estándar en función del origen de las señales YUV, las señales YPbPr pueden transportar informaciones en definición estándar o de alta definición. Existen, pues, dos juegos de ecuaciones diferentes de conversión.

En televisión de definición estándar, los coeficientes son los siguientes:

$$\begin{aligned} Y &= 0,299 R + 0,587 G + 0,114 B & R &= Y + 1,420 Pr \\ Pb &= -0,169 R - 0,331 G + 0,5 B & G &= Y - 0,344 Pb - 0,714 Pr \\ Pr &= 0,5 R - 0,419 G - 0,081 B & B &= Y + 1,772 Pb \end{aligned}$$

En televisión de alta definición, estas ecuaciones pasan a ser:

$$\begin{aligned} Y &= 0,213 R + 0,715 G + 0,072 B & R &= Y + 1,575 Pr \\ Pb &= -0,115 R - 0,385 G + 0,5 B & G &= Y - 0,187 Pb - 0,468 Pr \\ Pr &= 0,5 R - 0,454 G - 0,046 B & B &= Y + 1,856 Pb \end{aligned}$$

En cuanto al conversor descrito en este artículo, teniendo en cuenta el hecho de que la toma de euroconector no transporta señales de alta definición, las ecuaciones que han sido utilizadas son las correspondientes a la definición estándar.

Lista de los componentes

Resistencias (capa metálica del 1 %)

R1 = 80,6 Ω
 R2 = 86,6 Ω
 R3 = 76,8 Ω
 R4 = 1070 Ω
 R5 = 549 Ω
 R6 = 2940 Ω
 R7 = 324 Ω
 R8 = 150 Ω
 R9, R10, R13, R14, R15 = 301 Ω
 R11 = 133 Ω
 R12 = 174 Ω

R16 = 75 Ω
 R17 = 105 Ω
 R18 = 261 Ω
 R19 = 330 Ω

Condensadores

C1, C3 = 1000 $\mu\text{F}/25\text{ V}$, electrolítico radial
 C2, C4 = 100 $\mu\text{F}/25\text{ V}$, electrolítico radial
 C5, C6, C7, C8 = 1 nF cerámico

Semiconductores

IC1 = LT 1398CS

IC2 = LT 1399CS
 IC3 = 7805
 IC4 = 7905
 B1 = puente rectificador 100 V 1 A
 D1 = LED, tamaño y color a elección

Varios

TR1 = Transformador 240 V - 2 x 9 V, 3,2 VA
 F1 = Portafusible para circuito impreso y fusible de 100 mA temporizado

nancia mejor que 0,1 dB de 0 a 150 MHz. Es exactamente lo que nos hace falta para tratar nuestras señales de vídeo. Incluso si estos componentes solo están disponibles en encapsulados SMD, no son demasiado pequeños y permiten si manipulación cómoda y pueden ser soldados por simples aficionados a la electrónica con un sencillo soldador de punta fina.

Esquema eléctrico del conversor

Nuestro esquema, libremente adaptado de varios documentos y notas de aplicación de Linear Technology, utiliza cuatro amplificadores contenidos, por una parte en un LT1398, en IC1, que es un modelo doble, y por otra parte en un LT1399, en IC2, que es un modelo triple. IC1. A realiza la suma ponderada de las señales R, G (V) y B (A) para producir la señal de luminancia o la señal Y. Sabiendo que la ecuación que hay que realizar es la siguiente: $Y = 0,3 R + 0,59 G + 0,11 B$, se obtienen, muy fácilmente, los valores de las resistencias R4 a R7. IC1.A está cableado en un montaje como sumador inversor ponderado, cuya ganancia viene dada respectivamente por las relaciones: R4/R7 para la entrada R, R5/R7 para la entrada G (V) y, por último, R6/R7 para la entrada B (A).

Las resistencias R1 a R3 permiten fijar la impedancia de entrada de las tres entradas del conversor de 75 Ω , algo que es indispensable para una transmisión sin degradación de las señales vídeo.

La señal de luminancia generada por IC1.A está invertida, ya que estamos en presencia de un sumador inversor, por lo que es invertida de nuevo por IC2.B que, además, la amplifica en una razón de 2, teniendo en cuenta los valores respectivos de R13 y R8. La resistencia R16 y la impedancia de entrada de 75 Ω del equipo conectado al conversor se encargan de introducir una atenuación de 2, lo que asegura pues, al final, entre las entradas R, G (V) y B (A) y la salida Y, un respeto perfecto de la ecuación presentada más arriba.

La señal Pr debe respetar la relación $Pr = 0,71 (R - Y)$, para Pb la relación es $Pb = 0,56 (B - Y)$. Como son dos ecuaciones idénticas con un coeficiente parecido, se ha usado el mismo montaje para producir Pb y Pr; en este caso, con un amplificador restador realizado alrededor de IC2.A para Pr en IC2.C para Pb. Teniendo en cuenta los valores de R9 y R10 (y en correspondencia, R14 y R15), estos amplificadores proporcionan a su salida una señal igual a $2 \cdot (B - Y)$, para IC2.C, y $2 \cdot (R - Y)$ para IC2.A. Los coeficientes 0,56 y 0,71 presentes en las ecuaciones precedentes son obtenidos por medio de los puentes divisores realizados por R11 y R12 (y sus respectivas R17 y R18) y a la impedancia de entrada de 75 Ω del equipo conectado al conversor.

Los amplificadores deben ser alimentados con una tensión simétrica de 5 V, para ello se usa un transformador con toma media, seguido de dos reguladores integrados, según un esquema muy clásico.

Realización

Incluso si los valores de las resistencias visibles en el esquema nos pueden llegar a sorprender, ya que no pertenecen a la tradicional serie E12, ni a la E24, están perfectamente normalizadas. Si tu vendedor no las tiene en stock, puedes encontrarlas sin dificultad en RS o Farnell, por ejemplo, al igual que para LT1398CS y LT1399CS. ¡Atención! No compréis en ningún caso estos componentes con el sufijo GN, ya que el encapsulado sería aún más pequeño y no podríamos implantarlos sobre el circuito impreso (¡e incluso ni soldarlas!).

La placa que os proponemos [1] aloja la totalidad de los componentes del esquema eléctrico con el fin de minimizar el cableado necesario. Sólo quedan externos los conectores de las entradas y salidas con el fin de tener una mayor libertad en la elección de la caja en la que será montado el conversor. Como IC1 e IC2 tienen encapsulado SMD, son soldados en la cara de cobre del circuito impreso, lo que no debe presentar ninguna

dificultad en particular si usamos un soldador con una punta suficientemente fina. Los otros componentes se montan de forma clásica, en la cara y sitio que se ha reservado para ellos. Atención al momento de la colocación y a la correcta identificación de las resistencias que, como son de valores precisos, usan más anillos de colores para su código que las resistencias habituales. En caso de duda, no dudéis en recurrir al ohmetro.

Los conectores para las entradas y las salidas serán del tipo RCA, llamados también conectores "Cinch", ya que es la «norma» para todo el conexionado que transporta estas señales en el campo del vídeo para el gran público.

Utilización

El funcionamiento del montaje es, evidentemente, inmediato ya que no conlleva ningún ajuste. En caso de problema, basado en una ausencia total de señales o por señales muy deterioradas en las salidas Y, Pb y Pr y, si las señales de entrada provienen de un euroconector, debemos pensar en verificar que el equipo sobre el que se encuentra este conector está correctamente definido para suministrar las señales R, G (V) y B (A), y no sólo las señales compuestas, como a menudo es el caso por defecto. El euroconector puede transportar, de igual forma, tanto las señales R, G (V) y B (A) como la señal S-vídeo o la señal de vídeo compuesto, pero no lo hay una norma que lo imponga sistemáticamente. Todo depende del aparato sobre el que está instalado.

Debemos tener en cuenta también que algunos equipos, como los modelos de decodificadores Canal Satélite, por ejemplo, sólo proporcionan las señales R, G (V) y B (A) sobre uno de sus dos euroconectores.

(090639)

Enlaces en internet

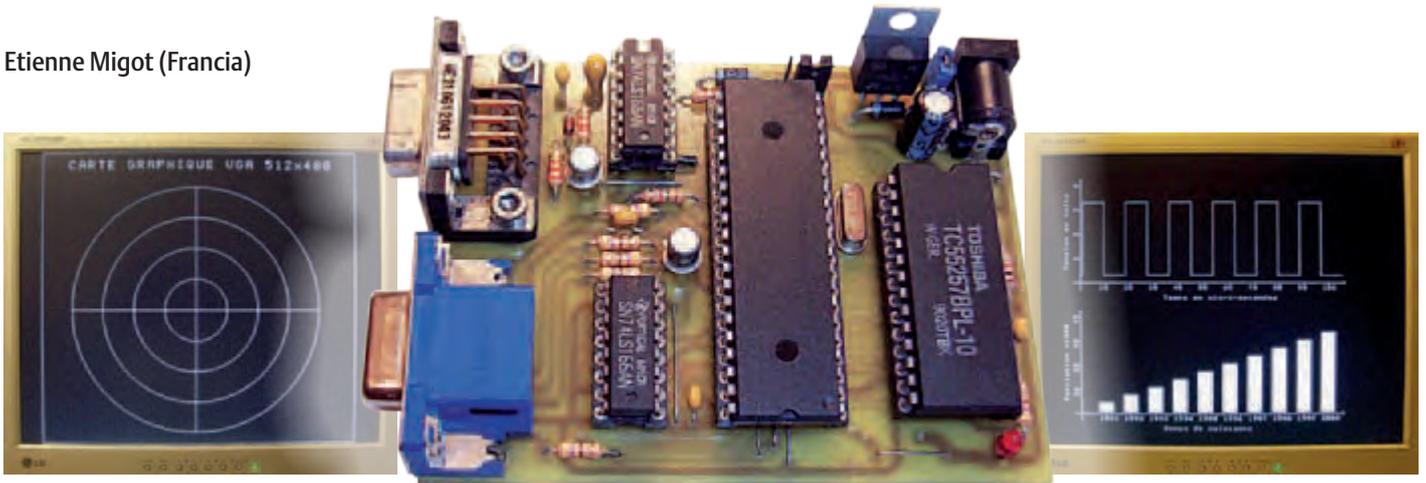
[1] www.elektor.es/090639

[2] www.tavernier-c.com

Tarjeta gráfica VGA

¡No tiren sus viejas pantallas!

Etienne Migot (Francia)



La necesidad de mostrar un gran volumen de información está presente en muchos proyectos, pero el tamaño del visualizador a menudo plantea un problema. Una solución es utilizar una vieja pantalla de ordenador a 14 ó 15 pulgadas, en desuso, pero que aún funciona. La placa VGA descrita en este artículo permite hacer exactamente esto y es compatible con todos los microcontroladores que disponen de un puerto de serie.

En general, el diseñador de un sistema con microcontrolador se limita a usar un visualizador alfanumérico clásico, de hasta cuatro líneas de 40 caracteres, pero donde no es posible ver ningún gráfico y donde el precio es elevado por encima de las dos líneas. Un visualizador gráfico, VGA monocromo o color, es aún más caro y más difícil de integrar en el proyecto. También es posible utilizar un ordenador como un sistema de visualización. Esta última solución se utiliza para disponer de una visualización cómoda, pero adolece de varios inconvenientes, en particular el hecho de monopolizar el ordenador cuando la visualización debe usarse continuamente, o la dificultad de conseguir el programa «que se adapta» (por ejemplo para la visualización de curvas en tiempo real). La idea de este proyecto nació de un hecho simple: la mejora continua de las características técnicas de las pantallas de ordenador viene acompañada del desecho, casi sistemático, de los antiguos modelos. Esto afecta a casi la totalidad de las pantallas de 14 pulgadas y cada vez más a las pantallas

de 15 pulgadas, aunque tengan prestaciones más que reseñables.

¡No tiréis más vuestras viejas pantallas! Os proponemos reciclarlas como visualizadores de alta gama, gracias a la realización de la placa VGA que os proponemos en este artículo. Se conecta con cualquier microcontrolador a través de un enlace serie, ofrece una puesta en marcha simple, a base de comandos de alto nivel, y es fácil para construir, ya que utiliza únicamente componentes de montaje convencional.

Arquitectura global

El convertor RS-232/VGA se establece alrededor de un microcontrolador de Microchip (un dsPIC30F3011) al que se le ha asociado una memoria RAM estándar, de un tamaño de 32 KB, que contiene los píxeles mostrados en la pantalla. El sinóptico de la **Figura 1** muestra los enlaces entre las diferentes funciones:

- Direcciones A0 a A14: direccionamiento en lectura y en escritura de 32 KB de RAM;

- Datos D0 a D7: datos a mostrar sobre 8 bits (1 bit = 1 pixel);
- WE y OE sirven para escribir o para leer los datos en la memoria RAM;
- SCK establece la frecuencia de 25 MHz para los píxeles de salida de la FIFO;
- Hsync y Vsync son las señales de sincronización de Línea y de Cuadro;
- RX-RS232 es el terminal de enlace por el que llegan los caracteres que hay que mostrar y los comandos en modo texto y gráfico.

Los valores del cristal de cuarzo y del multiplicador del PLL interno del dsPIC son tales que éste último trabaja a una frecuencia interna de 100 MHz, es decir, un tiempo de ciclo de 40 ns (cuatro períodos por ciclo). La señal vídeo que sale de la FIFO, así como las señales de sincronización, son llevadas directamente al conector VGA de salida, la adaptación de nivel se realiza por simples resistencias que hacen la función de puente divisor con las resistencias de entrada de la pantalla.

Nota. Los Proyectos de Lectores mostrados están basados en la información suministrada únicamente por el(los) autor(es). El uso de esquemas eléctricos con el estilo de Elektor y otras ilustraciones en este artículo, no implica que el proyecto haya pasado por los Laboratorios de Elektor y se haya hecho una réplica para verificar su funcionamiento.

La resolución disponible es de tipo VGA monocromo, limitada a 512 x 480 píxeles por los 32 KB de RAM. Para pasar a 640 x 480, ésta debería tener un tamaño mínimo de 38,4 KB ($640 \times 480 / 8$). La resolución VGA completa también habría necesitado una entrada/salida suplementaria para conectar la dirección A15, lo que no es posible, ya que el dsPIC no tiene más entradas/salidas libres. Esta elección habría conducido pues a la elección de un encapsulado de tamaño superior, no disponible en versión DIL, con todas las complicaciones de montaje que ello conlleva. Por las mismas razones, la versión monocromo ha sido prioritaria con relación a la versión en color, que habría necesitado, por lo menos, tres veces más memoria RAM y más terminales suplementarios sobre el micro.

Señal de vídeo

La visualización sobre pantalla es una sucesión de imágenes (cuadros), donde cada imagen está compuesta de una sucesión de líneas que contienen, ellas mismas, una sucesión de píxeles. Para simplificar, la señal vídeo puede ser considerada como la yuxtaposición de tres componentes principales:

- Una señal de sincronización de trama (cuadro), que indica el instante de reanudación del barrido del inicio de pantalla;
- Una señal de sincronización de línea, que da la marca para pasar al principio de línea siguiente;
- Los niveles de los píxeles que hay que mostrar, las únicas informaciones visibles en la pantalla.

Las señales de sincronización, en reposo, están a nivel alto. Para la visualización de los píxeles el nivel bajo corresponde al negro. La señal de sincronización de trama define el tiempo entre el principio de una imagen (esquina superior izquierda de la pantalla) y el fin de la misma (esquina inferior derecha). En general, el "refresco" de la pantalla se hace a 60 Hz, con lo que la sincronización de la trama se produce cada 16,67 ms. Su duración se corresponde con el tiempo de dos líneas. La trama está compuesta de 525 líneas, con en el siguiente orden de aparición en la pantalla (ver **Figura 2**):

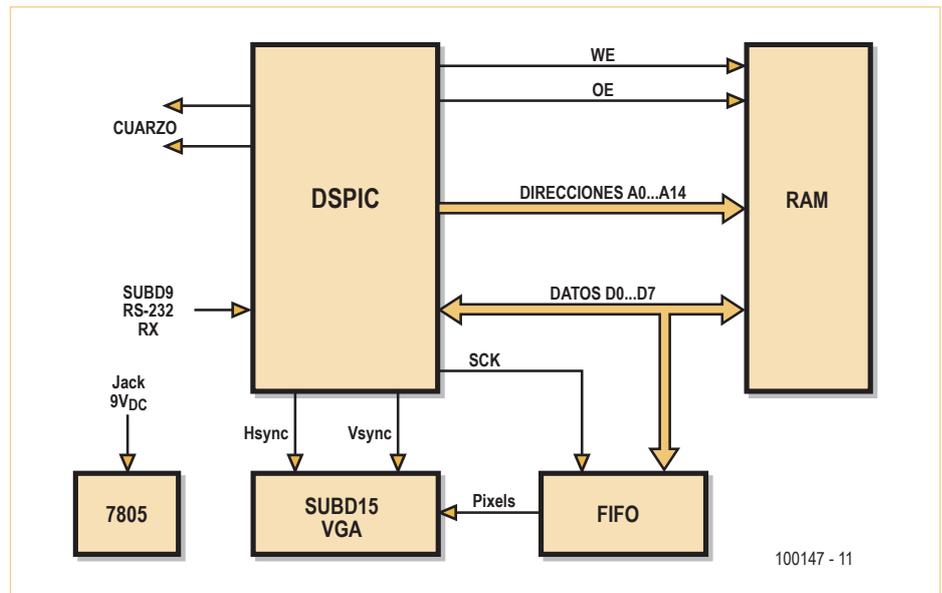


Figura 1. Arquitectura general de la placa VGA.

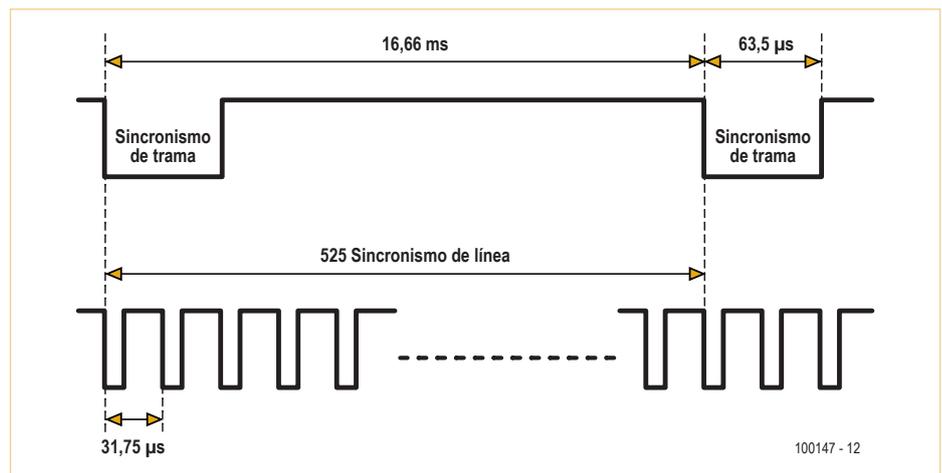


Figura 2. Cronograma de una trama.

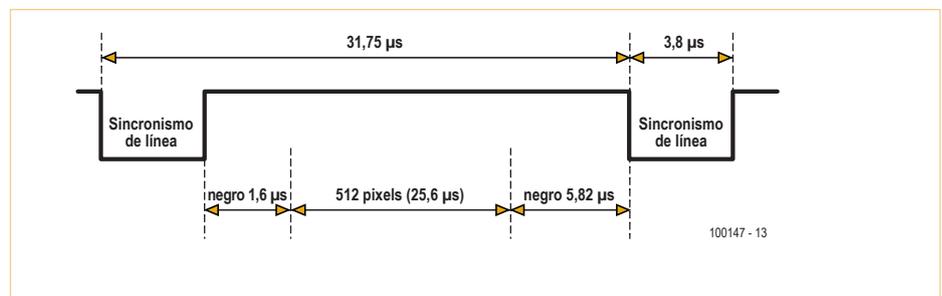


Figura 3. Cronograma de una línea.

- la señal de sincronismos de trama (sincro-trama), de una duración de dos líneas;
- 32 líneas negras;
- 480 líneas que contienen los píxeles que componen la única parte visible en la pantalla;
- 11 líneas negras.

Coordenadas texto y gráfico

La placa gestiona los modos texto (60 líneas de 64 caracteres de 8 bits) y gráfico (512 x 480 puntos) de forma simultánea. La coordenada (0,0) está en la esquina inferior izquierda de la pantalla, tanto en modo gráfico como en modo texto. Las coordenadas en modo texto van de (0,0) a (63,59), mientras que en modo gráfico van de (0,0) a (511,479). Ciertos comandos gestionan las coordenadas de tipo texto en horizontal y las de tipo gráfico en vertical, es decir, de (0,0) a (63,479).

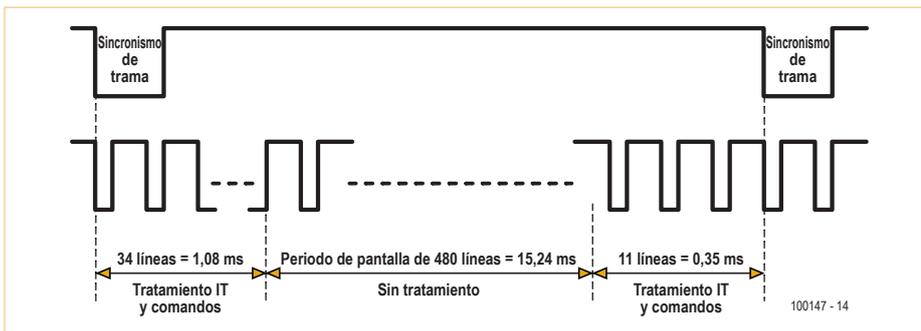
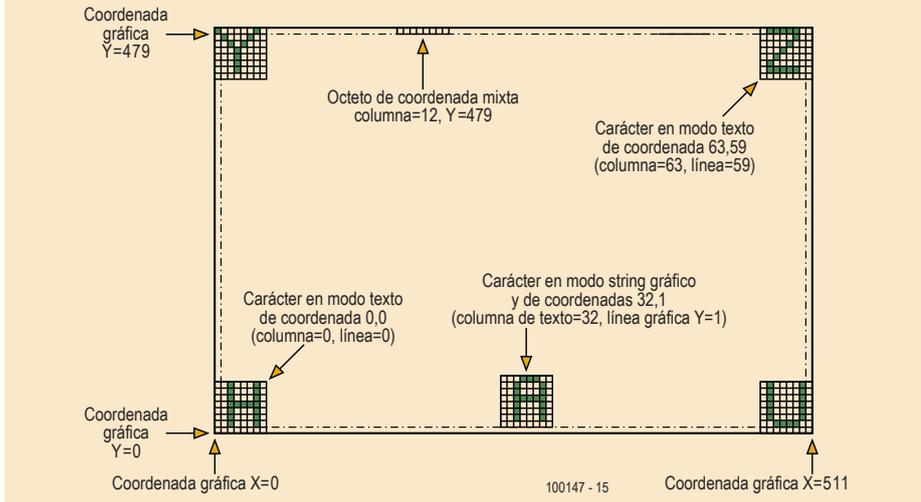


Figura 4. Reparto tratamiento/visualización.

La señal de sincronización de línea define el tiempo entre el principio de una línea (izquierda de la pantalla) y el fin de la misma (derecha de la pantalla). La señal de sincronismo de línea se produce cada 31,75 μ s (16,66 ms / 525) y su duración es de 3,8 μ s. La composición de cada línea, detallada en la **Figura 3**, es la siguiente:

- una sincronización de línea de una duración de 3,8 μ s;
- un período de negro, antes de los píxeles, de cerca de 1,6 μ s;
- 640 píxeles de 40 ns cada uno, es decir una duración total de 25,6 μ s;
- un período de negro después de los píxeles de unos 0,7 μ s.

Las pantallas son bastante tolerantes sobre los sincronismos, de forma que las duracio-

nes de los sincronismos pueden variar un 1 % sin ninguna consecuencia. En el seno de una línea, el reparto del tiempo entre las zonas de negro y los píxeles es todavía más tolerante. En cambio, el instante de aparición del primer píxel, con relación a la señal del sincronismo de línea, debe ser extremadamente preciso, bajo el riesgo de ver aparecer unas ondulaciones verticales sobre la pantalla.

En nuestro caso, sólo se visualizan 512 píxeles de los 640 aceptados por la norma VGA, lo que significa que 128 píxeles serán «negros». Dichos píxeles están colocados al final de línea, el período de negro después de los píxeles será, pues, de 5,82 μ s (0,7 μ s + 128 x 40 ns). Por lo tanto, la parte visible de la pantalla no está centrada pues, sino que aparece desplazada a la izquierda.

Arquitectura del software

La parte del programa es dividida en tres partes:

- La generación de las señales de sincronismo y la lectura de la entrada RS-232;
- La visualización de los píxeles (parte de vídeo visible);
- El descifrado de los comandos y su tratamiento.

La generación de las señales de sincronismo y de visualización de los píxeles debe ser precisa para que la visualización en la pantalla sea perfectamente estable. Esto se realiza completamente con el control de una interrupción del Temporizador (IT), adaptada con una gran cantidad de instrucciones *nop* y que ocupa más del 90 % del tiempo CPU.

La descodificación de los comandos, los cálculos gráficos y la escritura en la memoria RAM de los píxeles, deben contentarse con el 10 % restante del tiempo de uso de la CPU (ver **Figura 4**). Es aquí donde interviene toda la potencia del DSP, no sólo por su rapidez de ejecución, sino también por la utilización del *DSP engine* interno para los cálculos matemáticos (multiplicación y división de 32 bits).

En ciertos casos, los cálculos consumen tal cantidad de tiempo que hemos tenido que usar una argucia: el «vuelo de trama» (“frame stealing”) que consiste en no mostrar píxeles durante la duración de una trama completa y, así, disponer del 100 % del tiempo de CPU durante 16,66 ms. Sin embargo, no hemos abusado de esta técnica, ya que el ojo percibe la ausencia de una trama en la pantalla, lo que provoca una sensación visual desagradable. Por lo tanto, hemos limitado su uso a los comandos de relleno de pantalla en 2D, los que más tiempo consumen.

La lectura de la entrada RX del puerto serie se hace con cada barrido de línea, lo que permite una velocidad de transmisión importante. El valor programado en el encendido es de 9.600 baudios. Podemos cambiar la velocidad según nos convenga, hasta 115 kbaudios, por un comando dedicado. Un programa “buffer”, de tipo FIFO, de un tamaño de 512 octetos, permite apilar los caracteres y los comandos mientras esperan su tratamiento, que sólo puede hacerse fuera del período de visualización. El tratamiento del programa buffer FIFO se hace una en cada trama, las instrucciones

recibidas son tratadas en el orden en que llegan. Es posible, pues, enviar varias decenas de caracteres y/o de comandos al buffer. El límite se sitúa a nivel de la potencia de procesamiento del DSP y del tiempo que se le ha concedido para tratar el conjunto de estos comandos. No se ha previsto ningún retorno de información del DSP hacia el puerto serie, debido a la falta señales de entrada/salida. Por lo tanto, el "apilado" de los comandos debe ser manejado con precauciones para no saturar al DSP, teniendo en cuenta que no existe ningún "cortafuegos" a nivel del Buffer FIFO de recepción. El tratamiento de un comando se inicia después de la visualización del último pixel de una línea y se detiene en el primer pixel de la línea siguiente. Se puede fraccionar sobre varias tramas, si el tiempo de tratamiento sobrepasa el equivalente a 45 líneas (1,43 ms), que es el caso particular de los comandos gráficos. El programa está disponible en [1].

Esquema eléctrico

El esquema eléctrico (ver Figura 5) es la traducción directa del diagrama de bloques. Un diodo LED permite visualizar la presencia de los +5 V. La entrada RX del interfaz RS-232 está en reposo a nivel alto (+5 V). Esta entrada se activa bien por señales de 0 V/5 V sobre el conector K5 (conexión a un microcontrolador), o bien por señales de ±12 V sobre el conector K3 (conexión a un ordenador). La alimentación +5 V puede provenir directamente de nuestra aplicación (en este caso, el puente K2 no debe estar montado) a través de K5 o de K3 (el terminal 9 de K3 ha sido "reclutado" para permitirnos encaminar los +5 V desde nuestra placa). Si no, tendremos que pasar por una alimentación externa de +9 V_{CC}.

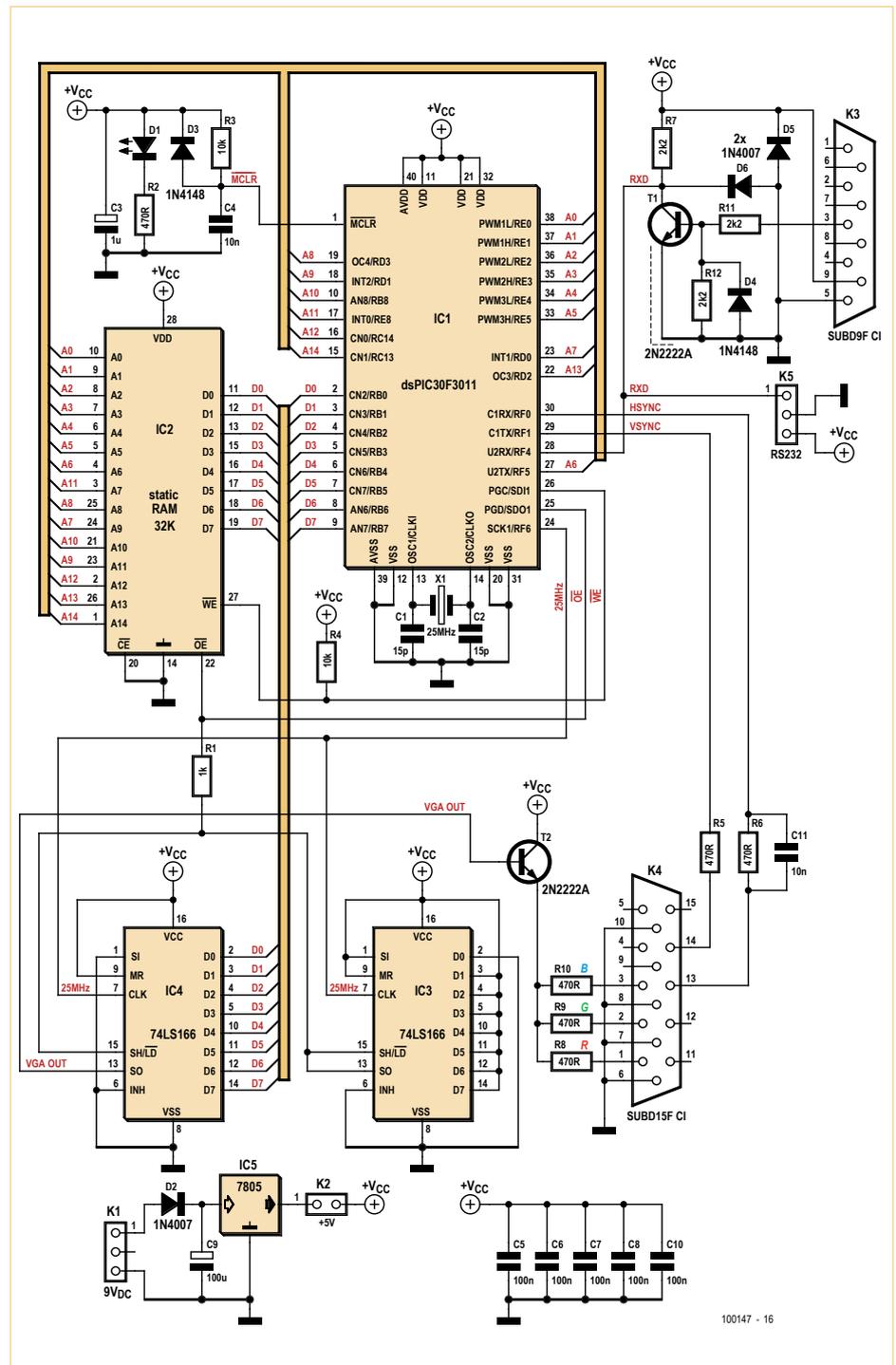
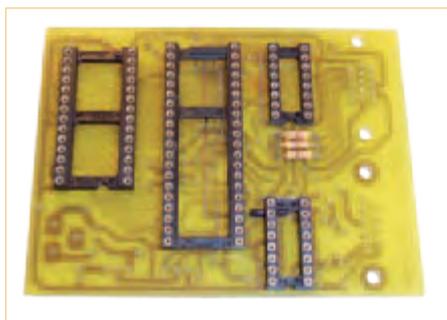


Figura 5. El esquema eléctrico de la placa VGA. ¡Presta atención a las conexiones A7, A9 y A11 de IC2!



Circuito impreso

El circuito impreso puede ser realizado a una sola cara, con puentes en hilo, o a doble cara, con plano de masa. Las pistas son bastante anchas, por lo que no debemos prever ninguna dificultad por este lado. Las señales que circulan por la placa son señales lógicas de 25 MHz, por lo que es importante cuidar la reali-

zación del circuito. Con una placa de una sola cara, pueden aparecer en la pantalla algunas rayas parásitas, pero el circuito impreso es más fácil de realizar. Con placa de doble cara, habrá que conectar el plano de cobre situado en la cara de componentes a un solo punto de referencia de masa (por ejemplo el terminal 8 de IC4).

Tabla 1. Dos opciones para la conexión serie.

Placa VGA	RS-232 (PC)	TTL (uC)
K3 terminal 3	TX	-
K3 terminal 5	GND	-
K5 terminal 1	-	TX
K5 terminal 2	-	GND

Tabla 2. Dos opciones para alimentar la placa VGA.

Placa VGA	+9 V (externa)	+5 V (aplicación)
K1 terminal 1	+9 V	-
K1 terminal 3	GND	-
K5 terminal 3	-	+5 V
K5 terminal 2	-	GND

Montaje de los componentes

Primero se montan los puentes con hilos (atención al cortocircuito con el puente 1 presente bajo IC3, es aconsejable enfundarlo antes de montarlo), luego las resistencias, los diodos, los zócalos de los CIs (opcionales, pero aconsejados, sobre todo para el microcontrolador IC1) y, finalmente, los condensadores y después el resto de componentes. El circuito de reinicio (“reset”) del DSP (R3 / D3 / C4) se justifica sólo si alimentamos la placa con una alimentación +5 V cuyo tiempo de establecimiento, en el encendido, es demasiado lento. En la gran mayoría de los casos, podrá ser reemplazado por un puente colocado en lugar y posición de R3.

Los componentes R11/R12/D4/T1/K3 sirven únicamente para el enlace RS-232. El condensador C11, que permite desplazar ligeramente la señal de sincronización de línea (en el borde izquierdo de la pantalla), puede ser omitido si se revela que es necesario.

Las resistencias R8, R9 y R10 fijan, respectivamente, el nivel de rojo, de verde y de azul al nivel máximo con un valor de 470 Ω. Si cableamos las tres resistencias, conseguimos blanco. En función del valor de estas tres resistencias (entre 470 Ω y unos millares de ohm), es posible mostrar cualquier color. ¡Si montamos solamente R9, conseguiremos un verde al más puro estilo monitor de los años 80! Atención, ciertas pantallas LCD muy recientes miden las impedancias sobre los terminales RVB de K4 y, por lo tanto, no soportan bien la ausencia de resistencia. Para paliar este fenómeno, basta con conectar las entradas R, V o A no utilizadas a la masa de vídeo, a través de una resistencia de 470 Ω.

Los condensadores C1, C2 y C6 se alojan bajo el zócalo del dsPIC (IC1, ver **Figura 6**), ¡componente que se recomienda encarecidamente! En ausencia de este zócalo, habrá que soldar estos tres condensadores en la cara de pistas.

El conector sub-D de 15 terminales podemos recuperarlo de una vieja placa vídeo de PC. También es posible no usar este conector, pero la conexión entre el cable de vídeo y el circuito impreso deberá ser extremadamente cuidadosa, particularmente a nivel de los diferentes blindajes de los cables.

La referencia de RAM es poco importante, hay que escoger un modelo 32 KB, de bajo consumo, con un tiempo de acceso inferior o igual a 100 ns. Incluso hicimos funcionar la placa con una RAM 120 ns. Hemos obtenido buenos resultados con una RAM TC55257BPL-10 (Toshiba), una HY62C256LP-10 (Hyundai) y una MB84256-12L (Fujitsu). Con este tipo de RAM, la placa consume cerca de 50 mA.

Las FIFO del tipo 166 pueden ser indiferentemente un 74HC166 TTL LS o CMOS. Para el dsPIC se aconseja insistentemente la versión dsPIC30F3011-30.

Las **Tablas 1 y 2** resumen las conexiones de la placa en el lado del puerto serie y la alimentación.

Prueba de la placa

En el encendido, el conversor RS-232/VGA se inicializa en modo terminal, con 9600 baudios, modo “scroll” y cursor inicializado en la esquina baja izquierda de la pantalla. La resolución de texto es de 60 líneas por 64 columnas, con caracteres de 8 x 8 píxeles. En este modo, el conversor sólo puede recibir texto y los códigos de control CR y LF, así como el código de control “modo fijo”. El juego de caracteres comprende los códigos ASCII estándar entre <espacio> (0x20) y el carácter tilde (~, 0x7E), más un juego semi-gráfico comprendido entre 0x80 y 0xFF.

Se aceptan varios tipos de comandos como códigos de control clásicos de tipo terminal (RC, BS, VT, LF, etc.), así como los caracteres a visualizar y los comandos de tipo Escape (ESC) que permiten explotar el modo VGA. Con la recepción del código “modo fijo”, la placa pasa a modo texto y gráfico. En ese momento, puede recibir texto y los códigos de control de un octeto comprendido entre 0x01 y 0x20, así como los comandos de alto nivel que permiten trazar los objetos gráficos. El texto y los gráficos pueden estar totalmente mezclados.

Un modo de demostración permite ver funcionar la placa sin envío de comandos. Conectamos la placa a una pantalla VGA a través de K4 y la colocamos en modo “demo” (con un puente sobre los terminales 1 y 2 de K5). Conectamos una alimentación +9 V

sobre K1, con lo que el LED debe encenderse y la demostración debe iniciarse.

Podemos centrar la imagen en nuestra pantalla utilizando el ajuste de posición horizontal. Mejor aún, utilicemos el ajuste de reloj, si nuestra pantalla posee uno, lo que permitirá que la imagen ocupe toda la pantalla.

Para comprobar la placa con un ordenador, conectaremos la placa a una pantalla VGA a través de K4, quitaremos (si es necesario) el puente sobre K5, conectaremos un PC por RS-232 con K3, y conectaremos una alimentación de +9 V sobre K1. Ejecutamos un terminal ASCII en el PC (aconsejamos el excelente programa RealTerm) configurado a 9.600 baudios, sin paridad, con 8 bits de datos, un bit de stop y ningún control de flujo. Pulsaremos los caracteres en el teclado del PC y éstos deben aparecer en la pantalla, en la línea más inferior. ¿No disponemos de un puerto COM en nuestro PC? Ningún problema, cualquier cable adaptador USB/RS-232 comercial nos hará esta función.

Si estas dos pruebas se pasan con éxito, nuestra tarjeta gráfica está operativa. Para aprovechar todas sus funciones es indispensable leer el manual de usuario [1] que detalla el formato de los comandos. Aquí encontraremos ejemplos que nos mostrarán cómo dibujar objetos gráficos en unos segundos y de qué modo podemos crear y previsualizar, con un simple fichero de texto, nuestras aplicaciones gráficas, sin escribir una sola línea de código para el microcontrolador.

Formato de los comandos y de los códigos de control

La placa acepta dos formatos de comandos (ASCII y binario) que están compuestos por varios octetos. El formato ASCII es el más adaptado al envío de comandos con un terminal en el PC. Permite un rápido aprendizaje y visualizar el resultado de un comando sin escribir código.

El formato binario está mejor adaptado a los comandos que provienen de un microcontrolador. Sin embargo, ambos formatos pueden ser utilizados, bien directamente desde el teclado o bien a través de nuestra aplicación, nosotros elegimos.

Los códigos de control están compuestos de un solo carácter cuyo valor está situado

Soporte técnico

En el manual técnico disponemos de numerosas informaciones y ejemplos comentados. El autor está a nuestra disposición para toda pregunta relacionada con la realización de este proyecto (circuito impreso, programación del dsPIC, evoluciones del proyecto), por correo electrónico, en la dirección etiennes33@yahoo.fr

entre 0x01 y 0x1B, por lo que algunos son realmente difíciles de enviar directamente desde el teclado del PC. Sin embargo, pueden ser almacenados en ficheros texto y enviados a través del terminal. Estos comandos permiten la gestión de la posición de visualización (cursor), de alternar entre los modos "scroll" o fijo, y la gestión global de la pantalla (ON, OFF, borrado).

El acceso a los comandos de alto nivel se hace por secuencias de tipo ESC (0x1B). Los comandos de alto nivel utilizan coordenadas de tipo texto, gráfico, o mixto (ver cuadro específico). Todos los comandos comienzan con el carácter ESC (0x1B), seguido de una letra que determina el comando a utilizar, más un número de caracteres que depende del tipo de comando. Estos comandos permiten manipular objetos de texto y gráficos:

- fijar la posición del cursor;
- dibujar un círculo;
- rellenar una zona con un motivo;
- trazar una línea;
- encender/apagar un pixel;
- trazar un rectángulo;
- escribir una cadena de caracteres.

Para más detalle sobre los códigos de control y los comandos, nos ayudaremos del manual de usuario [1].

Los comandos y los códigos de control pueden ser almacenados en ficheros de texto y

enviados a la placa de forma secuencial, lo que permite construir, comprobar y ajustar una buena parte de un proyecto gráfico sin tener que compilar la más mínima línea de código en el microcontrolador.

Conclusión

El campo de utilización de este montaje es muy amplio, desde el simple monitor de depuración hasta la visualización simul-

tánea de curvas gráficas. Esta placa VGA nunca reemplazará a un visualizador LCD gráfico en los sistemas embebidos, pero permitirá a los diseñadores disponer de una superficie de visualización confortable por un precio irrisorio.

(100147)

Enlaces en internet

[1] www.elektor.es/100147

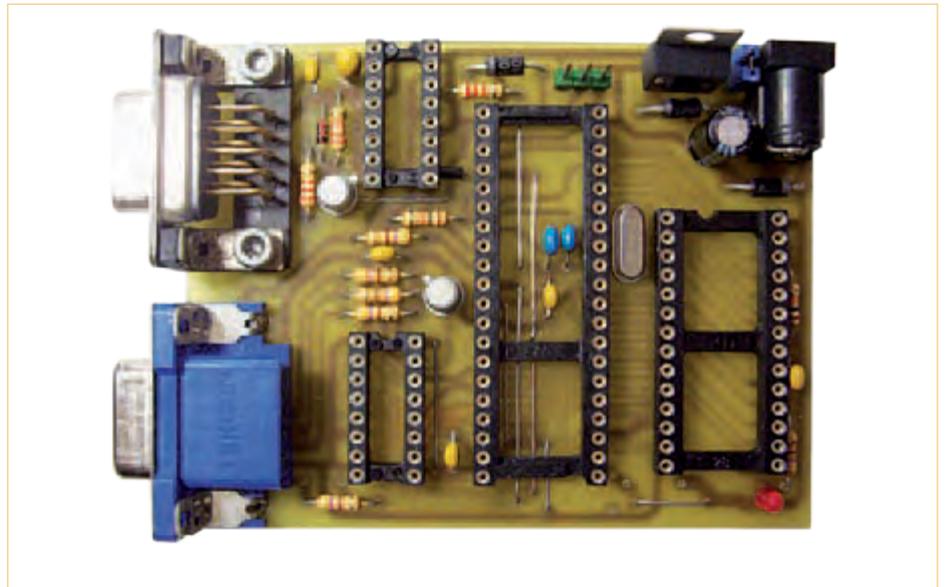


Figura 6. Un componente puede ocultar a otros. Se recomienda encarecidamente el uso de un zócalo para IC1.

Publicidad

EURO
CIRCUITS

**The European reference for
PCB prototypes & small series**

See us live at National Electronics Week
12-13 April 2011 – stand N412

Register for a free entry ticket via our web site.

www.eurocircuits.com

El Robot Proton de Elektor

Plataforma polifacética de experimentación y de educación

Bart Huyskens (Belgica)

Elektor presenta con el Proton una plataforma polifacética de robótica apta para estudiantes, aficionados y profesionales. El robot puede funcionar con diferentes familias de microcontroladores y ofrece una amplia selección en todo tipo de sensores y actuadores. ¡No dejes escapar esta oportunidad e invierte en una combinación única de conocimiento y placer!

El nuevo robot Proton que presentamos aquí, es un sistema de construcción propia que puede ser ampliado hacia todo tipo de direcciones. El robot se vende completamente terminado, pero también se puede adquirir por partes. De esta manera se puede adaptar fácilmente el sistema a los deseos personales y al presupuesto de cada uno. En la **figura 1** puedes ver un resumen del robot Proton con sus características más importantes.

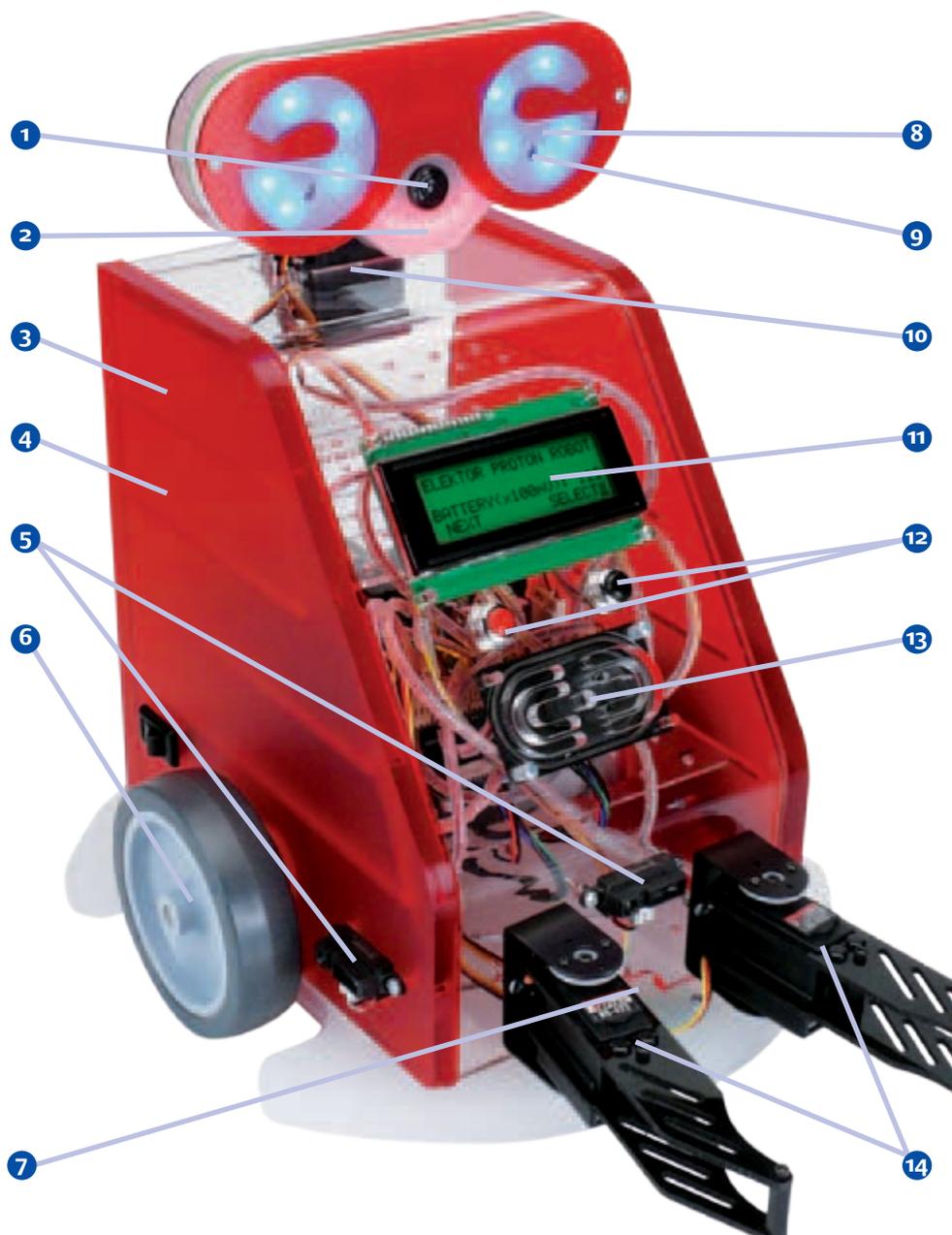
Elige tu propio microcontrolador

La característica más importante es sin duda el hecho de que el **propio** usuario puede determinar qué microcontrolador se usa como corazón del robot y qué lenguaje y/o programador desea utilizar para eso. Durante la adquisición del robot puedes elegir entre una placa "addon" para el PIC16F887 o una placa para el AVR ATMEGA32, pero con una placa de experimentación y un mínimo de soldadura puedes controlar este robot casi con cualquier μC (que disponga de suficientes E/S digitales, unas entradas analógicas, I²C y RS232). No se requiere en absoluto ninguna experiencia en I²C, RS232 o entradas analógicas. Se incluyen macros para simplificar considerablemente la configuración que se necesita.

Como puedes observar en el resumen de la figura 1 y en el esquema de bloques de la figura 2 la mayoría de los componentes se controlan a través del bus I²C. El protocolo I²C es muy sencillo y la mayoría de los microcontroladores llevan incorporados un módulo hardware responsable de la comunicación I²C.

Varios lenguajes de programación

Se han elaborado ejemplos de programación de todos los módulos del robot Proton en Flowcode y en C, tanto para el PIC como para el AVR. El compilador C para el PIC es el HITECH-C Lite, y para el AVR, el compilador es el WINAVR



GCC. Ambas herramientas son gratuitas, pero también puedes optar por otros compiladores y lenguajes. El diseñador ha creado un fichero de cabeceras personalizado para ambos compiladores C, que simplifica mucho la programación en C. Para Flowcode también existe un juego de macros que simplifica los ajustes más complejos. En ambos casos todo el código es libre y por lo tanto completamente adaptable por el usuario.

El propio diseñador ha utilizado el AVRISPMK2 y el PICKIT2 como programador, los dos son buenos programadores de bajo coste puestos a la venta por Atmel y Microchip. Pero también se puede utilizar otro hardware de programación.

Amplio Soporte

Alrededor del robot Proton se organizarán varias actividades de soporte para los usuarios:

Manual: Para empezar hay un manual (en inglés por el momento) muy amplio que proporciona a todo el mundo – desde el princi-

piante al profesional – toda la información relevante para programar los distintos componentes de este robot. Se describe ampliamente cada componente y se dan para cada uno de ellos 4 ejemplos completamente elaborados, que muestran cómo se programa este módulo (en Flowcode y en C): FC-PIC, C-PIC, FC-AVR, C-AVR.

Foro + sitio Web: En el sitio Web de Elektor habrá una página (en inglés por el momento) dedicada únicamente al robot Proton. Ahí encontrarás siempre las versiones más recientes del manual, consejos, trucos, descargas y un foro donde puedes entrar en contacto con colegas y el diseñador del robot Proton. Aquí puedes compartir también código, esquemas, fotos y videos con otros.

Desafío: te retamos a diseñar tus propios proyectos “add-on” y conectarlos a este robot. La placa madre dispone de conectores para E/S adicionales, A/D, RS232, SPI e I²C y en el robot hay sitio más que suficiente para incorporar tus propios proyectos. Controla el robot a través de tu móvil, a través de Bluetooth, a través de WiFi, haz que los robots se comuniquen entre sí. Monta una cámara y tra-

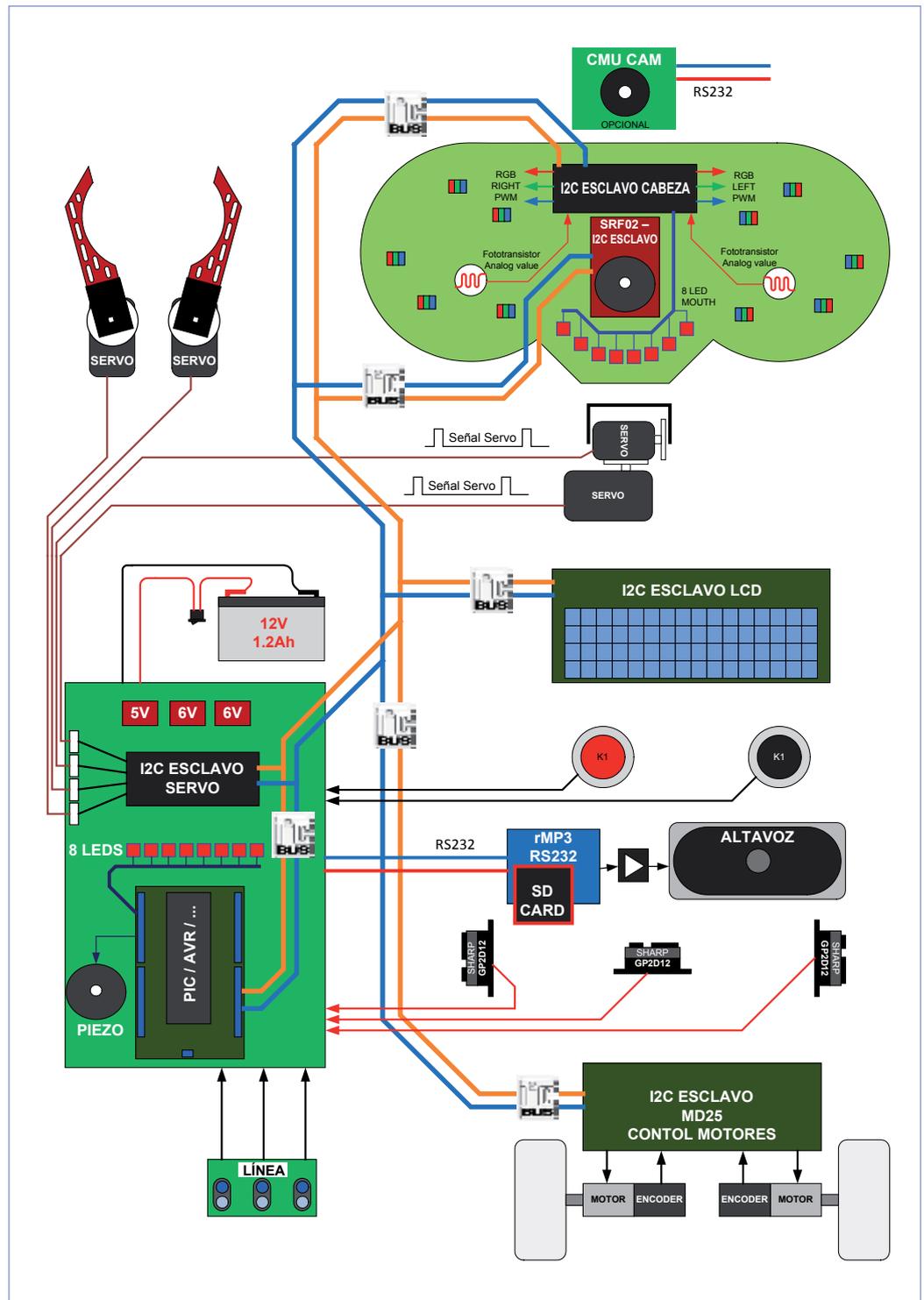
1	Sensor US de distancia	Un medidor de distancia de ultrasonidos que funciona a través de sencillas órdenes I ² C. El valor de la medición se presenta en cm, pulgadas o μs. Función: La medición de distancias desde 4 hasta 300 cm.
2	8 LED para la boca	8 LED forman la boca y pueden ser controlados de forma independiente en el modo normal; En el modo audio funcionan como indicadores de intensidad de la música del módulo de audio. La boca se mueve entonces con el habla o la música. Función: Representar una reacción robótica cuando se perciben sonidos.
3	Altavoz piezoeléctrico	Un pequeño altavoz con una alta impedancia, conectado a una salida digital. Puede reproducir todas las frecuencias audibles. Función: Beep-sonidos – alarmas – tonos de llamada.
4	8 LED en el cuerpo	Conectados a 8 salidas digitales del μC. Función: Depuración, primeros pasos de programación.
5	3 sensores IR de distancia	3 sensores IR analógicos pueden medir distancias desde 4 hasta 80 cm. Función: Evitar o detectar obstáculos.
6.	Módulo controlador de motor	Este sólido módulo controlador de motor de corriente continua se controla mediante el protocolo I ² C. Los codificadores de rueda de los motores proporcionan una realimentación continua al módulo de motor sobre la posición y el ángulo. El μC puede pedir estos valores en cualquier momento, al igual que la tensión de la batería y el consumo de cada motor. Función: Establecer la velocidad y dirección de giro de ambos motores mediante órdenes sencillas.
7	3 sensores de línea	3 sensores infrarrojos en la parte de abajo del robot, conectados a 3 entradas digitales del μC. Función: Seguir líneas negras o blancas.
8	Ojos LED	5 LED de RGB en cada ojo, atenuables independientemente mediante PWM para poder crear así todo tipo de colores. Función: Representación de efectos luminosos.
9	2 Fototransistores	2 Fototransistores en la cabeza, que funcionan a través de I ² C. Función: Medir la intensidad de la luz y localizar una fuente luminosa.
10	2 Servomotores	Estos 2 motores muevan la cabeza del robot mediante órdenes I ² C. Función: Posicionamiento horizontal o vertical de la cabeza.
11	LCD	Los mensajes de texto aparecen en un LCD grande de color azul con 4 x 20 caracteres. Los programas y macros suministrados de ejemplo hacen que el control sea muy sencillo. Función: Mostrar los valores de los sensores medidos o el menú de selección.
12	Pulsador de color rojo y negro	2 pulsadores conectados a 2 entradas digitales del μC. Función: Selección de las opciones en el menú de selección.
13	Módulo de audio	Reproduce directamente ficheros MP3 y WAV desde una tarjeta SD, se controla a través de un juego de comandos RS232. Función: Reproducción de ficheros mp3, wav, ... en la tarjeta SD (entre otras cosas para hacer que el robot hable), guardar datos y valores de mediciones en la tarjeta SD, generar datos para el ecualizador.
14	Dispositivo de agarre	Dispositivo de agarre opcional. Función: coger y mover latas y otros objetos.

baja con reconocimiento de imágenes o haz que tu robot reaccione a comandos vocales... El cielo es el límite. Documenta tu proyecto "add-on" y quizás sea seleccionado para ser publicado en Elektor o para un estupendo premio durante uno de los eventos organizados alrededor del robot Proton. En estos eventos queremos organizar, entre otras cosas, competiciones donde haya cada día un "desafío" especial en el foco de atención.

Paquete de construcción o completamente montado

Este robot se puede encargar como paquete de construcción completo, y también se pueden adquirir por separado cada una de las partes. Un amplio manual de ensamblaje con fotos e instrucciones te ayudarán en la construcción. Todas las placas están completamente ensambladas y probadas, pero aún queda por montar todo el conjunto y soldar algunos cables. Esto no es nada complicado, sin embargo viene bien tener un poco de experiencia en soldadura. También puedes pedir el robot Proton completamente construido y provisto de un amplio programa de pruebas. ¡Mira en el sitio Web para ver todas las posibilidades!

(110263)



Más información y opciones de encargo:

www.elektor.es/proton

Video en Youtube:

<http://www.youtube.com/watch?v=4WiH3LCzYJk>

El autor

El robot Proton de Elektor ha sido diseñado por Bart Huyskens. Bart es un profesor apasionado de la electrónica y los sistemas embebidos en el Instituto St.-Jozef de Schoten (Holanda), y ya ha diseñado varios robots, entre los que se encuentran los exitosos 'Formula Flowcode robot' y 'Robu-robot'. El desarrollo del Proton ha costado más de dos años de trabajo, algo que se puede ver en el resultado.

Regulador de Intensidad de Luz DMX512 de 1 canal

Desarrollado usando E-Blocks y Flowcode 4 para micros PIC

Per Stegelmann (Dinamarca)

DMX512 es el estándar de la industria para equipos de iluminación en escenarios y teatros. Por suerte el estándar está bien documentado, por lo que ayudándonos de este artículo, de estilo tutorial, nada debería impedirnos construir nosotros mismos un regulador de intensidad compatible con el DMX512, perfeccionándolo y encontrando aplicaciones que van más allá de la iluminación y que nos permitirán aprender la programación para PIC y C a lo largo del camino.

El regulador de intensidad DMX512 tratado aquí fue desarrollado a partir del **Probador DMX512 Portátil** del número de Elektor de junio de 2002 [1]. El probador demostró ser un dispositivo valioso ya que podíamos seleccionar cualquier canal con el que quisiésemos trabajar (con hasta 480), así como cambiar el carácter de 'Principio de Trama' ("Start of Frame", SOF) usado, permitiéndonos probar otros equipos DMX512, además de reguladores de intensidad.

El microcontrolador usado en el Regulador de Intensidad DMX512 es un PIC16F88, con su UART serie incorporada, que es ideal para la función del receptor DMX. El 16F88 también dispone del tradicional módulo temporizador/contador, TMR0, que es usado aquí para controlar el disparo del triac que, al final, gobierna la tensión eficaz aplicada a la carga conectada, observando el paso por cero de la tensión alterna, por supuesto, con la ayuda de un

detector. Cuando se detecta un paso por cero, el temporizador del PIC, TMR0, es cargado con un valor que controla el instante de disparo del triac, donde efectivamente 0x00h = potencia cero y 0xFFh = máxima potencia. Realmente, cuando escribimos 0x00h hacia la lámpara, ésta no está totalmente apagada, sino que queda un pequeño brillo permanente para ampliar el tiempo de vida de la lámpara y conseguir que responda así más rápidamente. La rampa que finalmente gobierna la intensidad de lámpara puede ser cambiada con los programas internos del PIC ("firmware").

Circuitos y sub-circuitos

El circuito, cuyo esquema eléctrico se muestra en la **Figura 1**, puede ser considerado como formado por un cierto número de sub-circuitos, identificados como fuente de alimentación,

Prestaciones

- Configuración BCD de la dirección del DMX – los 512 canales son seleccionables.
- Indicación por LED de una dirección DMX incorrecta seleccionada. (parpadeo lento del LED).
- Desconexión del módulo ante un sobrecalentamiento.
- Indicación por LED de desconexión térmica (parpadeo rápido del LED).
- Todo programado con Flowcode v. 4.

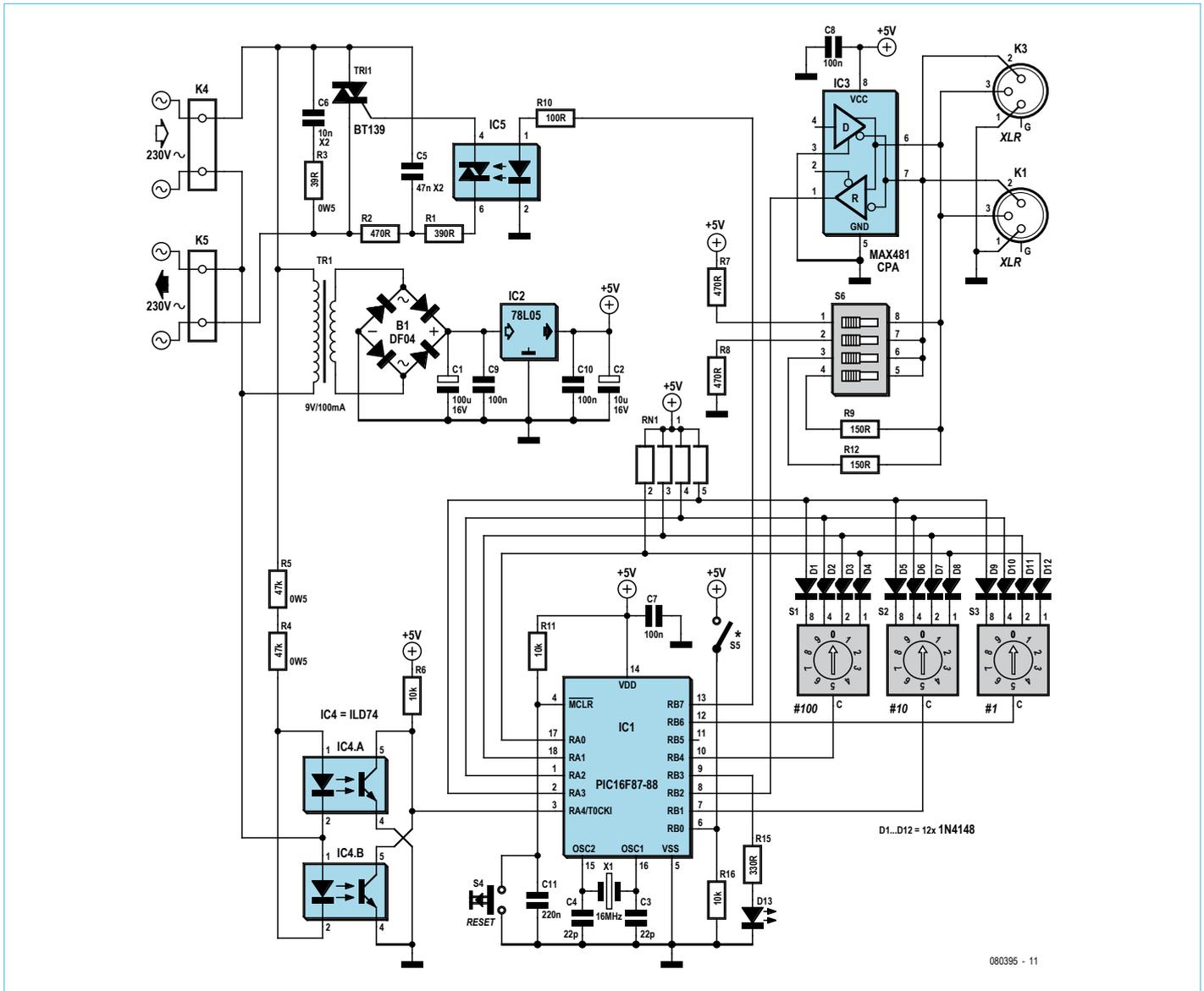


Figura 1. Esquema eléctrico del circuito del Regulador de Intensidad DMX512 desarrollado con Flowcode.

microcontrolador, detector de paso por cero, interfaz de línea de alimentación AC, selector de direcciones DMX y controlador RS-485. Vamos a echar una ojeada a los mimos.

La fuente de alimentación es muy sencilla con su transformador de tensión (Tr1), el puente rectificador (B1), el condensador de

almacenamiento de 100 μF (C1), el regulador de tensión 78L05 (IC2), los condensadores amortiguamiento y desacoplo (de 100 nF y 10 μF ; C9, C10, C2), Nota: Tr1 debe ser elegido para adaptarse a la tensión de red local (230 VAC o 110-127 VAC). El microcontrolador para este proyecto es un PIC16F87/88 que trabaja a 16 MHz. Tiene

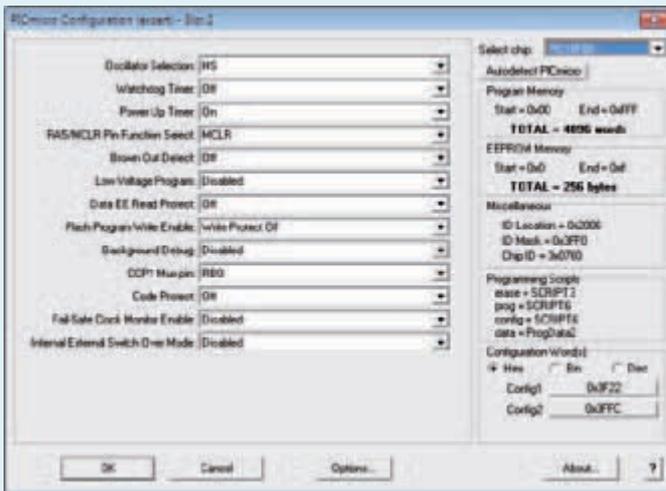
Productos y Servicios de Elektor

- Multiprogramador de PIC de E-blocks: # EBoo6
- Flowcode 4 para PC V4 versión de estudiantes: # TEFLCST4

- El programa de Flowcode (.fcf), programa C, código fuente, ficheros hexadecimales (descarga gratuita): # o80395-11.zip.
- Enlaces en el artículo.
- Componentes accesibles a través de www.elektor.es/o80395

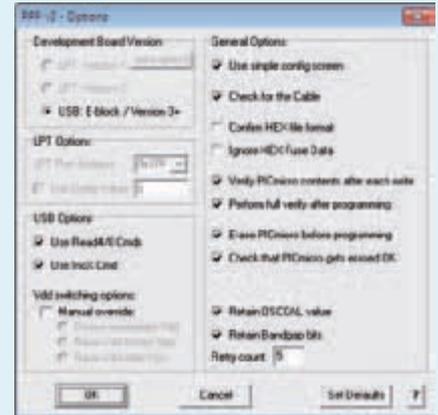
Programación del microcontrolador

La programación del microcontrolador PIC 16F88 es muy sencilla. Si disponemos de un programador que trabaje con Flowcode sólo tenemos que configurar chip, como se muestra en la captura de pantalla de más abajo y, a continuación, pulsar sobre el icono “**Compile to chip**”. En cambio, si no disponemos de un programador que trabaje con Flowcode tendremos que usar la opción “**Compile to HEX**” y utilizar nuestro propio programador para programar el componente.



Encontramos un problema cuando programábamos un PIC por primera vez. Parecía como si todo fuera bien, pero resultó que el código no fue programado sobre el microcontrolador. Después de muchas tentativas desesperadas el problema fue identificado como una mala configuración en la opciones de pantalla del PPP v3.

El culpable resultó ser el campo “**Confirm HEX file format**”. Marcando dicha opción y diciendo No al fichero hexadecimal del PIC18 en el momento de la programación, todo funcionó bien. ¡Después, la opción fue desmarcada de nuevo y la configuración continuó funcionando! Qué había erróneo sigue siendo un misterio, pero podéis intentarlo si encontramos algún problema al programar el componente, ¿tal vez es sólo un problema de W7?



un circuito RC conectado a su terminal /MCLR, lo que permite conectar un botón de reinicio (reset). La línea de puerto RB0 es usada para la función de protección térmica (S5 es el conmutador térmico), y TMR0 para la detección de paso por cero y temporización del instante de disparo del triac. La línea de puerto RB2 es el terminal de entrada de datos del DMX, y RB5 el terminal TX (no usado aquí). El circuito de paso por cero está formado por un doble optoacoplador ILD74 (IC4) y un divisor de tensión compuesto por R4 y R5. El circuito detecta ambos pasos por cero.

El circuito de alimentación AC está construido alrededor de un optotriac (IC5), del tipo MOC3023, que controla un triac de potencia BT139 para el control de la energía a la carga. Este circuito ha sido obtenido de una nota de aplicación de Fairchild.

El selector de direcciones DMX comprende tres conmutadores BCD, cuatro resistencias de “pull-up” y 12 diodos. Por multiplexación de los terminales comunes de los conmutadores BCD se puede leer el valor de cada conmutador.

El controlador RS485 está formado tan sólo por un CI, del tipo MAX481CPA, un micro-interruptor DIP de 4 canales y un par de resistencias. Su configuración es solo para recibir datos.

Desarrollo del programa, paso a paso

Durante el desarrollo de los programas internos para el PIC (“firmware”) del Regulador de Intensidad DMX512, el autor realizó varios pasos, que han sido documentados y descritos más abajo para que nuestros lectores puedan seguir y apreciar la facilidad de uso de Flowcode.

Paso1. El código para leer la dirección de canal DMX fue colocado en un bucle que probaba los conmutadores y hacía parpadear los LEDs si la dirección DMX no era válida. La dirección debía estar en el rango de 1 a 512 para ser válida. Una vez que esto funcionaba, la función fue convertida en una macro de Flowcode de forma que pudiese ser llamada cada vez que fuese necesario desde el

programa principal. El valor de cada conmutador se lee colocando a nivel bajo la línea de un terminal del puerto B, habilitando realmente el conmutador BCD conectado, de modo que el valor puede ser leído por el puerto A.

Paso2. La siguiente cosa era la parte del receptor DMX. Este paso era un poco más complicado de realizar ya que hace uso del componente estándar RS232 de Flowcode y un bloque de código en C. La llamada a la macro RS232 se utiliza para configurar la velocidad del receptor (250 Kbits/s) y para habilitar el hardware. El bloque de código C (ver **Figura 2**) contiene el código que se encarga de recibir los datos DMX de entrada y verificar la ausencia de errores. El formato de datos DMX viene definido como (de forma secuencial y sumamente simplificado):

1. una Ruptura (“breal”) en la línea DMX (1 milisegundo).
2. un “Inicio de trama” (“Start of Frame”) (0x00h para reguladores de intensidad).
3. los valores reales del canal DMX; esto puede llegar hasta 512 bytes de datos pero no todos los equipos envían todos los canales.

Se detecta una condición de Ruptura en la línea DMX comprobando los errores de trama en los datos de entrada (RCSTA = 0). Cuando no se ha recibido ningún error de trama, el siguiente byte de datos debe ser un Inicio de Trama (0x00h), en cualquier otro caso el receptor es reiniciado (DMX_STATE = 0) y se espera la llegada de otra Ruptura en la línea DMX. Si se produce un Inicio de Trama, el contador del canal de DMX se establece a 1 y la señal DMX_STATE pasa a 1. Ahora, cuando se alcanza la dirección del canal DMX seleccionado, la señal DMX_VALUE es actualizada con la señal RX_DATA, la señal DMX_STATE es puesta al cero y el bucle del receptor puede comenzar otra vez.

Paso 3. Con la parte de receptor en funcionamiento, ya era posible seguir con la parte real software del regulador de intensidad. Ésta usa la interrupción del TMR0 en su desbordamiento y se

usa en la detección del paso por cero, así como un temporizador que define el instante de disparo del triac. Esto se consigue por medio de un componente de interrupción de Flowcode que habilita la interrupción de TMR0, eligiendo TMR0 como fuente de interrupción, T0CK1 como fuente de reloj, y el cambio de nivel bajo a nivel alto como el reloj, configurando el prescaler a 1:1; ver la captura de pantalla de la **Figura 3**. El uso de un componente en código C para preestablecer el TMR0 a 0xFFh habilitará el detector de paso por cero como un desbordamiento que ocurrirá en el primer paso por cero de la tensión de línea AC. Esto, a su vez, llamará a la macro denominada 'Fire_triack', que contiene un componente de decisión en su inicio que decide qué camino tomará el programa. Si la señal Enable_trigger = 1, el triac será disparado, Enable_trigger es reiniciada a cero, y TMR0 está listo para un nuevo desbordamiento en el siguiente paso por cero. Si Enable_trigger = 0, TMR0 es cargado con un valor que producirá un desbordamiento cuando el triac tenga que ser disparado.

Paso 4. Pronto, se hicieron necesarios los cálculos de valor de TMR0. Con el microcontrolador trabajando a 16 MHz, el ciclo de reloj dura $16\text{ MHz} / 4 = 4\text{ MHz} = 250\text{ ns}$. Ahora el prescaler de TMR0 es configurado a 1:256, lo que da un valor de $256 \times 250\text{ ns} = 64\text{ }\mu\text{s}$, siendo éste el reloj básico para TMR0. Si ahora tomamos $256 \times 64\text{ }\mu\text{s} = 16,38\text{ milisegundos}$, éste es el período máximo que TMR0 puede medir, que aquí es más que suficiente, ya que sólo necesitamos contar hasta, aproximadamente, 10 milisegundos (medio ciclo de la frecuencia de línea AC de 50 Hz, si trabajamos con una frecuencia de línea de 60 Hz tendremos que modificar el programa). Por consiguiente, TMR0 tiene que ser precargado con un valor que acortará el tiempo, este valor puede ser calculado como: $10\text{ ms} / 64\text{ }\mu\text{s} = 156$. Es decir se requiere un valor de precarga de $(256 - 156) = 100$. Añadiendo este valor de precarga a nuestro valor DMX (el valor de DMX está dividido por 2) conseguimos el valor de precarga real de TMR0, esto se calcula en el bloque de código C de Flowcode.

Paso 5. La última cosa que hemos añadido ha sido el circuito de protección térmica. Este elemento protegerá al montaje contra daños térmicos, usando un conmutador de temperaturas que se cierra a una temperatura de, aproximadamente, 60-80 °C, en caso contrario, el conmutador permanece abierto. Si el comportamiento que deseamos no es éste, tendremos que adaptar el programa en consecuencia.

Estructura del programa

El programa para el regulador de intensidad DMX fue desarrollado con **Flowcode 4 para PICMicro** y funciona como se describe más abajo. El programa de Flowcode completo desarrollado para el proyecto está disponible, gratis, en la página web asignada a este proyecto [2]. ¿Lo consideramos práctico? ¡Pues aquí vamos!.

El programa tiene tres "sub-macros", la primera lee los conmutadores BCD para obtener la dirección del canal DMX a ser usado y la segunda es una macro de interrupción de TMR0 que detecta el paso por cero y controla el disparo del triac. La última macro es una función de apagado térmico que si es llamada, esta macro corta la alimentación a la carga. Sólo es llamada y activada si se ha conectado un conmutador térmico (S5, NO) entre R80 y +5 V. La lectura de los conmutadores BCD se realiza al principio del programa por lo que, si se cambia la

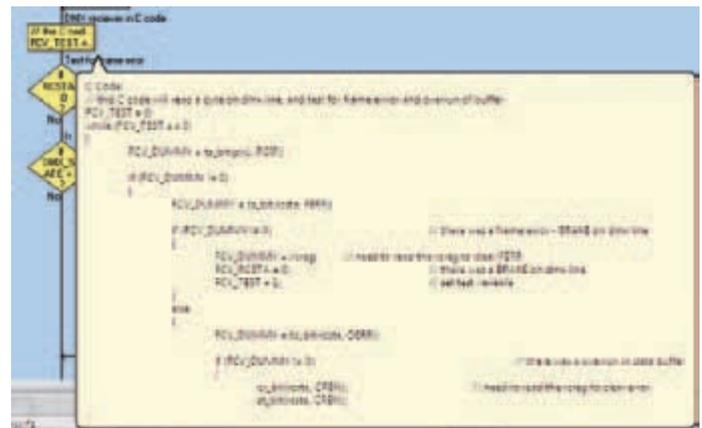


Figura 2. La casilla de código C controla la lectura de datos del DMX512 y la detección de error de trama/desbordamiento.

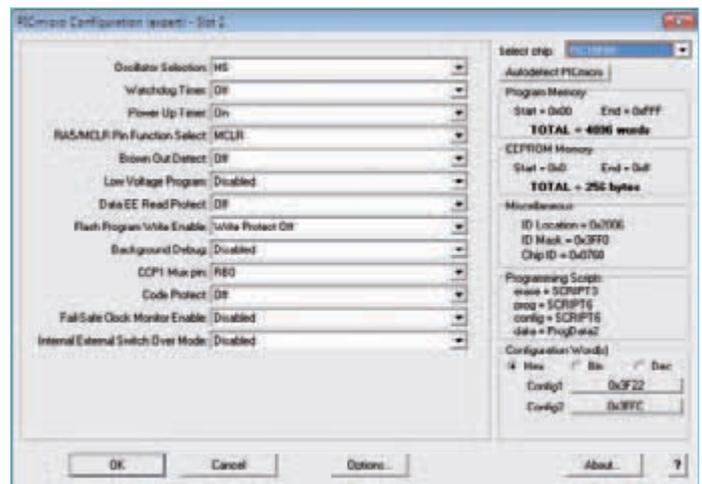


Figura 3. Método en Flowcode para configurar el temporizador TMR0 para interrupciones.

configuración del canal, tendremos que reiniciar el Regulador de Intensidad o quitar la alimentación de red del módulo otra vez.

En el programa principal, la aplicación configura el puerto serie haciendo una lectura "fantasma". En este momento se lee la dirección de canal DMX y TMR0 es inicializado como una interrupción por desbordamiento, a la vez que cuenta la entrada del flanco ascendente en el terminal A4. Además, el registro de TMR0 es pre-configurado con 0xFFh, lo que provoca que se genere una interrupción cuando hay un flanco ascendente en este terminal. Esta es la forma en que se detecta el paso por cero.

Ahora el programa entra en un 'bucle infinito', aquí es donde se realiza la recepción de byte de datos DMX real (una de las oficinas centrales de Apple, en Cupertino CA, está en la dirección: *1, Bucle Infinito*). En primer lugar, el programa tiene que detectar una 'ruptura' en la línea DMX. Esto se realiza en el bucle más externo, donde los errores de trama de los bytes recibidos son detectados. Una vez que se ha recibido los bytes sin errores de trama, la siguiente cosa de probar es verificar que se trata del byte de Inicio de Trama (SOF), el cual, para módulos reguladores de intensidad ('dimmers'), es igual a 0x00h. Si es un byte SOF, entonces la siguiente cosa de hacer es contar los bytes recibidos hasta alcanzar la dirección DMX seleccionada y, a continuación, copiar el dato DMX recibido al registro usado en la macro TMR0, ¡eso es todo! Una parte de esta rutina se muestra en la **Figura 3**.

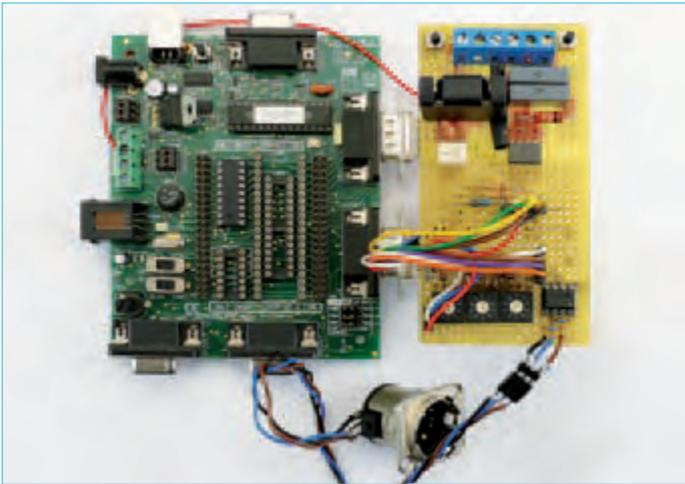


Figura 4. Construcción experimental del Regulador de Intensidad DMX en un trozo de placa de montaje rápido, conectada al Multi-Programador PIC de E-Blocks.

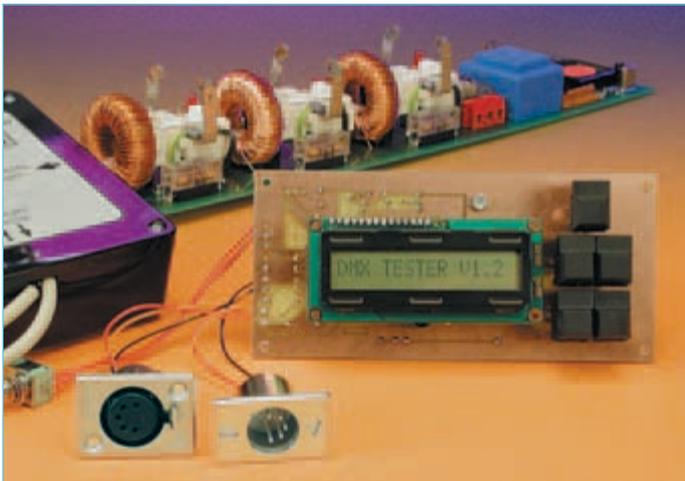


Figura 5. El Probador DMX de Elektor descrito en la edición de junio de 2002.

Dentro de la macro de interrupción TMR0, cuando se produce una interrupción TMR0, el programa verifica si se trata de un desbordamiento de paso por cero o de una interrupción de desbordamiento del temporizador TMR0. Si se ha producido una interrupción de paso por cero el registro del temporizador TMR0 es completa con el valor DMX después del cálculo, y TMR0 es reiniciado de nuevo, esta vez como temporizador. La siguiente

interrupción de desbordamiento marcará realmente el disparo del triac, que se hace con tres pulsos de disparo para asegurar la activación correcta en cualquier momento, en el semi-período de la señal AC. Ahora, de nuevo, TMR0 es habilitado otra vez, pero esta vez para contar el flanco ascendente de A4 y pre-configurarlo con 0xFFh. Esto completa la macro.

Construcción

La **Figura 4** muestra la primera configuración para probar la idea. La placa de montaje rápido ("breadboard") contiene un detector de paso por cero, un interfaz de línea AC, un interfaz RS485 y tres conmutadores BCD para poner la dirección de canal DMX. Se ha conectado un LED al puerto B (PB.3), en el conector sub-D, que es usado para la visualización de estado y depuración. La placa y el MultiProgramador EB006 [3] son alimentados externamente a través de un regulador de tensión de red de 9V y 250 mA de salida. El circuito de la placa de montaje rápido ha sido usado para varios experimentos, no sólo con microcontroladores PIC, sino también de la casa Atmel y otros componentes, ya que está totalmente aislado de la tensión de red AC a través de un optoacoplador y un controlador "optotriac". Por si acaso, tenemos que advertir que el laboratorio de Elektor no han diseñado ninguna placa de circuito impreso para este proyecto.

La placa experimental del regulador de Intensidad está conectada al MultiProgramador PIC de Bloques Electrónicos usando conectores sub-D de 9 terminales conectados a los puertos A y B. El probador DMX de Elektor está dibujado en la **Figura 5**, principalmente para recordar otro brillante diseño de Elektor.

La caja a elegir para este proyecto dependerá generalmente de las condiciones en las que el regulador de intensidad va a ser usado. Si es solo para el uso interior, la caja no tiene que ser de clase IP65 (impermeable), pero si debe ser usado al aire libre se impone una caja IP65. Además, debemos tomar todas las precauciones relevantes con respecto a la seguridad eléctrica cuando se instale la placa en una caja y se realice el conexionado.

(080395)

Enlaces y Referencias en Internet

- [1] Probador DMX512, Elektor de junio 2002. www.elektor.com/010203
- [2] www.elektor.es/080395
- [3] www.elektor.es/e-blocks

Sobre el autor

Per Stegmann (44) vive en Dinamarca. Formado como un técnico electrónico, su profesión actual es la de construir y reparar equipos de prueba para la producción de audífonos. Per escribe: "Cuando uno de mis colegas me preguntó si yo había visto alguna vez un programa para desarrollar código de microcontroladores que usaba iconos para crear dicho código, me mostró una versión de demostración ("demo") de Flowcode, la herramienta software para E-Blocks. Lo probé y pronto me vi enganchado:- era una forma completamente nueva de hacer el desarrollo de código. Antes odiaba escribir algo de código en lenguaje ensamblador y C, pero esto no lo había probado nunca antes. Yo ya había visto alguna publicidad de Flowcode, pero creía que no sería constructivo o capaz de generar código útil. Como resultado de las pruebas, Flowcode demostró realmente ser una valiosa herramienta generando código totalmente válido, al mismo tiempo que creaba la documentación de mi programa de forma sencillísima: sólo tienes que escribir el diagrama de flujo. Así pues, compré una licencia y un Multiprogramador de PIC de E-Blocks y, unas semanas más tarde, ya tenía mi primera aplicación ejecutándose."

Generador de sonidos marinos

Electrónica tranquilizante

El sonido del mar (o mejor dicho: el rompiente) tiene un efecto relajante y tranquilizador sobre nosotros. En estos tiempos ajetreados y apresurados mucha gente experimenta la necesidad de sentir esto también en casa, para, por ejemplo, dormir mejor por la noche. Es fácil simular tales sonidos del rompiente con un poco de electrónica bien compacta.

Harry Baggen (redacción Holanda)



Productos y servicios de Elektor

- Placa: número 100922-1
- Placa completamente construida y probada: número 100922-91
- Disponible caja especial en forma de concha
- Diseño de la placa (descarga gratuita): número 100922-1
- Toda la información disponible en www.elektor.es/100922

El susurro del mar tiene de alguna forma un efecto tranquilizador en la mayoría de la gente. La razón de esto ha sido investigada por una cantidad innumerable de científicos, pero hasta ahora nadie ha podido encontrar una explicación coherente. Aunque realmente no importa, el hecho es que muchos se benefician de la escucha de los sonidos marinos. El sonido marino puede ser una buena alternativa a las pastillas para dormir, especialmente para la gente que sufre insomnio. Y también cuando no lo sufres directamente, es simplemente muy agradable dormir por la noche con el suave sonido de la subida y bajada de la rompiente. ¡Seguro que por la mañana habrás descansado mejor!

Para reproducir el sonido marino no necesitamos mucha electrónica. En el pasado Elektor se publicaron ya varios de estos

circuitos. Como en los últimos tiempos ha aumentado el interés por un generador de sonidos marinos, hemos revisado de nuevo la última versión (de la edición de enero de 1996) y hemos creado con ella una versión más moderna y muy compacta.

Ruido, cuadrado y triángulo

Al hablar del murmullo del mar, piensas directamente en ruido, algo que queremos evitar a toda costa en circuitos electrónicos. Pero aquí queremos justamente eso, generar ruido. Para eso hemos utilizado una receta de confianza, es decir, un transistor del cual se utilice la unión base/emisor en la dirección de bloqueo, de modo que empiece a funcionar como diodo zener. De esta tarea se encarga T2 en el esquema de la **figura 1**. El transistor recibe la corriente zener de la resistencia R15 y la tensión

zener se encuentra en 7...9 V. A continuación, la señal del ruido pasa a través de C7, a la etapa amplificadora construida alrededor de T1 con una elevada amplificación. La amplificación se puede ajustar con el potenciómetro P2. Luego la señal pasa a través de C10 a un pequeño amplificador de potencia construido de forma semi discreta con IC2.A, T3/T4 y componentes circundantes. Este amplificador puede proporcionar suficiente potencia como para controlar directamente un pequeño altavoz, gracias a la etapa de salida push-pull construida de forma discreta, y hacer audible el ruido en un cuarto de estar o un dormitorio. Vale, ahora podemos escuchar el ruido, pero esto no es tan especial. Hay que modular el ruido de tal forma que surja el típico sonido del murmullo. Para eso nos hemos ayudado de tres operacionales que se encuentran

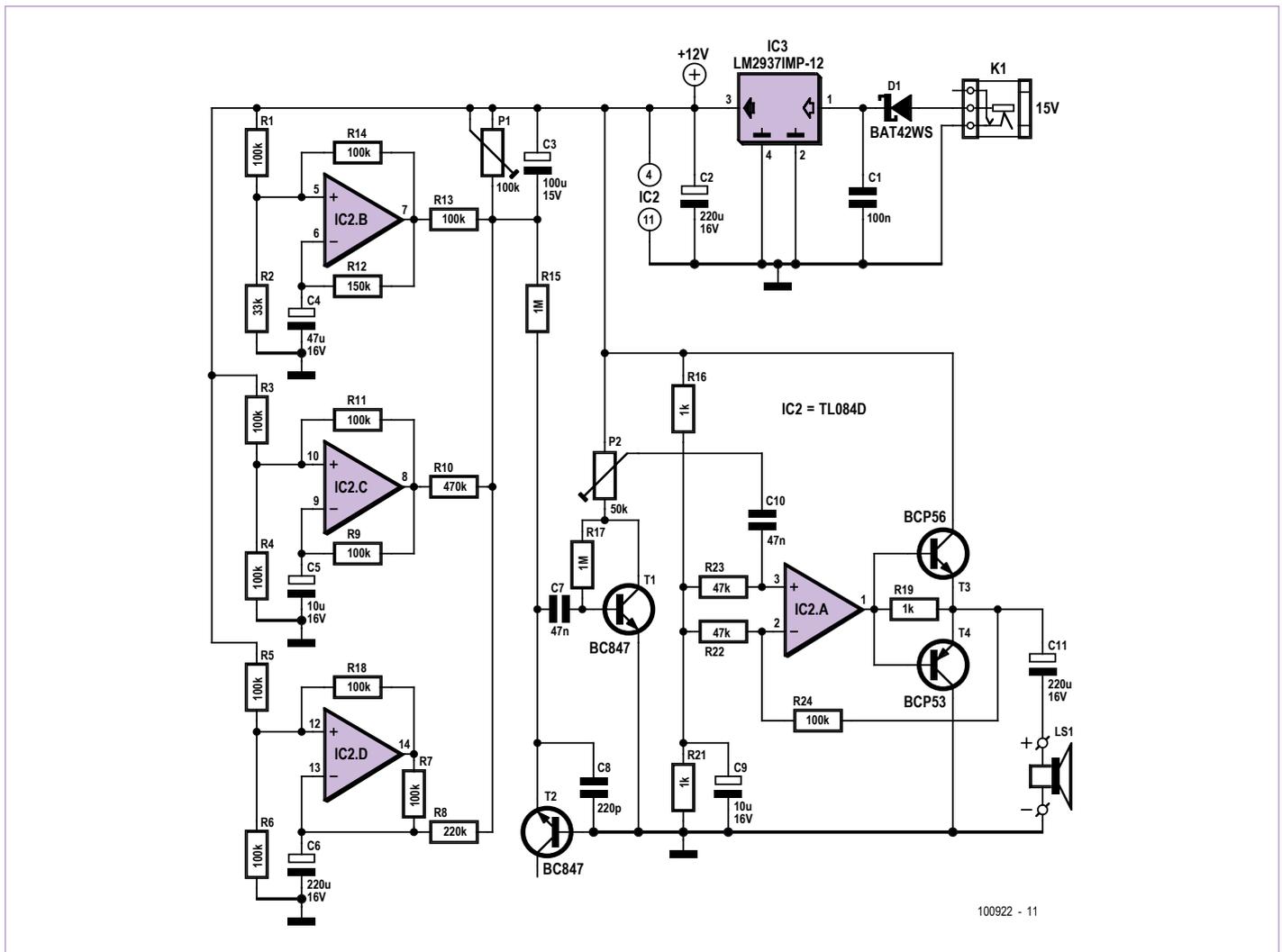


Figura 1. El circuito del generador de sonidos marinos consta de tres generadores de onda cuadrada, una fuente de ruido, una etapa de amplificación y una de potencia.

con IC2.A en un único encapsulado. Estos tres operacionales (IC2.B, IC2.C e IC2.D) están todos configurados como generadores de ondas cuadradas, donde sus frecuencias están muy lejos las unas y de las otras. De esto se encargan los elementos que determinan los tiempos R12/C4, R9/C5 y R7/C6. Los dos osciladores de la parte superior (alrededor de IC2.B e IC2.C) generan una onda cuadrada que es dirigida a C3 a través de R13 y R10. Este condensador electrolítico integra los cambios de tensiones, de modo que surja un tipo de onda triangular. Esta tensión triangular hace que la corriente que pasa por T2 cambie rítmicamente de nivel y con esto también el ruido generado, de modo que simule así la crecida y bajada de una ola. El oscilador de la parte inferior genera una frecuencia muy baja para la cual C3 casi no funciona como inte-

grador. Por eso no se ha conectado la salida de este operacional con C3 a través de R8, sino con el punto nodal de R7 y C6, donde ya existe una tensión triangular. Como C6 está conectado a la tensión de alimentación positiva a través de R8 y P1, el desarrollo de la tensión sobre el condensador electrolítico C6 se parece más a un diente de sierra que a un triángulo. Esta señal simula el efecto de las olas grandes que de vez en cuando llegan a la playa. Como los tres osciladores funcionan de forma asincrónica y a frecuencias diferentes debido a las diferencias en su dimensionado, la forma de onda resultante es bastante aleatoria. Damos algunos valores como información: El periodo de la onda cuadrada generada por IC2.B, es casi 10 s, el de IC2.C asciende a unos 1,5 s e el de IC2.D es el más lento con casi 1 minuto. La suma de todas

estas señales crea en el punto nodal de P1/ R15 una tensión lentamente variable entre unos 9 y 11 V. En realidad el mar genera un espectro más amplio de sonidos, tanto con frecuencias muy bajas como con frecuencias muy altas, pero como el alcance de frecuencia del pequeño altavoz utilizado aquí es bastante limitado, no nos hemos tomado la molestia de simularlo. El estabilizador de tensión de baja caída IC3 se encarga de que todo el circuito sea provisto de una tensión de alimentación estable de 12 V. Se ha añadido D1 como protección contra el cambio de polaridad, de modo que nada se rompa en caso de que la polaridad del adaptador sea incorrecta. La tensión de entrada puede estar entre 15 y 25 V, el consumo es de 100 mA máximo.

Lista de materiales

Resistencias (SMD):

R1,R3,R4...R7,R9,R11,R13,R14,R18,R24 = 100 k Ω (0805)
 R2 = 33 k Ω (0805)
 R8 = 220 k Ω (0805)
 R10 = 470 k Ω (0805)
 R12 = 150 k Ω (0805)
 R15,R17 = 1 M Ω (0805)
 R16,R19,R21 = 1 k Ω (0805)
 R22,R23 = 47 k Ω (0805)
 P1 = 100 k Ω de ajuste (3306W)
 P2 = 50 k Ω de ajuste (3306W)

Condensadores (SMD):

C1 = 100 nF (0805)
 C2,C3 = 100 μ F/16 V (case-d)
 C4 = 47 μ F/16 V (case-d)
 C5,C9 = 10 μ F/16 V (case-b)
 C6,C11 = 220 μ F/16 V(case-e)
 C7,C10 = 47 nF (0805)
 C8 = 220 pF

Semiconductores (SMD):

D1 = BAT42W
 T1,T2 = BC847
 T4 = BCP53
 T3 = BCP56

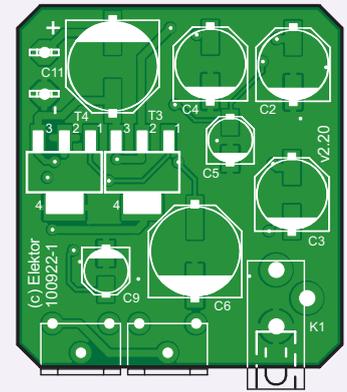
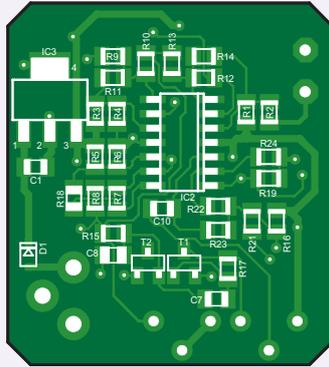


Figura 2. La placa del generador de sonidos marinos (representada aquí a un 150%) es ultra compacta gracias a la utilización de componentes SMD.

IC3 = LM2937IMP-12/NOPB
 IC2 = TL084ACD

Varios:

K1 = conector de la tensión de alimentación (CUI PJ-007)

LS1 = altavoz en miniatura, por ejemplo, Kingstate KDMG20008 (código Farnell 1502730)

Placa: 100922-1

Placa completamente construida: 100922-91 (ver www.elektor.es/100922)

Una placa pequeña

Se ha optado por que todos los componentes sean del tipo SMD, para que el circuito sea lo más pequeño posible y facilitar así su montaje en una caja. Por eso las dimensiones de la placa (ver **figura 2**) han quedado limitadas a unos 28 x 32 mm. Eso hace que la construcción manual sea más difícil, pero como ofrecemos la placa completamente ensamblada por un precio módico, la mayor parte de los interesados en este circuito no tendrán ningún problema. Quien aún así quiera intentarlo por su cuenta, puede descargar el diseño de la placa de nuestro sitio web de forma gratuita o encargar ahí la placa suelta.

Como ya hemos comentado anteriormente, puedes incorporar la placa en casi cualquier caja disponible gracias a sus limitadas dimensiones. El tamaño depende principalmente de la dimensiones del altavoz disponible. Además, el diseñador industrial Rein van der Mast ha diseñado para este proyecto una caja especial en forma de concha que los que interesados pueden pedir (ver enlace a la página web de este proyecto). Con esta caja

se puede crear un bonito adorno que tenga un sitio fijo, por ejemplo, encima de la mesilla de noche.

Lo más prudente es probar primero la placa construida y hacer los ajustes necesarios antes de montarla en una caja. Para eso debes conectar un adaptador de red ade-

cuado (15 V/100 mA es suficiente) y el altavoz. Coloca primero ambos potenciómetros a mitad de su recorrido y escucha si sale ruido del altavoz. P2 funciona como regulador del volumen, puedes girarlo hasta que suene lo suficientemente alto (o bajo) a tu gusto. Luego se ajusta P1 de tal manera que

se escuche claramente el sonido de la crecida y de la bajada. También en pasajes “meditativos” se debe percibir algún ruido. Si todo funciona como debe, puedes incorporar la placa en la caja y empezar a disfrutar del sonido marino.

Hola, ¿Sigues ahí? Espera un momentito antes de dormir, ¡acabas de empezar a leer Elektor!

(100922)



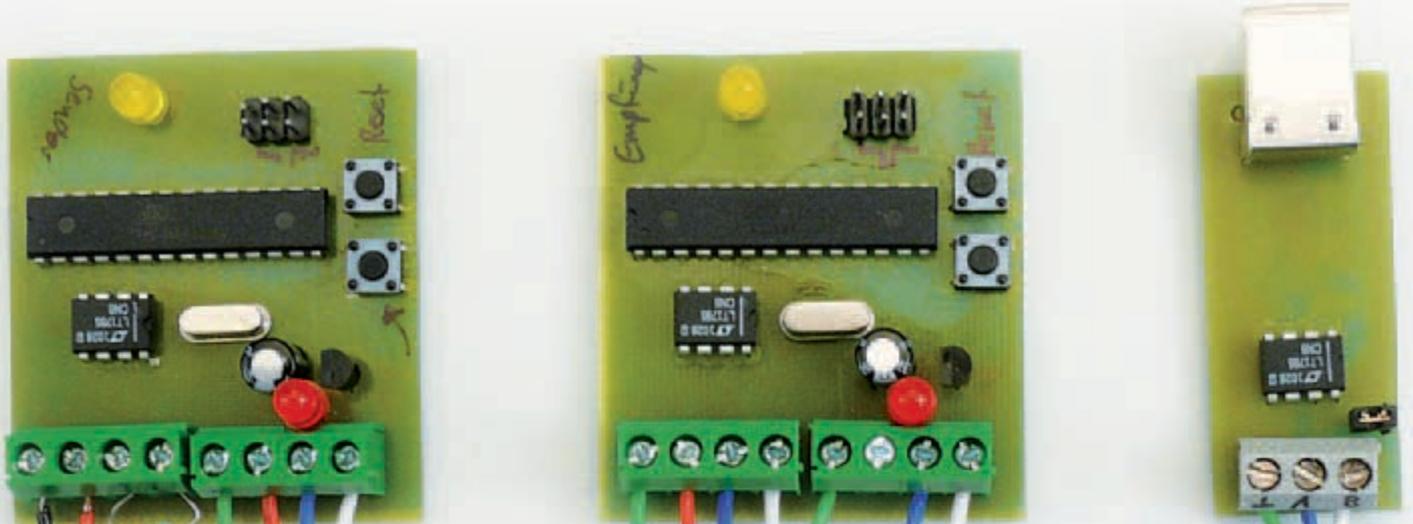
Figura 3. Una impresión de la concha especialmente diseñada por Rein van der Mast de SOLide para el generador de sonidos marinos.

Literatura:

Zeegeluid-generator, Elektor enero 1996 (www.elektor.nl/950125)

¡Que viene el bus! (5)

Ya casi está: primera tarjeta y software



Tras muchas consideraciones teóricas en los anteriores artículos de la serie, ¡finalmente pudimos enviar algunos bytes por líneas auténticas del bus! El hardware lo forman dos nodos de prueba con ATmega88 y un compacto convertidor USB/RS485, y aparte algo de BASCOM y Visual Basic. ¡Ya estaba listo nuestro material de prueba para los primeros experimentos! Aprendimos un montón, como siempre, pero lo verdaderamente importante era sentar las bases de futuros desarrollos de software y hardware.

Jens Nickel (Alemania)

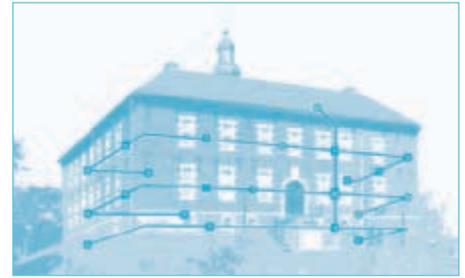
En la última parte de la serie describimos un sencillo protocolo para la comunicación del bus, en el cual un “mensaje” constaría básicamente de 16 bytes [1]. El primero tiene el valor $10101010_{\text{bin}} = AA_{\text{hex}} = 170$, que se utiliza para sincronizar. El segundo byte incluye los datos de direccionamiento y propósito de los datos, y si los últimos dos bytes se utilizarán para el control de errores. La cosa iba bastante bien, pero desde que escribí el anterior artículo, dormirse en los laureles habría resultado un problema. En el grupo de mails formado por lectores especialmente interesados (incluyendo algunos desarrolladores de bus con mucha experiencia) aparecieron nuevas propuestas y sugerencias propias o de otros proyectos. Ya que el RS485 (el cual describimos a menudo) no dispone de control de colisiones “integrado”, surgió un animado diálogo sobre cómo queríamos evitar dichas colisiones en el futuro. Cada vez más hablamos

sobre la posibilidad de utilizar planificadores, mediante los cuales un nodo de red asigna al resto de dispositivos del bus ciertos “slots emisores”. El más sencillo de estos sistemas parece ser una estructura del tipo pregunta-respuesta, en la que el planificador primero pregunta con un mensaje, y el nodo correspondiente responde con otro. Después pasa al siguiente nodo de la serie, y así en adelante. Sin embargo, el modo bautizado por mí como “Round-Robin-Ping-Pong” todavía tiene un problema: cuando hay que preguntar a 32 dispositivos del bus, en la serie sólo podremos hacerlo una vez por segundo (considerando que nuestra tasa de transferencia low-speed es 9600 baudios). La otras discusiones radicaban en la robustez (¿y si el planificador falla?), el tiempo de cálculo de los dispositivos del bus (hay que escuchar a cada uno continuamente) y el enrutamiento de mensajes en segmentos de bus conectados entre sí.

Primeros prototipos

La experiencia me dice que en estas discusiones teóricas siempre viene bien un poco de ensayo-error, y cuando ves que algo funciona desde el principio, esto supone un enorme empuje a la motivación. ¡Había que lograr finalmente los primeros prototipos y software! Del tema del planificador nos ocuparíamos posteriormente (para cada uno de los problemas anteriores ya habíamos estado discutiendo posibles soluciones). Ahora lo que queríamos era mandar finalmente unos cuantos bits por las líneas. No obstante, todavía tuve que retrasar esto un poco, porque como redactores nos encontrábamos a medias preparando la edición de abril y el laboratorio estaba momentáneamente envuelto en otros muchos proyectos...

Afortunadamente Günter Gerold, aparte de integrante del equipo ElektorBus, siempre ha sido hábil a la hora de ponerse manos



a obra... Puedo decir que en laboratorio de Günter rara vez el soldador está frío, y en cuanto a diseñar tarjetas pasa más o menos lo mismo. Sólo unos cuantos días después de haber recibido los componentes, me encontré con un paquete sobre mi mesa, que incluía dos tarjetas completamente montadas, obtenidas de los esquemas publicados en la anterior edición. Y no sólo eso, ¡sino que Günter también había ruteado, fabricado y montado una pequeña tarjeta convertidora USB/RS485! Aparte de eso, este fan confeso de Basic programó unas cuantas líneas en BASCOM, con el fin de enviar un byte por el bus. También diseñó un software de test en Visual Basic, con el fin de comprobar si el PC recibía los bytes desde el convertidor correctamente.

Nodos de prueba y convertidor USB

Las imágenes de las tarjetas se incluyen en el paquete de descarga que puede encontrarse en la web de este artículo [2]. El montaje no resulta en conjunto demasiado difícil, este fan de los SMD ha utilizado únicamente esta clase de “miniaturas” para los componentes pasivos. El esquema del circuito del nodo de prueba puede verse en la **figura 1** (en la foto al comienzo del artículo también pueden verse los prototipos). Como conector ISP utilizamos un conector tipo pin header estándar de 2x3, hemos de prestar atención a toda costa que el frente del conector del programador AVR-ISP apunte en la dirección del controlador.

El circuito del convertidor USB/RS485 (véase la **figura 2**) está basado en la hoja de datos del chip convertidor USB/TTL FT232R de FTDI [3]. El esquema del circuito no resulta muy difícil de comprender, en comparación con el circuito de nuestro nodo de prueba anteriormente descrito. Los pines DI y RO del driver RS485 se conectan en este caso con TXD y RXD del FT232R, que se encarga de toda la comunicación vía USB. El pin CBUS2 de este último chip entrega por defecto la señal TXDEN, con la cual es posible el funcionamiento en modo half-duplex. Cuando se reciben datos por el puerto USB el pin DE del LT1785 se pone a nivel alto, con lo que pueden enviarse datos al bus RS485. Esta misma configuración de pines ya la utilizamos en nuestro convertidor full-duplex “modificado” USB/RS485 de la edición de

diciembre de 2010 [4]. En el pin CBUS3 puede obtenerse la señal PWREN#, que durante el funcionamiento normal del USB se encuentra a nivel bajo [3]. Así el receptor del LT1785 está siempre activo y el componente USB continuamente a la escucha. Mediante el jumper se activa la resistencia de salida de 120 Ω entre las líneas de bus A y B, en las cuales se conectan sendas cle-

mas. El tercer contacto de dichas clemas es la masa de la tarjeta, conectada con la del USB, que se corresponde con el potencial de tierra. La importancia de esta conexión –sin la cual en condiciones normales el bus podría igualmente funcionar– se explica en el cuadro de texto adjunto. Para montar el FT232R hace falta cierta habilidad y experiencia con el soldador,

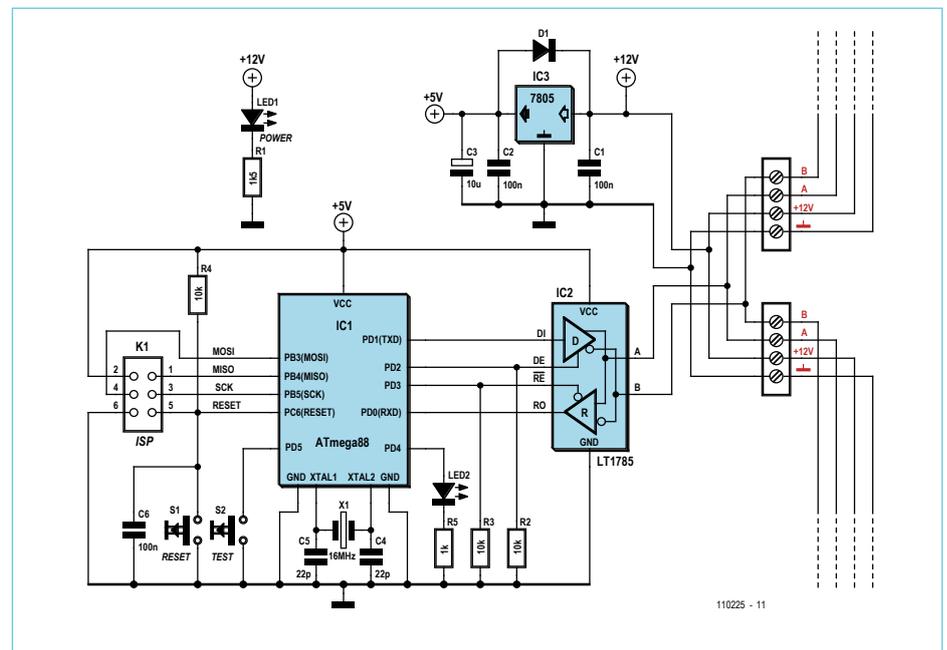


Figura 1. El circuito del nodo de prueba mostrado en la anterior edición. Para cada línea del bus existen dos clemas de tornillo, y así poder direccionar las señales.

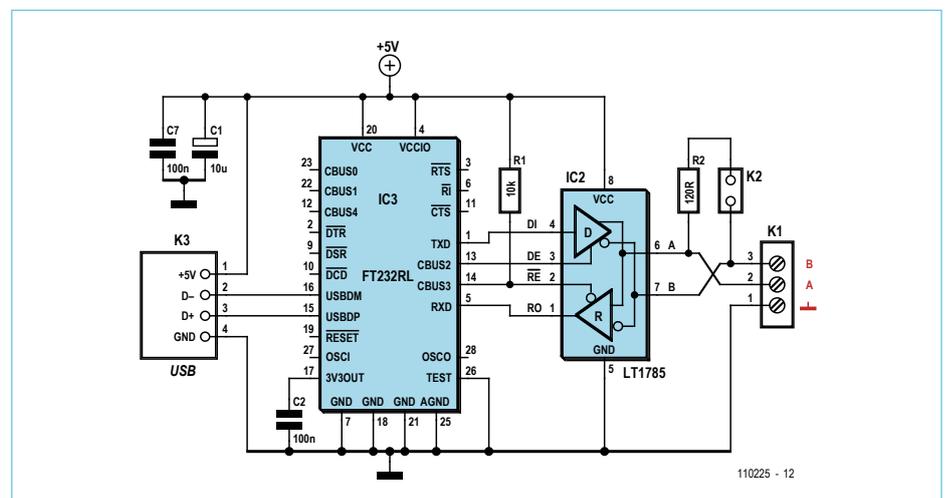


Figura 2. Esquema del convertidor USB/RS485 basado en el archiconocido FT232R.

Código: software en BASCOM (muestra)

```

Readeeprom Ownaddress , 2      'Addresses in EEPROM
Readeeprom Otheraddress , 3

Do
  If Button = Pressed Then      'Button alias Portpin PD5
    If Buttonstatus = Released Then Sendeventflag = True
    Buttonstatus = Pressed
  Else
    Buttonstatus = Released
  End If

  If Sendeventflag = True Then
    Driver = Enabled            'Portpin PD2=1
    Receiverstop = Enabled     'Portpin PD3=1

    Set Ucsr0a.6                'flag (see below) must be (re)set

    Sendmessage(1) = 170        'Byte 0 in ElektorMessageProtocol (EMP)
    Sendmessage(4) = Otheraddress 'EMP Byte 3 = address-byte of receiver
    Sendmessage(6) = Ownaddress  'EMP Byte 5 = address-byte of sender
    Sendmessage(7) = 10          'EMP Byte 6 = first databyte of sender
    Printbin Sendmessage(1) ; 16 'send 16 bytes message

  Do
    Loop Until Ucsr0a.6 = 1     'wait until all bytes sent

    Driver = Disabled
    Receiverstop = Disabled

    Sendeventflag = False
  End If

  If Receivedeventflag = True Then 'an incoming message?
    Commandbyte = Receivedmessage(7) 'first data-byte

    If Commandbyte = 10 Then
      Toggle Testled
    End If

    Receivedeventflag = False
  End If

  Waitms 10

Loop

Onrxcomplete:                'Interrupt on first incoming byte

Startbyte = Udr
If Startbyte = 170 Then
  Inputbin Receivedmessage(2) ; 15 'Read following 15 bytes
  If Receivedmessage(4) = Ownaddress Then
    Receivedeventflag = True      'incoming message
  End If
End If

Return

```

pues este componente sólo está disponible en versión SMD (**figura 3**). Sin embargo, quizá en un futuro Elektor ofrezca una versión del convertidor montada parcialmente, y a lo mejor es posible hasta utilizar el conocido cable adaptador USB/TTL de FTDI (que ya integra un chip FT232R) [5].

Primeros tests

Para las primeras pruebas los nodos se conectaron con cable estándar (véase la foto: línea de bus A = azul, B = blanco, +12 V = rojo, masa = verde), los conectores atornillados hicieron un buen servicio. Además, uno de los nodos tenía contacto directo con el convertidor USB/

RS485. Si configuramos el PC como dispositivo del bus 0, podemos nombrar a los nodos de prueba con el ATmega simplemente “Node 1” y “Node 2”. ¡En cualquier caso no podemos olvidar la resistencia de salida al final de cada uno de los buses! En la parte del convertidor basta con activar el jumper, en la otra sencillamente atornillamos una resistencia entre los conectores A y B del nodo de prueba 2.

Para experimentar necesitábamos un programador compatible AVR-ISP y un entorno de desarrollo. Como Günter ya había programado unas cuantas líneas de prueba en BASCOM, continué con este IDE, pero prometiéndome que también seguiría con el código en C. La instalación de BASCOM no es en absoluto difícil, en la página del fabricante podemos encontrar una buena guía [6]. La versión gratuita, con un código limitado a 4 KB, es más que suficiente (como comparación: el controlador tiene una flash de 8 KB, de los cuales el firmware sólo ocupa aproximadamente el 10 %).

Para programarlo basta con el AVRISP-mkII de Atmel. Ya que BASCOM trabaja con él, tendremos que instalar el driver LibUSB (open source y por lo tanto gratis) [7]. En la ayuda de BASCOM podemos encontrar un manual [6].

Después me puse a desarrollar la primera versión del firmware capaz de enviar y recibir mensajes de 16 bytes utilizando el ElektorMessageProtocol (EMP). Posteriormente me dediqué al software de PC con Visual Basic (el entorno de desarrollo más apto es Visual Studio Express de Microsoft, que también es gratis [8], quien lo prefiera también puede utilizar el IDE de código abierto Sharp Develop [9], el cual utilizó Günter). La instalación de Visual Studio se detalla en [10].

Durante el desarrollo tuve que afrontar algunas adversidades: el PC recibía constantemente bytes fantasmas, a veces el bus funcionaba él solo, otras veces sólo con el programador conectado (!)... Menos mal que mi colega Thijs (que ya se ha encargado de algunos proyectos de audio) y yo finalmente tuvimos éxito, en el verdadero sentido de la palabra todos estos efectos se originaban desde la propia “masa” del asunto (véase el cuadro “Grounding for Dummies”).

Firmware

El resultado de nuestros esfuerzos puede descargarse como código fuente desde la



Figura 3. Montar el FT232R no resulta especialmente fácil, pero ya hemos pensado una solución para principiantes.

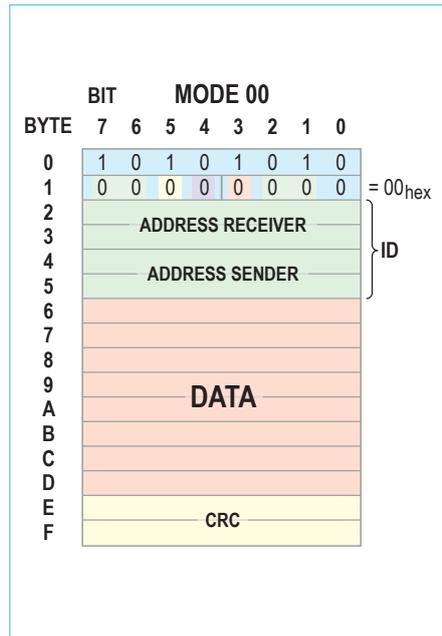


Figura 4. Composición de los mensajes de 16 Bytes (posterior al 00_{hex} específico del ElektorMessageProtocol). El CRC todavía no se ha implementado.



Figura 5. Captura del software para PC. Nuestra pequeña central puede activar los LEDs de los nodos y notificar los mensajes que circulan por el bus.

página web. En ambos nodos de prueba se utiliza el mismo firmware, tras pulsar el botón de cada nodo se envía un mensaje de 16 bytes por el bus. De acuerdo con el EMP consta de los siguientes bytes (ver la figura 4): un byte de inicio (= 170), un byte de modo (= 0), en cada caso dos bytes con las direcciones del receptor y del emisor, así como el verdadero paquete de datos útiles (el CRC todavía no se ha implementado, por simplicidad se ha dejado este valor a 0). Como datos útiles sólo se envía el byte "10". Tanto la propia dirección como la de los otros nodos están programadas fijamente con BASCOM en EEPROM del ATmega (el bit de fusible *High G* ha de ponerse a 0 = *Preserve EEPROM...*, de modo que la EEPROM no se sobrescriba otra vez con cada actualización del firmware).

El código muestra un extracto de la primera versión del firmware. Cuando uno de los nodos recibe el primer byte de un mensaje, en el controlador se lanza una interrupción *URXC (UART RX complete)*. Entonces el programa salta a la etiqueta *Onrxcomplete*. En las líneas siguientes se leen del puerto los otros 15 bytes mediante el comando *Inputbin*, desde el array de byte *Receivedmessage*. Después se activa un bit de flag interno llamado *Receivedeventflag* (no confundir con el flag de registro del ATmega). Éste sirve para indicar al programa principal que se ha recibido un mensaje y que sus datos están a la espera de ser procesados.

El verdadero programa principal consiste en un bucle infinito que integra tres partes. La primera parte comprueba si se ha pulsado el botón. Si es así, se activa *Sendeventflag*, específica del programa, lo cual significa que debido a un evento interno ha de enviarse un mensaje. Otro bit llamado *Buttonstatus* se encarga de evitar los rebotes en el pulsador.

Las otras dos partes del programa comprueban si *Receivedeventflag* o *Sendeventflag* han sido activadas. En el primer caso se procesa el mensaje. Si el byte 6 del mensaje recibido es igual a 10, entonces el LED de prueba del nodo pasará de encendido a apagado o viceversa (aclaración: BASCOM cuenta los arrays desde 1, el byte 6 corresponde por lo tanto al lugar 7 en el array de bytes). Con *Sendeventflag* activada se recopila el mensaje para ser enviado. Los bytes de dirección correctos para el receptor (el otro nodo) y los del emisor permanecen en la EEPROM, como ya se dijo antes. Antes de enviar el comando ha de ponerse a nivel alto la entrada DE del driver, esto se hace en BASCOM mediante el comando *PORTD.2 = 1* (en [11] puede encontrar un breve curso sobre BASCOM). Para que al programar no tengamos que estar continuamente consultando el esquema del circuito, al inicio del programa he fijado algunas constantes y alias, de forma que el desarrollo sea un poco más comprensible: *Driver = Enabled*. Al enviar mediante

el comando *Printbin* se comprueba con un pequeño bucle si todos los bytes han abandonado el buffer de salida (bit 6 = *TXC*, en el registro *UCSROA* del ATmega88), sólo tras cumplirse esto el pin DE del LT1785 volverá a nivel bajo.

Software de PC

Para que la comunicación con el PC funcione, primero hemos de instalar el driver VCP (puerto COM virtual) para el chip FT232R, que podemos encontrar en la web de FTDI [12], disponemos de una guía en [13]. Después podremos ver qué bytes está recibiendo nuestro convertidor a través del bus sirviéndonos de un programa terminal en el PC (que todavía no he comprobado). Nuestro primer software de PC se basa en el programa VB.NET de Günter para probar el convertidor montado por él mismo. Tuvo primero que averiguar cómo se utilizaba la clase *.NET SerialPort* [14], para configurar un puerto COM (virtual) y poder recibir caracteres. Me ha resultado de gran ayuda el libro "Serial Port Complete" de Jan Axelson [15]. Jan, al cual reclutamos para nuestra lista de mails sobre el bus, describe aquí las bases de los puertos COM, RS232 y RS485, junto a un montón de consejos prácticos e indicaciones acerca de aislamiento, cableado, terminales y muchos más. Para programar la parte del PC Jan utilizó *.NET-Framework*, los códigos se encuentran tanto en VB como en C#.

Grounding for Dummies...

...podría traducirse como “masas para principiantes”. Me costó –considerando la electrónica una prioridad más que un mero hobby– más de una hora percatarme de que la masa del bus (el cable verde) tenía que ir a tierra irremediabilmente. En caso contrario, una de las líneas del bus (azul o blanca) no podía tener toma de tierra!

Esto último ocurría cuando queríamos controlar rápidamente lo que sucedía en el bus mediante un osciloscopio, y ambas conexiones de la sonda estaban conectadas a ambas líneas del bus. El problema está en que el driver LT1785 ha de poner ambas líneas a la misma tensión de 2,5 V (Common Mode Voltage) cuando el bus se encuentra en estado de reposo, en lugar de a sus masas (en caso contrario no habría espacio suficiente para los bits que se representan en negativo o positivo mediante diferencias de tensión entre las líneas del bus) [16]. La masa de la tarjeta convertidora USB está conectada a la del USB del ordenador, y por consiguiente a la toma de tierra, con lo que al final las líneas de bus también comparten esta misma toma de tierra. Entonces el LT1785 del convertidor ya no podía trabajar debidamente “en negativo” ni enviar los bits a través del bus.

El potencial de tierra ha de pasar de la tercera clema del convertidor a la masa del bus (y por lo tanto a la masa de los otros dispositivos del bus). El que no se lo crea, que utilice para

alimentar un nodo de prueba una de las pequeñas fuentes de alimentación conmutadas que tenía a mano en el laboratorio de Elektor al principio de los experimentos. Estas fuentes no tienen toma de tierra, sino masa flotante, oscilando alegremente arriba y abajo como un reloj de 50 Hz (respecto del potencial de tierra); y obviamente alterando la componente continua en la tensión del bus. Al conectar los nodos de prueba alimentados por el bus entre ellos, esto sigue sin ser un problema. Pero ojo, en el bus todavía tenemos el convertidor USB limpiamente conectado a la toma de tierra, con lo que las oscilaciones en las líneas del bus son interpretadas por el driver RS485 como bits y bytes. Resultado: ¡datos sucios en el software de PC! Con una línea de masa entre el convertidor USB/RS485 y el resto de nodos del bus nos quitamos un problema de en medio.

La tercera clema fue idea de John Dammeyer de nuestro equipo del bus; según John esta conexión de masa también tiene una función de protección adicional. Si debido a un error en el cableado se conecta una alta tensión continua con el potencial de tierra en un nodo (a masa, y por lo tanto a las líneas del bus), lo único que protege el USB del ordenador es el driver RS485 del convertidor. Lo peor que puede pasar es que una línea de masa adicional conectada al potencial de tierra se achicharre, ¡y esto siempre resulta más barato que tener que cambiar la placa base del PC!

Una descripción detallada del software de PC no entraría dentro de los límites del artículo, pero el código fuente sí está debidamente comentado. *ComPort* es el objeto que utilizamos, forma parte de la clase *SerialPort*. El evento *DataReceived* [14] basta para entrar en la subrutina *Receiver*, la cual procesa los mensajes entrantes. Mediante la propiedad *ComPort.ReceivedBytesThreshold* configuramos previamente que este evento sólo ha de lanzarse cuando se reciben 16 bytes, o sea, un mensaje completo (una de las excelentes ventajas de nuestro tamaño de mensaje fijo).

Lamentablemente, en la rutina *Receiver* no podemos cambiar los elementos de control directamente con un formulario para visualizar las noticias entrantes. Esto ha de hacerse en una rutina independiente, a la cual se llama en *Sub Receiver* mediante

el método *BeginInvoke*; esta rutina ordena los bytes recibidos en un array. He aquí una explicación más avanzada: el evento *DataReceived* se desarrolla en un hilo propio, y sólo pueden transmitirse datos al hilo principal de este modo. Los principiantes pueden simplemente dar por zanjado este tema y dejar las correspondientes líneas de código sin tocar. En la rutina *ShowMessage* se procesan y muestran los mensajes.

Funcionamiento

Tras iniciar el software de PC lo primero es seleccionar el puerto COM en la Combobox; al presionar el botón *Connect* se iniciará la conexión. Ahora el software escuchará al bus y mostrará los mensajes con una nota de la hora en un cuadro de texto (también los enviados por él mismo). En el medio del formulario se representan ambos nodos de

prueba, cada uno con su Test-LED y Test-Button. Pulsar el botón de “Node 2” se corresponde con pulsarlo en la propia tarjeta, así que el LED del otro nodo se encenderá o apagará.

El software puede ampliarse fácilmente. Para concluir he integrado un pequeño acuse de recibo por parte del receptor: cuando el LED se encuentra encendido, envía un mensaje con el byte “11” de vuelta; al apagar el LED de prueba se envía un “12”. Podemos comprobar en el PC el estado real de los LEDs, actuando casi como una mini central domótica. ¡Y todavía faltan muchas ampliaciones más!

(110225)

¡Forma parte del desarrollo! ¡Cualquier consejo o sugerencia es bienvenido en el mail de nuestra redacción, redaktion@elektor.de!

Enlaces en Internet

- | | |
|--|--|
| [1] www.elektor.es/110012 | [9] www.sharppdevelop.net/opensource/sd/ |
| [2] www.elektor.es/110225 | [10] www.elektor.es/100539 |
| [3] www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf | [11] www.elektor.es/080330 |
| [4] www.elektor.es/100369 | [12] www.ftdichip.com/Drivers/VCP.htm |
| [5] www.elektor.es/080213 | [13] www.ftdichip.com/Support/Documents/InstallGuides.htm |
| [6] http://avrhelp.mcselec.com/index.html | [14] http://msdn.microsoft.com/en-us/library/system.io.ports.serialport.aspx |
| [7] http://sourceforge.net/projects/libusb-win32/ | [15] www.lvr.com/spc.htm |
| [8] www.microsoft.com/express/Downloads/#2010-Visual-Basic | [16] http://cds.linear.com/docs/Datasheet/178591fc.pdf |

¡Ayúdame! No encuentro una solución...

Thijs Beckers (laboratorio de Elektor)

En el laboratorio no siempre estamos enfrascados con nuestros circuitos. También surgen preguntas técnicas que encuentran un camino hacia el corazón de la empresa. A pesar de nuestro esfuerzo por probar lo mejor posible los diseños publicados, de escribir los artículos lo más claramente posible y de poner a disposición la mayor cantidad de material adicional a través de nuestra página Web, siempre puede ocurrir que algo vaya mal. Y entonces empieza lógicamente el flujo de preguntas. La mayoría nos llega a través del formulario de preguntas de nuestro sitio Web [1]. Un “buen” ejemplo de esto es lo que pasó con el multiefectos de la edición de diciembre de 2010. Con las prisas por cumplir con la fecha tope del artículo, pintamos en el esquema varios condensadores electrolíticos al revés (C23, C25, C30 en C32), como consecuencia el resultado fue que también obtuvieron la polaridad incorrecta en la impresión de la placa. En principio, esto no es muy grave; ya que como no se trata de altas tensiones, en principio los condensadores electrolíticos no dan ningún problema. Un problema mucho más grande fue que la programación del primer lote de los ATmega8 de la placa principal salió mal por parte de nuestro proveedor. Estaban mal algunas configuraciones de los fusibles, por lo que el integrado intentaba utilizar su reloj interno, mientras debería trabajar con el

Mi amplificador de 1970 empieza a dar ruido. ¿Me puedes mandar una actualización y una placa nueva?

reloj externo de 8 MHz. El resultado fue tan sólo un cuadro parpadeante en el display. Las consecuencias: varios clientes desilusionados y mucho trabajo adicional para el laboratorio, el servicio de atención al cliente y el departamento logístico. Por supuesto que reparamos estos errores lo más rápidamente posible después de su descubrimiento. Muchas otras preguntas técnicas tratan sobre diseños más sencillos como: “¿Dónde tengo que colocar el condensador C15?” y “¿Qué tipo de estaño tengo que utilizar?”. Este tipo de preguntas son cosas de coser y cantar para nuestro compañero Jan Visser. Pero algunas preguntas nos ponen en apuros. “¿Cuál es la diferencia entre un Atmega y un núcleo 8051?”, “Me gustaría utilizar el circuito xx de Elektor, pero con xx salidas más, una alimentación de 12 V y con una memoria de 512 kB en el procesador... ¿Qué tengo que cambiar?” y “Busco un sensor determinado para poder medir... ¿Puedes indicarme uno que no sea demasiado caro y se pueda adquirir preferiblemente en Conrad?”. Como podréis comprender, nos falta tiempo para formular una respuesta a este tipo de preguntas específicas. La diferencia entre un procesador Atmega y un 8051 no se puede

explicar en pocas palabras sin un profundo conocimiento de ambas familias de procesadores, y además se puede encontrar abundante información en ambos campos para poder tomar una decisión sobre con cuál empezar.

Una pregunta que se escucha frecuentemente es dónde está a la venta algún componente. Las tiendas de electrónica mantienen un stock reducido de componentes, especialmente SMD. Lo que en este caso se puede preguntar al dependiente es si él puede encargar varios componentes para ti. Por supuesto que puedes ir con la lista de materiales de Elektor y preguntar si quiere pedirlo en Farnell. Por lo general, sólo se pueden pedir sueltos los componentes especiales; los componentes “estándares” hay que pedirlos en múltiplos de 10, 25 ó incluso 100. Esto no supone generalmente ningún problema para los componentes que cuestan varios céntimos por unidad, pero si se trata

¿Podéis diseñar un “news ticker” para el antiguo molino reconstruido de nuestro pueblo igual que un “LED-spinner”?

de un condensador de 0,25€ por unidad, del cual hay que pedir un mínimo de 10 y hay que hacer esto para cinco valores diferentes, entonces el precio empieza a subir bastante. Así es que hay que mirar un poco cómo hacerlo lo mejor posible. Aparte de esto, muchas veces se organizan campañas de compra,

principalmente a través de foros visitados por aficionados – gente que se encuentra con el mismo problema, en Holanda está Circuits Online [2], entre otros sitios.

A propósito de la disponibilidad de los materiales: Nos esforzamos por elegir los mejores componentes corrientes posibles. Sin embargo, no podemos evitar que algunos componentes después de tres años (o a veces antes) sean declarados ‘obsoletos’ por el fabricante. El mundo de la electrónica es casi igual de rápido que el mundo de la informática: un circuito de hace 10 años procede casi de la prehistoria y los integrados, fundamentalmente, son difíciles de adquirir en general...

Para que la tramitación de preguntas sea la más rápida y completa posible, hemos añadido al formulario online de preguntas [1] algunos campos que se deben rellenar. El número, el título del artículo, el mes y el año de publicación nos ayudan ya bastante. Si además describes bien el problema o la pregunta, probablemente podamos ayudarte directamente y no tengamos que pedir más información por no tener claro lo que quieres saber. Gracias por adelantado de parte del equipo del laboratorio.

(110131)

Enlaces Web

[1] www.elektor.es/contact

[2] www.circuitsonline.nl

Medir la temperatura en un santiamén (2)

Thijs Beckers (Redacción NL)

Algo imposible de hacer con un termómetro IR como el de la prueba del mes pasado, es cosa de coser y cantar para una cámara termográfica, también llamada cámara IR. Con motivo de la prueba, Vincent van der Walle (foto 1), ingeniero de ventas de KWx B.V. [1], uno de los distribuidores holandeses de FLIR [2], vino a darnos una explicación sobre cámaras termográficas. Claro está que una demostración de varias cámaras termográficas desarrolladas por FLIR formaba parte de esto. Quizás estas cámaras tengan un precio demasiado elevado para un aficionado, el modelo más barato tiene un precio justo por debajo de 1000 € (el FLIR i3), pero si se utilizan en departamentos de investigación y desarrollo, la inversión se recupera enseguida incluso en el modelo más caro. ¿Cómo? Sigue leyendo...

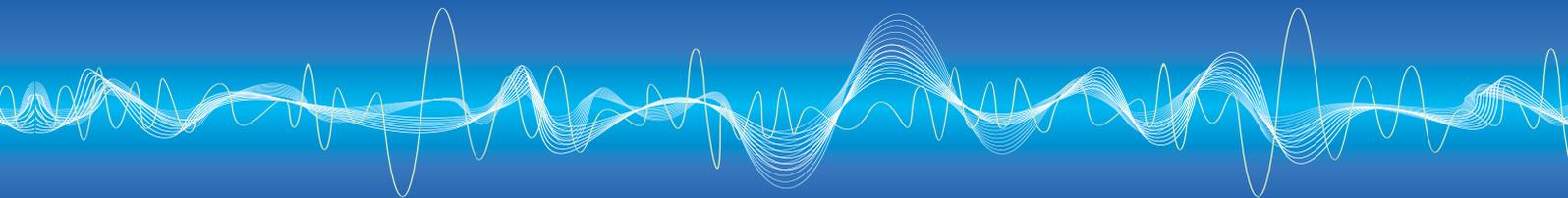


Primero es importante saber lo que estás midiendo. Esto ya se planteó en la prueba de los termómetros IR que publicamos en Elektor el mes pasado. La emisividad de un objeto tiene una influencia especial sobre la temperatura medida (mira también el recuadro “emisividad” en el artículo de la prueba del mes pasado). Lo que es difícil de ver con los termómetros IR, se ve fácil y rápidamente con una cámara termográfica: las mediciones a una sencilla lata llena de agua que acaba de hervir muestra rápidamente el “problema” de la medición de temperatura con un termómetro IR (ver foto 2). Si no se adapta la emisividad al material (lata, en este caso, con un factor de emisividad de unos 0,15), el valor medido se desvía tanto, que de hecho resulta inservible. Nuestro modelo de referencia, el Fluke 572, indicaba un valor de 34,5 °C si no se ajustaba la emisividad.

En un termómetro IR es difícil ver este error de medición. Pero en una cámara termográfica (que utiliza de hecho el mismo sensor que un termómetro IR, pero hay múltiples sensores y están puestos en una matriz) se ve directamente que aquí pasa algo “raro” (ver foto 3). La lata parece estar bastante fría (el punto de medición del centro asciende a 28,9 °C), mientras el agua y los trozos de cinta aislante que hay en el borde se iluminan fuertemente como indicación de que estas áreas tienen una mayor temperatura; unos 72 °C si miramos la escala en la parte inferior de la pantalla.

Lo que la cámara “ve” en realidad cuando está dirigida hacia la lata, es la emisión de calor reflejada del entorno de la lata y una parte del calor de la propia lata (por eso la temperatura es un poquito más alta) en lugar de contar sólo con la temperatura de la propia lata. Los trozos de cinta aislante de color sí proporcionan el resultado correcto de medición. Aquí no importa el color de la cinta, tal y como se puede apreciar también en la foto. Únicamente entran en juego el material y la emisividad específica (y ésta es unos 0,95 en el caso de los trozos de cinta). De modo que con una cámara IR verás rápidamente dónde falla la medición y dónde puedes medir correctamente.





Pero a pesar de esto hay que tener cuidado y saber bien lo que se hace. De esta forma dirigimos, a modo de prueba, las cámaras IR hacia los disipadores de una fuente doble, de las cuales una parte proporcionaba 2 amperios en 8 ohmios y la otra parte 0,5 A (también en 8 ohmios). Los disipadores eran bastante opacos, y a pesar de los diferentes colores elegidos por el fabricante de los disipadores, las cámaras IR no tuvieron ningún problema en mostrar una imagen correcta de la escala de la temperatura sobre los disipadores (ver **foto 4**). El disipador de la derecha está claramente más caliente y parece que estuviera ardiendo en la pantalla de la cámara (el disipador alcanzaba unos 55 grados). Lo que en cambio nos llamó la atención, fue que la parte superior de los disipadores parecían estar más fríos (o fríos) si colocábamos la cámara a la misma altura que la fuente de alimentación (es decir directamente detrás de la fuente). ¿Qué es lo que pasaba? La parte superior de los disipadores de aluminio funcionaban como espejo y reflejaban la temperatura de un muro relativamente frío detrás de la fuente (igual que funciona como espejo el firme asfaltado en un día caloroso cuando miras hacia la lejanía). El mismo efecto se puede ver bien en la **foto 5**: La imagen térmica de la lata con agua caliente se refleja en la superficie de la mesa, mientras que la temperatura de la mesa es más elevada sólo directamente por debajo de la lata.

Por desgracia, este tipo de errores de medición sólo se pueden observar —y no se pueden evitar. No es algo que se pueda medir en una especie de “cámara anecoica” de irradiación de temperatura para evitar este tipo de reflexiones, igual que se hace con las pruebas de sonido. Cada objeto tiene una irradiación térmica y una influencia sobre la medición. Por eso hay que pensar bien lo que estás midiendo y con una cámara IR es mucho más fácil interpretar el resultado que con un termómetro IR.

Quizás te preguntes para qué le sirve una cámara IR a un departamento de diseño. La respuesta es sencilla: Las cámaras IR pueden determinar la temperatura con mucha precisión de aquello que se presenta delante de su lente. Eso es muy útil en la búsqueda de fallos de la placa impresa. Dirige la cámara hacia la placa y verás directamente dónde se aumenta la temperatura (demasiado) y dónde probablemente existe un fallo (de diseño). La cámara IR FLIR T425 de la foto 6 era incluso tan precisa, que pudimos ver la “huella IR” del silicio del integrado FV1 del multiefectos digital (Elektor de septiembre de 2010) en la parte exterior del encapsu-

lado, aunque la temperatura sólo estaba a dos grados por encima de la temperatura ambiental (temperatura del integrado: 27 °C). Para el lector observador: en la parte de la izquierda de la pantalla de la cámara, también se pueden observar dos estabilizadores de tensión que han cogido ya colorcito.

Si te cansas por un momento de los circuitos eléctricos, siempre puedes imitar al CSI con una cámara como esta... Una huella dactilar en la mesa deja, por lo menos, durante cinco minutos un rastro térmico, eso se puede ver perfectamente con esa cámara. Generalmente se pueden trazar las pisadas con facilidad, especialmente en la alfombra. Sigue al colega que acaba de pasar e intenta averiguar de dónde acaba de venir...

Generalmente puedes ver exactamente en qué punto ha caído un objeto y dónde ha tocado el suelo. En realidad eso es sólo física elemental, ya que cuando un objeto toca el suelo, su energía cinética (de movimiento) se convierte en calor. Y eso se puede ver perfectamente con la cámara IR.

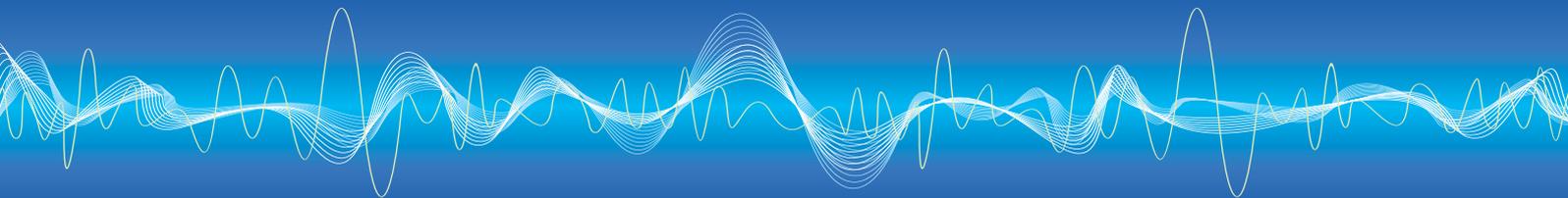
(110273)

Enlaces Web:

[1] www.kwx.nl

[2] www.flir.com





Problemas con Skype

Thijs Beckers (Redacción NL)

En primera instancia quedamos contentos con el resultado de las pruebas de nuestro circuito de reunión online. Las señales de los micrófonos llegaban correctamente al canal izquierdo y derecho de nuestro PC. Las pruebas con el Windows Live Messenger transcurrieron felizmente y nos escucharon perfectamente en el otro lado. Todo funcionaba como debía.

Sin embargo, cuando hablamos mediante Skype con nuestro redactor francés, a través del portátil de nuestro jefe de redacción internacional, empezamos a tener problemas. Sólo funcionaba un micrófono. ¿Teníamos una rotura en un cable? ¿Estaba bien configurado el software del sistema de sonido del portátil? ¿Era un problema de la configuración en Skype? ¿Se había agotado la pila de 9V? No, todo estaba bien configurado y funcionaba como debía. También sustituímos los cables para mayor seguridad, pero el problema persistía. Intercambiamos los módulos y nos llamó la atención que fuera ahora el otro módulo el que parecía dejar de funcionar. Eso nos hizo que pensar. En Windows Live Messenger todo funcionaba perfectamente. En Skype sólo funcionaba el canal de la izquierda y el intercambio de los módulos no servía de nada. Uno más uno es dos y una búsqueda rápida en Internet nos aclaró que no fuimos los únicos en observar esta “característica” de Skype. Era imposible transmitir señales en estéreo. Tuvimos que admitirlo, Windows Live Messenger también transmitía únicamente la señal

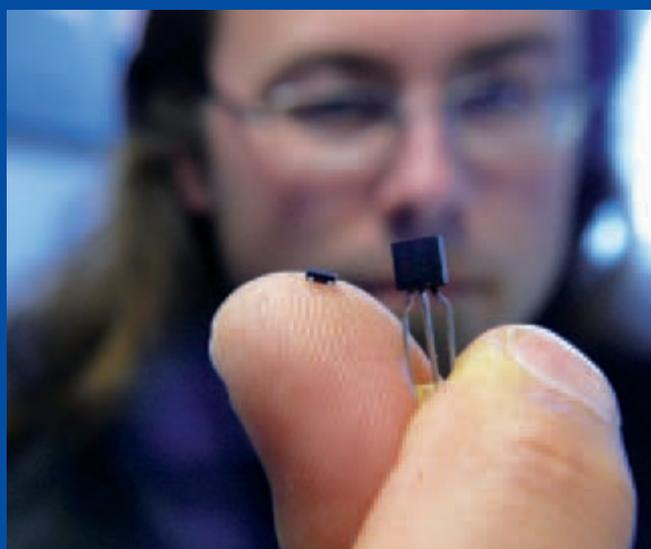


monofónica, pero para eso sumaba correctamente las señales de los canales de entrada de la izquierda y la derecha. Si nos fiamos de varios foros, ¡Skype no hacía eso, por lo menos, desde 2005! Seguramente hay una buena razón para ello...

Finalmente, para poder utilizar nuestro sistema con Skype, tuvimos que entregar las señales de ambos canales a la entrada izquierda del sobremesa o portátil. La solución más sencilla es hacer una suma a través de dos resistencias. Esto no es muy correcto, pero sí es efectivo. Una breve prueba nos confirmó su buen funcionamiento y en un recuadro aparte de este artículo se explica cómo construir este “cable Skype”.

¡Lo que bien termina, bien hecho está!

(110314)



Thijs Beckers (Redacción NL)

A los electrónicos normalmente no les gusta el ruido. Cuanto menos se oiga, mejor. Pero durante el diseño del generador de sonido marino descrito en algún otro sitio de esta edición sucedió por una vez al contrario. En ese diseño necesitábamos un transistor que hiciera el mayor ruido posible. Si coges el esquema y le echas un vistazo: T2 funciona como fuente de ruido. El circuito original que data de 1996 utilizaba un BC547 y una fuente de alimentación de 12 V. El diseñador Chris Vossen

Problemas de ruido

prefería una fuente de alimentación más baja, de modo que el circuito podría funcionar también con una pila de 9 V. El problema aquí fue el transistor T2 generador de ruido, que se había sustituido en el nuevo circuito por uno en formato SMD, un BC847. Este necesitaba — igual que el BC547 original — por lo menos una tensión de 10,5 V para poder generar el ruido suficiente, así que una sencilla fuente de alimentación con una pila de 9 V no suministraría tensión suficiente. Se intentó también con un BC850, que necesita un poco menos de tensión para que el circuito funcionase como fuente de ruido, pero por desgracia también aquí una tensión de 9 V para la fuente resultó ser demasiado baja.

También podríamos haber cambiado completamente de rumbo y diseñar drásticamente otro circuito, con, por ejemplo, un convertidor boost para subir la tensión de alimentación de 9V, o utilizar un diodo zener y calcular a continuación la nueva configuración del circuito, pero la redacción, el departamento de diseño y la imprenta se estaban dando leña el uno al otro por las fechas tope, así que se tenía que tomar una decisión... Por tanto no habría fuente de alimentación de 9 V. Al fin y al cabo, con un adaptador tampoco tienes que tener miedo de que el circuito se desconecte porque la pila esté agotada.

(110280)

Herramientas de desarrollo de bolsillo

¿Qué utilidad tienen las aplicaciones electrónicas para tablet-PCs y smartphones?



La oferta de herramientas de software para desarrollo electrónico es tan amplia como variada. Además, el ordenador no es la única ayuda técnica del electrónico, en el área de los smartphones y los tablet-PCs existe todo un mundo de aplicaciones. Pero queríamos saber con más exactitud: ¿qué se nos ofrece y cómo de útiles son los programas?

Christopher Rausch (Redacción Alemania)

Actualmente no podemos concebir el día a día de un desarrollador sin el software de PC para el diseño y simulación de circuitos electrónicos. Algunos productos como Proteus y Pspice son imprescindibles para la mayoría de lectores. Estos programas generalmente proporcionan resultados muy buenos en la simulación [1], sin embargo su precio suele ser muy elevado y requieren un PC. Junto al software para ordenadores, en Internet también podemos encontrar multitud de aplicaciones electrónicas para smartphones y tablet-PCs. Dichas aplicaciones nor-

malmente son muy asequibles, y en algunos casos incluso gratuitas. Los dispositivos móviles actuales disponen de una gran potencia de cálculo y a menudo, como ocurre en los smartphones y tablet-PCs, se controlan exclusivamente mediante una pantalla táctil. Por ello nos preguntamos: ¿qué hacen estas aplicaciones y qué funcionalidad le ofrecen al usuario final? ¿Qué posibilidades e inconvenientes derivan del concepto de "pantalla táctil" y de las dimensiones relativamente pequeñas en los displays?

Para poder responder a estas preguntas, hemos puesto sobre el banco de pruebas

dos aplicaciones para smartphones y para el iPad de Apple. En cuanto a las de los smartphones, elegimos aquellas que utilizan el sistema operativo Android. Este sistema es muy común y ofrece a los desarrolladores de software la libertad de la cual carece el iPhone, por ejemplo. Mientras que los desarrolladores para productos de Apple no pueden ejecutar su software gratuitamente y es necesario que el fabricante lo pruebe primero sobre hardware real, para los desarrolladores de Android no existen este tipo de barreras. Tampoco hace falta que el propietario venda el software desde la tienda oficial. En nuestros tests nos

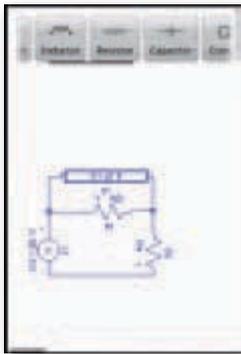


Figura 1. Divisor de tensión con Droid Tesla.

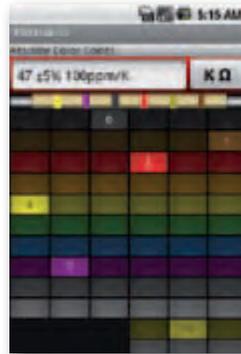


Figura 2. Conversor del código de colores de Electronica.

hemos servido del emulador de Android para Windows (ver cuadro de texto) del SDK (Software Development Kit) oficial de Android [2].

En el cuadro de texto se muestra un resumen de otras aplicaciones electrónicas que no han sido probadas. La elección de estas aplicaciones para smartphones está basada en una lista del experto en Android Stefan Schwark.

Herramientas de desarrollo tamaño bolsillo

La oferta de aplicaciones electrónicas para smartphones con Android abarca desde libros de consulta digitales hasta calculadoras para dimensionar los componentes, así como software para procesar y simular circuitos electrónicos. Frente a otros displays pequeños en comparación, los smartphones ofrecen claridad y utilidad en las aplicaciones. Sin embargo, dada su movilidad están predestinados a servir meramente como medio de apoyo.

La primera aplicación que hemos analizado se llama Droid Tesla [3]. Este software (figura 1) es un simulador de circuitos. Mediante elementos modulares puede diseñarse el diagrama deseado y posteriormente comenzar la simulación. El simulador es capaz de reproducir resistencias, capacidades, inductancias e impedancias complejas, así como medidores tanto de tensión como de corriente. Para alimentar el circuito están disponibles una fuente de corriente y otra de tensión. La tensión y la corriente se especifican mediante sus componentes real e imaginaria, lo cual es relativamente inusual. Lamentablemente no permite representar gráficamente los resultados de la simulación. Los valores correspondientes sólo se pueden medir utilizando los voltímetros y amperímetros virtuales. La utilización de elementos modulares resulta cómoda y ahorra espacio, aunque la conexión de componentes individuales a veces requiere cierta habilidad. El pequeño display de los smartphones en comparación

con un monitor de PC dificulta el diseño y revisión de esquemas con cierto tamaño.

No obstante la mayor parte de aplicaciones electrónicas para smartphones con Android son calculadoras o libros de referencia para componentes o partes de circuitos utilizados a menudo. Al repertorio estándar de estos programas pertenecen los decodificadores del código de color de las resistencias y una calculadora para la ley de Ohm, los divisores de tensión, los circuitos resonantes y otros muchos circuitos básicos en la electrónica.

La segunda aplicación en nuestro test recibe el nombre de "Electronica" [4] y pertenece a esta categoría de herramientas de apoyo electrónicas. Consta de diferentes calculadoras, un intérprete del código (figura 2), así como información sobre integrados y asignación de pines. En cuanto a funcionalidad, este software pertenece a la gama más reducida, pero funciona sin complicaciones y los conduce rápidamente al resultado deseado. A las herramientas individuales se accede mediante un menú de categorías y su diseño goza de gran claridad.

Diseño de circuitos con el iPad

El iPad es un tablet PC –similar a un smartphone– que funciona casi exclusivamente mediante pantalla táctil, y dispone de un display sustancialmente más grande (de 9,7 pulgadas). Para este dispositivo también existe una amplia oferta de aplicaciones electrónicas. El espectro abarca desde aplicaciones de cálculo a bases de datos de componentes, hasta simuladores de circuitos. También aquí se plantea la pregunta de si el concepto de funcionamiento completo mediante la pantalla táctil resulta práctico en el software electrónico disponible para el iPad. Para el test hemos seleccionado y probado dos simuladores de circuitos para éste. Con la ayuda de este software diseñamos pequeños circuitos por nuestra cuenta, los guardamos y posteriormente los simulamos. Finalmente los resultados se muestran a modo de gráfico.

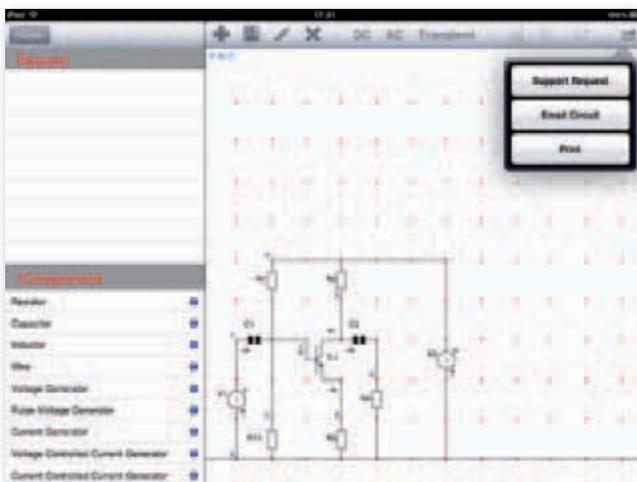
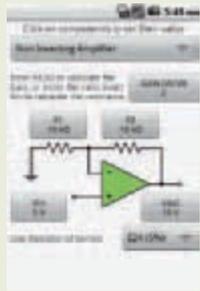


Figura 3. Transistor conectado en el iPad.

La primera aplicación que utilizamos fue el software Circuit Lab HD [5] (figura 3).

Otras aplicaciones para Android e iPad		
		<p>DCircuit Lab HD es el equivalente a Circuit Lab HD para componentes digitales. Pueden analizarse de un vistazo circuitos combinacionales y secuenciales según fabricantes, de forma parcial o al completo (http://sites.google.com/site/lurvill/dcircuitlabhd).</p>
		<p>Elektor Electronic Toolbox: esta herramienta supone una importante ayuda electrónica. Incluye multitud de calculadoras, conversores, hojas de datos de integrados, esquemas de circuitos y asignación de pines en conectores (http://itunes.apple.com/es/app/id339158729?mt=8).</p>
		<p>Ohm's Calculator: esta sencilla calculadora permite realizar cálculos basados en la ley de Ohm. Un sistema alternativo para la entrada de datos aceleraría la entrada de los valores deseados (http://smartchef.org/rantekinc/).</p>
		<p>Scientific Electronics Circuit Calculator for iPad: compilación de calculadoras electrónicas y fórmulas para electrónicos en su trabajo diario. Cada cambio en los valores de los componentes genera un nuevo cálculo, con lo que los resultados se leen rápidamente (www.syclabs.com/CircuitCalciPad.html).</p>
		<p>Elec Ref es una compilación de herramientas de cálculo para cables, transformadores y motores. Entre otros, incluye información y datos de los distintos tipos de cables (www.trunnion.info/eleceref.html).</p>
		<p>Electrical Wiring Pro: esta ToolBox ofrece al usuario información y herramientas en el tema del "cableado". Podemos encontrar una lista completa de lo que incluye en la web del fabricante (www.intineo.com/electrical_wiring_pro.html).</p>
		<p>Audio Test Tone Generator: esta herramienta ofrece un generador de señales senoidales, triangulares, rectangulares y de diente de sierra, así como distintos tipos de ruidos, con salida tanto gráfica como acústica (www.digitalantics.co.uk/android/audio-test-tone-generator).</p>
		<p>ElectroDroid: ElectroDroid ofrece algunas herramientas e información (sobre la asignación de pines según distintos fabricantes) para electrónicos (http://demisoft.altervista.org/_siti_interni/electrodroid/).</p>

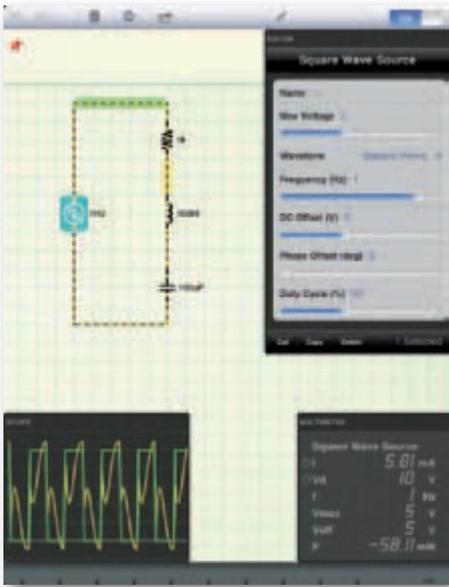


Figura 4. Simulación de circuitos con iCircuit.

El diseño de los esquemas de circuitos funciona, como casi todo en el iPad, mediante la pantalla táctil. Primero especificamos

con los dedos los puntos en los cuales se situarán los componentes. Posteriormente seleccionamos de un menú el componente a situar en el circuito. Al contrario que una solución del tipo “drag and drop”, este método requiere cierta práctica. El abanico de componentes incluye resistencias, condensadores e inductancias, así como elementos semiconductores como diodos, amplificadores operacionales y transistores bipolares. Para pequeños circuitos analógicos esta biblioteca de componentes es más que suficiente. Aparte, también dispone de fuentes de alimentación de corriente continua o alterna (reguladas y sin regular). Los valores de los componentes pueden cambiarse sirviéndonos del teclado virtual del iPad. Una vez terminado el circuito podremos iniciar la simulación, y las distintas ten-

siones y corrientes se mostrarán gráfica o numéricamente. Puede seleccionarse entre simulación DC, AC o transitoria (el desarrollo en un cierto espacio de tiempo). El software representa los resultados como curvas, pudiendo elegirse entre muchos tamaños según las distintas visualizaciones. Desafortunadamente, los diagramas carecen de marcas para la escala, lo cual dificulta la interpretación de los resultados y hace que sea menos precisa.

Simulación divertida

iCircuit [6] es, al igual que la aplicación probada anteriormente, otro simulador de circuitos. Sin embargo, ofrece al usuario la posibilidad de seguir modificando el circuito durante la simulación y ver cómo los cambios afectan a las gráficas actuales. La creación de

Publicidad

Crea sistemas electrónicos complejos en minutos utilizando Flowcode 4



Diseña – Simula - Descarga



Flowcode es uno de los lenguajes de programación gráfica más avanzados del mundo para microcontroladores (PIC, AVR, ARM y, ahora también, dsPIC/PIC24). La gran ventaja de Flowcode es que permite a aquellos que tienen poca experiencia crear sistemas electrónicos complejos en minutos. El interfaz gráfico de desarrollo de Flowcode permite a los usuarios construir un sistema electrónico completo en pantalla, desarrollar un programa basado en diagramas de flujo estándar, simular el sistema y generar el código hexadecimal para microcontroladores PIC, microcontroladores AVR y microcontroladores ARM.



¡NUEVO!
Flowcode 4 para
dsPIC/PIC24

Convencete tu mismo.
Versión de demostración,
más información y pedidos en
www.elektor.es/flowcode

circuitos en iCircuit se hace mediante “drag and drop”, o sea, que se arrastran los componentes correspondientes desde un menú hasta su sitio correcto, y del mismo modo se conectan entre sí. Los valores de los componentes pueden introducirse mediante el teclado virtual del iPad o cambiarse con el dedo en una barra (**figura 4**). El hecho de no tener que detener la simulación del circuito electrónico dota al software de un obvio carácter educativo. Junto a los componentes pasivos estándar, la paleta incluye también interruptores, semiconductores como diodos, transistores bipolares y de efecto de campo, convertidores A/D y D/A así como algunos circuitos lógicos simples. La visualización de los resultados de simulación se hace mediante osciloscopios virtuales. Éstos pueden activarse u ocultarse según los necesitamos. Adicionalmente también puede representarse el flujo de la corriente con velocidad y sentido, así como la distribución de la carga en las pistas del circuito.

Conclusión

El desarrollo y simulación de circuitos en smartphones todavía resulta difícil. El motivo principal es el pequeño tamaño del display en dispositivos móviles, en

Enlaces

- [1] www.elektor.es/100359
- [2] <http://developer.android.com/guide/developing/tools/emulator.html>
- [3] <https://market.android.com/details?id=org.vlada.droidtesla>
- [4] <https://market.android.com/details?id=com.hacsoft.electronica>
- [5] <http://sites.google.com/site/lurvill/circuitlabhd>
- [6] <http://icircuitapp.com/>
- [7] <http://developer.android.com/sdk/index.html>

comparación con un monitor de PC. En esquemas grandes resulta fácil perderse. También su funcionalidad revela algunas carencias. Tampoco hemos de olvidar que la mayoría de software disponible es gratuito. En cuanto a calculadoras y herramientas de ayuda secundaria, los smartphones cumplen mejor su tarea. En este caso el tamaño de los displays es suficiente para mostrar todos los contenidos de forma clara y práctica. Además, gracias a las ventajas que ofrece su movilidad pueden convertirse en una completa ayuda para la marcha.

Las aplicaciones para iPad aquí mostradas han dado buen resultado a la hora de diseñar y simular pequeños circuitos. Sin embargo, no pueden ofrecer las funciones necesarias para desarrollos electrónicos profesionales. Y esto refiriéndonos tanto a los componentes que pueden utilizarse como al análisis de los resultados en la simulación. No obstante, aquí también hemos de

tener en cuenta que este software cuesta una mínima parte de lo que vale el profesional para PC. Con la pantalla táctil el software se vuelve más intuitivo, pero carece de la precisión y exactitud requerida habitualmente a la hora de diseñar los esquemas. Para colegiales, estudiantes y aficionados, iCircuit es perfectamente válido. Gracias a la posibilidad de simulación permanente y la visualización adicional de los flujos de corriente en el circuito, el software es perfecto para experimentar y aprender.

(110230)

Sobre el autor

Christopher Rausch es estudiante del máster en telecomunicaciones (especializado en electrotecnia) en la escuela RWTH de Aachen. Actualmente está realizando unas prácticas en la redacción de Elektor.

Emulador de Android



El emulador forma parte del Android Software Development Kit [7], lógicamente es capaz de simular un smartphone con

Android en el ordenador. Para el desarrollador de software que pretende introducirse en la programación de dispositivos con Android supone toda una ventaja poder probar rápidamente y de forma sencilla sus propios programas sin necesidad de hardware específico.

El usuario puede descargar nuevas aplicaciones mediante un browser preinstalado o instalar en el sistema vía consola de comandos un paquete de Android (archivo APK) preexistente o recién creado. Para ejecutar estos paquetes en el smartphone emulado, los usuarios de Windows tendrán que abrir aparte del emulador una consola de DOS (command.exe) y cambiar con instrucciones en DOS (por ejemplo “cd <pfad>”) el directorio en el cual se iniciará el emulador. La verdadera instalación funciona sólo mediante el comando “adb install <dateipfad>”, implementada directamente desde la consola. Como parámetro hemos de introducir el directorio del archivo con la extensión “.apk” inclusive. Tras la instalación encontraremos el nuevo programa en el menú de aplicaciones del smartphone virtual.

Curso de audio DSP

Parte 1: tratamiento de señales de audio con un DSP

El tratamiento del audio es la disciplina técnica dentro del propio área de tratamiento de señales en la que más se ha progresado en aproximadamente 30 años de aplicación. En este curso no sólo mostramos las características y los métodos de programación de un procesador de audio digital, sino también 3 aplicaciones que se sirven de un circuito procesador de señal asequible pero de gran calidad.

Alexander Potchinkov (Alemania)

Al igual que el amplificador operacional es un componente universal en el tratamiento de señales analógico, el procesador de señales lo es para el tratamiento digital. Esta comparación no lo aclara del todo, pues un procesador de señales digital no sólo tiene más pines de conexión que un operacional, sino que también es mucho más complejo y potente que éste. La serie de artículos comienza con una breve introducción al tratamiento digital de señales analógicas de audio. Sigue la idea del procesador digital DSP56374 de Freescale, en su encapsulado de 52 pines cómodo de soldar, con la introducción sobre cómo adaptar este procesador especial a la tarea para la cual ha sido diseñado, el tratamiento digital de señales de audio. Ofrece una frecuencia de procesador de 150 MHz y una aritmética de señal refinada acorde con su potencia de cálculo, gracias a la cual podremos alcanzar un tratamiento del audio realmente complejo. Junto con la tarjeta DSP desarrollada para esta serie de artículos se presentan tres proyectos prácticos:

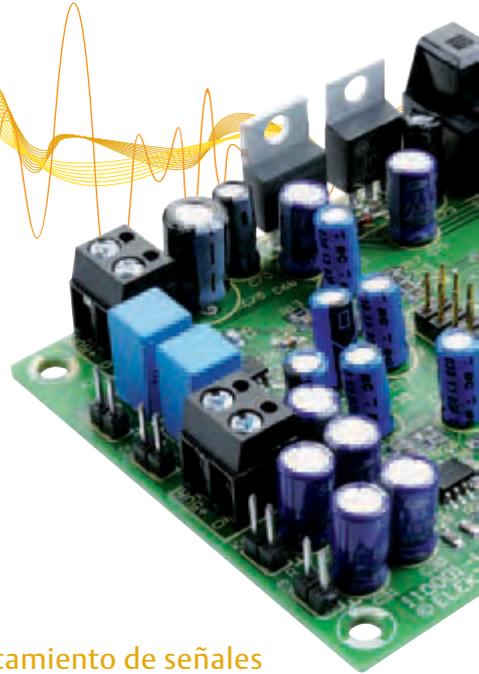
- Generador de audio digital para la obtención de señales senoidales con escasa distorsión, ruido blanco y rosa, así como de terceras y octavas.
- Medidor del valor de pico en la modulación con display LED.
- Procesador dinámico con compresor, limitador y puerta de ruido.

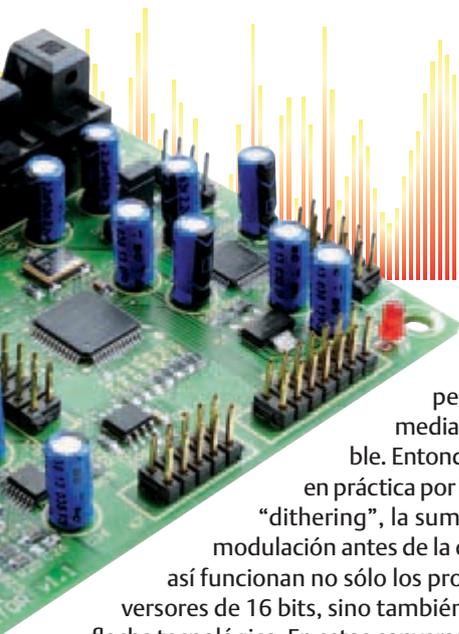
Tratamiento digital de señales de audio analógicas

Un micrófono capta señales analógicas, y un altavoz también funciona siempre con este tipo de señales. Durante décadas la técnica de audio desde el micrófono hasta el altavoz ha sido analógica. Incluso la tecnología HiFi desde los años 60 hasta los 80 ha sido puramente analógica y los resultados obtenidos han sido plenamente satisfactorios. El CD como sustituto del disco de vinilo se introdujo en el mercado en 1982, eclipsándolo por motivos tanto

cuantitativos como económicos. Primero necesitábamos tener lo mínimo en cuanto a tratamiento digital de señales de audio. Hacía falta, dicho de forma sencilla, transformar la señal analógica del micrófono mediante un conversor analógico-digital (ADC), grabar la señal digital obtenida en un CD, leerla con el reproductor de CD y volver a convertirla a señal analógica con un conversor digital-analógico (DAC). Podríamos decir que esta cadena de transferencia tiene etapas de entrada y salida analógicas.

Como el lector ya sabe, el CD funciona con una resolución de 16 bits. No existen tareas triviales, el conversor (tanto el ADC como el DAC) han de tener la suficiente calidad, lo cual no recae únicamente en los conversores en sí mismos, sino que para su funcionamiento requiere de filtros analógicos, el de *antialiasing* antes del ADC y el de *reconstrucción* tras el DAC. Esto ya lo sabían desde hacía tiempo los desarrolladores del CD en Philips, y tuvieron una idea para realizar un inteligente y elegante tratamiento digital de señales. Para decirlo sin complicaciones, Philips logró obtener una calidad de 16 bits con un DAC de 14, y lo consiguió con el sobre-muestreo (*oversampling*) y el *noise-shaping* en un filtro digital previo, o sea, todo mediante auténtico tratamiento digital. Al respecto hemos de mencionar un problema fundamental del tratamiento de señales digitales. El tratamiento de señales analógicas tiene la deseable característica de que con baja modulación las distorsiones no lineales son muy pequeñas, y con alta modulación son grandes. Esto se corresponde con nuestro oído, pues al igual oímos mejor con baja modulación que con alta. En el tratamiento de señales digitales lamentablemente ocurre lo contrario. Con una baja modulación, en proporción a la señal útil tendremos una gran parte de distorsiones no lineales, y con alta modulación una muy baja. Esto ha llevado a ingenieros y matemáticos a intentar que sea lo más "parecido a lo analógico", en el sentido en que con baja modulación las distorsiones no lineales se perciban





peor o sencillamente filtrándolas mediante el rango de frecuencia audible. Entonces llegaron las técnicas puestas en práctica por Philips, complementadas con el “dithering”, la suma de señales de ruido con baja modulación antes de la cuantificación. Por el momento así funcionan no sólo los procesadores de señales con convertidores de 16 bits, sino también los de 24, que son la punta de flecha tecnológica. En estos convertidores casi encontramos por completo tratamiento de señales de audio digital, y sólo una pequeña parte analógico.

Procesadores digitales de señal

Un DSP es un tipo especial de procesador que entre sus características incorpora muchas de las típicas exigidas para el tratamiento

digital de señales. Éstos son capaces de procesar en tiempo real algoritmos repetitivos de forma estructurada y bastante sencilla, que generalmente tienen multitud de pasos con distintos cálculos. El reloj de proceso de los algoritmos es la frecuencia de muestreo de las señales digitales, que por ejemplo en el CD se especifica de 44,1 kHz. Un algoritmo DSP ha de implementarse por completo en aproximadamente 22,7 μ s, lo necesario para el buen funcionamiento de un puerto de audio.

Arquitectura del procesador

La figura 1 muestra el diagrama de bloques del DSP56374. La arquitectura del procesador muestra multitud de buses de datos y direcciones, así como muchos segmentos de memoria en los que podemos reconocer una *arquitectura Harvard*, en la que pueden hacerse varias lecturas de la memoria a la vez. Los DSPs son procesadores de

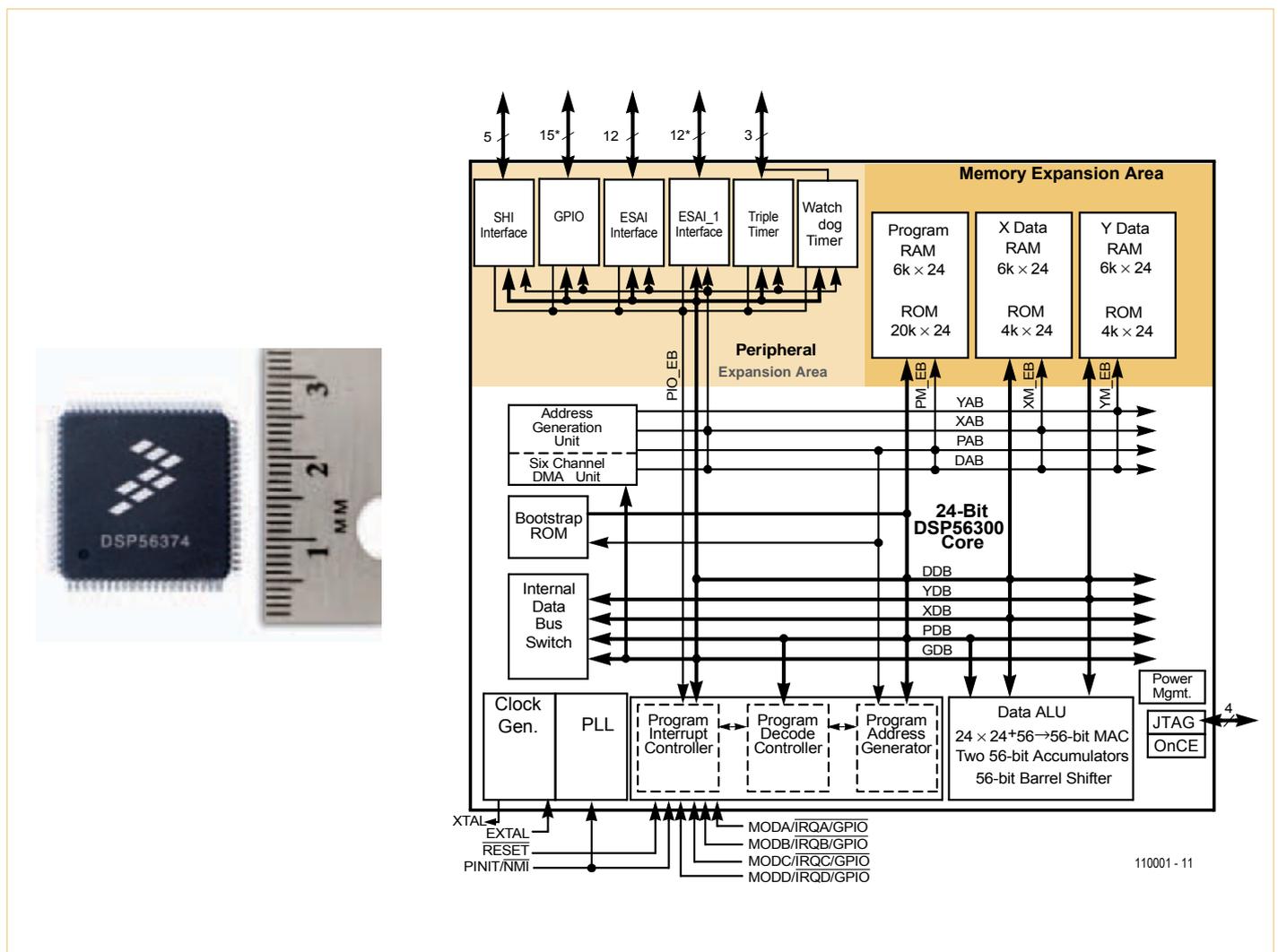


Figura 1. Diagrama de bloques del procesador digital de señal DSP56374.

registros. Los operandos y resultados de la mayoría de cálculos aritméticos se almacenan en registros y se leen de éstos. Por un lado, permite operaciones simultáneas sin microprogramas, y por otro las rutinas de carga y escritura necesarias entre las áreas de memoria y registros, que debido a la arquitectura de hardware y las *Address-Generation-Units* (AGUs) es capaz de realizar múltiples operaciones

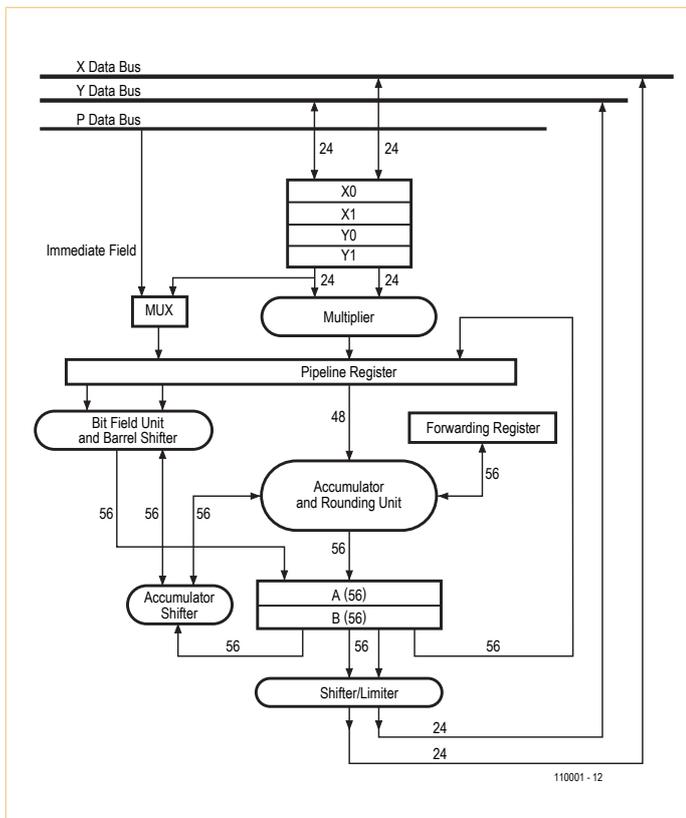


Figura 2. Vía de datos del procesador de audio Symphony.

aritméticas al mismo tiempo. Las AGUs son calculadoras que obtienen las direcciones de memoria según los modos típicos de direccionamiento DSP independientemente del tratamiento de los datos. El bloque SHI (*Serial Host Interface*) se trata de un puerto síncrono serie, que puede funcionar en modo SPI o I²C. El DSP puede trabajar en ambos modos, tanto master del bus como slave. En este puerto puede conectarse por ejemplo un microcontrolador. El bloque GPIO, *General Purpose Input Output*, ofrece conexiones para tareas de propósito general, como aquellas que pueden encontrarse en todo procesador. Ambos bloques ESAI y ESAI_1, *Enhanced Serial Audio Interface*, proporcionan seis puertos de datos para audio junto a las conexiones necesarias para la sincronización. En el DSP de 52 pines DSP56374 el bloque ESAI_1 no está disponible. Los tres temporizadores del DSP y el de watchdog, mediante el cual puede comprobarse si el DSP está realmente trabajando o se encuentra

Filtro FIR con el DSP56374

El formato numérico para representar las señales del DSP se trata de dos *decimales con complemento a dos o decimales cortos*, lo cual en cifras binarias corresponde a que la coma esté directamente tras el bit de signo. Los coeficientes del filtro, que siempre son decimales, se leen en words de 24 bits desde la memoria del DSP, y determinan las características del filtro. También existe otro software de diseño de filtros que no sólo permite hacerlos de esta clase, sino también los llamados filtros IIR. Aparte, también hay filtros con realimentación de la señal, aptos para el diseño directo de los famosos filtros analógicos.

La figura 3 muestra el diagrama de flujo de la señal para el filtro, con N=5. Se trata de una representación gráfica de la ecuación en diferencias

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + b_3x(n-3) + b_4x(n-4),$$

que nos da el método de cálculo del filtro.

La cajita cuadrada marcada con z-1 simboliza el almacenamiento de un valor de señal de 24 bits. El triángulo representa la multiplicación de una señal por un factor de 24 bits, en el caso de un coeficiente de filtro. El círculo con el símbolo (+) simboliza la suma de dos valores de 48 bits.

La ventana temporal de valor n contiene a la vez cinco valores la señal de entrada, x(n), x(n-1), x(n-2), x(n-3) y x(n-4), que se generan, por ejemplo, mediante un conversor ADC con words de 24 bits. x(n) es el valor actual de la señal (muestra), los valores x(n-k) son los intervalos consecutivos separados o valores anteriores de la señal. El cálculo de la señal de salida real y(n) requiere cinco operaciones MAC así como la inicialización del acumulador a cero. Han de seguirse los pasos siguientes, para lo cual puede utilizarse un bucle continuo:

```

a := 0;
a := a + b0 * x(n)
a := a + b1 * x(n-1)
a := a + b2 * x(n-2)
a := a + b3 * x(n-3)
a := a + b4 * x(n-4)
    
```

“bloqueado”, han de obtenerse de tres conexiones externas. El grupo *Memory Expansion Area* incluye tres bancos de RAM, X e Y para datos, así como P-RAM para el programa. Las unidades ROM no son importantes para nosotros. Pueden configurarse los tamaños de las particiones individuales X, Y y P. El bloque AGU contiene las dos unidades computacionales verdaderamente potentes y flexibles para calcular dos direcciones en un único ciclo de procesador. La ROM de Bootstrap incluye el código para arrancar el DSP. Éste puede arrancar por ejemplo en serie mediante el SHI. Con ambos bloques *Clock Generator* y *PLL* se obtiene el reloj del procesador. El DSP permite utilizar un generador de reloj con menor frecuencia, que puede multiplicarse hasta una frecuencia máxima de 150 MHz en el procesador. Mediante el bloque *Program Interrupt Controller* se gestiona el número de interrupciones internas. Muchas fuentes de interrupciones tienen su origen en los puertos. A éstos les siguen los

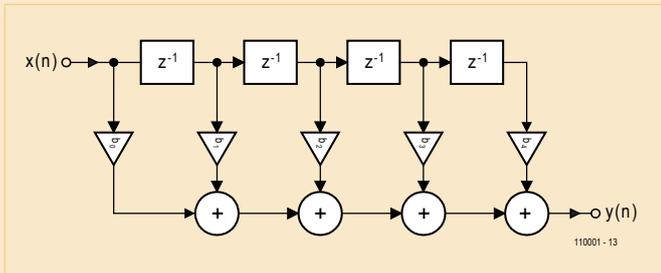


Figura 3. Diagrama de flujo de la señal para el filtro FIR.

Antes de analizar el código DSP, primero hemos de explicar los dos sectores de memoria necesarios. En la RAM X se almacenan con direccionamiento lineal los coeficientes hasta $N=5$ con una dirección base aleatoria. En la RAM Y se encuentran siempre se encuentra la ventana temporal actual. El direccionamiento del buffer circular se gestiona con un cálculo del módulo N. Se llama *cbase* a la dirección base del rango de memoria para coeficientes y *dpntr* a una célula de almacenamiento especial de la RAM Y, que alberga puntero al área de entre las distintas llamadas del filtro. Ésta célula contiene al principio la dirección base del rango de memoria para datos.

```

    move    #N-1,M4                ; modulo N indexing
    ...
Audioloop
    ...
    movep  x:RX0,y1                ; read audio sample
    move   #cbase,R0               ; set coefficient pointer
    move   y:dpntr,x0
    move   x0,R4 ; load data pointer
(1)  clr  a x:(R0)+,x0 y1:y:(R4)+ ; read 1st. coeff, save 1st data
(2)  rep  #N-1
(3)  mac  y1,x0,a x:(R0)+,x0 y1:y:(R4)+
(4)  macr y1,x0,a (R4)-           ; requantisation
    move  a,x:TX0                 ; write audio sample
    move  R4, x0
    move  x0, y:dpntr             ; save data pointer
    jmp  Audioloop

```

En la instrucción (1) no sólo se inicializa el acumulador a cero, sino que también se mueven dos datos y se incrementan los punteros (con el puntero R4 también se realizan cálculos de módulo). El coeficiente de filtro b_0 se guarda en el registro x_0 y los datos de audio actuales en el buffer circular. Para las siguientes instrucciones *mac* podrán combinarse los coeficientes del filtro con dichos datos actuales, que se encuentran en el registro y_1 . La instrucción (2) inicia un bucle de hardware cuyo contenido se ejecuta $(N-1)$ veces. Con la instrucción de cuenta de bucles (3) se calcula y acumula un producto parcial, y además se escriben los coeficientes y datos para el próximo producto en los registros x_0 e y_1 . El resultado de estos productos parciales se deposita en el acumulador, a modo de word con un tamaño de 56 bits. La instrucción *macr* de (4) implementa el cálculo y acumulación del producto parcial y recuantifica el resultado de 56 bits a 24, como lo necesita el DAC. Al mismo tiempo el puntero al área de datos se disminuye en 1.

bloques de control de ejecución en el programa. A la derecha encontramos el bloque *Data ALU* o “vía de datos”. Aquí es donde se lleva a cabo el tratamiento digital de señales. Finalmente encontramos el bloque *JTAG/ONCE*, mediante el cual, utilizando un software en el PC (el debugger), pueden cargarse o depurarse programas DSP.

Vía de datos

La *vía de datos* está representada en la **figura 2**. El corazón de dicha vía de datos es la unidad MAC. MAC significa *Multiply And Accumulate*, donde se hacen cálculos del tipo $a:=a+x*y$. Este cálculo es uno de los más importantes en el tratamiento digital de señales, gracias al cual se generan los filtros digitales, entre otros. El acumulador almacena el producto $x*y$, para lo que hace falta un multiplicador, un sumador y un registro acumulador con sus correspondientes conexiones y longitud de palabra. En la figura pueden verse los dos

operandos para el desarrollo del producto mediante dos buses de datos (buses X e Y), puede cargarse en cuatro registros de 24 bits, x_0, x_1, y_0 e y_1 (también se refieren a él como pequeño *Register File*). El DSP funciona mediante dos registros acumuladores a y b , que están conectados con los buses. Los resultados intermedios de cada acumulación no se almacenan en la memoria. Si ha de implementarse una de estas operaciones MAC en el algoritmo de un filtro digital, se le permite al DSP obtener los valores de la señal y los coeficientes de filtrado para el próximo cálculo en la operación MAC siguiente, a la vez que se desarrolla la operación MAC actual. En la imagen puede verse cómo el DSP opera con datos de 24 bits y no sólo utiliza resultados de las multiplicaciones de 48 bits sin perder la cuenta de la acumulación, sino que su registro acumulador de 56 bits, con 8 bits adicionales, resulta de especial ayuda, permitiendo cifras de mayor tamaño.

El registro de operandos está conectado mediante dos vías de 24 bits con el multiplicador fraccional por hardware (“Fractional-Hardware-Multiplier”), que implementa las multiplicaciones en cada ciclo del procesador. En nuestro caso se trata de resultados de multiplicación de 48 bits, que o bien son procesados con la ALU en la *Accumulator and Rounding Unit* o se almacenan mediante los buses de datos en la memoria del DSP. La *Accumulator and Rounding Unit* sirve ante todo para acumular, y forma junto con el multiplicador la unidad MAC. El procesador de esta unidad también puede utilizarse para sumar y restar señales o para las operaciones lógicas AND, OR y XOR. Mediante la unidad *Bit Field Unit and Barrel Shifter* se ejecutan operaciones con bits agrupados y conversiones de multibits, que siempre se ejecutan en un ciclo de procesador. El shifter y el limitador, que pueden verse en la imagen bajo el registro acumulador, sirven por ejemplo para recuantificar los resultados de multiplicación para una palabra de 24 bits del DSPs. Finalmente hemos de decir que el DSP puede realizar cálculos con operandos “longword”, para lo cual han de combinarse en cada caso los dos registros de 24 bits x_0 y x_1 , junto con y_0 e y_1 , para dar los registros de 48 bits x e y . Una instrucción

```
add x,b
```

da lugar a una suma de 48 bits, y la instrucción

```
add a,b
```

a una de 56 bits.

Para programar la vía de datos ha de darse otra indicación. Los comandos `mac` y `mpy` (`mpy` significa multiplicación) son las llamadas instrucciones de tres direcciones. Mediante

```
mac x0,y0,a 0      mpy x1,y1,b
```

se especifican tres direcciones, las de ambos registros del operando y la del registro del resultado. La suma y la resta son instrucciones de dos direcciones. Mediante

```
add x0,a      0      sub y0,b
```

se carga uno de los dos operandos en ambos registros acumuladores, ya que ninguna de las dos entradas de la ALU en la *Accumulator and Rounding Unit* del registro de operandos puede ser utilizada. El operando en el registro acumulador se sobrescribe con el resultado del cálculo. Dada esta rutina de carga han de ejecutarse dos instrucciones en cada caso para realizar las instrucciones de dos direcciones.

Algunas técnicas especiales de cálculo, como la aritmética de saturación (saturando al valor máximo y al mínimo del rango), servirán para asegurarnos de que el DSP puede comportarse como si estuviese sobrecargado, al igual que los aparatos analógicos a los que nos habíamos. Este tipo de comportamientos han de modelarse en otro procesador con software.

Control y Address Generation Unit

Para incrementar la utilidad de los DSPs también existen tareas especiales. Al respecto cabe mencionar ante todo el bucle de hardware “zero overhead loop”, cuyos bucles de cuenta son capaces de trabajar sin apenas instrucciones casi. Siendo específicos, esto significa que sólo son necesarios unos pocos ciclos del procesador para ejecutar el bucle de hardware, y dentro de este sólo está incluido el tratamiento de la señal. El bucle de hardware con una sola instrucción (por ejemplo la aritmética de la señal con posible movimiento simultáneo de datos) es necesario para el filtro FIR, cuyo programa DSP analizaremos más tarde. Aquí los requerimientos de inicialización del bucle son especialmente bajos, pudiendo utilizarse casi toda la potencia de cálculo del DSP para el filtro FIR. Con una frecuencia de muestreo de 48 kHz y una de procesador de 150 MHz puede ejecutarse un filtro FIR con una longitud de 3000. Esto no sólo requiere la posibilidad de darse transferencias de datos simultáneas, sino también tener el direccionamiento de la memoria listo en segundo plano sin utilizar la vía de datos.

Para el direccionamiento indirecto mediante punteros, el DSP cuenta con ocho de ellos, de R0 a R7, con registros. Las características de la aritmética del direccionamiento y las separaciones entre incrementos y decrementos para cada puntero se fijan en registros funcionamiento, N y M (llamados *registro de offset* y *registro modificador*). Los 24 registros de la AGU son de 24 bits y también pueden utilizarse para resultados intermedios o llevar la cuenta de la duración de un bucle.

En el registro N pueden albergarse valores para incrementar o disminuir los punteros. De este modo por ejemplo podemos leer de un buffer circular N pulsos de reloj previos al valor de la señal, y así implementar un retardo de ésta, entre otros para compensar las diferencias de periodo entre los altavoces de un conjunto múltiple y el oyente.

El registro M sirve para definir la clase de aritmética de punteros utilizada. Aparte, el DSP dispone junto al direccionamiento lineal también del típico modo DSP por módulo y bit-reverse. El direccionamiento por módulo lo necesitamos para el buffer circular, en el cual mediante filtros digitales se actualiza con cada ciclo del reloj los cortes temporales de la señal (ventana temporal). Una aritmética de módulo N está limitada a N números, de 0 a N-1, que podemos imaginarnos ordenadas en un anillo.

Interrupciones

Las interrupciones de hardware se disparan desde el núcleo del procesador DSP, las etapas DSP y otras fuentes externas, y las interrupciones de software desde el programa DSP en ejecución. En muchos casos éstas pueden etiquetarse con distintos valores de prioridad, para separar las tareas importantes de las triviales.

Una gran diferencia del GPP (General Purpose Processor) es que en la tabla de vectores de interrupción no se encuentran las direcciones de inicio de la rutina de servicio de interrupción, Interrupt-Service-Routine o ISR, sino que para cada interrupción existen dos comandos. Este hecho hace posible establecer dos clases de interrupciones:

Sobre el autor

Alexander Potchinkov es catedrático del área de tratamiento digital de señales en la Universidad Técnica de Kaiserslautern y dirige una empresa de ingeniería para el tratamiento de señales de audio. Junto

a los DSPs y sus algoritmos también ocupa su tiempo con los amplificadores de válvulas y la simulación en SPICE.

- interrupción rápida, con una o dos instrucciones, sin comando de retorno
- interrupción larga, que provoca un salto dentro de una ISR de cierta duración, ejecutándose en lugar del programa actual.

Una de las tareas especialmente importantes de un DSP es leer datos de otros puertos. Lo necesario para esto es la correspondiente instrucción `MOVE` y la dirección de destino en la RAM del DSP, que puede procesarse como una interrupción rápida. Esto evita el back up necesario del estado del procesador en el stack.

Puerto de audio del DSP56374

El puerto de audio incluye seis líneas de control y sincronización, así como seis de datos. La tasa del reloj para enviar y recibir puede ser distinta. Entre otros, podemos activar el formato de audio más conocido, el I²S. Pueden utilizarse hasta seis salidas de audio y hasta cuatro entradas, mientras que la suma de ambas no sea mayor que seis. A la vista del programador sólo están seis registros de 24 bits en

la parte emisora, en la cual se escribe la señal de audio sincronizada según la tasa del reloj en el programa DSP y cuatro registros de 24 bits, de los cuales se lee dicho programa.

Próximamente...

Con esta descripción de la construcción interna del DSP damos por terminada la primera parte de nuestro curso de AudioDSP. En la próxima parte abordaremos la programación de la familia DSP563xx.

(110001)

Enlace:

www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DSP56374

Publicidad

DVD-ROM Elektor 2010

➔ Todos los artículos del año 2010

El DVD Elektor 2010 contiene todos los artículos publicados durante el año 2010 en las ediciones en español, inglés, francés, holandés y alemán de la revista Elektor en formato PDF. Los artículos conservan el formato original con el que aparecieron publicados en la revista y se visualizan con Adobe Reader.

El DVD incluye un motor de búsqueda exhaustiva para localizar palabras clave en cualquier artículo.



Nuevo



Incluye artículos
en español

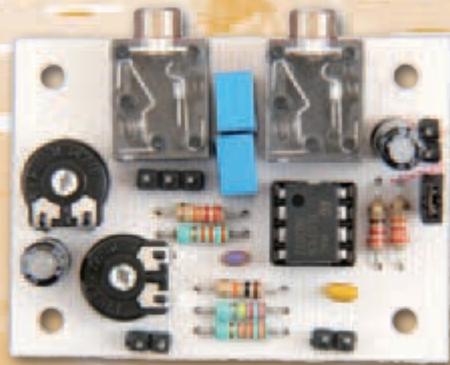
ISBN 978-90-5381-267-9 • 27,50 €

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

Más información y pedidos en www.elektor.es/tienda

¿Hola? ¿Puedes oírnos?

Un mejor entendimiento en las conferencias online



El mundo online se hace cada vez más grande, en tanto que la sensación de distancia se reduce cada vez más. Chatear (por video) con los hijos (o los padres) que vivan lejos, reuniones internacionales donde el departamento de producción en Taiwán pueda conversar con la oficina central de Londres a través de una conexión online... Esto empieza a ser la cosa más normal del mundo. Para reuniones con pocos participantes, basta generalmente con una webcam o un portátil con un micrófono incorporado, pero para grupos más grandes o en espacios acústicamente menos óptimos (con, por ejemplo, mucho eco) el grado de inteligibilidad es generalmente muy bajo, especialmente en el otro lado de la línea. Para solventar este problema hemos creado este sencillo diseño.

Thijs Beckers (Redacción Holanda)

Cada vez hay más empresas que utilizan (video) conferencias online, y Elektor no es una excepción. El problema aquí suele ser

la inteligibilidad. Cuando hay un gran grupo delante del micrófono (del portátil), puede ocurrir que los colegas online no puedan

seguir bien la conversación. Por supuesto que la calidad del micrófono juega un papel importante, pero en espacios más grandes

Características

- Utilizable con cualquier PC o portátil con entrada de línea o de micrófono.
- Fácil manejo.
- Los dos módulos son fáciles de interconectar.
- El patrón de registro de los micrófonos se puede adaptar a todas las circunstancias.
- Construcción sencilla con componentes estándares.

que no son óptimos acústicamente (hay mucho eco), es mucho más acertado trabajar con varios micrófonos. Y Elektor no sería Elektor si nosotros mismos no intentásemos resolver este problema. Este artículo describe el primer resultado: un sencillo sistema de reuniones.

¿Qué es lo que hace?

Vamos a trabajar con micrófonos. O mejor dicho: con cápsulas electret, ya que son fáciles de adquirir y no son tan caras como los micrófonos “decentes”. Además son más aptas para el objetivo que perseguimos. Para nuestra aplicación nos pareció acertado utilizar un patrón de registro bidireccional. Así pudimos colocar el micrófono en el centro de la mesa de reuniones y registrar a cuatro personas al mismo tiempo, mientras se evitaba el resto del ruido emitido hacia el micrófono (ver figura 1) de la mejor forma posible. Para eso hicimos uso de un “truco”: montando dos cápsulas microfónicas con una característica omnidireccional [1] de espaldas una contra la otra y amplificando sus señales de forma invertida, obtuvimos el patrón de registro deseado.

Ya describimos hace tiempo esta configuración en Elektor, fue en el número 3 de 2003 [2]. Ahí hablamos también de la distancia entre cápsulas: No puede ser demasiado grande. Esta distancia determina la frecuencia de corte donde la amplificación empieza a disminuir a mayor frecuencia. Si esta distancia es igual a la mitad de la longitud de onda de la frecuencia, entonces las señales eléctricas de las cápsulas se encuentran desplazadas en fase exactamente 180° la una respecto a la otra. El amplificador restador sigue todavía amplificando la señal. La amplificación disminuye si aumenta la frecuencia y la distancia entre las membranas es superior a la mitad de la longitud de onda. Cuando se encuentran las membranas a una distancia de 2 cm, por ejemplo, la frecuencia de corte asciende a:

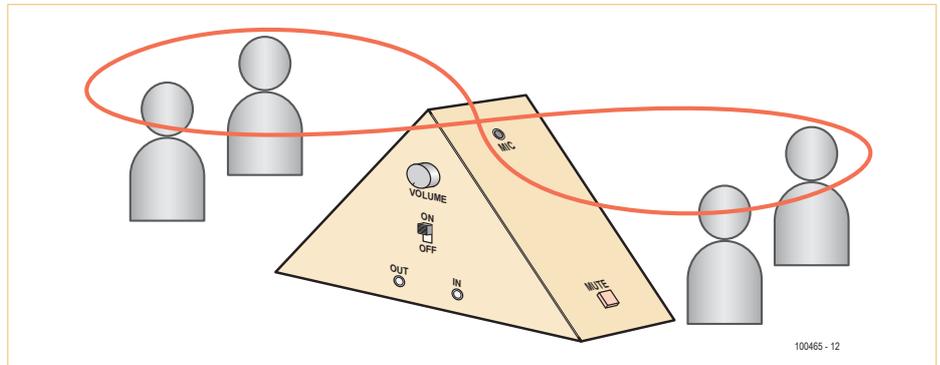


Figura 1. Para concentrar al máximo el sonido registrado de las personas que hay en la mesa, y evitar otros sonidos indeseados y ecos, hemos construido nuestro propio micrófono con la característica bidireccional.

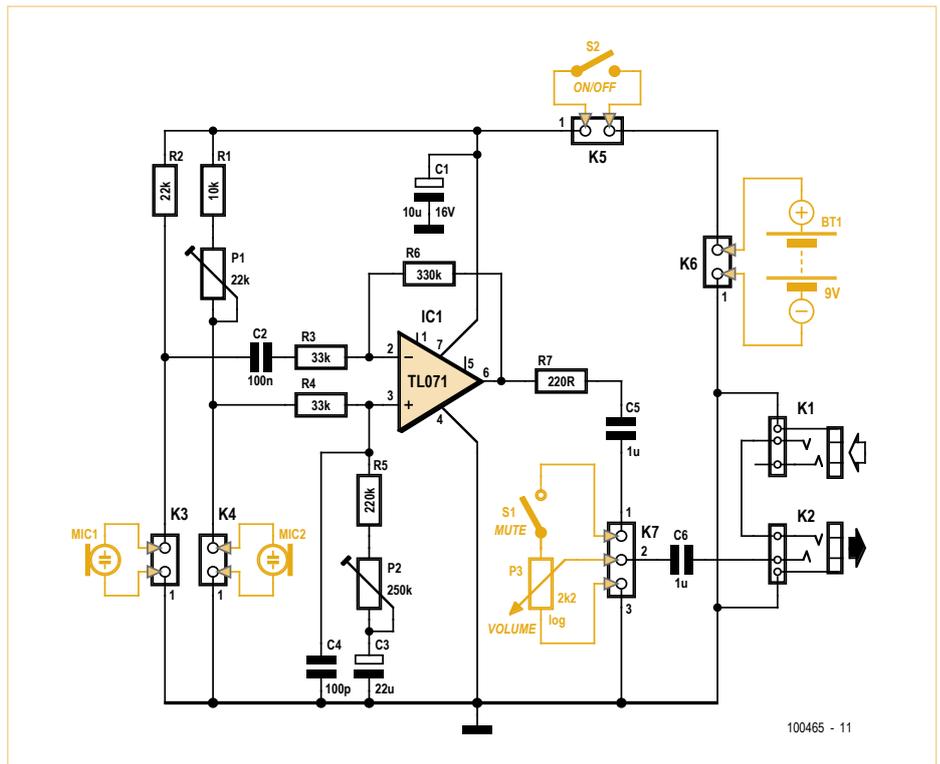


Figura 2. El esquema de nuestro sistema de reuniones contiene solo los componentes más necesarios. Casi no se puede ser más minimalista.

Productos y servicios de Elektor:

- Placa nº. 100465-1 disponible a través de www.elektor.es/100465
- Toda la información está disponible en www.elektor.es/100465
- Los ficheros pdf y Eagle del diseño de la placa impresa, se pueden descargar de forma gratuita en www.elektor.es/100465



nos causa problemas

No hemos probado todo el software de chateo o de conferencia, pero resulta que hay diferencias entre la señal de sonido que utiliza el software. Windows Live Messenger suma correctamente las señales de la izquierda y de la derecha, mientras Skype sólo transmite la señal del canal de la izquierda. De esta forma sólo se habría transmitido en nuestro circuito una única señal de micrófono, mientras que se entregan dos.

Para permitir que nuestro circuito pueda funcionar con dos módulos de micrófono y Skype, hemos construido un “cable Skype” lo más sencillo posible – del mismo estilo que el del diseño del circuito. En la figura 3 puedes ver como lo hemos hecho.

Mediante dos resistencias de 470 ohmios superponemos ambas señales y las enviamos mezcladas por el cable. En el lado del PC o portátil conectamos únicamente la señal mezclada a la entrada del canal de la izquierda.

Ambos circuitos se influyen mutuamente en alguna medida, pero en la práctica no se notará casi nada. Por amor a la sencillez, nos parece una solución perfecta.

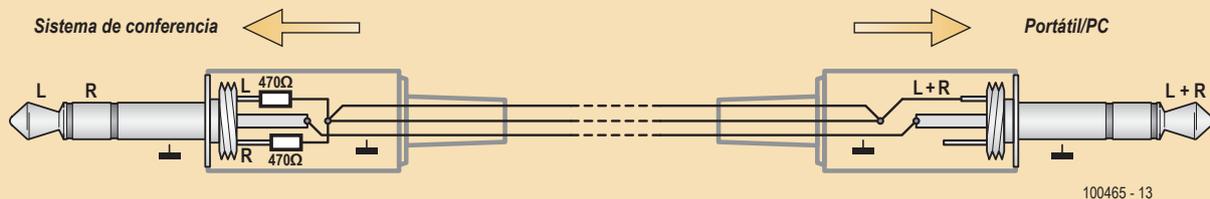


Figura 3. El “cable Skype” realiza una sencilla suma de la señal del canal izquierdo y derecho.

$$f = \frac{v}{\lambda} = \frac{330 \text{ m/s}}{0,04 \text{ m}} = 8,25 \text{ kHz}$$

(v = velocidad del sonido en el aire)

Esto es más que de sobra para el habla. Para mantener un diseño sencillo, optamos, en primera instancia, por la conexión de (un máximo de) dos módulos idénticos. Un módulo se conecta al otro con un conector mini-jack, y el otro se conecta a su vez a la entrada de línea del PC o sobremesa con otro conector mini-jack. Después de esto, resultaría muy práctico poder ajustar el volumen. Por eso incluimos un sencillo potenciómetro en el diseño. Tampoco podía faltar el botón mute — para cuando quieras discutir algo sin molestar en la reunión.

¿Cómo funciona?

El punto de partida del diseño de este circuito fue un sencillo esquema con el que hacer fácilmente este circuito. Efectivamente ha quedado sencillo: ver figura 2. Tan sólo un único operacional y un puñado de resistencias, condensadores y conexiones. De todas formas vamos a repasarlo contigo.

Las cápsulas microfónicas se conectan a través de conectores. Así puedes montar fácilmente las cápsulas en un lugar de la caja. En el croquis de la figura 1 puedes observar que hemos colocado las cápsulas lo más alto posible en una caja triangular (en la parte posterior hay otra cápsula a la misma altura). Lo mejor es colocarlas lo más cerca posible unas de otras. Para proveer los FET de las cápsulas de tensión, las salidas de las cápsulas están conectadas a la tensión de alimentación a través de R1, R2 y P1 según las reglas. Los valores se han elegido de tal forma que aproximadamente la mitad de la tensión de alimentación se encuentra en la entrada del operacional. Esto no es tan importante para la señal de salida del MIC1, ya que ésta se desacopla en continua en la entrada del amplificador con C2. Es tanto más importante para la señal de salida de MIC2, ya que con esto se determina la configuración de la tensión continua de todo el circuito. Hay que ajustar P1 de tal forma que la tensión en la salida del operacional sea igual a la mitad de la tensión de alimentación, es decir 4,5 V.

Con las resistencias R3...R6 y P2 se determina el factor de amplificación del operacional. Nosotros optamos por una ampliación de 10 veces, que en la práctica resulta ser

más que suficiente. La fórmula de la amplificación es la siguiente:

$$V_{\text{out}} = -v_1 \left(\frac{R_6}{R_3} \right) + v_2 \left(\frac{R_5 + P_2}{R_4 + R_5 + P_2} \right) \left(1 + \frac{R_6}{R_3} \right)$$

donde v_1 y v_2 representan las tensiones de las entradas del operacional.

Tal y como se puede observar ya en la fórmula, P2 influye sobre el factor de amplificación aplicado sobre la señal de MIC2. Con esto se pueden anular las diferencias entre las cápsulas y variar la característica del micrófono formado por dos cápsulas desde una característica cardioide o supercardioide hasta un patrón de registro bidireccional. Si las cápsulas fueran exactamente iguales y P2 + R5 fuera exactamente igual a R6, entonces surgiría una característica bidireccional perfecta. C4 evita tendencias de oscilación del operacional y R7 protege la salida de un cortocircuito. C5 desacopla el potenciómetro de volumen de la tensión continua que se encuentra en la entrada del operacional. También para el potenciómetro del volumen (P3) se ha optado por un conector, de modo que sea más fácil ubicarlo en una caja. Entonces el botón mute se coloca directamente en serie con el conector.

Para interconectar los módulos, se han interconectado K1 y K2 de tal forma que la señal de salida de un módulo siempre se encuentre sobre la punta del conector jack y la señal del otro módulo pasada por el primero siempre se encuentre sobre el anillo de dicho conector. Así no importa cómo se conectan los módulos y siempre podrás encontrar las señales de los módulos en los canales de la izquierda y derecha de la salida.

Para limitar la cantidad de cables que se encuentran sobre la mesa de reunión, hemos optado por una pila de 9 V como alimentación del circuito. El consumo del circuito sólo asciende a unos 2 mA, así que con una pila recargable de 9 V de 250 mAh aguantará unas 125 horas.

¿Cómo se construye?

Hemos diseñado una pequeña placa (ver figura 4) para el circuito. El diseño está disponible en formato pdf y eagle en el sitio Web de Elektor [3]. La construcción es cosa de coser y cantar. Los conectores K3...K7 son opcionales; por supuesto que puedes soldar los hilos directamente a la placa.

La construcción mecánica de este proyecto sí requiere alguna atención especial. La fijación de las cápsulas microfónicas, en especial, puede resultar un poco más difícil. Habrá que fijarlas de forma “desacoplada”, ya que por el contrario transmitirían demasiado sonido de contacto; en otras palabras: cualquier golpe en la mesa se amplifica no solo por el desplazamiento “normal” del aire sino también a través del sonido que produce el contacto, lo que resulta ser una influencia bastante negativa para la calidad del sonido/inteligibilidad. Una buena solución es cubrir las cápsulas con gomaespuma y pegarlas luego en la caja. La caja triangular presentada en la figura 1 era una idea que teníamos, pero no lo hemos calculado. Nuestros prototipos consistían en dos tablas de madera fijadas una a otra en forma de T invertida, donde las cápsulas micro estaban montadas en un trozo de gomaespuma sobre la tabla que estaba colocada verticalmente.

Conectar y ajustar

Ajusta la tensión de salida del operacional con ayuda de P1 a la mitad de la tensión de

Lista de materiales

Resistencias:

R1 = 10 k Ω
 R2 = 22 k Ω
 R3, R4 = 33 k Ω
 R5 = 220 k Ω
 R6 = 330 k Ω
 R7 = 220 Ω
 P1 = 22 k Ω de ajuste, horizontal
 P2 = 250 k Ω de ajuste, horizontal
 P3 = 2k2 log

Condensadores:

C1 = 10 μ F
 C2 = 100 nF
 C3 = 22 μ F
 C4 = 100 pF
 C5, C6 = 1 μ F

Semiconductores:

IC1 = TL071

Varios:

K1, K2 = conector jack hembra de 3,5 mm, Lumberg 1503 09, por ejemplo Farnell # 1243244
 K3...K6 = conector de 2 vías
 K7 = conector de 3 vías
 Conector hembra para K3...K7
 S1 = pulsador cerrado en reposo
 S2 = interruptor de vía única
 Cápsula microfónica electret, por ejemplo Panasonic MCE-2020U ó WM-61A
 Placa EPS 100465-1, ver www.elektor.es

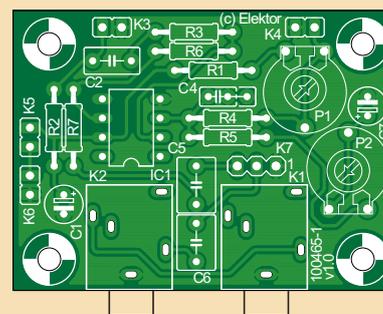
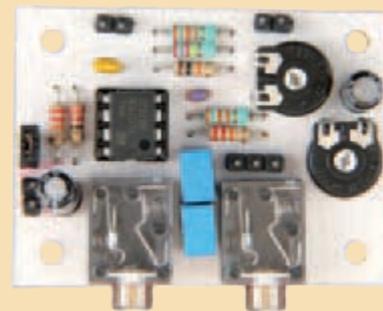


Figura 4. A un circuito sencillo le corresponde una placa sencilla.



alimentación (4,5 V). La característica de direccionamiento de las cápsulas combinadas se puede variar con P2 entre la forma cardioide o supercardioide y bidireccional. La forma bidireccional es la mejor en situaciones como las de la figura 1, un característica cardioide o supercardioide es mejor cuando solamente hay que captar el sonido de un lado del módulo.

Interconecta los módulos utilizando el conector K1 de un módulo y el K2 del otro. Después, puedes conectar el módulo del que aún queda libre K2 con la entrada de línea del PC o del portátil. No olvides indicar que utilizas la entrada de línea como fuente de entrada del sonido en el programa con el que mantienes la reunión online.

Aparte de eso, en las pruebas con el popular programa Skype, resultó que el programa no sumaba las entradas de línea, sino que

sólo utilizaba el canal de la izquierda, mientras otros programas como es el Windows Live Messenger suman y transmiten correctamente las señales de los canales de la izquierda y de la derecha. Con nuestro circuito sólo se habría transmitido el sonido de un módulo. Para solucionarlo hemos construido un cable Skype (ver el recuadro).

(100465)

Enlaces Web

- [1] <http://es.wikipedia.org/wiki/Micrófono>, y hacer scroll hacia abajo)
- [2] www.elektor.com/010123
- [3] www.elektor.es/100465

GSM, SMS, CALL IDentificación

El ATM18 nunca
más estará solo...

¡TIENES
UN
MENSAJE!

Grégory Ester (Francia)

«Controlar a distancia», tres palabras que constituyen muy a menudo un titular muy atractivo y con razón. Deseamos dominar la máquina, controlar la situación, nos sentimos muy poderosos. Con este montaje, nos bastará con, bien mostrar la “tarjeta de paso” gracias a la presentación de nuestro número de teléfono al ATM18, o bien enviar directamente un SMS a nuestra placa preferida con el fin de que actúe en consecuencia.

La bella (ATM18) y la bestia (GM862-GPS)

El GM862-GPS de Telit [1] ha sido concebido idealmente para la realización de aplicaciones de control a distancia. El módulo integra GPS en 20 canales SIRF III, así como un modem GSM/GPRS de cuatro bandas, con alojamiento para insertar una tarjeta SIM. Habrá que añadirle una antena para la parte GPS (hay que señalar que la función GPS no es utilizada aquí) y una segunda antena para el GSM. En Internet [2] hay disponible una placa de prueba o de evaluación, especialmente concebida para recibir este módulo. Para el tratamiento de los datos provenientes del módulo GM682-GPS, utilizaremos la ATM18 [3] y a su fiel pantalla LCD bifilar [4], así como la placa de relés [5] que pertenece a la misma familia. Con el fin de añadir rápidamente a nuestro proyecto una función de señalización, utilizaremos nuestro familiar pixel RVA gigante, *BlinkM MaxM*, controlado por I²C, al que ya hicimos sus pruebas en el artículo «Talk Show» [6].

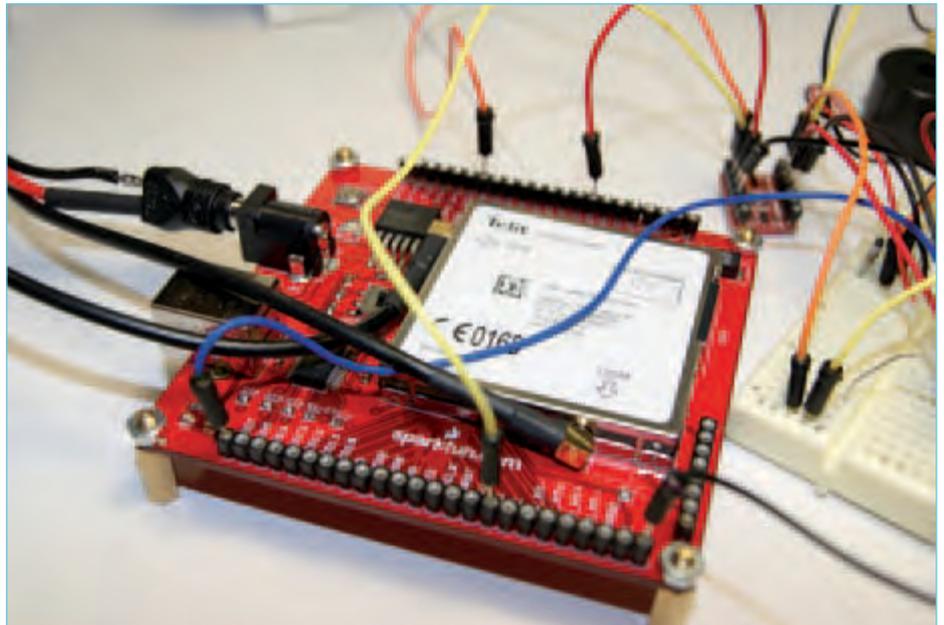


Figura 1. Placa GM862-GPS cómodamente instalada.

Domestiquemos a la bestia...

Antes de poner en marcha el módulo GM862-GPS, soldaremos los soportes tuli-

pas sobre la pequeña placa de prueba (ver **Figura 1**), que le permitirán un enlace fácil hacia la ATM18. Una vez hecha esta opera-

ción, conectaremos la antena GPS antes de insertar el módulo GM862-GPS sobre su soporte, luego conectaremos la antena GSM.

Productos y servicios Elektor

- Placa “ocho relés”: tienda online ref. 071035-72
- Placa Puerto de extensión: tienda online ref. 071035-95
- Placa controlador ATM18: tienda online ref. 071035-91
- Placa base ATM18: tienda online ref. 071035-92
- Visualizador bifilar: tienda online ref. 071035-93
- Microcódigo (descarga gratuita) en www.elektor.es/110139

Colocaremos el microinterruptor sobre la posición DTR HI (modo programación del GM862-GPS no activado).

Nos proveeremos de un cable USB A macho a USB B macho con el fin de conectar la placa al PC. Es posible que sea necesaria la instalación del controlador "FTDI" [7].

Ponemos en funcionamiento el modem pulsando durante 1 s en el botón pulsador ON/OFF, como solemos hacer con nuestro teléfono móvil. En este estado el LED rojo POWER está encendido y el LED STAT_LED parpadea a intervalos de 1 s, indicando que el GM862-GPS no está registrado en la red. En efecto, aún no hemos insertado ninguna tarjeta SIM, por el momento.

A partir de ahora van a comenzar las cosas serias. El modem responderá a los comandos enviados por su conexión serie, con la condición de que éstos últimas empiecen siempre por los caracteres ASCII AT (abreviación de ATención) y acaben por un "retorno de carro" <CR>. Con el fin de evitar llenar este artículo con la descripción de numerosos comandos "AT", al final del artículo se han reunido todos en la **Tabla 1**, ya indexada. A la llegada de un comando de escritura o de lectura, la respuesta está siempre visible bajo la forma: <CR><LF>OK<CR><LF>, si el comando ha sido reconocido, o <CR><LF>ERROR<CR><LF> si no lo ha sido (normalmente, con una descripción corta del error).

Comencemos por comprobar la conexión que acabamos de establecer físicamente entre nuestro GM862-GPS y el ordenador, sobre el cual nos habremos cuidado de instalar el famoso terminal *Hercules* [8] configurado, por el momento, a 115.200 baudios. La tarjeta SIM no la insertaremos por ahora. Antes de enviar nuestro primer comando, vamos a activar la visualización de los caracteres especiales bajo su forma hexadecimal. Para hacerlo, colocaremos el cursor del ratón en la ventana activa *Received/Sent data* y, con un pequeño clic en el botón derecho de ratón se nos mostrará un menú contextual. Seleccionamos *Special Chars* y, después, *Hexadecimal*.

La **Figura 2** nos indica que el enlace está perfectamente establecido. Hay que señalar que el comando 01 que acabamos de enviar es repetido automáticamente por el modem, así pues, el *ECO* va seguido de la

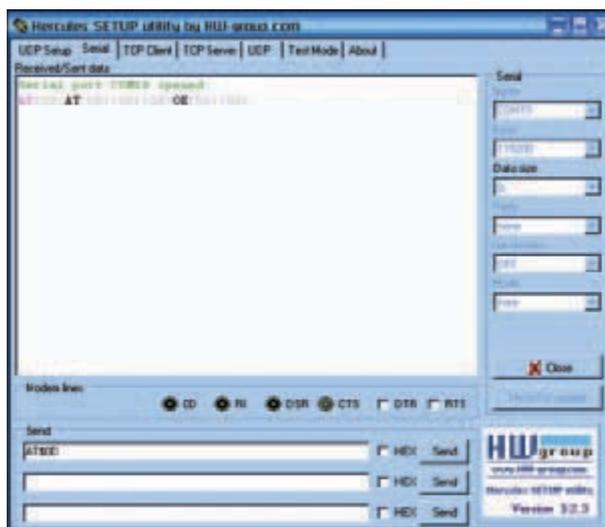


Figura 2. ¡Hércules, el retorno!

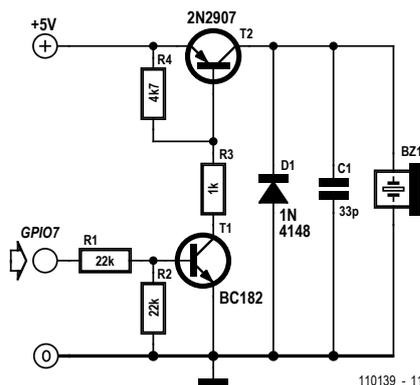


Figura 3. Un pequeño puñado de componentes para tener sonido.

respuesta esperada. Con el fin de no sobrecargar nuestra tabla, no mostraremos los caracteres especiales, por lo que el comando 02 es idéntico al comando 03 y el módulo GM862-GPS ha sido fabricado, en efecto, por Telit (comandos 03 y 04).

El comando 06 nos informa sobre la intensidad de la señal útil recibida, $rss_i = 1$ corresponde a -111 dBm. Sabiendo que el paso es de 2 dBm, el valor leído $rss_i = 11$ se corresponde a una atenuación de -91 dBm. Cuanto más grande sea RSSI mejor es el nivel de recepción.

La bestia va a emitir un sonido

El módulo GM862-GPS dispone de varias entradas/salidas de uso general. Vamos a estudiar el terminal marcado como GPIO7, que puede ser configurado como entrada

o como salida, pero también acepta el modo ALTERNATE FUNCTION que, en particular para este terminal, corresponde al modo *Buzzer Output*. El comando 07 nos permite activar el terminal GPIO7 en el modo citado. El comando 08 nos permite verificar la configuración de todos los terminales. Ahora basta con indicar al módulo GM862-GPS que, a partir de ahora, los sonidos deberán ser orientados hacia el terminal GPIO7, que acaba de ser configurado con este fin. Es el comando 09 el que nos permite hacerlo.

Se necesita un puñado de componentes para adaptar la señal que sale de GPIO7 hacia el zumbador ("buzzer") (ver **Figura 3**). Este pequeño montaje puede realizarse fácilmente sobre una placa de experimentación del tipo Labdec u otro modelo. Los

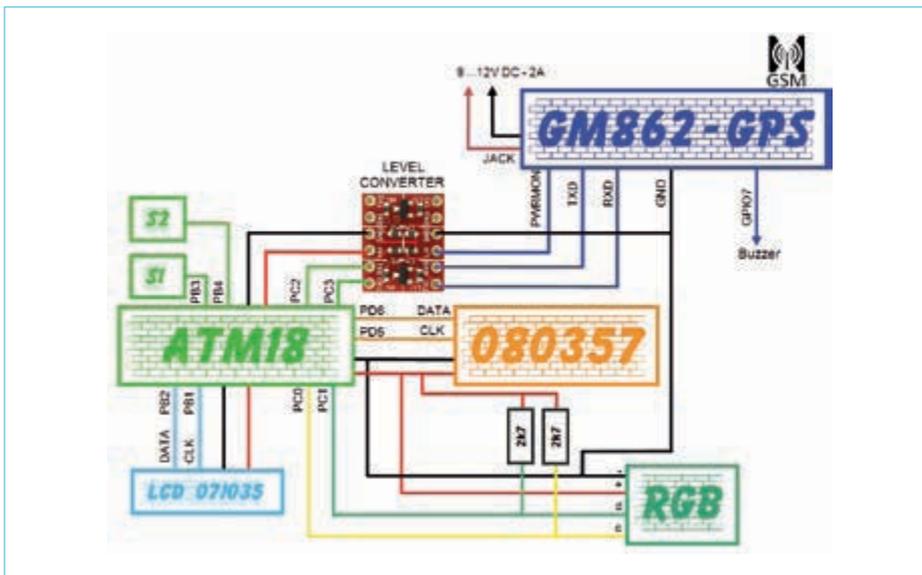


Figura 4. Diagrama del cableado del conjunto.

5V y GND estarán conectados a la alimentación de 5 V de nuestra placa ATM18.

Antes de presentar la placa ATM18 al módulo GM862-GPS vamos a ajustar la velocidad de transferencia del enlace serie a 9.600 baudios a través del comando 12.

Quitaremos la tensión de alimentación del módulo desconectando el cable USB de PC y después insertaremos la tarjeta SIM. En lo sucesivo es preferible alimentar la placa del modem con una tensión de alimentación comprendida entre 9 y 12 V, de 2 A. Podemos volver a conectar el cable USB para continuar la comunicación con nuestro modem, gracias al terminal a Hércules. La placa lo permite ya que la alimentación externa es prioritaria frente a la tensión proveniente del puerto USB del ordenador.

Antes de ir más lejos, necesitamos usar el comando 14 con el fin de pedir el código PIN de la tarjeta SIM. Si el código es correcto, la validación es seguida de una señal acústica y, si el GM862-GPS está registrado en la red, el diodo LED indicador LED_STAT parpadeará a intervalos de 3 s.

Ejecutamos de nuevo los comandos 07 y 09 con el fin de activar el timbre, después llamamos a nuestro modem marcando su número. El timbre suena y el LED_STAT queda encendido. Al mismo tiempo, un mensaje RING es enviado sobre el terminal.

Si el timbre por defecto no nos gusta, podemos modificarlo con el comando 15.

La bestia va a expresarse

Sin volver a leer las explicaciones precedentes, intentemos configurar el terminal

GPIO7 para que reciba los tonos de señalización.

La respuesta reenviada por el módulo después de haber ejecutado el comando 18 nos indica que la tarjeta SIM (SM) puede servir, al mismo tiempo, para leer, escribir, enviar y almacenar los mensajes SMS. El comando 19 indica que todos los comandos de lectura y escritura se hacen sobre la tarjeta SIM que dispone de 50 emplazamientos, donde tres mensajes están presentes actualmente en la memoria.

Para enviar un SMS, activaremos en un primer momento el modo TEXT por medio del comando 20. Cuando enviamos un SMS, primero pasa por su centro de mensajería (SMSC), por lo que es necesario que este número esté presente en alguna parte. Normalmente, este parámetro ya viene inscrito y el comando 21 nos dará la respuesta.

Comando 22: el envío del mensaje se hace en dos etapas. Primero hay que informar del número del destinatario, a continuación, escribir el mensaje y validarlo con "Ctrl-Z" (\$1A). El mensaje se envía directamente a la red sin ser almacenado en nuestra tarjeta SIM.

En el momento de la recepción de un SMS y para que el GM862-GPS (EM significa *Equipo Móvil*) pueda indicar al terminal Hércules (ET: *Equipo Terminal*) la presencia de un mensaje y, sobre todo, su emplazamiento, necesitaremos usar el comando 23. En lo sucesivo si el módulo GM862-GPS recibe un SMS, el terminal recibirá un mensaje formateado del tipo: +CMTI: "SM", 5, lo que significa que hemos recibido un nuevo

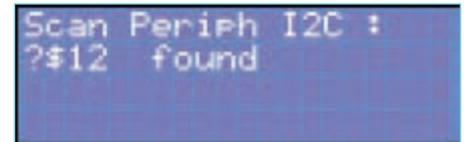


Figura 5.

mensaje y que ha sido almacenado en la posición 5 de nuestra tarjeta SIM. Para leerlo, nada más simple: ejecutamos el comando 24. ¿Ya lo hemos leído? Pues ya podemos borrarlo con el comando 25.

La bella y la bestia: el reencuentro

Uno de los dos micro-programas descritos más abajo podrá ser cargado en memoria del microcontrolador ATmega88 de nuestra placa ATM18. El cableado de todos los protagonistas seguirá el diagrama de conexionado de la **Figura 4**. Los puntos de contactos RX-O y TX-I de la placa de prueba roja que acoge nuestro modem deben estar abiertos (dos pequeños toques con nuestro soldador bastan suprimir los puntos de contacto). Al alimentar todo el conjunto el modem GM862-GPS se activa mediante una presión larga sobre el botón pulsador ON/OFF, como lo haríamos con nuestro teléfono móvil.

El primer programa (110139-I_GM862-GPS_ATM18_P1.bas [9]) nos permite invertir el estado del relé 1 de nuestro montaje, a condición de que seamos quien tiene ese derecho. Para conseguirlo, basta con modificar las constantes `Const Admin_phone = "06XXXXXXXX"` y `Const Code_pin = "XXXX"`, rellenándolas con nuestro número de teléfono y nuestro código PIN. Está claro que, con el fin de que podamos ser identificados como un llamante no oculto, debemos activar la presentación de nuestro número para mostrar la "carta de identificación" al ATM18. ¡Compilaremos, cargaremos y todo estará listo!

Si el número reconocido es el del administrador, el pixel gigante BlinkM MaxM brillará en color verde, si no es será el rojo el color emitido.

Una vez el programa está en ejecución, he aquí lo que pasa cronológicamente:

En un primer momento, y después de haber recuperado automáticamente la dirección



Tabla 1. Los trabajos de Hércules

Índice	Comando	Respuesta obtenida	Significado de la respuesta
	AT+CGMR\$0D	07.02.403 OK	Versión de Software
	AT\$GPSSW?\$0D	\$GPSSW: GSW3.5.0Telit_3.5.00.00-C23P2.00a OK	Versión del Software GPS
	AT##SELINT?\$0D	#SELINT: 2 OK	Interfaz AT Estilo 2
01	AT\$0d	{0D}{0A}OK{0D}{0A}	Enlace establecido
02	AT+CGMI\$0D	AT+CGMI{0D}{0D}{0A}Telit{0D}{0A}{0D}{0A}OK{0D}{0A}	Fabricante: Telit
03	AT+CGMI\$0D	Telit OK	Fabricante: Telit
04	AT+CGMM\$0D	GM862-GPS OK	Modelo : GM862-GPS
05	AT+CGSN\$0D	358278006334593 OK	Código IMEI : 358278.00.633459.3
06	AT+CSQ\$0D	+CSQ: 11,0 OK	+CSQ : rssi = 11, tasa de error inferior a 0,2%
07	AT##GPIO=7,0,2\$0D	OK	El terminal 7 está configurado en modo <i>Buzzer Output</i>
08	AT##GPIO?\$0D	Muestra el estado lógico de todas las Entradas/Salidas	
09	AT##SRP=3\$0D	OK	Salida del timbre hacia el buzzer
10	AT##SRP?\$0D	#SRP: 3 OK	Salida del timbre configurada en modo 3
11	AT\$GPSSAV\$0D	OK	Almacenamiento de los parámetros del GPS
12	AT+IPR=9600\$0D	OK	Puerto serie ajustado a 9600 baudios
13	AT##SHDN\$0D	OK acompañado de una señal sonora	Modem desconectado de la red, seguido de su apagado
14	AT+CPIN=7453\$0D	OK	El código PIN ha sido solicitado
15	AT##SRS=7,10\$0D et AT##SRS=0,0\$0D	OK OK	Reproduce la 7ª melodía de las 32 disponibles durante 10 s (maxi = 60 s) y almacena este nuevo timbre como melodía por defecto
16	AT+CCLK="11/02/04,20:48:00+04"\$0D	OK	Actualización de fecha y hora: 20h48m00s 4fev2011 GMT+1
17	AT+CCLK?\$0D	+CCLK: "11/02/04,22:06:22" OK	Lectura de la fecha y hora
18	AT+CPMS=?\$0D	+CPMS: ("ME","SM"), ("SM"), ("SM") OK	¿Memorias de almacenamiento disponibles?
19	AT+CPMS?\$0D	+CPMS: "SM",3,50,"SM",3,50,"SM",3,50 OK	Preferencias actuales para el almacenamiento de los mensajes
20	AT+CMGF=1\$0D	OK	Activación del modo TEXT para el envío de los SMS
21	AT+CSCA?\$0D	+CSCA: "+33609001390",145 OK	Número del centro de mensajería SMSC
22	AT+CMGS=0682834725\$0D HELLO BB\$1A	> _ +CMGS: 3 OK	Escriba su SMS El SMS ha sido enviado
23	AT+CNMI=1,1\$0D	OK	Configuración posible para la recepción de un SMS
24	AT+CMGR=5\$0D	+CMGR: "REC READ","+33682834725","","11/02/05,21:57:49+04" Hola desde BB OK	El mensaje almacenado en la posición 5 ha sido leído
25	AT+CMGD=5\$0D	OK	El mensaje almacenado en la posición 5 ha sido borrado
26	AT+CMGR=5\$0D	+CMS ERROR: 321	Índice de memoria no Válido, no hay (más) mensajes para leer en esta posición.

```
ATM18 and GM862-GPS
> Relays : 00000000
GM862-GPS READY
SIM PIN OK
```

Figura 6.

```
ATM18 and GM862-GPS
> Relays : 00000000
GM862-GPS READY
SIM PIN ALREADY OK
```

Figura 7.

```
GM862-GPS ON NETWORK
PLAYING TONE 7 OK?
```

Figura 8.

```
GM862-GPS ON NETWORK
PLAYING TONE 11 OK?
```

Figura 9.

```
GM862-GPS ON NETWORK
PLAYING TONE 11 OK?
RINGING TONE 11 DONE
YOU CAN CALL ATM18 !
```

Figura 10.

```
CALL WAITING...
```

Figura 11.

PC del módulo BlinkM MaxM (ver **Figura 5**), el programa procede a la prueba de los periféricos. El módulo RGB difunde una luz roja y los ocho relés se activan durante un pequeño instante.

El eco de los comandos enviados está desactivado. El código PIN es solicitado, seguido de una temporización de algunos segundos y, a continuación se prueba la conexión y se inicializa el módulo GM862-GPS (ver **Figura 6**). Si el código PIN ya ha sido anotado en un inicio previo, el mensaje de la **Figura 7** es el que se muestra.

En este estado, la pulsación del botón S1 de la placa ATM18 nos permite pasar a la pantalla siguiente (ver **Figura 8**) para elegir el tipo de tono (la primera línea nos indica si nuestro GM862-GPS “encuentra la red”). Si mantenemos pulsado S2 durante los 1,5 s que siguen a la visualización de esta pantalla, se reproducirán los 32 tonos disponibles, brevemente, a intervalos de 1,5

s. Soltaremos S2 cuando oigamos el tono que nos interesa/gusta. Si mantenemos S2 pulsado durante un tiempo mayor a 1,5 s, el paso al tono siguiente se hará 8 s más tarde, lo que nos permitirá oír el tono en su totalidad. S2 sin pulsar nos permite pasar a la etapa siguiente (ver **Figura 10**). ¡Tan pronto como se muestre la pantalla correspondiente a la **Figura 11**, ya podemos llamar al ATM18!

Desde el teléfono registrado como administrador, marcaremos el número correspondiente a la tarjeta SIM insertada en el GM862-GPS, el tono que acabamos de seleccionar será el que suene, el relé 1 cambia su estado y el LED verde brilla (ver **Figura 12**). Espere un minuto para poder hacer otra llamada. Llamemos ahora a nuestro montaje desde de un teléfono con otro número. ¡Aparecerá el mensaje de la **Figura 13** aparece y esta vez es el LED rojo el que brilla con gran intensidad!

Enviar SMS al ATM18

El montaje precedente tiene la ventaja de no costarnos un solo céntimo en llamadas. Por otra parte, es muy posible y muy simple hacer evolucionar el programa. En efecto, como sabemos que a cada llamada se transmite la cadena *RING* al ATM18, podemos imaginar como condición la cuenta del número de tonalidades para la activación de un relé en particular. La segunda evolución, para las personas que padezcan de sordera, podemos pensar también en una adaptación del programa que atribuya un color de emisión sobre el pixel RVA, diferente y único, en función de la persona (o teléfono) que llame. El visitante en cuestión no tiene que usar ya el intercomunicador y, una vez delante de la puerta de entrada del edificio, le bastará con llamar el montaje situado en el piso de la persona sorda, su presencia será señalada por una luz de color que le ha sido reservada previamente.



```
YOU RECEIVED A CALL
From : 0682834725
HELLO ADMINISTRATOR
> Relays : 00000001
```

Figura 12. Llamada recibida del administrador.



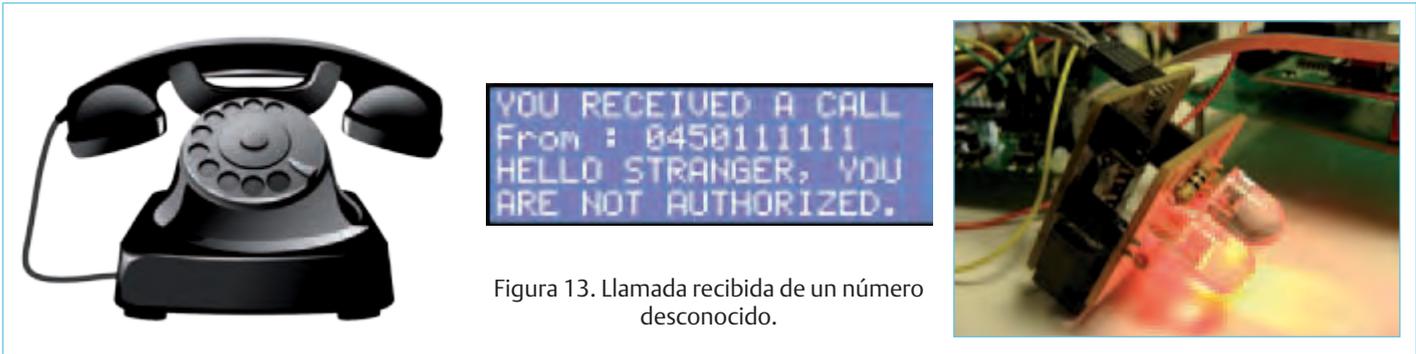


Figura 13. Llamada recibida de un número desconocido.

El segundo programa (110139-I_GM862-GPS_ATM18_P2.bas [9]) va más lejos ya que nos va a permitirle enviar un SMS a nuestro montaje con el fin de controlar los ocho relés de una vez y, por lo tanto, imponerles un estado lógico "0" (no activado) o "1" (activado). Además, nos será enviado un acuse de recibo para confirmarnos la ejecución correcta de nuestra demanda. Debemos señalar también que si sólo deseamos interrogar a nuestro montaje para recibir solo el estado lógico de todos los relés, sin activar ninguno, es posible. Pero, ¿cómo lo hacemos? Y ¿cómo funciona? En el inicio, el funcionamiento es idéntico al montaje precedente, el eco de los comandos enviados está desactivado, el código PIN es solicitado seguido de una temporización de algunos segundos. Seguidamente, comprobamos la conexión y se inicializa el módulo GM862-GPS. Así pues, tenemos una pantalla como la de la **Figura 6** o **7**. Una pulsación sobre S1 nos permite pasar a la pantalla siguiente (ver **Figura 14**). Se verifica el estado de la red y en la segunda línea se muestra si, naturalmente, estamos registrados en la mencionada red. ¡La placa ATM18 y el módulo GM862-GPS están a nuestras órdenes!

Sin esperar más, cogemos nuestro teléfono, el mismo que hace un rato, y enviamos el SMS "00001111" a nuestro montaje. ¡Ti-La-Ti! ¿Lo oímos? ¡El GM862-GPS reacciona! Al mismo tiempo, la pantalla LCD es rellena con un mensaje que indica que se ha recibido un SMS y que se ha almacenado en zona memoria nº 1 (ver **Figura 15**). El contenido del SMS que hemos enviado se muestra sobre la cuarta línea y los relés RE1 a RE4 se activan. La acción continua con el envío del acuse de recibo señalizado sobre la cuarta línea por el mensaje *REPORT SENT!*, con lo que recibiremos en nuestro teléfono móvil un SMS como prueba del buen desarrollo de la acción. ¡No han pasado ni 30 segundos! Enviemos ahora el mensaje siguiente, finalizando con el carácter siguiente, al ATM18: « R ». Esta vez tendremos, sobre la cuarta línea de la pantalla LCD, el mensaje *REPORT ?* (ver **Figura 16**) que indica que sólo deseamos recibir el estado lógico de todos los relés, pero que pedimos ninguna acción. Naturalmente, recibiremos un SMS de retorno en respuesta a nuestra petición (ver **Figura 17**). ¡Como mínimo, es una gran satisfacción el poder controlar a la placa ATM18!

(110139)

Enlaces en internet

- [1] www.telit.com/en/products/gsm-gprs.php?p_ac=show&p=7
- [2] www.sparkfun.com/products/281
- [3] www.elektor.es/atm18
- [4] www.elektor.es/071035
- [5] www.elektor.es/080357
- [6] www.elektor.es/100360
- [7] www.ftdichip.com/FTDrivers.htm
- [8] www.hw-group.com/products/hercules/index_en.html
- [9] www.elektor.es/110139

El módulo GM862-GPS, así como las antenas GPS y GSM, la placa de prueba, el módulo BlinkM MaxM y la placa de conversión de nivel se pueden localizar, entre otros sitios, en Lextronic, www.lextronic.fr



Figura 14.

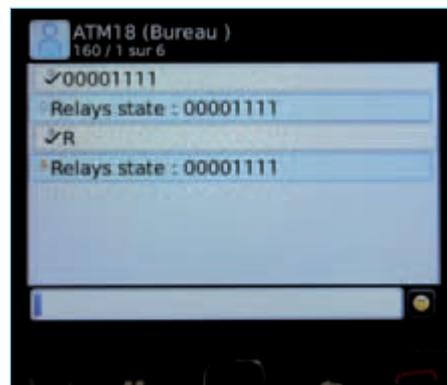


Figura 18. La pantalla del teléfono, testigo de los intercambios que acaban de efectuarse.



Figura 16.



Figura 15.



Figura 17.

El último toque

Diseño gratuito de cubiertas frontales profesionales

Giel Dols (Holanda)

El detalle que más salta a la vista de un amplificador, aparato de medida u otro aparato electrónico recién construido, es sin duda su “interfaz de usuario”: la cubierta frontal. La creación de una cubierta frontal atractiva de un aparato electrónico — es decir: con un acabado profesional —, es un obstáculo difícil de salvar para muchos electrónicos. Aparte de tener un cierto conocimiento del material, hay que disponer también de sierra eléctrica, taladro y fresa y lo que es más importante, saber utilizarlos.

El fabricante alemán de cajas y cubiertas frontales, Schaeffer AG, ofrece una solución eficiente para este problema. Con la ayuda del programa Front Panel Designer que se puede descargar gratuitamente, todo el mundo puede diseñar cubiertas frontales y encargar su fabricación.

El programa está disponible en alemán, francés e inglés y hay versiones para Windows, Linux y Mac OS [1].

En este artículo vamos a explicar, a grandes rasgos, lo fácil que es diseñar una cubierta frontal profesional con Front Panel Designer y encargar su fabricación. El software dispone de una ayuda muy buena proporcionando una explicación perfecta en cada situación. De modo que, descargue el software e instálalo. Como ejemplo diseñaremos una cubierta frontal para un armario de 19” en donde ubicaremos el preamplificador [2] que permite conectar una guitarra eléctrica al multiefectos [3].

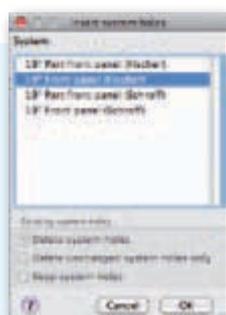
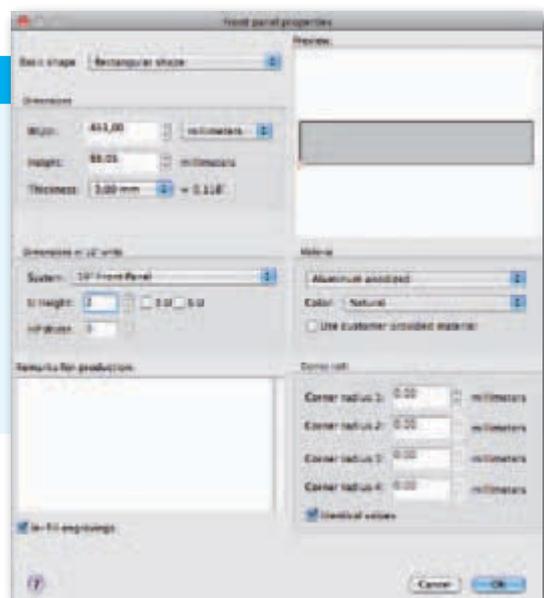
Paso 1: Las características de la cubierta frontal

La opción de menú **File -> New** abre una ventana donde hay que definir todas características de la cubierta frontal: forma básica, longitud, anchura y grosor.

Una opción especial es la elección de cubiertas frontales destinadas a armarios de 19” donde se indica la altura en “unidades”.

Por supuesto que tenemos que indicar también el tipo de material con el que se tiene que fabricar la cubierta frontal y el (posible) color que debe tener. Aquí tenemos la opción de elegir entre aluminio (de diferentes acabados) y plexiglás.

Una última opción que se puede definir es el radio de redondeo de las esquinas de la cubierta frontal.



Paso 2: Agujeros, agujeros y agujeros

Está claro que se ha pensado bien en ello: una cubierta frontal de 19” tiene que venir equipada de agujeros de montaje en los sitios correctos. El software se encarga de esto. Se han predefinido los agujeros de montaje para dos grandes fabricantes. A través del menú **Insert -> System holes** se llega al sitio donde elegir la opción deseada.

La parte del manejo del multiefectos requiere un hueco para un display de cristal líquido y cinco agujeros para los ejes de K1 y S1 hasta S4. También debe de haber cuatro agujeros para poder fijar la placa detrás de la cubierta frontal. Nuestro preamplificador requiere cinco agujeros: tres para los potenciómetros JP3, JP4 y JP5 y dos para los conectores jack de chasis de 6,3 mm, uno para la conexión de la guitarra y otro para el interruptor mute de pie.

Elige **Insert -> Drill hole** y pincha la cubierta frontal en el lugar dónde tiene que ir el agujero. Entonces aparecerá el menú del taladro. Ahí definimos cómo tiene que ser el agujero deseado. Seleccionamos un agujero de 3 mm destinado a tornillos avellanados tipo M3 (*counter sunk allen screws*). El software rellena el resto de la información relevante y muestra en una ventana cómo aparecerán una serie de cosas en una sección transversal.

Si haces clic en **Select mode** sobre una parte de la cubierta frontal, aparecerá en la esquina superior de la izquierda de la pantalla, todas las características de ese elemento. Ahí puedes ajustar todas las características. También puedes introducir las coordenadas exactas de los ejes X e Y de un elemento determinado (en este caso nuestro agujero).

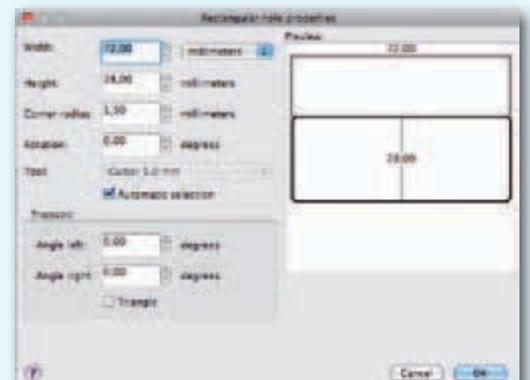
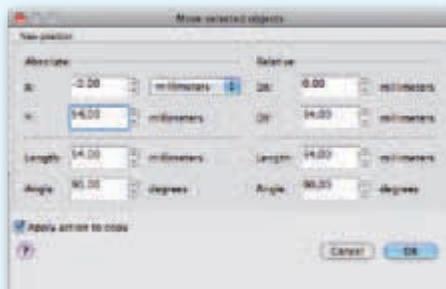
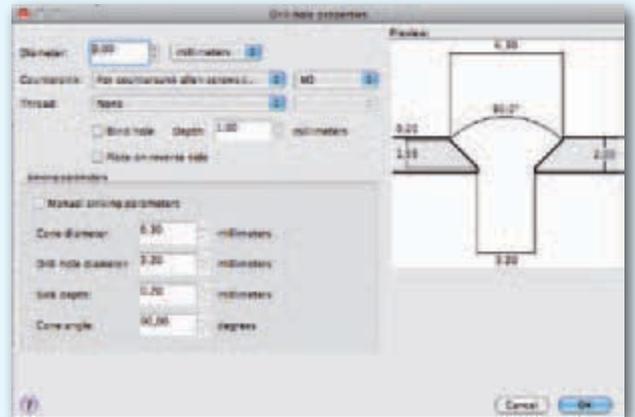
Estas coordenadas son relativas al sitio de nuestro punto cero. Este se encuentra por defecto en la esquina inferior de la izquierda de la cubierta frontal. Para evitar que haya que calcular las coordenadas de cada elemento de la cubierta frontal, existe la posibilidad de desplazar el punto cero. Para eso pinchamos primero en **Set origin** y después en el sitio (después de hacer mucho zoom) donde queremos tener el punto cero. Si colocamos el punto cero sobre un agujero, la colocación en el sitio correcto de los demás agujeros y del hueco para el display LCD será más fácil.

La opción **move** también se utiliza para la colocación de los demás agujeros. Para eso hay que seleccionar el agujero colocado e introducir el desplazamiento del segundo agujero (en nuestro caso 54 mm) a través de **Selection -> Move**. No olvides activar aquí la opción "Apply action to copy", ¡ya que en caso contrario se desplaza el elemento seleccionado en vez de copiarlo! Selecciona entonces ambos agujeros y repite la acción; ahora con un desplazamiento horizontal de 85 mm.

Ahora hay que dibujar el hueco del display LCD. A través de **Insert -> Rectangular cutout** llegamos a un menú para introducir las características de nuestra "ventana", en nuestro caso: la longitud, la anchura y el radio de los redondeos de las esquinas. Selecciona la ventana e introduce las coordenadas correctas en la ventana 'Object properties' (en la parte superior de la izquierda) para el centro de la ventana del display deseado.

Coloca después los agujeros de los cuatro botones de mando S1 hasta S4 y el eje de K1. La coordenada del eje Y es la misma para todos estos agujeros (ya que están colocados en una línea). Determinamos las coordenadas del eje X midiéndolas en la placa impresa original.

El software dispone ya de una biblioteca limitada, pero útil, de huecos para ventiladores, interruptores, conectores sub-D, AMP y SCART de algunos fabricantes conocidos. Se puede acceder a estos elementos especiales a través de **Insert -> Macro object**. En el momento de seleccionar un elemento aparece una ventana con la previsualización por debajo de la biblioteca de macros. El software ofrece también la posibilidad de seleccionar uno mismo los elementos o combinaciones de elementos, y guardarlos en una librería de usuario. Selecciona para eso **Selection -> Create macro**.

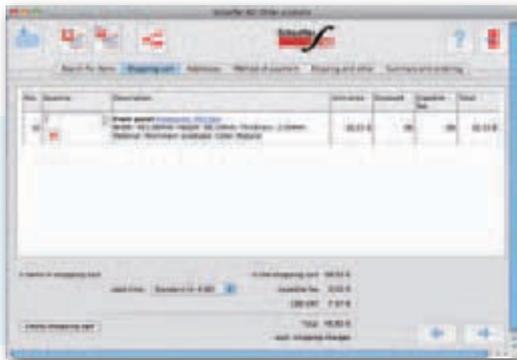
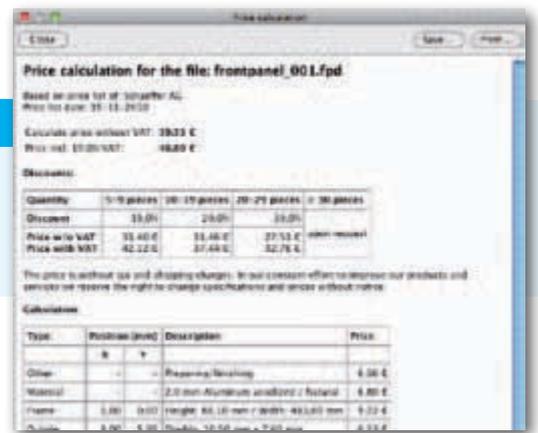


Paso 3: Texto

La opción del menú **Insert -> Text engraving** abre una ventana donde se puede grabar el texto deseado y sus características. Cada trozo de texto definido por separado puede ser seleccionado para colocarlo exactamente en su sitio o cambiarlo.

Paso 4: ¿Cuánto cuesta?

Pinchando sobre el icono colocado en el extremo derecho, marcada claramente con dos monedas, se abre una ventana con el precio de la cubierta frontal. Los costes se especifican hasta el más mínimo detalle y también aparece un resumen con precios por lote. ¡Afortunadamente aquí no hay sitio para sorpresas!



Paso 5: Vamos a pedir...

El último paso antes de montar la nueva cubierta frontal diseñada en nuestro aparato, es por supuesto: el pedido. Esta opción también está perfectamente integrada en el software. A través de la opción **Order -> Order current front panel** (o: **Start ordering program**) se inicia un interfaz con el que se tramita el pedido online, donde debes introducir todos los datos relevantes.

Enlaces Web

[1] **Front Panel designer software**

www.schaeffer-ag.de/en/download/front-panel-designer.html

[2] **Guitarra para el multiefectos**

www.elektor.es/100923

[3] **Multiefectos digital**

www.elektor.es/090835

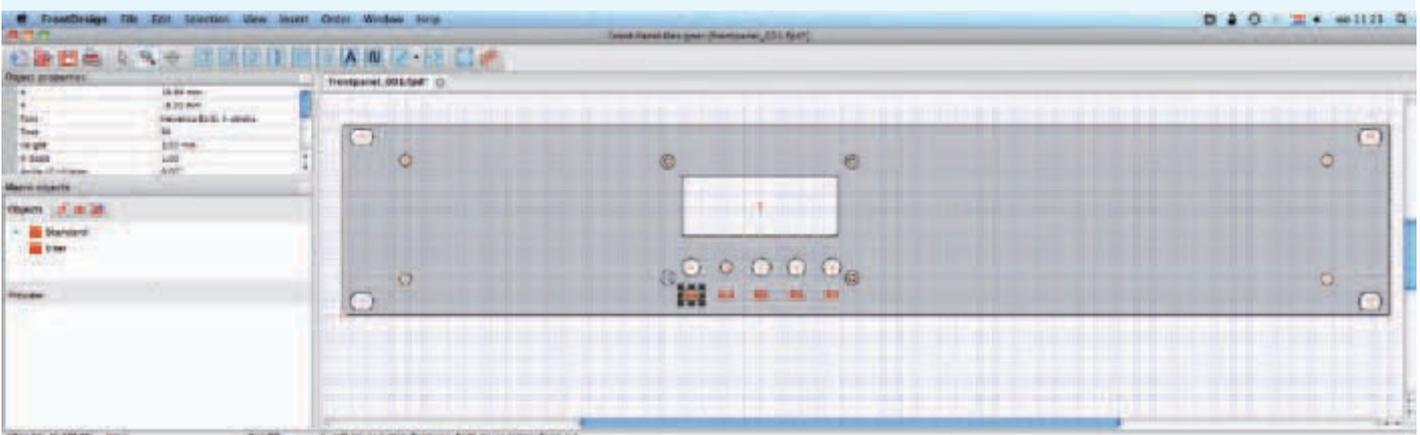
[4] **Descarga de la cubierta frontal**

www.elektor.es/110238

(110238)

¿Probar?

Durante la evaluación de este software diseñamos (parte de) una cubierta frontal. De modo que aún no está terminada. Todavía hay que colocar los agujeros del preamplificador al igual que los huecos de los dos conectores jack de chasis de 6,3 mm. ¡Te invitamos a probarlo tú mismo! Puedes descargar la cubierta frontal – que aún hay que terminar de diseñar [4] de nuestro sitio web.





Osciloscopio de Almacenamiento Tektronix 564 (1963)



Jan Buiting (Editor de Elektor Reino Unido/EE.UU)

Es difícil que podamos considerarnos alguien experimentado en electrónica si no reconocemos al instante un osciloscopio Tektronix de la serie 500. Estas bestias, en sus cajas de color azul brillante, estuvieron en todos los laboratorios de electrónica, colegios y otros ‘sitios científicos’ durante más de dos décadas, antes de ser desechadas o entregadas como parte del pago por el cambio a equipos fabricados ya con transistores. Muchos usuarios todavía se ofenden por esto. En un sitio web dedicado a los equipos de prueba Tektronix basados en tubos de vacío, la palabra ‘transistor’ está prohibida, donde sólo se ha usado una vez “trnsst”.

Es fácil ver por qué los osciloscopios de la serie 500 de Tektronix remueven aún sentimientos nostálgicos de los años 60 y 70, en los días de ‘colegio y laboratorio’. En primer lugar, estos equipos estaban contruidos de manera increíblemente sólida, comparada con los productos de las generaciones actuales, con sus tapas de plástico, sus mandos de goma y sus textos arcanos en la pantalla.

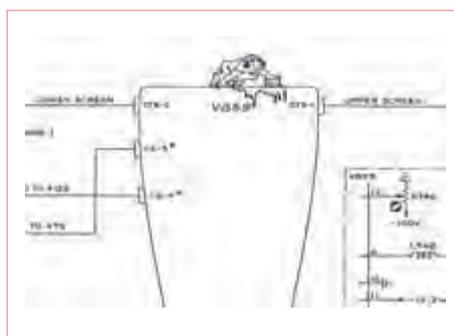
Añadamos a esto el peso enorme (el modelo 564, sin módulos conectados, pesaba 13,8 kgs/30,5 libras), el ruido del ventilador extractor (no presente en todos los modelos), la imagen de forma cortante en la pantalla CRT, los famosos módulos de conexión rápida de ‘clic-clac’ y el sonido de la gran cantidad de mandos rojos y negros del panel frontal. Y no disponían de elementos ‘borrables’ en la memoria del equipo.

En los años 70 y 80 mi osciloscopio Tek era (creo) un modelo 535, con un módulo de conexión de dos canales. Comparado con el montaje DIY (de ‘Hágalo Usted Mismo’) que mis amigos usaban, el rango de frecuencia de mi 535 era tal que me vi arrastrado por toda la provincia, en el coche de mi padre, con la maleta de ‘Osciloscopio portátil’ y todo ello para hacer complejas medidas y observar escurridizos “glitches” (pequeñas señales parásitas de los circuitos digitales), generalmente en proyectos de Elektor con CIs CMOS. También lo usé para calentar mi pequeño taller que tenía en el ático. Nunca conseguí hacer completamente estable, en su barrido vertical, a uno de los canales de su módulo vertical, a pesar de haber intercambiado las válvulas. Fue la primera vez que vi un “nuvistor” (pequeña válvula hecha sólo de metal y cerámica) y los contactos dorados.

Hablando de contactos, la mayor parte de los componentes y cables de los osciloscopios de la serie Tek 500, estaban soldados con estaño con un contenido de 3 % de plata para garantizar una conexión que durase una vida. Uno de los paneles de alimentación de HV dentro del osciloscopio tenía un carrito plástico con estaño plateado para posibles reparaciones. Con 20 años, escribí para que me hiciesen un nuevo suministro de tan preciado estaño (intentando usar las palabras más amables de mi inglés “americano”). Después de, aproximadamente, un mes recibí un paquete, con dos carretes dentro, de la planta europea de Tektronix, en Guernesey, Cannel Islands. También había una carta adjunta con el consejo cortés de cómo soldar así como dos ‘notas de revisión’. Mi madre dijo que aquello era un milagro y “como los Soldados en 1945 – siempre de buen humor”; a lo que contesté “Considero que es sólo un servicio amable”.

Según los manuales del instrumento, los módulos de conexionado pueden ser cambiados sin apagar el instrumento principal, así pues, la “conexión en caliente” no es un invento de los años 90. Nunca me atreví a probar esto por miedo a dañar mi valioso 535 y necesitar pedir prestado otro osciloscopio para repararlo. Después de unos 10 años lo vendí y con un cierto beneficio.

El modelo 564 que se muestra aquí es bastante más ligero de peso y más pequeño, comparado con el 535. Conseguí este instrumento





de mi vecino, a cambio de un balde de poda el otoño pasado. El instrumento fue rescatado de un montón de diversos bienes domésticos y mobiliario, en un sótano ubicado en el edificio del

piso de sus padres. Como módulos conectables, el instrumento disponía de una base de tiempos 2B67 y del amplificador 3A6 de doble traza. Una vez hecha la limpieza habitual, aspirar el polvo y un encendido cuidadoso del 564, decidí hacer una búsqueda por Internet de 'Tektronix vintage' (los 'Tektronix clásicos'). Para mi asombro, me encontré con un método de cómo limpiar a fondo un osciloscopio, aplicado realmente por los propios técnicos de Tektronix en el Centro de Mantenimiento de Fábrica y se ofrecía como un servicio a los clientes. Sorprendentemente, los instrumentos eran 'lavados' usando agua destilada caliente y un detergente de limpieza de cristalería de laboratorio, con los componentes críticos sellados en plástico. El método fue mantenido y publicado hasta finales de los años 70, para la serie 7000 con transistores y hasta CIs [1]. El quid no era un lavado adecuado, sino un secado controlado del instrumento en un horno, durante 24 horas, a una temperatura baja. Tek escribe de forma escueta: *Coloque el instrumento (con los módulos conectables lavados instalados) en el horno y déjelo secar durante, al menos, 24 horas (El horno hace de lugar de almacenamiento adecuado hasta que el equipo tenga que ser usado. Más tiempo de secado también es bueno).* El lavado y el secado van seguidos de una etapa de lubricación y engrasado de conmutadores, mandos y motores, usando el kit de Lubricación de Tektronix p/n 003-0342-01 (¡qué más!). Decidí lavar mi coche en vez del 564.

La apertura de la caja de un osciloscopio Tektronix de válvulas se parece a la apertura del capó de un coche americano de los años 30, con un gran motor en el morro. Cada una de las dos tapas perforadas parcialmente, en los lados del instrumento, puede ser retirada con tan solo quitar dos tornillos. Una docena de válvulas nos dan la bienvenida, polvo, etiquetas de PATENTES impresas en el chasis y el olor de la electrónica añeja, 'estilo americano'. En algunos casos, los módulos de conexión contienen más válvulas y transistores que el chasis principal.

Mi 564 resultó que funcionó a la primera en un canal (traza). Se trata de un osciloscopio de almacenamiento con cero (sin) RAM. En su lugar se usan pantallas de almacenamiento, los cañones de inundación y los electrodos colimación, dentro del CRT diseñado con el objetivo de proporcionar una memoria de visualización primitiva. Una vez que vemos la señal usted tenemos que 'congelarla' en el monitor CRT, para ello utiliza los controles ERASE, INTENSITY y STORE en una cierta secuencia. Aparece una imagen no demasiado definida que es retenida en el tubo de fósforo (según Tek) durante, al menos, una hora. Mi 564 solo pudo mantener la (pobre) imagen durante, aproximadamente, 15 minutos antes

de que hubiera desaparecido completamente. Es posible que sea necesario hacer alguna re-calibración, por lo que estoy planeando escribir a Oregon para una sustitución del CRT y otro poco de estaño de repuesto. La imagen almacenada se puede borrar moviendo la palanca a la posición 'ERASE', con lo que la pantalla muestra un verde brillante. La traza almacenada puede ser usada para una comparación de buena/mala con una traza en vivo, así como para hacer una fotografía. Esta última aplicación es tratada en detalle en el Manual de Instrucciones del 564.

Según la buena tradición estadounidense, los empleados jubilados de Tektronix tienen anécdotas agradables para compartir como: el correo que se recibía en la fábrica de los usuarios de equipos, en lo que se describió cortésmente como "las zonas del mundo menos avanzadas tecnológicamente". Aunque los sobres iban dirigidos a: *Tektronix Inc., Portland, Oregon, EE. UU, VERTICALES*, dicha correspondencia fue debidamente entregada por el servicio de correos estadounidense.

El Manual de Instrucciones que viene con el 564 es un producto de ingeniería electrónica de EE.UU, típico años 60. Extenso, completo y agradablemente ilustrado, con descripciones completas del funcionamiento eléctrico del instrumento y con guía de mantenimiento y reparación. Francamente no esperaba encontrar ningún tema interesante en el manual pero, en la última página, la *Mujer Lavadora del 564*, de repente aparece limpiando de la pantalla del CRT V859. El humor de los ingenieros. En la página ClassicTek.org podemos encontrar otros comentarios graciosos, como *el Vaquero 7A26*, *el Hombre Inhibido 184* y *el Alpinista 502*. Comentario: después de todo, mi madre tenía razón.

(100920)

Enlaces en Internet

- [1] www.classictek.org
- [2] <http://www.r-type.org/exhib/aaa0273.htm>

Para lectura adicional

El Museo de los osciloscopios Tek:
<http://www.chiark.greenend.org.uk/scopes/tek.html>

BarryTech:
<http://www.barrytech.com/tektronix/vintage/tekvintage.html>

Museo de Tek:
<http://tekmuseum.ebaman.com/main.html#tline>