

Programas/Juegos/Montajes/Código Máquina

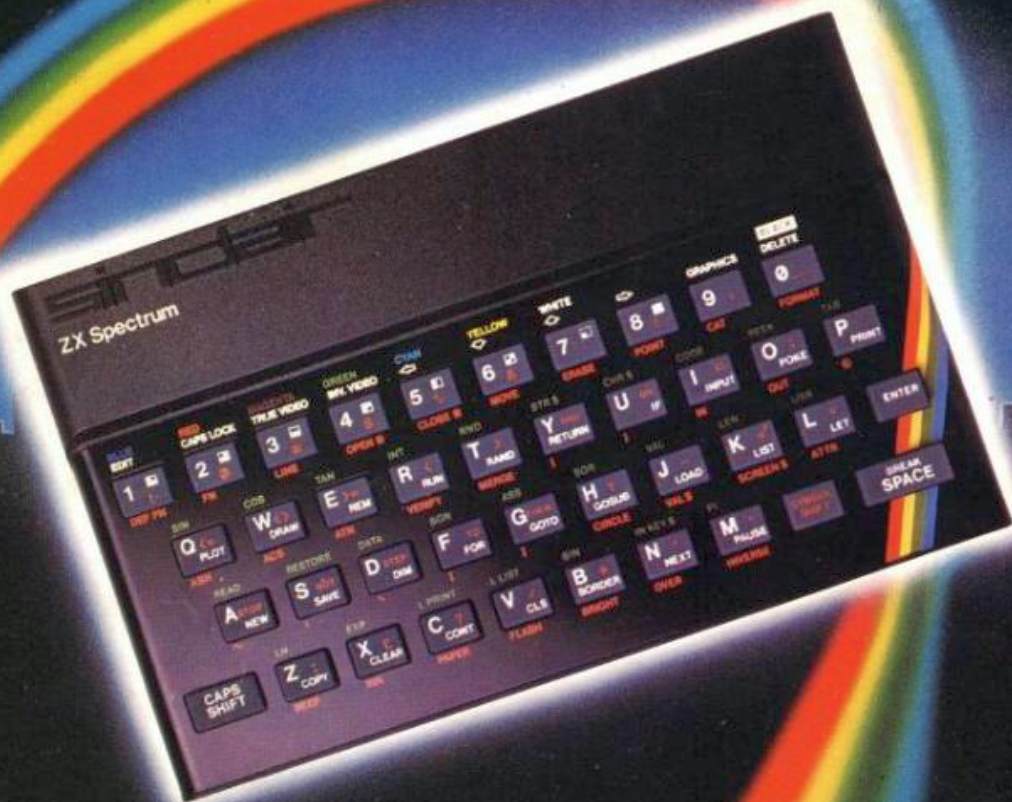
Año 1 / No. 3 - 200 Ptas.



ZX
REVISTA PARA LOS USUARIOS
DE ORDENADORES SINCLAIR

**REVISTA PARA LOS USUARIOS
DE ORDENADORES SINCLAIR**

EL SPECTRUM POR DENTRO



16 K: 39.900 Ptas.
48 K: 52.000 Ptas.

sinclair ZX Spectrum

El ordenador de todos para todo.



DISTRIBUIDOR
EXCLUSIVO:

INVESTRONICA

Central Comercial: TOMAS BRETON 60. TEL: 468 03 00. TELER 23359 (HCOE MADRID)
Delegación Cataluña: MONTANER 565. TEL: 212 68 00. BARCELONA



REVISTA PARA LOS USUARIOS DE ORDENADORES SINCLAIR

Este número 3 de ZX trae unas cuantas novedades. En primer lugar, dieciséis páginas más (sin variar el precio de venta). Ello nos ha permitido crear nuevas secciones. Pero la más destacable es la publicación de los primeros programas enviados por los lectores que han acudido a nuestra convocatoria del número anterior. Y, lo más importante, aunque no debamos sorprendernos por ello, es el alto número de lectores jóvenes que nos llaman, nos escriben y envían programas elaborados por ellos.

Hasta el próximo mes.

HARDWARE

Abrimos este número de la revista destapando el Spectrum y mostrando sus "tripas". Al hacer un recorrido por las interioridades del ordenador más popular del mercado, intentamos dar a conocer la misión específica de cada circuito integrado. Para comprender mejor la máquina que tenemos entre manos.

Página 4

PROGRAMAS

Crece el número de programas que presentamos. En su mayor parte son juegos —el mejor modo de entusiasmarse con las posibilidades de los ordenadores Sinclair— pero hay también programas educativos.

Página 22

COMENTARIOS

Iniciamos la andadura de esta nueva sección analizando tres programas de entre los muchos que se venden para el ZX Spectrum.

Página 12

MONTAJES

Esta sección vuelve a las fuentes. El tema de este número, "La soldadura bien hecha", ha sido escrito pensando en quienes no presumen de "manitas" (e incluso para algunos que sí presumen), con el objeto de permitirles avanzar en la elaboración de sus propios periféricos.

Página 46

LIBROS

La bibliografía en castellano sobre el Spectrum, casi inexistente hace poco tiempo, crece a ritmo acelerado. Aquí comentamos el libro escrito por Tim Langnell que acaba de salir en Ediciones Noray.

Página 17

NOTICIAS

¿Cómo es el nuevo ordenador de Sinclair, bautizado QL? A los lectores picados por la curiosidad, les ofrecemos una primera descripción de este equipo que, por características y precio, nos promete otra revolución en el mercado.

Página 11

LECTORES

Llueven cartas a nuestra Redacción. A las que saludan y felicitan, les agradecemos colectivamente. A quienes nos formulan consultas, esperamos responderles en este número.

Página 14

SOFTWARE

Quien pretenda sacar todo el partido posible de la capacidad de programación que le brinda su Spectrum, deberá conocer su código máquina. Iniciamos en este número la publicación de una serie de artículos escritos por Juan Martínez Velarde. Primer capítulo: la organización de la memoria.

Página 50

ZX es una publicación de Ediciones y Suscripciones, S. A. • Presidente: Fernando Bolín • Jerez, 3. Telfs.: (91) 250 15 93 - 458 76 02, Madrid-16 • Director: Norberto Gallego • Redacción: Alejandro Diges, Aníbal Pardo, Simeón Cruz, Gumersindo García, José Luis Durá jr • Diseño: R. Segura • Administración. Gerente de Circulación y Ventas: Luis Carrero • Suscripciones: Antonio Zurdo (91) 457 26 17 • Producción: Miguel Onieva • Publicidad Madrid: Telf. (91) 457 45 66 • Publicidad Barcelona: Tallers, 62-64, Barcelona-1. Telf. (93) 302 36 48 • Distribuye: Sociedad General Española de Librería. Avda. Valdelaparra, s/n. Alcobendas, Madrid • Imprime: Heroes, S. A. Torrelara, 8. Madrid-16 • Depósito Legal: M. 37.432-1983.

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Sinclair ni de sus representantes.

EL SPECTRUM POR DENTRO

Una vez que se ha conseguido un cierto dominio y comprensión sobre el funcionamiento de nuestro Spectrum (o de cualquier otro microordenador), lo más normales que uno quede maravillado de lo que un dispositivo bastante menor que una caja de zapatos puede hacer.

Casi indefectiblemente surge la incógnita, un algo que nos subyuga y, nos impulsa a cuestionarnos cómo lo hace. Los más osados, quizá los menos, alguna vez se habrán valido del modesto destornillador para observarle las vergüenzas al amado microordenador. El resultado más común es que, lejos de disipar las dudas que nos atormentan, surjan otras nuevas y más enrevesadas, tal vez más complejas.

Un recuerdo brota como la lucecita de una casa distante en la noche oscura del bosque. Una foto que alguien puso en el capítulo 9 del manual de introducción. Cuando estudiábamos el manual con fruición, no llegó a llamar demasiado nuestra atención. Pero ahora, tal vez nos lo podría aclarar todo. La foto va acompañada por leyendas, y un breve texto comenta la imagen del interior del Spectrum.

Tal vez a muchos lectores la sucinta explicación les deje satisfecha la curiosidad. Estamos seguros de que a muchos no.

La intención de este artículo es describir con mayor minuciosidad los pormenores del diseño del Spectrum. Sin embargo, no se pretende en ningún momento facilitar las pautas para que el usuario manitas pueda repararse su equipo. Ese trabajo deberá quedar siempre confiado a las expertas manos de los servicios técnicos especializados.

Las pautas que el lector encontrará aquí pueden ser lo suficientemente válidas para ir familiarizándose con el funcionamiento y la arquitectura de los

ordenadores, pues otros sistemas de precio superior se basan en principios iguales, bastante similares.

También este mayor conocimiento de lo que hace un ordenador, y de qué se vale para hacerlo, puede revelarse como una poderosa herramienta de evaluación a la hora de construir accesorios o seleccionar dispositivos periféricos antes de su compra. Los principios en que se basa el Spectrum para su funcionamiento dejarán de ser una cruel incógnita, después de leer estas líneas.

Vamos a emplear tres modos de descripción del sistema.

El primero es el diagrama de bloques, donde los rectángulos dibujados representarán un componente, o conjunto de componentes, con una misión perfectamente definida. Un conjunto de líneas unen entre sí a los diferentes bloques, representando los caminos por los cuales las señales binarias (unos y ceros, o si se prefiere tensión y no tensión eléctrica) pasan de uno a otro y en qué sentido.

El segundo es conocido familiarmente por los técnicos como esquema teórico. Aquí cada componente electrónico dibujado tiene una correspondencia física con un componente real, y las conexiones eléctricas que los unen aparecen representados por líneas. Por último, las fotografías nos ofrecen una visión real de cómo es el componente y qué lugar ocupa en la placa de circuito impreso, que se oculta bajo la carcasa de plástico del Spectrum.

Se hará referencia a uno de los tres o a su conjunto, según resulte, haciendo necesario a lo largo de la descripción.

Entramos en materia. La descripción, en principio, del diagrama de bloques, nos dará una clara panorámica de cada elemento del conjunto. Después pasaremos al detalle.

El microprocesador es el rectángulo

alargado que se observa en la parte izquierda del dibujo, donde pone Z-80. Tiene capacidad para hacer muchas cosas con las señales binarias, pero hay que instruirle cuidadosamente de cómo hacerlas. De momento esta idea de base nos sirve para continuar.

Como el Spectrum trabaja con el lenguaje de alto nivel, conocido por BASIC, se necesitará un programa capaz de interpretar lo que vamos introduciendo por el teclado en forma de líneas de programa escritas en dicho "idioma". Después es necesario convertirlo en una serie de instrucciones simples que entienda y ejecute el microprocesador, en cuyo interior se encuentra la unidad central de proceso (UCP). Este programa se encuentra grabado en la memoria ROM, rectángulo dibujado inmediatamente a la derecha del Z-80. En ella, aparte de toda esta información, se encuentran otras cosas de gran utilidad, que veremos más adelante.

La ROM viene grabada desde fábrica y es inalterable por parte del usuario, a no ser que voluntariamente se sustituya por otra desarrollada para tal fin.

Cuando tecleamos nuestro programa en BASIC, se va almacenando en otro tipo de memoria llamada RAM (Memoria de Acceso Aleatorio). A esta clase de memoria muchas veces se le califica de volátil, puesto que se puede hacer desaparecer toda la información contenida en ella de una manera extremadamente sencilla, simplemente desconectando la alimentación. También se pueden variar sus contenidos durante la programación, o en el transcurso de la ejecución de un programa de aplicación.

Un rectángulo, situado prácticamente en el centro del diagrama, representa a un bloque de los 16 Kbytes de RAM, la versión mínima del Spectrum. Otro bloque situado justo encima muestra un multiplexor de direcciones, que se en-

carga de decodificar las direcciones de memoria a las que quiere acceder el sistema, activando los *chips* de RAM que correspondan a la misma.

Un rectángulo de trazos en punto y raya representa el bloque de memoria RAM, que contiene los 32 Kbytes adicionales de memoria que completan al sistema de 48 Kbytes. En él aparece otro multiplexor de direcciones, que opera de manera similar al anterior, decodificando las direcciones de memoria de este bloque a las que se accede. Otro bloque, titulado control de lectura/escritura controla si el acceso a la RAM se efectúa para leer o para escribir datos en ella.

Uno de los mayores logros conseguidos primero en el ZX-81, posteriormente en el Spectrum y otras marcas de microordenador, es la ULA (*Uncommitted Logic Array*), cuya misión es sustituir un gran número de circuitos integrados, que contendrían puertas lógicas. En este circuito integrado, de diseño especial,

con destino a un modelo determinado de microordenador, se basa en gran medida el bajo costo y el reducido tamaño de tales sistemas.

La UCP debe comunicarse con el mundo exterior. Por un lado obtiene una entrada a partir del teclado, por otro envía datos de salida al televisor, al altavoz y al *cassette* (del que también recibe entrada de datos). La ULA ayuda a la UCP en dichas tareas. Ataca al bloque modulador de UHF, que envía las señales de vídeo en color al televisor y a un *interface* que ajusta los niveles de las señales enviadas y recibidas desde el *cassette*.

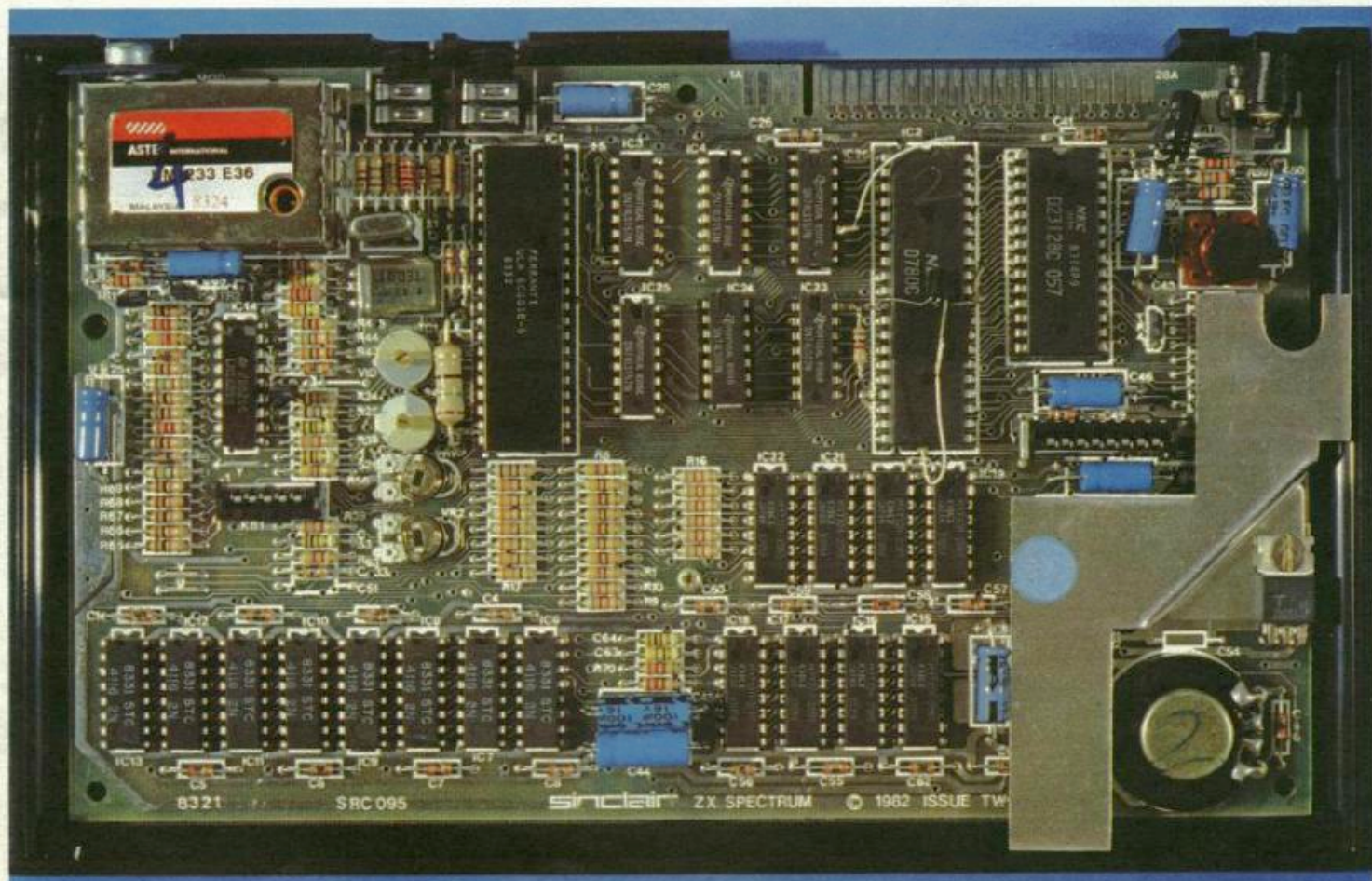
En definitiva, la ULA coordina estas funciones de Entrada/Salida. Las líneas gruesas representan a los buses, que son un conjunto de líneas que actúan como vía de comunicación. Existen tres buses principales. El bus de direcciones está compuesto por 16 líneas (2 bytes), que sirven para direccionar hasta 64 K (65,536) de direcciones de memoria. El

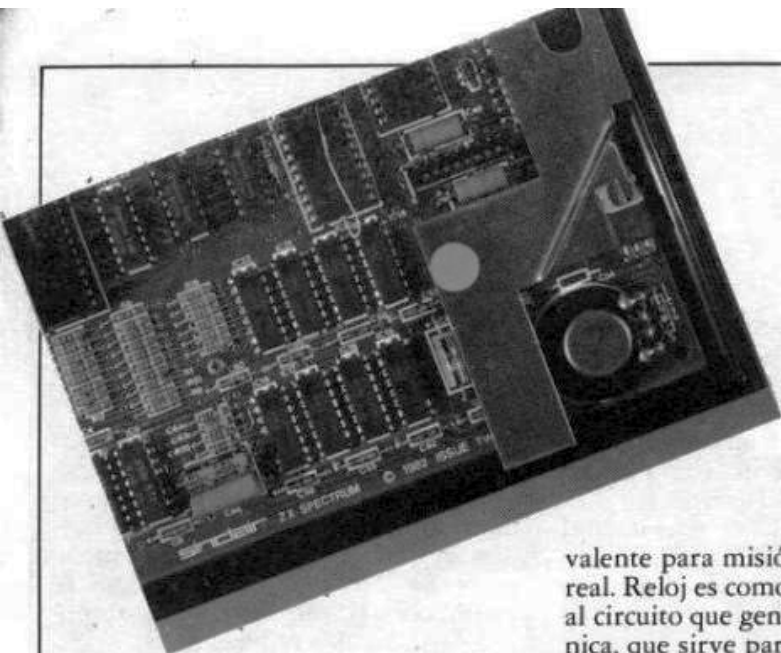
bus de datos, que tiene 8 líneas (1 byte), sirve para transferir un byte de datos entre los diferentes bloques del ordenador. El bus de control, compuesto por 5 líneas, se utiliza para que circulen por él las señales de control del sistema.

El bus de datos y el de control son bidireccionales, esto es, las señales pueden ser enviadas hacia y desde el microprocesador. El bus de direcciones es unidireccional, van desde el microprocesador a los demás bloques.

Por otro lado, los tres buses pueden ser accedidos directamente desde el exterior mediante el conector que aparece en la ranura posterior del Spectrum. Es el llamado *slot* de expansión, que se utiliza para la conexión de dispositivos periféricos y accesorios. Para terminar la descripción del diagrama, observemos en la parte inferior izquierda el bloque que constituye la fuente de alimentación estabilizada. Partiendo de los 9 voltios no estabilizados, que proporciona la fuente de alimentación ex-

Aspecto general que ofrece la tarjeta de circuito impreso sobre la cual van dispuestos los componentes que materializan al Spectrum.





Detalle del altavoz, aleta de refrigeración, 7805 y los 8 chips de expansión de la RAM.

terna, produce tensiones de distintos valores para el funcionamiento del ordenador. Para ir siguiendo la correspondencia entre componentes es preciso identificarlos en la foto y después en el esquema teórico, de acuerdo con los pies de escritura que aparecen en cada uno.

El microprocesador es el *chip* que va encapsulado en una pastilla negra y tiene un total de 40 patillas. Es, en definitiva, el componente más importante del circuito, por cuanto tiene capacidad para ejecutar todo el conjunto de acciones que hacen del Spectrum un ordenador. Puede llevar a cabo hasta 158 diferentes, conocidas como el juego de instrucciones.

Existen distintos microprocesadores en el mercado, fabricados por un reducido número de firmas. En el caso de Sinclair, la elección ha recaído sobre el Z-80A de Zilog, uno de los más populares entre los diseñadores de microordenadores.

El Z-80A va ejecutando obediente-mente las instrucciones en el orden que la marcan los programas existentes en las memorias RAM y ROM. Una línea de programa BASIC contenida en la RAM es interpretada en la ROM y esta libera una serie de instrucciones simples, que el microprocesador puede entender, en un determinado orden.

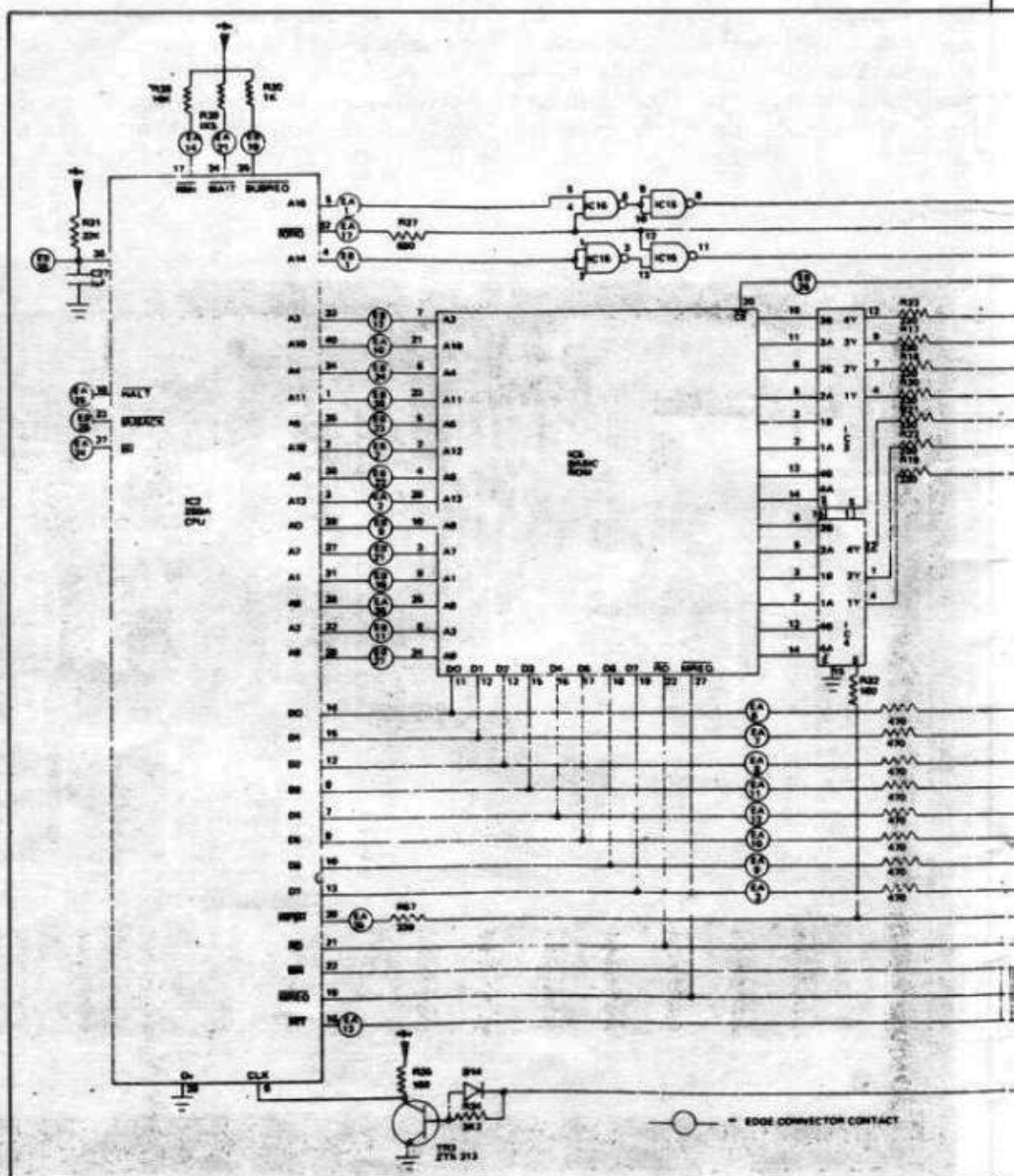
La UCP puede, por medio de su juego de instrucciones, llevar a cabo distintos tipos de tareas, que serían: Lectura de datos a partir de un dispositivo de entrada. Enviar (escribir) datos hacia un dispositivo de salida. Leer datos existentes en la memoria del sistema. Escribir datos en la memoria, y efectuar operaciones aritmético-lógicas con los datos previamente capturados.

Para llevar a buen término todas estas operaciones es necesario establecer un mecanismo que sincronice el orden en que se efectúan los pasos necesarios para ejecutar cada instrucción. Un reloj sería el instrumento equi-

valente para misión parecida en la vida real. Reloj es como se ha dado en llamar al circuito que genera una señal electrónica, que sirve para el control del ordenador. Se utiliza un cristal de cuarzo que vibra con una frecuencia extremadamente estable y proporciona el cronometraje más idóneo.

El Z-80A puede aceptar señales de hasta 4 MHz (4 millones de señales por segundo). Sin embargo, por motivos de diseño, la señal empleada en el Spectrum es de 3,5 MHz. Cada vez que esta señal produce una variación de tensión, de 0 a 1 y de 1 a 0 (lógicos), el microprocesador acepta como orden para realizar el siguiente ciclo de la instrucción que se está ejecutando.

Por las 40 patillas del encapsulado, el microprocesador envía o recibe (o ambas cosas) señales eléctricas con un cometido específico. Dieciséis de ellas están conectadas al bus de direcciones, poniendo en forma de unos y ceros



lógicos (tensión o notensión) la dirección de memoria o de los dispositivos de Entrada/Salida a los que desea acceder. Estas patillas, y algunas otras, ofrecen lo que se llama salida triestado, en vista de que pueden tener tres estados diferentes: el cero y el uno lógicos y un estado de resistencia eléctrica infinita, que en otras palabras produce el mismo efecto que si las patillas estuvieran desconectadas del circuito. Se designan de A0 hasta A15.

Otras ocho patillas, bautizadas D0-D7, están destinadas a la entrada y salida de datos y conectadas al bus que lleva el mismo nombre. Son también

triestado e intercambian los bytes de datos con la memoria y los dispositivos de E/S.

Otra patilla sirve para indicar a la memoria que la señal presente en el bus de direcciones es correcta y que el microprocesador desea acceder a ella.

La indicación de si la operación que desea efectuar la UCP es leer o escribir un dato, es llevada a cabo por otra patilla. El resto de las patillas están destinadas a operaciones de interrupción del funcionamiento del microprocesador, al refresco de las memorias RAM dinámicas, a la puesta a cero del sistema, la indicación de que se desea

efectuar una operación de E/S con un periférico y las señales de reloj.

Según vimos anteriormente, ambas versiones del Spectrum disponen básicamente de 16 Kbytes de memoria RAM, ampliándose en 32 Kbytes más para el modelo de 48 K. En ella se contienen varias cosas: un espacio de trabajo, que el ordenador utiliza mientras ejecuta los programas, diversas variables requeridas por el BASIC, la información sobre el contenido de lo que aparece en la pantalla del televisor, los gráficos que pueda haber definido el usuario y, por último, el programa en BASIC.

El tipo de memoria RAM empleado es del tipo llamado dinámica. Su comportamiento es análogo al de un condensador eléctrico: cada celdilla que contiene un bit debe ser "refrescada" al menos una vez cada 2 milisegundos; en caso contrario, como ocurre con un condensador real (tiene corrientes de fuga), se perderá irremisiblemente la información contenida. El microprocesador lleva prevista tal eventualidad. De un modo que resulta totalmente transparente, tanto para el programador como para el funcionamiento del ordenador en sí, el microprocesador accede, una a una, a todas las direcciones de memoria, leyéndolas y volviendo a escribir su contenido original, restituyéndolo antes de su pérdida total. Esto es efectuado por el microprocesador continuamente, en determinada fase del ciclo de ejecución de cada instrucción.

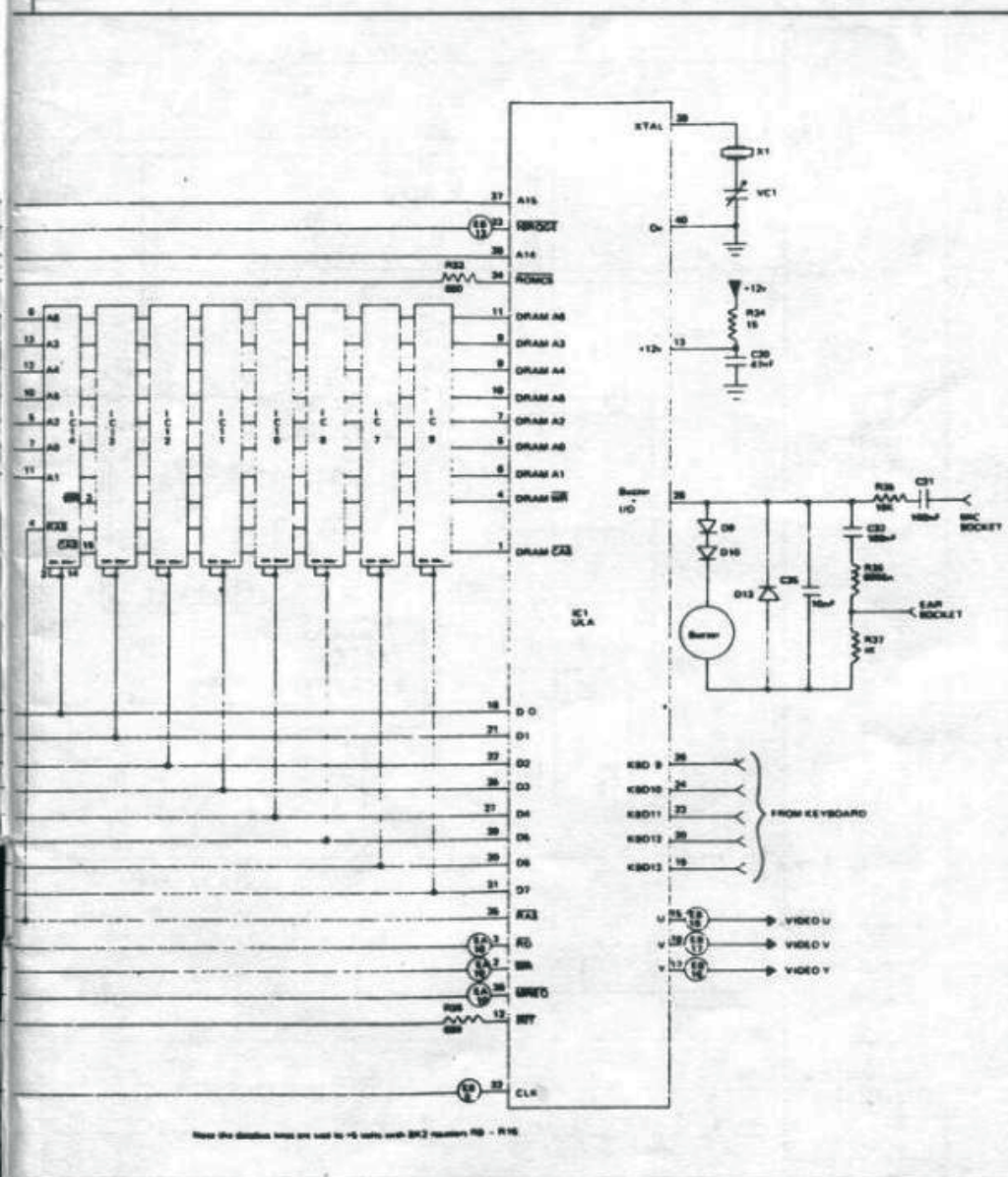
La ULA también accede a la memoria para tomar de ella la información, que ha de ser visualizada en la pantalla en determinados intervalos de tiempo.

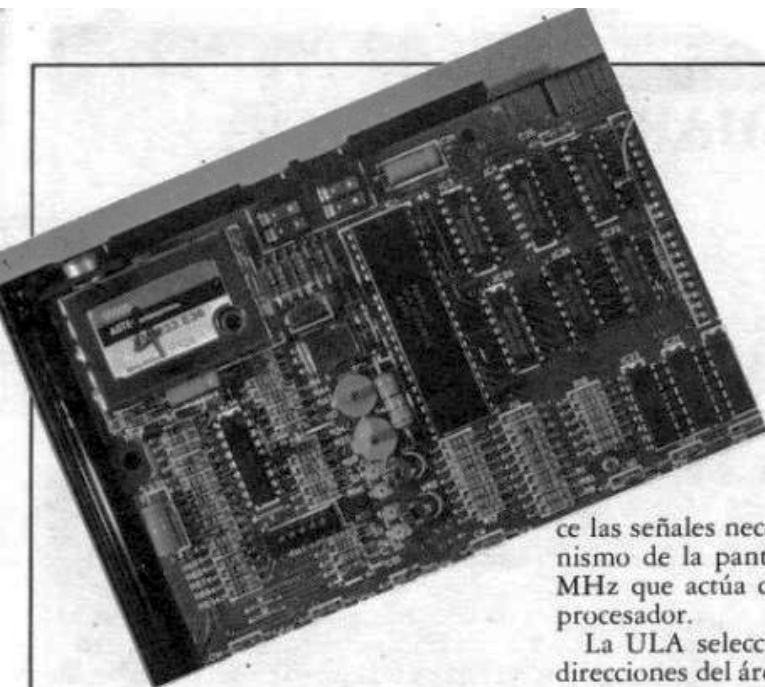
Los 16 Kbytes de RAM están formados por 8 chips de memoria. Tienen una capacidad de 16 Kbits cada uno, que multiplicado por 8 producen los bytes indicados. En el mapa de memoria ocupan las direcciones hexadecimales comprendidas entre 4000 y 7FFF.

En el modelo de 48 K, los 32 Kbytes adicionales ocupan las direcciones 8000 a FFFF y se consiguen con otros 8 chips de memoria dinámica, sólo que esta vez son de 32 Kbits.

Los 16 Kbytes más bajos en el mapa de memoria son ocupados por la memoria ROM, las direcciones comprendidas entre 0000 y 3FFF (hexadecimales). Físicamente se trata de un chip encapsulado en una pastilla de 28 patillas y contiene el programa monitor. Es un

Esquema teórico del Spectrum, los circuitos integrados van representados en forma de bloque generalmente.





Detalle de la circuitería de video, con el modulador de UHF, el cristal de cuarzo y la ULA (el CI de mayor tamaño).

programa escrito en el código máquina del Z-80 y se divide en tres partes principales: las rutinas de Entrada/Salida, el intérprete del BASIC y el manejo de expresiones (sintaxis).

Las patillas reservadas a las direcciones son solamente 14, puesto que con una palabra de ese número de bits se pueden direccionar perfectamente los 16 K de memoria ($2^{14}=16\text{ K}$). Las 8 reservadas para los datos proporcionarán el byte escrito en la dirección de memoria accedida. Otras patillas que le dicen a la ROM si realmente la UCP se está dirigiendo a ella.

La ROM viene grabada desde un principio y es inalterable, revelándose imposible cambiar sus contenidos por más que se desconecte la alimentación o se intente desde el teclado.

Aparece una interesante línea, accesible desde el slot de expansión, etiquetada ROMCS que, cuando se le aplica la señal uno lógico (+5V en este caso), anula el funcionamiento de la ROM en tanto permanezca esta señal. Tiene bastante importancia a la hora de diseñar determinados accesorios que necesiten por alguna causa bloquear la ROM temporal o definitivamente durante su actuación.

La ULA vimos que sustitúa por sí sola a un elevado número de chips más simples. Es un chip diseñado específicamente para el Spectrum y se encarga de todas las tareas de Entrada/Salida del ordenador. Entre sus funciones incluye la generación de la señal de video, que aparece en la pantalla; la lectura del teclado y las operaciones de lectura y escritura en el cassette. También se encarga de que el zumbador acústico funcione adecuadamente.

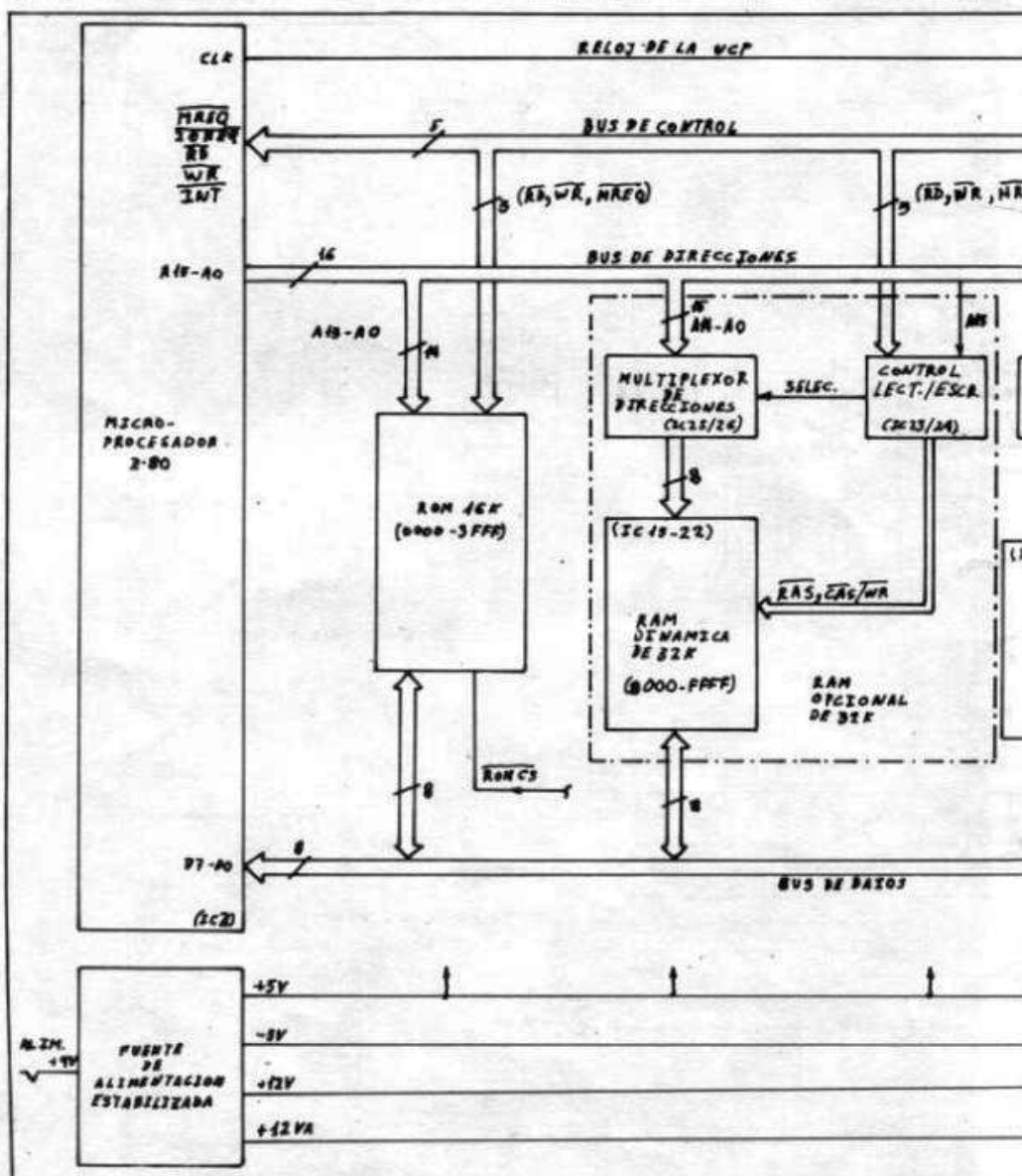
Un circuito de la ULA diseñado para mantener la oscilación permanentemente en el cristal de cuarzo, se encarga de obtener una señal de 14 MHz que, una vez tratada adecuadamente, produ-

ce las señales necesarias para el sincronismo de la pantalla y la señal de 3,5 MHz que actúa como reloj del microprocesador.

La ULA selecciona directamente las direcciones del área de la RAM destinada a contener la información de la pantalla. En ella hay 6 Kbytes reservados a tal fin, organizados como mapa de memoria o, lo que es igual, a cada una de esas direcciones de memoria le corresponde un punto de la pantalla y viceversa. La ULA tiene la habilidad de

poder detener el funcionamiento del reloj de 3,5 MHz durante la generación de la imagen, engañando a la UCP. La razón es simple, los tiempos necesarios para formar la imagen deben ser extremadamente precisos y en un momento dado puede aparecer el conflicto. Podría ocurrir que simultáneamente la UCP desee hacer algo en las direcciones de memoria correspondientes a la imagen y que la ULA tenga necesidad de tomar el dato a visualizar situado en la misma dirección. En tal caso, la preferencia es de la ULA. Si quien espera es la UCP, no hay efectos exteriores aparentes. En cambio, si espera la ULA puede crearse una inestabilidad en la imagen.

Podemos resumir esta actuación diciendo que la ULA genera las señales de campo y de línea para el televisor, y



HARDWARE

entre los topes de sincronismo va metiendo las señales de video correspondientes a cada línea, proporcionando la imagen final.

El modulador de crominancia es controlado para que genere las señales en los 8 colores seleccionables. La última etapa de esta cadena es un modular de UHF, que produce la señal de alta frecuencia que ataca la entrada de antena del televisor, produciéndose los gráficos de alta resolución y el formato de 32 por 24 caracteres.

El teclado es, en esencia, una rejilla de 8 por 5 hilos. Los dispuestos en sentido vertical están aislados de los horizontales. La intersección de cada uno de ellos puede conectarse eléctricamente por por medio de las teclas. La información sobre si una tecla ha sido presionada, le

llega al sistema por medio de los bits 11 a 15 del bus de direcciones y cinco conexiones a la ULA. Esta lee el teclado unas 50 veces por segundo, para cerciorarse de que ha sido presionada alguna tecla. El proceso se realiza aprovechando el instante anterior al inicio de una nueva imagen (se generan también 50 cuadros por segundo en la norma europea de TV utilizada en España), que es el momento más idóneo para interrumpir el funcionamiento de la UCP.

Los cables planos que aparecen bajo la carcasa del Spectrum son las conexiones físicas de esa matriz de filas y columnas del teclado con la circuitería del ordenador. Para guardar (SAVE) y volver a recuperar (LOAD) los programas en *cassette*, se hace necesario un *interface* que adecúe los niveles de las

tensiones empleadas por el ordenador y el dispositivo de *cassettes*, para obtener un perfecto entendimiento. Este trabajo lo efectúa la ULA, que transfiere la información entre los *jacks* de auricular y micrófono y el bus de datos. Para ello tiene que ser capaz de realizar determinado tipo de conversiones de las señales analógicas (continuas) en digitales (unos y ceros).

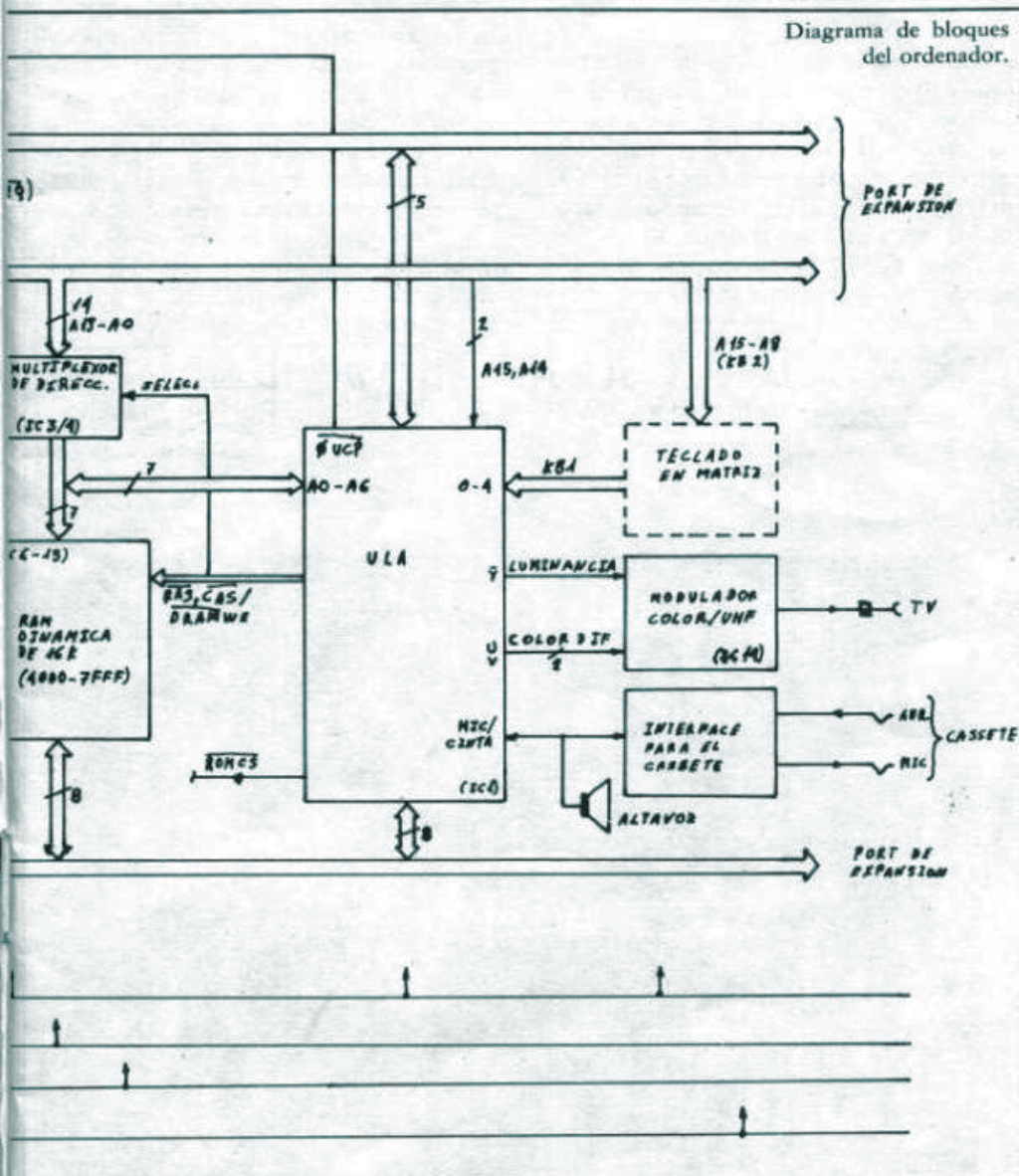
Durante el LOAD, la UCP realiza ciclos de lectura sucesivos, leyendo lo que el *cassette* proporciona a través del conector de auricular a la línea 6 del bus de datos (conviene recordar que los datos son leídos y grabados en serie, es decir, un bit detrás de otro). Para SAVE, la UCP repite sucesivos ciclos de escritura, enviando los datos a la entrada del micrófono por la línea de datos número 3 del bus.

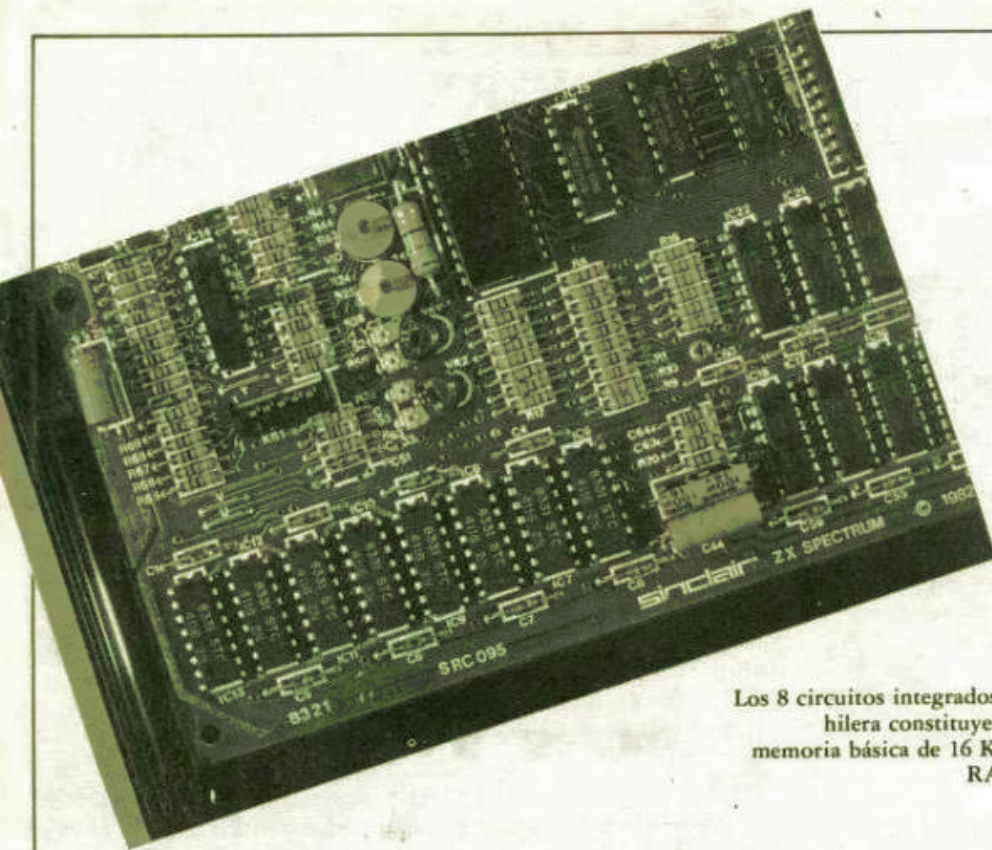
Observando el esquema, puede parecer paradójico que ambas conexiones al *cassette* y el zumbador estén conectadas a la patilla 28 de la ULA. Sin embargo, operan de manera independiente entre sí. Las operaciones antes descritas, al igual que la excitación del zumbador, se producen por configuración de la dirección 254, utilizada como *port*, empleándose los bits de datos 3 y 6 para el *cassette* y el 4 maneja al zumbador. Unos pocos componentes asociados se ocupan de que no haya conflicto y que interfieran su funcionamiento. De todas formas, si se conecta un amplificador a los conectores MIC o EAR, se escuchará la señal de audio generada en el zumbador, pero con mayor volumen.

La fuente de alimentación se encarga de proporcionarle al ordenador el suministro de corriente que necesita para su funcionamiento. El *ZX power pack* consiste en un transformador reductor de tensión que, después de rectificarla, produce una salida de 9 voltios no regulados.

Sin embargo, los circuitos integrados del Spectrum necesitan otras tensiones totalmente estabilizadas. Los circuitos lógicos integrados, incluyéndose el microprocesador, la ULA y el modulador de UHF necesitan +5 voltios. La RAM dinámica precisa +5 V, -5 V y +12 V. La ULA necesita una segunda alimentación de +12 V, igual que el modulador de color.

La mayor parte de estos circuitos son extremadamente sensibles a los cambios de tensión. Si es algo mayor de lo requerido, se corre el riesgo de inutilizar





Los 8 circuitos integrados en hilera constituyen la memoria básica de 16 K de RAM.

los circuitos integrados. Si, por el contrario, es menor, las señales lógicas pueden ser tan vagas que no serán interpretadas correctamente como unos y ceros. Por otro lado, conociendo las extremadas velocidades con que operan los circuitos, una caída de tensión que nos pasaría desapercibida, podría conducir a errores.

La fuente de +5 V queda controlada por un regulador integrado de conocida experiencia, el 7805, circuito integrado que estabiliza la tensión a los voltios requeridos, y está encapsulado en una pequeña cajita plástica de la que surgen 3 patillas, semejando un transistor. Dos condensadores electrolíticos le ayudan en la tarea, almacenando pequeñas cantidades de carga eléctrica.

Las tensiones de -5 V y +12 V se producen de manera algo más compleja. Se recurre a un circuito conocido por inversor. A partir de las tensiones +5 V regulados y +9 V sin regular, se ataca al inversor, que entra en estado de oscilación, proporcionando corriente alterna, que adecuadamente rectificadora y alisada proporciona los +12 V. Rectificando esta señal en sentido inverso y estabilizándola con un diodo zener, se obtienen los -5 V y los -12 V requeridos. El principio básico del inversor es simple, una tensión continua se convierte en otra alterna de mayor valor, por medio de la oscilación, después con ayuda de diodos y condensadores de gran capacidad se rectifica y alisa, para obtener el resultado deseado.

Sobre la tarjeta de circuito impreso se

localiza una amplia aleta de aluminio, unida físicamente al 7805 por medio de un tornillo. Su misión consiste en disipar el calor generado por este regulador, que a veces puede ser captado por la mano desde el exterior cuando el ordenador lleva un rato trabajando.

Desde el slot de expansión se puede acceder a las tensiones de alimentación de +5 V, +9 V y +12 V, cuando se conectan dispositivos periféricos. En caso de ser necesarios más de 300 mA, habrá que utilizar otra fuente de alimen-

tación, para no sobrecargar la que suministra fluido al ordenador.

Cuando el usuario de un Spectrum de 16 K quiere ampliarlo hasta los 48 K, es necesario añadirle los 32 K restantes de RAM, enchufándolos a los zócalos que aparecen vacíos en la tarjeta principal, destinados a este fin.

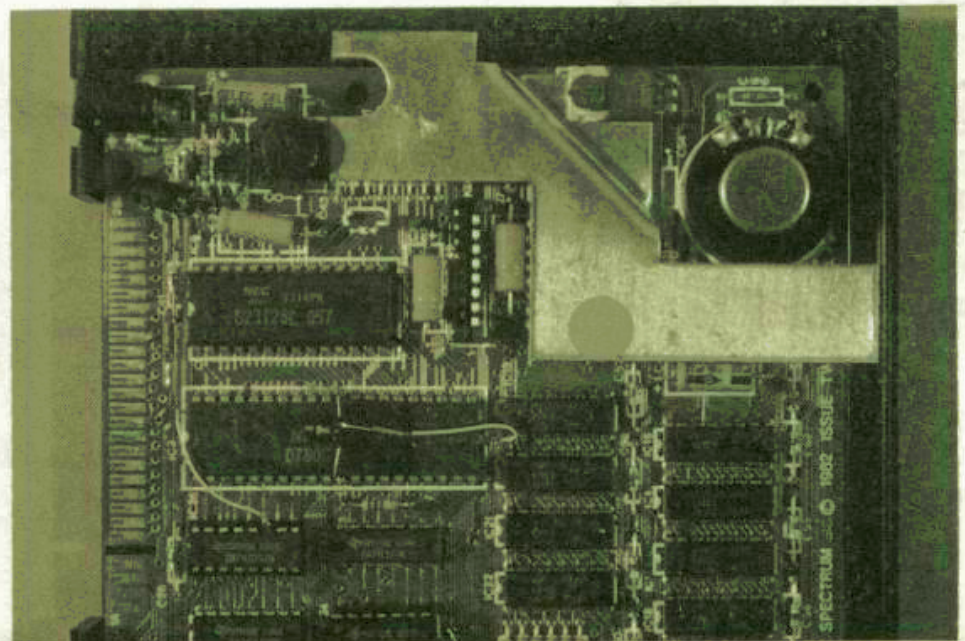
El slot para la expansión, situado en la parte posterior del Spectrum, proporciona acceso directo a los buses del sistema. A través de él se pueden conectar un número de elementos periféricos, tales como la impresora ZX, los interfaces para joysticks y cartuchos, el interface para el microdrive y la red local, y un sinnúmero de accesorios de otros fabricantes, tales como los proporcionados por Indescomp.

Cualquier usuario podría fabricarse sus propios periféricos y ampliaciones. Para ello se requiere un buen conocimiento del funcionamiento del microprocesador Z-80 y de su código máquina, que puede aprenderse en los manuales que edita el fabricante. Casi todas las señales existentes en los buses tienen que ver directamente con este chip.

En las páginas de esta revista iremos proporcionando algunos montajes que ampliarán las posibilidades de utilización del Spectrum (y del ZX-81 también).

Roberto Menéndez

El circuito integrado microprocesador, Z-80A, es el que lleva sobre sí el transistor tan pintorescamente colocado. El otro es la ROM.



No le ha faltado imaginación ni ambiciones a Sir Clive Sinclair a la hora de bautizar al nuevo ordenador salido del esfuerzo de su equipo de diseñadores. La novedad, presentada con bombos y platillos a la prensa británica el 12 de enero se llama, simplemente Sinclair QL (por *quantum leap*, cuya mejor traducción nos parece "salto cuántico". Efectivamente, Sinclair está convencido de que su nuevo modelo representa un paso de gigantes en el mercado de los ordenadores. Sus rivales son —y lo dice con una sonrisa— el IBM/PC y el Apple IIe, sólo que —siempre según sir Clive— estas máquinas se venden a precios cuatro veces superiores al del QL.

¿Qué precio? De momento, 399 libras que, traducidas a la actual paridad representan algo más de 550 dólares (nos abstenemos de hacer el cálculo en pesetas para no crear falsas ilusiones, pero cualquiera puede hacerlo). En esa misma línea de precios, en los mercados británico y americano, se venden ordenadores de prestaciones muy inferiores.

Se trata de un equipo de 16 bits, basado en el microprocesador Motorola 68000. Lleva 128 Kbytes de memoria RAM, pero Sinclair anuncia que saldrá simultáneamente una tarjeta de expansión que puede llevar la memoria del QL hasta 640 Kbytes. Dos *microdrives* se encargan del almacenamiento masivo.

Hay dos aspectos que constituyen una riesgosa apuesta de Sinclair. Uno es el sistema operativo propio, llamado QDOS, lo que le hace absolutamente incompatible con toda otra máquina existente hasta la fecha. Por otra parte, el lenguaje implementado es una versión exclusiva del BASIC, a la que se ha dado en llamar



Sinclair Super BASIC. Durante la rueda de prensa de presentación, sir Clive se sintió lo bastante eufórico como para comentar: "suele criticarse mucho al BASIC porque es un lenguaje con veinte años de antigüedad. Nosotros hemos corregido todos los errores que el BASIC clásico contiene, por eso lo llamamos Super BASIC".

Los cuatro programas disponibles de momento —*word processing*, tabla de

análisis financieros, base de datos y gráficos de negocios— han sido producidos para Sinclair por Psion.

A propósito del Spectrum conviene decir, que el QL no es compatible. En cambio, sí lo es, a nivel de conexión, con las dos *interfaces* y el *microdrive* lanzados recientemente por el mismo fabricante y que todavía no han llegado a España. En cambio, el QL no tiene *modem* ni facilidades para un disco du-

conectar con un disco duro que podría darle una capacidad de almacenamiento de masa aún más potente.

Como es tradicional en los productos desarrollados por Sinclair Research, el QL será fabricado por una firma externa, en este caso Data-tech, una filial de Thorn EMI, que prevé lanzar los primeros ejemplares en serie hacia finales de febrero, hasta alcanzar una producción mensual de 20.000 unidades.

Seguramente, volveremos a hablar de este ordenador ampliamente en estas páginas. Por ahora, sólo queremos destacar dos aspectos: 1) el QL no es una versión avanzada del Spectrum sino un producto radicalmente diferente; 2) esto no excluye la posibilidad de lanzamiento de un nuevo modelo, sin que se sepa con certidumbre si el QL es el célebre ZX83 de que tanto se ha hablado o si este último sigue esperando su hora.

Hizo falta esperar dieciocho meses para que Sinclair Research sacara al mercado (de momento sólo en Gran Bretaña, y en cantidades limitadas) sus prometidos *Microdrive* e *Interface 1* (ver ZX n.º 1). Los problemas han tardado menos en aparecer. El nuevo interface RS232 no está resultando tan estándar como se preveía, a juzgar por los comentarios que comienzan a llegarnos desde Londres. Al parecer, el problema reside en que el dispositivo puede transmitir datos perfectamente, pero la recepción de los mismos es complicada. Es decir, no encontrará problemas para utilizar una impresora, pero si se trata de conectar con un *modem* ya es otro cantar. La razón es simple: Sinclair, para reducir costos sin sacrificar márgenes de beneficio, optó por una solución de difícil solución técnica.

En concreto, el Interface 1 apenas lleva *hardware*, y todo el control se realiza por *software*, que resulta más barato. El Rx (*receive*), Tx (*transmit*), DTR y CTS están directamente conectados, pero la de-

codificación de señales se hace por programa a través de la ROM.

El primer problema planteado surge del hecho de que el Interface 1 presupone que cualquier *hardware* conectado con él usa el CTS para indicar el comienzo de un byte. Es decir, que el Spectrum siempre "pensará" que está en el comienzo de un byte, cuando puede estar en el medio. La única forma de evitarlo es transmitir los datos lo suficientemente despacio como para permitir al *software* la recogida de información.

El segundo problema, no menos importante, es que la decodificación se realiza por *software*. El Spectrum tendría que estar leyendo la información del port RS232 todo el tiempo. No hay tiempo para otra actividad. Por ejemplo, mientras se imprime un dato en pantalla, se habrá perdido la mitad del siguiente byte, porque sin DTR y CTS el sistema emisor desconoce que el Spectrum no está preparado para recibir información.

COMENTARIOS

aunque carezca de una espectacularidad que no necesite. Es una aplicación bastante útil para saber dónde se nos van las pesetas y poder controlar los gastos.

Calificación: ■■■■□

Programa: Horizontes

Tipo: Educativo

Autor: PSION

Distribuidor: Investrónica, S. A.

Formato: Cinta de cassette ZX Spectrum (16 ó 48 K)

Programa didáctico orientado a todos aquellos que son nuevos en el uso del Spectrum. Pretende dar una serie de conocimientos básicos sobre el manejo y posibilidades del ordenador.

El usuario puede cargar fácilmente la cinta en la máquina, ya que las instrucciones contenidas en la etiqueta de la cinta del cassette para la utilización correcta del programa no plantean complejidad alguna, con excepción de algunos errores. Una vez cargada la cinta en el ordenador, sólo es preciso esperar las instrucciones que dé el programa.

La cinta contiene dos partes que se corresponden con sus dos caras. La cara A comienza con una descripción del hardware del Spectrum. Esta parte técnica quizás resulte confusa para el nuevo usuario, pero para los objetivos que persigue el programa es suficiente. A continuación incluye cuatro lecciones

explicativas del teclado. Para el nuevo usuario este aprendizaje presentará más dificultad, debido a las diferentes funciones asignadas a cada tecla, sobre todo teniendo en cuenta que es probable



que no conozca el significado de las instrucciones del lenguaje BASIC. No obstante, estas lecciones están apoyadas por un diccionario de términos del lenguaje.

La cara B contiene una muestra de juegos que permite conocer la capacidad creativa del Spectrum. Los objetivos de esta segunda cara son: primero, que el usuario una vez aprendida la cara A, practique jugando y segundo, que conozca algunas de las aplicaciones prácticas y lúdicas de su ordenador.

En general, el programa cumple su objetivo, que es el de enseñar a manejar el ordenador al nuevo usuario. La presentación es atractiva e invita a adentrarse en el mundo del Spectrum. Es recomendable que el fabricante cuide más los errores en las instrucciones, pues esto puede crear confusión en el usuario.

Calificación: ■■■■□

MICRO **M** WORLD

EL MUNDO **sinclair** A TU ALCANCE

Descubre el apasionante mundo de la informática de mano de profesionales:

- Sinclair ZX - 81: 14.975,- Ptas.
- ZX Printer: 17.100,- Ptas.
- Sinclair ZX Spectrum (16 K): 39.900,- Ptas.
- Extensa variedad de Software nacional y de importación.
- (48 K): 52.000,- Ptas.
- Asesoramiento gratuito

CURSILLOS DE BASIC PARA EL SPECTRUM Y BIBLIOGRAFIA



REGALAMOS UN CASSETTE
CON 5 PROGRAMAS A
TODOS NUESTROS CLIENTES

MODESTO LAFUENTE, 63 · TELEFONO: 253 94 54 · MADRID · 3

Pregunta: Tengo un ZX81 y he intentado meter varios de vuestros programas, pero no han funcionado, como el Bowling para 1 Kbyte o el Fantasma y el de Un caso de probabilidad, que pone que es de un 1 Kbyte. El ordenador me da falta de memoria, pues no tengo amplificador, y además hace poco tiempo que la tengo y todavía no domino el BASIC.

Juan Carlos. Madrid.

Respuesta: Es extraño que no hayan funcionado por falta de memoria los programas que cita, ya que fueron preparados para 1 K. Nos parece posible que tal vez su ZX no funciona bien o, como usted mismo dice, al no dominar el BASIC haya metido algo "extraño" sin darse cuenta. Pero no es normal que ello ocurra en varios juegos. Pruebe con algún otro programa de los que requieren sólo 1 K de memoria; si le sigue fallando, es que su ordenador necesita una revisión. Aprovechamos la ocasión para tranquilizar a muchos lectores que nos llaman para plantearnos lo que suponen son errores en la transcripción de nuestros programas. Todos ellos han sido probados en la máquina que en cada caso se indica.

Pregunta: ¿Podrían incluir algún programa de Bingo en el próximo número.

Juan Cabezas. Madrid

Respuesta: Lo intentaremos. No es usted el primero en pedirnos un programa de este tipo, pero no tenemos ninguno disponible. Pensándolo bien, no es mala idea para otros lectores. Si hay alguno que nos puede proporcionar un programa, será bienvenido y recompensado.

Pregunta: ¿Podrían explicar cómo generar caracteres gráficos con el ZX Spectrum? Veo que los utilizan en los juegos de su revista, y no sé cómo meterlos en mi ordenador.

Gregorio. Salamanca

Respuesta: Este es uno de los puntos más oscuros en el manual del Spectrum y, en consecuencia, uno de los que más dudas suscitan.

En nuestro número 2 —que usted no había leído al enviarnos su carta— ya indicábamos ambas cosas: cómo generar caracteres gráficos y cómo utilizar un programa que trabaje con dichos caracteres.

No obstante, para mayor claridad y sencillez, y para que pueda ser útil a otros lectores de la revista, veamos otra forma de utilizar un programa preparado con caracteres gráficos, como ocurrirá casi siempre que utilizemos un programa de juegos.

Básicamente, tenemos dos opciones: generar el carácter gráfico previamente a la introducción del programa o introducir éste de forma posteriormente.

El primer paso es el que veíamos en el número anterior, lo cual requeriría tener ciertos conocimientos del lenguaje BASIC para localizar las instrucciones que daban lugar al carácter gráfico.

La segunda posibilidad es más simple, por cuanto no necesita tener conocimiento alguno del lenguaje. Es decir, se introduce el programa línea a línea. Cuando aparece un carácter gráfico se introduce de acuerdo con la información sobre notas gráficas, si bien no se representará dicho carácter ya que no se ha generado todavía. Si sigue así hasta terminar el programa y después, como siempre, se ejecuta dicho

programa (es decir, se presiona RUN y ENTER).

Veámoslo con un ejemplo:

Con la primera opción, la generación previa del carácter, puede introducir la instrucción 30 y siguientes, que son las que cargan el carácter gráfico correspondiente en la "a" y en la "b". Después, presiona RUN y ENTER e introduce posteriormente las restantes sentencias. Cuando llega a la 20 cambia al modo gráfico y presiona la "a" y la "b", con lo que obtendrá el carácter gráfico correspondiente.

Para la generación de ca-

racteres gráficos puede adoptar diversas modalidades (información en binario, decimal, con utilización de bucles...) como veíamos en el número anterior. Existe un camino más fácil: introducir directamente el programa. Cuando se encuentre con un carácter gráfico, lo introduce de acuerdo con la información sobre notas gráficas que acompaña el programa. No le aparecerá el carácter gráfico inicialmente.

Notas gráficas:
Línea 20 - Graphic

10 GO SUB 30

20 PRINT AT 0,0;"AB": STOP

```
30 POKE USA "a"+0,BIN 00011000
40 POKE USA "a"+1,BIN 00111100
50 POKE USA "a"+2,BIN 01111110
60 POKE USA "a"+3,BIN 11111111
70 POKE USA "a"+4,BIN 01111110
80 POKE USA "a"+5,BIN 00111100
90 POKE USA "a"+6,BIN 00011000
100 POKE USA "a"+7,BIN 00000000
110 POKE USA "b"+0,BIN 10000000
120 POKE USA "b"+1,BIN 11000000
130 POKE USA "b"+2,BIN 11100000
140 POKE USA "b"+3,BIN 11110000
150 POKE USA "b"+4,BIN 11111000
160 POKE USA "b"+5,BIN 11111100
170 POKE USA "b"+6,BIN 11111110
180 POKE USA "b"+7,BIN 11111111
```

Pero después, cuando ejecute el programa, su Spectrum realizará el cambio automáticamente. Compruébelo, pero no olvide pulsar la

letra correspondiente en modo gráfico (Graphic A y Graphic B en nuestro ejemplo).

10 GO SUB 30

20 PRINT AT 0,0;"AB": STOP

```
30 POKE USA "a"+0,BIN 00011000
40 POKE USA "a"+1,BIN 00111100
50 POKE USA "a"+2,BIN 01111110
60 POKE USA "a"+3,BIN 11111111
70 POKE USA "a"+4,BIN 01111110
80 POKE USA "a"+5,BIN 00111100
90 POKE USA "a"+6,BIN 00011000
100 POKE USA "a"+7,BIN 00000000
110 POKE USA "b"+0,BIN 10000000
120 POKE USA "b"+1,BIN 11000000
130 POKE USA "b"+2,BIN 11100000
140 POKE USA "b"+3,BIN 11110000
150 POKE USA "b"+4,BIN 11111000
160 POKE USA "b"+5,BIN 11111100
170 POKE USA "b"+6,BIN 11111110
180 POKE USA "b"+7,BIN 11111111
```


Pregunta: ¿Me podrían dar alguna dirección en España para la reparación de un Spectrum 48K, adquirido en Inglaterra hace más de un año y que, por tanto, no tiene garantía? Ruego su contestación por ser éste un tema en el que creo que estará interesada mucha gente que puede, o podrá, encontrarse en mi caso.

Carlos Contreras. Madrid

Respuesta: Lea la respuesta a otro lector en esta misma página. Que tenga suerte.

Pregunta: Tengo un ZX81 de 1 K y decidí ampliar la memoria, para lo cual utilicé memoria del tipo 2114 de 4 bits 1 K y un decodificador de direcciones del tipo 74154, aparte de los *buffer* de direcciones y datos necesarios para no cargar en exceso al bus del sistema. La capacidad inicial debería ser de 4 K que, unidos al K que ya posee el sistema, me proporcionan 5 K de RAM. Cuando terminé y conecté el ZX y mi placa, descubrí que el ordenador aceptaba mi nueva memoria (hice PEEK 16388 + 256 * PEEK 16389 Pero cuando intenté hacer algún programa no obtuve más que bloqueos del sistema. Después me di cuenta de que la RAM del

sistema está direccionada no sólo en su sitio correspondiente (de la 16384 a la 17407), sino que hay una especie de copia de la misma en todas las direcciones hasta la 32767.

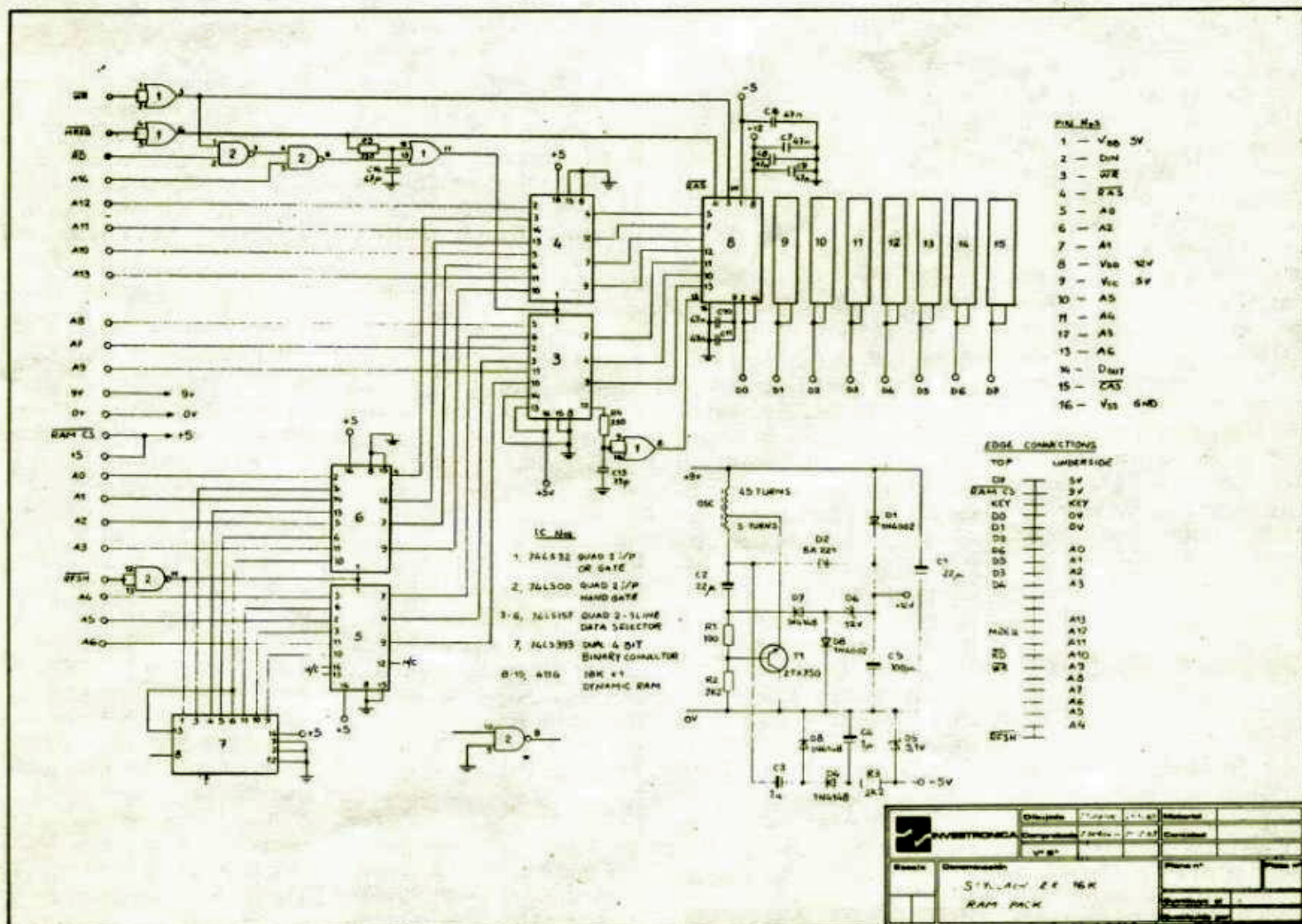
Creo que la solución está en inhibir la memoria interna del sistema, cuando esté direccionando a mis RAM's, usando para ello la patilla RAM CS del conector de expansión. Esta posibilidad también la he probado, pero sin obtener ningún resultado. Incluso probé a inhibir permanentemente esta RAM conectando el terminal permanentemente al positivo de la alimentación. Podría pensar que mi tarjeta está defectuosa, pero en este caso el ZX no aceptaría las

nuevas posiciones tal como indica.

¿Podrían decirme cómo solucionar este problema?

J. Mateu. Barcelona

Respuesta: Nos pone usted en un aprieto. Resulta imposible reconstruir teóricamente el camino seguido por usted, y tampoco podemos meternos a hacerlo en la práctica. En principio, la ampliación de memoria no debiera ofrecer tantos problemas. Para ayudarlo a resolverlos, reproducimos aquí el esquema de la ampliación de memoria de Sinclair ZX81 para 16 K, que nos ha sido facilitado por los técnicos de Investronica. Esperamos que le sirva de modelo para completar su experimento.



LECTORES

Pregunta: He buscado por toda Barcelona esta revista tan interesante, pero en todos los sitios estaba agotada. Les pido que me envíen el número 1. Me gustaría también que me enviaran un catálogo de todos los juegos de ZX Spectrum.

Xavier Ferreras i Escala.
Hospitalet (Barcelona)

Respuesta: Gracias por decir que ZX es una revista interesante. Entre todos lograremos hacerla realmente interesante, ya que en una revista como ésta la participación de los lectores es imprescindible.

Respecto a su petición del número 1, efectivamente se agotó a los pocos días de salir, y por lo que sabemos, algo parecido ha ocurrido con el número 2, a pesar de haber aumentado la tirada. De todos modos, ya está en la calle una reimpresión del primer número, que puede buscar en quioscos o pedir a nuestras oficinas.

Acerca de la segunda pregunta, por el momento no existe ningún catálogo completo del juego para Spectrum. Estamos preparando un *dossier* sobre el tema, recurriendo a las distintas fuentes, pero nos llevará tiempo. Si necesita alguno con urgencia, puede usted dirigirse a **Investrónica** (Tomás Bretón, 60, Madrid, o Muntaner, 565, Barcelona) y a **Indescomp** (Paseo de la Castellana, 179, Madrid).

años, y os ruego que si me lo podéis dar un poco simplificado, mejor.

José M. Macías Nieto.
Cáceres

Respuesta: Nos alegra comprobar el entusiasmo con el que, desde temprana edad, se acoge al Spectrum. Para atender a necesidades como la tuya ha nacido, precisamente, esta revista. Tu carta nos ha hecho recapacitar sobre la necesidad de incluir temas destinados a quienes, como tú, se incorporan al mundo de ZX. Estamos pensando en crear una sección Principantes, en la que se trataría de simplificar todo al máximo.

Pregunta: Poseo un ZX Spectrum que tiene estropeado el color por un fallo en la alimentación. Al ser adquirido en Andorra, no tengo facilidad para repararlo. ¿Pueden darme alguna idea de quién puede arreglarlo?

Joaquín Paredes Pardo.
Alicante

Respuesta: Son muchos los usuarios que, aprovechando el menor precio al que se vende el Spectrum en otros mercados, lo han traído por su cuenta. Hay también importadores, más o menos irregulares, que lo venden más barato, pero no dan garantías de reparación.

Esta situación está creando, pensamos, las condiciones para que aparezcan talleres de reparación independientes del distribuidor autorizado de Spectrum. Por ahora sólo conocemos una firma que se dedica a reparar ordenadores. Estos son sus datos: **Computest**, calle Balandro, 15, Alameda de Osuna, Madrid. Teléfono: (91) 416 73 85.

Pregunta: ¿Qué hay que hacer para conseguir el número 1 de ZX? ¿Me lo podéis enviar contra reembolso, o tendré que recurrir al soborno, chantaje u otras malas artes? He recorrido 19 quioscos y librerías distintas en Barcelona y otras dos poblaciones y no he podido encontrarlo.

Y para terminar, en tres lugares distintos del número 2 de ZX (páginas 3, 10 y 50) habláis del nuevo modelo de Sinclair. Incluso decís que para este año. ¿Qué tal si nos habláis un poco de de esto? ¿Nos compramos el Spectrum o esperamos?

Jaume Nosas Cánovas.
Castelldefels (Barcelona)

Respuesta: Como respondemos a otro lector, ya está a la venta la reimpresión del número 1, lo que ahorrará el tener que recurrir a malas artes para hacerse con él.

Respecto al nuevo modelo de Sinclair, debemos decirle que la empresa británica tiene un refinado gusto por el secreto. Se hablaba mucho el año pasado de un futuro modelo ZX 83 (que ahora debería llamarse ZX84), pero todo lo que se dice sobre sus características es pura especulación. Entre tanto, Sinclair acaba de sacar un ordenador para negocios, llamado QL, sobre el que informamos brevemente en este mismo número. Volveremos sobre el tema, naturalmente. En cualquier caso, no nos parece justificado dejar de comprar un Spectrum para esperar otro modelo. Si aplicáramos ese razonamiento, asistiríamos impasibles a los continuos avances de la tecnología.

Pregunta: ¿Las indicaciones al final de los programas de su revista significan que

sólo funcionan con ese tipo determinado de Sinclair? Por ejemplo, si pone 16K Spectrum, ¿sirve para 48 K Spectrum?

Miguel Angel Tristán
Burgos.
Madrid

Respuesta: Al final de cada programa se indican dos referencias: la capacidad mínima requerida y el ordenador. Así, cuando se indica 16 K Spectrum, quiere decir que ese programa está preparado para el ordenador Spectrum y requiere una capacidad mínima de 16 K de memoria, lo que significa que puede funcionar con más memoria, por ejemplo, 48 K, pero no en otro tipo de ordenador, como sería el caso de un ZX81.

Pregunta: Tengo un par de amigos y tengo dudas que me gustaría resolver: ¿Es compatible el ZX Microdrive con el ZX81? ¿Puede utilizarse un monitor de fósforo para el ZX81?

Manuel García Alvarez.
Barcelona

Respuesta: Respondiendo a la primera pregunta, hemos de decirle que el *microdrive* sólo puede utilizarse con el Spectrum, ya que necesita el *interface* específico para este ordenador. No obstante, existen en el mercado otros discos, incluso de mayor capacidad, que pueden conectarse con el ZX81. Pregunte a cualquier distribuidor, pero no es seguro que todos los importen de Inglaterra. Respecto al segundo interrogante, nuestra respuesta es positiva pero con reparos. Puede conectar el monitor de fósforo, pero para ello necesita un *interface* que ha de construirse, ya que no se distribuye ninguno de ese tipo en el mercado.

Pregunta: La carta que os escribo es para pedirlos que si me podéis dar información del ZX Sinclair Spectrum, de cómo se maneja, catálogos de programas, juegos, etc.

Yo soy un chico de quince

La versión española de Popular Computing

ORDENADOR POPULAR

LA REVISTA QUE INTERESA TANTO AL AFICIONADO COMO AL PROFESIONAL

Una publicación que informa con amenidad acerca de las novedades en el campo de las computadoras personales.

ORDENADOR POPULAR, la revista para el aficionado a la informática.

Ya está a la venta el N.º 10

Cómprela en su kiosco habitual o solicítela a:

**ORDENADOR
POPULAR**

Jerez, 3
Tel. 457 45 66
MADRID-16



ZX Spectrum. Qué es, para qué sirve y cómo se usa
 Tim Langdell
 Editorial Noray
 208 páginas
 1.100 pesetas



ordenador para su localización. Son de destacar en este capítulo los consejos que, aunque escasos, son verdaderamente instructivos y útiles, independientemente del nivel de conocimientos del lector, ya que analizan particularidades del BASIC del Spectrum.

La segunda parte del libro de Langdell, que ya conocíamos en su edición inglesa de Century Publishing, es la obligada referencia al mundo de los gráficos y del sonido. En ella se explica con gran detalle la generación de caracteres gráficos, objeto de estudio detallado en el que se muestran gran diversidad de ejemplos construidos con dichos caracteres gráficos (instrucción POKE) o bien con sucesiones de puntos (instrucciones PLOT, DRAW...).

El tema de los juegos ocupa la tercera parte. Se muestran algunos, junto con las principales instrucciones que los posibilitan. Se detalla con gran profundidad distintas formas de programar un juego (a través de claves en binario, utilizando la instrucción INKEY\$ para introducción de información, rutinas para el movimiento de figuras, etc.).

Pero, como bien sabemos, el Spectrum no sólo sirve para jugar. De las distintas posibilidades de aplicación nos habla la cuarta parte del libro, especial-

mente por lo que se refiere al campo de la educación. Es un capítulo muy corto, sin demasiada conexión con el tema principal del libro, pero que sirve para que se tenga presente la flexibilidad (o "versatilidad", que dicen los fabricantes) del Spectrum, susceptible de usarse para juegos, control, de *stocks*, aplicaciones científicas, etc. Es éste un tema que, sin duda, merecería un libro aparte.

Finalmente, bajo el ambicioso título de "usar el Spectrum hasta el límite", Langdell sólo consigue desbrozar el camino hacia ese límite que nadie conoce. Se nos describen distintos trucos para mejorar la programación (uso de comentarios, remuneración de líneas, incremento de velocidad en operaciones en hexadecimal, posibilidad de "ampliar" la memoria almacenando información en los bytes reservados para la generación de caracteres gráficos), y se analiza la compatibilidad entre distintos tipos de BASIC.

Finaliza el libro con dos apéndices: la tabla de códigos ASCII y el mapa de memoria del Spectrum (que, por cierto, no se molestaron en traducir).

Un buen libro, en suma, para todos los que deseen iniciarse en el mundo del Spectrum. Un buen complemento del manual, podría decirse. Pero no podemos menos que mencionar, como la nota más crítica de este comentario, un aspecto que revela precipitación de parte de los editores: los programas que ilustran el texto de Langdell están en inglés. Es una pena que la esmerada redacción del autor, respetada por el traductor Ramón Rovira, no esté acompañada con ejemplos (y apéndices) en castellano.

Este libro, con el que abrimos la nueva sección de comentarios bibliográficos, es sin duda un acierto por su eficacia didáctica de cara a todos aquellos que desconocen el lenguaje BASIC del Spectrum. Está dividido en cinco partes: introducción al ZX Spectrum; gráficos, color y sonido; juegos; aplicaciones serias y para la educación; usar el Spectrum hasta el límite.

En la primera parte se dan las nociones básicas del lenguaje y de la máquina. Se describen las instrucciones BASIC más importantes, como REM, PRINT, LET, FORNEXT, etc., haciendo especial hincapié en la utilización de los distintos "modos de pantalla" del

MICRO WORLD

ORDENADORES PERSONALES Y MICROORDENADORES DE GESTION

- SPECTRUM • KATSON
- ORIC-1 • APPLE
- NEW BRAIN • ALTOS

SOFTWARE STANDARD Y A MEDIDA

CURSILLOS Y FORMACION

En Madrid:

MODESTO LAFUENTE, 36. TEL. 253 94 54

CLUB sinclair de photo copy

Hazte socio del 1º Club de Informática de Galicia

Si quieres formar parte del CLUB SINCLAIR, rellena el cuestionario y envíalo a título personal a PHOTO COPY, c/ Teresa Herrera, 8. La Coruña

Puede recibirte noticias nuestras.

Apellido: _____
 Nombre: _____
 Dirección: _____
 Teléfono: _____
 Pasa: _____

MODELO DE SINCLAIR: ZX-81 ☐ ZX-Spectrum ☐

Recibo: ☐ SI ☐ NO ☐

photo copy DISTRIBUIDOR OFICIAL sinclair

COMPUTEST

REPARAMOS SU SPECTRUM CON GARANTIA

Llame al telf. (91) 416 73 85



OFERTA DE INTRODUCCION

Una revista para los usuarios de los ordenadores personales SINCLAIR.
Una publicación mensual que ayuda a obtener el máximo partido al ZX-81
y al SPECTRUM.

ZX le trae cada mes programas, juegos y montajes, además de reportajes
sobre programación, y la posibilidad de ganar premios realizando programas,
y otros temas siempre de gran interés.

**GRATIS
PARA USTED
si se suscribe a ZX.**

Una obra imprescindible
en la biblioteca de todo
poseedor de un ordenador
personal SINCLAIR Spectrum.

**Cómo programar
su Spectrum**, un regalo de 132
páginas, tamaño
210 x 270 mm., cuyo precio
de venta es de 850 Pts.

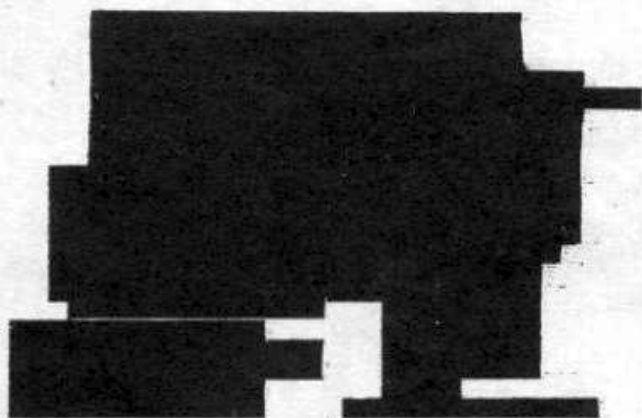


Aproveche ahora esta irrepetible oportunidad para suscribirse a **ZX**. Envíe hoy mismo
la tarjeta adjunta, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano.
Inmediatamente recibirá su primer ejemplar de **ZX** más el **REGALO** y así durante un año
(12 ejemplares).



Jerez, 3
Tel. 457 26 17
MADRID-16

Este es un pequeño programa de los de "sentarse y mirar". SU Spectrum puede crear una sucesión de figuras de distintos colores y tamaños si tiene el programa adecuado. Este lo es, aunque los hay mejores. ¿Recuerda los que le ofrecíamos en el número anterior? Procure probarlo en un televisor en color (16K Spectrum).



```
10 REM Demo
20 LET x=8*INT (RND*25)
30 LET y=8*INT (RND*18)
40 LET s=7+8*INT (RND*10)
50 LET w=7+8*INT (1+RND*15)
60 LET c=INT (RND*8)
70 POKE 23613,PEEK 23613-2
80 PLOT INK c;x,y+s
90 LET f=0: IF PEEK 23610<>255
THEN LET f=1
100 POKE 23610,255: POKE 23613,
PEEK 23613+2
110 IF f THEN GO TO 20
120 POKE 23613,PEEK 23613-2
130 DRAW INK c;w,0
140 POKE 23610,255: POKE 23613,
PEEK 23613+2
150 IF s<1 THEN GO TO 20
160 LET s=s-1
170 GO TO 70
```

Lo más probable es que usted conozca cuántos K tiene su ordenador. Pero si no lo sabe, o desea realizar un programa que se adapte a la memoria con la que está trabajando, encontrará que la instrucción que le ofrecemos a continuación no es ninguna tontería:

PEEK 23733

lo que devuelve el número 127 si su Spectrum tiene 16K o 255 si tiene el de 48K.

Basándose en la misma idea, la siguiente instrucción nos da la misma información, pero un poco más sofisticada:

PEEK 23732+256*PEEK 23733

lo que devuelve la posición más alta de memoria, es decir, 32767 si tiene el Spectrum de 16K, o 65535 si tiene el de 48K.

Los siguientes ejemplos ilustran su utilización:

```
10 LET m=PEEK 23732+256*PEEK 2
3733
20 PRINT AT 0,0;"La posición m
as alta de memoria es=";m
30 PRINT AT 3,0;"Luego, su Spe
ctrum tiene ";FLASH 1;16*(m=327
57)+48*(m=65535);"K."
```

La posición mas alta de memoria es= 65535

Luego, su Spectrum tiene 48K.

```
1 LET m=PEEK 23733
5 FOR j=1 TO 5
10 FOR i=1 TO 7
20 PRINT INK i;AT 5,14;m
25 PAUSE 20
30 NEXT i
40 NEXT j
50 IF m=255 THEN PRINT AT 15,0
INK 0;"Su Spectrum tiene 48K."
60 IF m=127 THEN PRINT AT 15,0
INK 0;"Su Spectrum tiene 16K."
```

Una buena utilización puede ser, como decíamos, adaptar un programa para que —según la memoria con la que trabaje— pueda correr tanto en un Spectrum de 16K como de 48K. Veamos un pequeño ejemplo:

```
10 REM Adaptacion dimension va
riables segun memoria.
20 LET memo=PEEK 23732+256*PEE
K 23733
30 LET x=25-10*(memo=32767)
40 LET y=25-10*(memo=32767)
50 DIM v(x,y)
```

Seguro que alguna vez estaba introduciendo un programa y de repente su Spectrum o ZX le dio un susto de muerte: "Out of memory!". Como medicina preventiva le recomendamos que cuando vaya a introducir un programa "extenso" o disponga de poca memoria utilice la siguiente introducción:

(PEEK 23730+256*PEEK 23731)-(PEEK 23653+256*PEEK 23654)

Para probarla hemos realizado un pequeño ejemplo, en el que le pedimos en dos partes distintas del programa que nos informe sobre la memoria disponible. He aquí el resultado obtenido con un Spectrum de 48 K:

```
1 LET memoria=(PEEK 23730+256
*PEEK 23731)-(PEEK 23653+256*PEE
K 23654)
2 PRINT "Memoria disponible d
espues de introducir la linea
1.....";memoria
5 DIM v(40): FOR n=1 TO 40: L
ET v(n)=RND*100: NEXT n
10 PRINT "Memoria disponible
al final del programa....."
15 LET memoria=(PEEK 23730+256
*PEEK 23731)-(PEEK 23653+256*PEE
K 23654): PRINT memoria
Memoria disponible despues de
introducir la linea 1.....
... 41196
Memoria disponible al final del
programa.....
... 40959
```


IDEAS

Como ya habrá visto, con Spectrum no tiene problemas para introducir cualquier instrucción que se le hubiese olvidado: o la pone en una instrucción ya existente, o bien crea una nueva. En el primer caso, puede no quedarle el programa demasiado claro, y en el segundo no quedará estéticamente bonito (así, verá la instrucción 10, 12, 15, 100...). Para evitar esto hay ordenadores que incluyen la instrucción "Remunerar". Como su Spectrum no la tiene, hay una rutina que puede sustituirla:

```
9990>REM Renumeracion.
9991 LET l=100: LET n=23755
9992 IF (256*PEEK n+PEEK (n+1))>
=9990 THEN STOP
9993 POKE n,INT (l/256): POKE (n
+1),l-INT (l/256)*256
9994 LET l=l+10: LET n=n+3+PEEK
(n+2)+256*PEEK (n+3)+1
9995 GO TO 9992
```

Pero, ¡cuidado!, no lo hace todo. Sólo renumera el número de las instrucciones, colocándolas de diez en diez, partiendo de 100, pero no renumera las "bifurcaciones" o "saltos" que se producen con el GOTO, que, lógicamente, deberá ir a otra instrucción. En el siguiente ejemplo le mostramos su funcionamiento y sus limitaciones (observe la línea 15 antes y

después de renumerar. el GOTO 25 debe de cambiarse a GOTO 140).

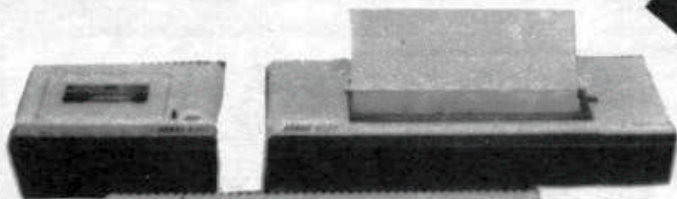
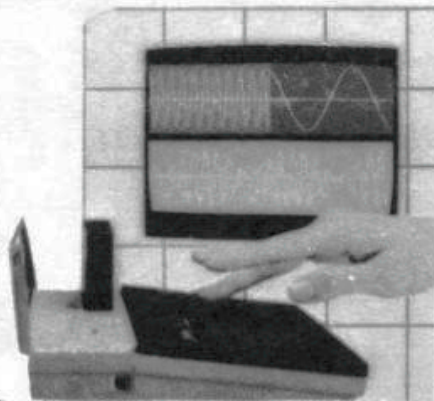
```
1 REM Antes de renumerar.
10 FOR i=1 TO 10
15 IF RND>.6 THEN GO TO 25
20 NEXT i
25 STOP
9990 REM Renumeracion.
9991 LET l=100: LET n=23755
9992 IF (256*PEEK n+PEEK (n+1))>
=9990 THEN STOP
9993 POKE n,INT (l/256): POKE (n
+1),l-INT (l/256)*256
9994 LET l=l+10: LET n=n+3+PEEK
(n+2)+256*PEEK (n+3)+1
9995 GO TO 9992
```

RUN 9990

```
100 REM Despues de renumerar.
110 FOR i=1 TO 10
120 IF RND>.6 THEN GO TO 25
130 NEXT i
140 STOP
9990 REM Renumeracion.
9991 LET l=100: LET n=23755
9992 IF (256*PEEK n+PEEK (n+1))>
=9990 THEN STOP
9993 POKE n,INT (l/256): POKE (n
+1),l-INT (l/256)*256
9994 LET l=l+10: LET n=n+3+PEEK
(n+2)+256*PEEK (n+3)+1
9995 GO TO 9992
```

electrónica
LUGO
ORDENADORES .

DRAGON 32



ATARI

SORD

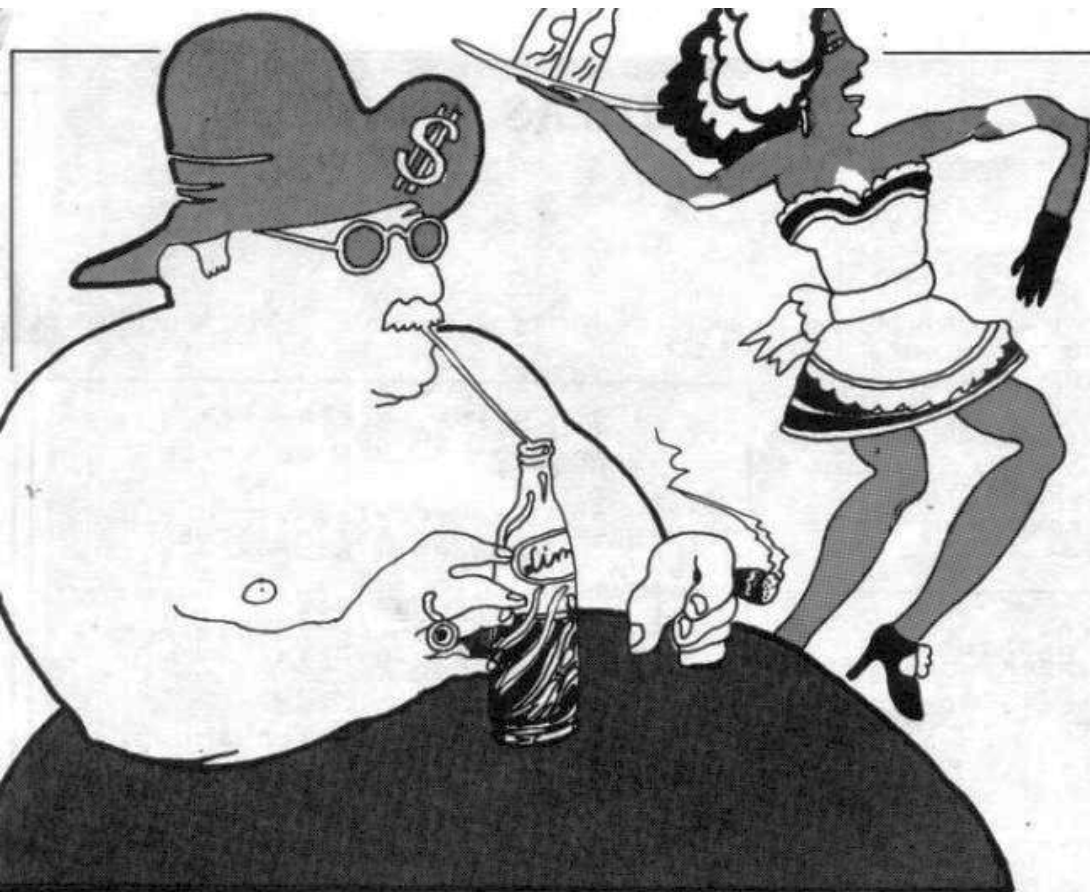
ZX

sinclair



**CABLES ESPECIALES
CONECTORES
PROGRAMAS**

**BARQUILLO. 40
MADRID - 4**



Es un juego clásico, bien conocido por los usuarios de Apple, ahora adaptado para ZX81.

Adelántese a la competencia. ¿No advirtió que ahora nadie se pone a vender refrescos en la calle? Es el mejor momento para planificar sus negocios. Ni siquiera tiene que hacer números para ver cuánto invierte. Su ZX81 ha decidido que la cifra óptima es de 25 millones. Pero le deja lo más difícil: ¿cuánto gasta en publicidad?, ¿cuánto producir?, ¿a qué precio ponemos el vaso? Además, su ZX le informará eficazmente de las condiciones climatológicas. ¿No hace falta ningún master de empresas para sacar un dinerito!

(16 K-ZX81)

LIMONADA

```

278 PRINT "L I M O N A D
A"
279 PRINT "-----"
280 PRINT AT 10,0;"NECESITAS IN
STRUCCIONES ? (S/N)"
290 LET A$=INKEY$
300 IF A$="S" THEN GOTO 3000
310 IF A$="N" THEN GOTO 330
320 GOTO 290
330 CLS
340 PRINT AT 10,0;"CUANTOS JUGA
DORES (2-4) ?"
350 INPUT P
360 IF P<2 OR P>4 THEN GOTO 350
361 CLS
362 PRINT AT 10,0;"CUANTOS DIAS
QUIERES JUGAR (MAXIMO 14) ?"
363 INPUT AD
364 IF AD<1 OR AD>14 THEN COTO
363
369 DIM M(P)
370 DIM A(P)
371 DIM R(P)
372 DIM L(P)
373 DIM B(P)
374 DIM Q(P)
375 DIM I(P)
376 FOR F=1 TO P
377 LET M(F)=25
378 NEXT F
379 FOR F=1 TO P
380 LET A(F)=0
381 NEXT F
382 FOR F=1 TO P
383 LET R(F)=0
384 NEXT F
385 FOR F=1 TO P
386 LET L(F)=0
387 NEXT F
388 FOR F=1 TO P
389 LET B(F)=0
390 NEXT F
391 FOR E=1 TO AD
392 PRINT "L I M O N A D
A DIA "E;AT 2,0;"
393
394 LET W=INT (RND*4)
395 IF W=0 OR W=3 THEN LET W$="
395
396 IF W=1 THEN LET W$="LLUVIA"

```

```

537 IF W=2 THEN LET W$="NIEVE"
538 PRINT AT 10,10;W$
539 PRINT AT 15,0;"INFORME METE
REOLOGICO DE VILLALTIMONES
540 FOR U=1 TO 100
541 NEXT U
542 LET AP=INT (RND*3)+1
543 LET BP=INT ((RND*4)+1)*0.5
544 FOR F=1 TO P
545 CLS
546 PRINT "L I M O N A D
A"
547 PRINT "-----"
548 PRINT "JUGADOR "F
549 PRINT "SALDO "M(F)
550 PRINT "ANUNCIOS ?
CADA UNO CUESTA
£",AP
551 INPUT A(F)
552 PRINT AT 12,15;A(F)
553 PRINT AT 13,0;"
554 LET M(F)=M(F)-(A(F)*AP)
555 PRINT AT 9,7;M(F);"
556 PRINT AT 15,0;"BOTELLAS ?
CADA BOTELLA
CONTIENE 10 VASOS"
557 PRINT "CADA BOTELLA CUESTA
£",BP
558 INPUT B(F)
559 PRINT AT 15,10;B(F)
560 LET M(F)=M(F)-(B(F)*BP)
561 PRINT AT 9,7;M(F);"
562 PRINT AT 16,0;"
563
564 PRINT "PRECIO DE VE
NTA POR VASO ?"
565 INPUT L(F)
566 LET L(F)=INT ABS L(F)
567 PRINT AT 18,20;L(F)
568 PRINT AT 19,0;"
569 FOR T=1 TO 100
570 NEXT T
571 NEXT F
572 IF W$="SOL" THEN LET G=175
573 IF W$="LLUVIA" THEN LET G=7
574
575 IF W$="NIEVE" THEN LET G=10

```


PROGRAMAS

```

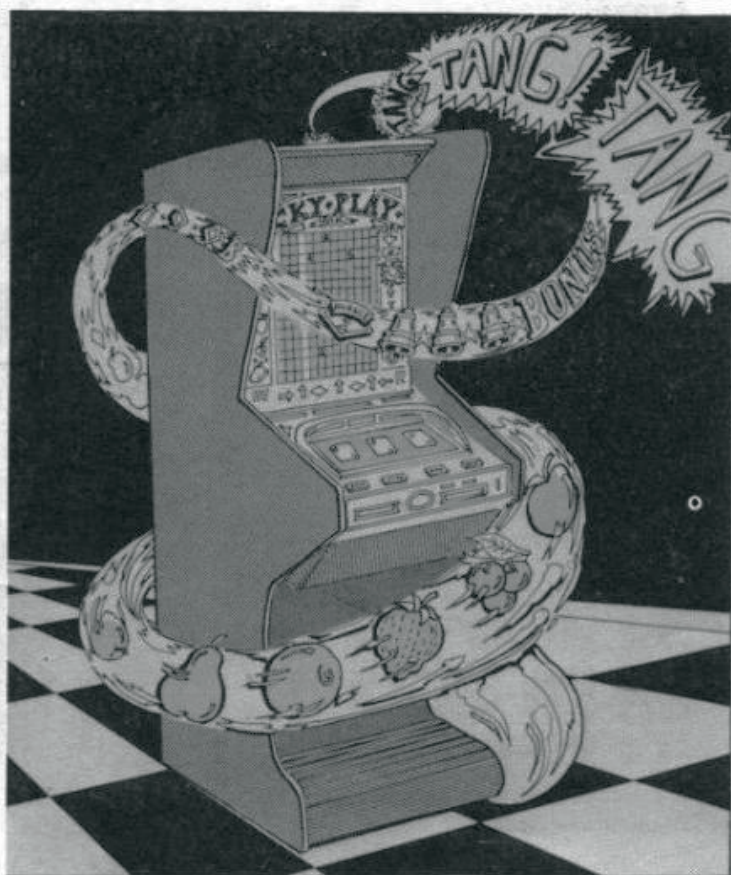
900 CLS
910 PRINT "L I M O N A D
A D I A "
920 PRINT " "
921 PRINT AT 6,9;U$
925 LET V=INT (RND*50)+G
930 PRINT AT 6,9;U$
940 PRINT AT 10,0;"HOY HAN HABI
DO "U;" CLIENTES"
950 FOR U=1 TO 125
960 NEXT U
970 CLS
980 DIM S(P)
990 FOR C=1 TO P
1000 LET S(C)=(B(C)*10)
1010 NEXT C
1020 FOR C=1 TO P
1030 LET I=U*10
1040 LET S(C)=I/L(C)
1050 IF L(C)>30 AND L(C)<=50 THEN
N LET S(C)=INT (RND*(U/(U-2)))
1060 IF L(C)>50 THEN LET S(C)=0
1070 LET Z=(RND*(A(C)*5))
1080 LET S(C)=S(C)+(Z*2)
1085 LET S(C)=INT (S(C)*0.75)
1090 NEXT C
1095 LET N=0
1100 FOR O=1 TO P
1110 LET N=N+S(O)
1120 NEXT O
1130 IF N>V THEN LET M=N-V
1140 IF N>V THEN LET M=INT (M/P)
1150 IF N>V THEN FOR Y=1 TO P
1160 IF N>V THEN LET S(Y)=S(Y)-M
1170 IF N>V THEN NEXT Y
1180 FOR H=1 TO P
1190 IF S(H)>0(H) THEN LET S(H)=
0(H)
1195 IF S(H)<0 THEN LET S(H)=0
1196 IF S(H)>V THEN LET S(H)=V
1200 NEXT H
1210 FOR J=1 TO P
1220 PRINT "L I M O N
A D A "
1230 PRINT " "

```

```

1241 LET MY=INT (S(J)*L(J))/100
1242 LET M(J)=M(J)+MY
1250 PRINT " " "JUGADOR "J," "52
1260 PRINT " " "HAS VENDIDO "S(
J)," "VENDIDOS"
1265 PRINT " " "TENIAS LIMONADA"
PARA "O(J)
1266 PRINT " " "PRECIO DE CADA U
1267 PRINT " " "ANUNCIO "A(J)
1270 PRINT " " "BENEFICIOS "MY
-BP*B(J)-AP*A(J)
1280 FOR U=1 TO 100
1290 NEXT U
1300 CLS
1310 NEXT J
1320 NEXT E
1330 GOSUB 10
1335 PRINT AT 5,19;
1340 FOR U=1 TO P
1345 PRINT TAB 19;"JUGADOR "CHR
$(U+155)
1350 PRINT TAB 21;"E";M(U)
1360 PRINT
1365 PRINT
1370 NEXT U
1380 PRINT AT 3,0;"OTRA PARTIDA
(S/N)?"
1390 LET X$=INKEY$
1400 IF X$="S" THEN GOTO 330
1410 IF X$="N" THEN STOP
1420 GOTO 1390
8995 SAVE "LIMONADA"
8999 RUN
9000 CLS
9010 PRINT "L I M O N
A D A "
9020 PRINT "EL OBJETIVO DE
L JUEGO ES GANAR EL MAXIMO DINE
RO POSIBLE."
9025 PRINT "TIENES TU PROPIO PUE
STO DE LIMONADA EN LA PLAYA
DE VILLA-LIMONES.VENDES LIMON
AA PARA INTENTAR SACAR EL MA
YOR BENEFI-CIO.AL PRINCIPIO TIE
NES £25 PARA COMPRAR BOTELLA
S."
9030 PRINT "PUEDES ANUNCIAR TU L
IMONADA PERO ESTO TE COSTARA DINE
RO."
9040 PRINT AT 21,0;"PULSA CUALQU
IER TECLA"
9050 IF INKEY$="" THEN GOTO 9050
9060 GOTO 330
9500 FOR F=1 TO 5
9510 NEXT F
9520 RETURN

```



TRAGAPERRAS

Supongamos que acaba de echar unas moneditas en la máquina tragaperras de su cafetería favorita y, de camino a casa, se ha preguntado, ¿por qué no hacer un programa igual con el Spectrum? Veamos que necesitamos para ello.

- Tendremos que mostrar una imagen gráfica que rotando al azar nos dé diferentes combinaciones. Ello lo podemos conseguir con la definición de caracteres gráficos.

- Necesitaremos poner un límite a la cantidad de dinero.

- Habrá que informar al jugador de las combinaciones con las que puede obtener dinero y qué cantidad

puede ganar en cada caso. Al mismo tiempo, podemos informarle por pantalla del dinero que posee en cada momento.

- Para empezar a jugar podemos seleccionar una jugada inicial, es decir, una combinación particular. Después iremos seleccionando las nuevas combinaciones.

- Con cada combinación definitiva obtenida habrá que ver si coincide con las combinaciones de premio.

- Finalmente, si todavía hay dinero, habrá que saber si el jugador desea seguir apostando o quiere retirarse.

En definitiva, hemos reducido un problema más o menos complejo a una serie de pasos más sencillos. Es-



quemáticamente tendríamos:

Una vez que ya tenemos claros los pasos a realizar, podemos empezar a realizar el programa. Para ello, podemos realizar una subrutina para cada uno de estos pasos y separarlos con el uso de espacios en blanco para una mejor identificación.

Tal y como vemos en el listado, las subrutinas principales son:

Línea	Concepto
15	Bucle principal de conexión con subrutinas.
1000	Selecciona nueva combinación a partir de la jugada principal.
2000	Selecciona combinación inicial.
3000	Muestra las combinaciones con premio.
4000	Comprobación ganancias según combinación.
5000	Sonido acústico para aviso de combinación con premio.
7000	Inicialización de variables.
8000	Definición de caracteres gráficos.
9000	Aviso de ausencia de dinero.

El programa está hecho para el Spectrum de 16 K, con la utilización de los siguientes caracteres gráficos:

Línea			Graphic	AAA
"	3000	—		CC
"	3010	—		DD
"	3020	—		A
"	3030	—		

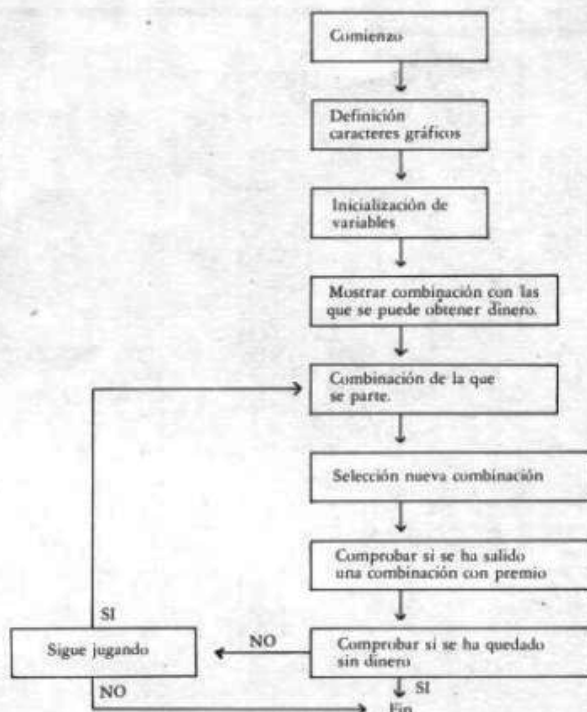
```

10 REM Tragaperras.

15 GO SUB 5000
20 GO SUB 7000
35 GO SUB 3000
40 GO SUB 2000
45 LET m=m-10
50 GO SUB 1000
60 GO SUB 4000
65 IF m<=0 THEN GO TO 9000

70 PRINT AT 20,0;"Tiene ";m;
"p
80 PRINT "Otra jugada(s/n) "
90 LET a$=INKEY$
100 IF a$="" THEN GO TO 90
110 IF a$="s" OR a$="S" THEN GO TO 40
120 PRINT AT 21,0;"Se lleva ";m;"p"
130 STOP
1000 FOR i=d TO d+33
1010 PRINT AT 10,12;CHR$(p);" "
;CHR$(q);" ";CHR$(r)

```



```

1020 POKE udg,i
1030 NEXT i
1040 POKE udg,d
1050 RETURN

2000 LET p=144+INT (RND*4)
2010 LET q=144+INT (RND*4)
2020 LET r=144+INT (RND*4)
2030 RETURN

3000 PRINT AT 2,0;"♦♦♦ gana 50p"
3010 PRINT "?? gana 25p"
3020 PRINT "?■ gana 10p"
3030 PRINT "◆?? gana 5p"
3040 RETURN

4000 IF p=144 AND q=144 AND r=144 THEN LET m=m+50: GO TO 5000
4010 IF (p=146)+(q=146)+(r=146)=2 THEN LET m=m+25: BEEP .1,20
4020 IF (p=147)+(q=147)+(r=147)=2 THEN LET m=m+10: BEEP .1,20: RETURN
4030 IF (p=144)+(q=144)+(r=144)=2 THEN LET m=m+5: BEEP .1,20: RETURN
4040 RETURN

5000 FOR i=1 TO 20
5010 BEEP .01,20
5020 NEXT i
5030 RETURN

7000 LET udg=23675
7010 LET d=PEEK udg
7020 LET m=100
7030 RETURN

8000 POKE USR "a"+0,BIN 00000000
8010 POKE USR "a"+1,BIN 00011000
8020 POKE USR "a"+2,BIN 00111100
8030 POKE USR "a"+3,BIN 01111110
8040 POKE USR "a"+4,BIN 11111111

```


PROGRAMAS

```

8050 POKE USR "a"+5,BIN 01111110
8060 POKE USR "a"+6,BIN 001111100
8070 POKE USR "a"+7,BIN 0001110000
8080 POKE USR "a"+0,BIN 000000000
8090 POKE USR "a"+1,BIN 0001110000
8100 POKE USR "a"+2,BIN 001111100
8110 POKE USR "a"+3,BIN 01111110
8120 POKE USR "a"+4,BIN 11111111
8130 POKE USR "a"+5,BIN 000000000
8140 POKE USR "a"+6,BIN 000000000
8150 POKE USR "a"+7,BIN 000000000
8160 POKE USR "a"+0,BIN 000000000
8170 POKE USR "a"+1,BIN 001111100
8180 POKE USR "a"+2,BIN 01111110
8190 POKE USR "a"+3,BIN 01111110
8200 POKE USR "a"+4,BIN 01111110
8210 POKE USR "a"+5,BIN 001111100
8220 POKE USR "a"+6,BIN 000000000
8230 POKE USR "a"+7,BIN 000000000
8240 POKE USR "a"+0,BIN 000000000
8250 POKE USR "a"+1,BIN 01111110
8260 POKE USR "a"+2,BIN 01111110
8270 POKE USR "a"+3,BIN 01111110
8280 POKE USR "a"+4,BIN 01111110
8290 POKE USR "a"+5,BIN 01111110
8300 POKE USR "a"+6,BIN 01111110
8310 POKE USR "a"+7,BIN 000000000
8320 POKE USR "d"+1,BIN 01111110
8330 POKE USR "d"+2,BIN 01111110
8340 POKE USR "d"+3,BIN 01111110
8350 POKE USR "d"+4,BIN 01111110
8360 POKE USR "d"+5,BIN 01111110
8370 POKE USR "d"+6,BIN 01111110
8380 POKE USR "d"+7,BIN 000000000
8500 FOR i=0 TO 7
8510 POKE USR "e"+i,PEEK (USR "a
+i)

```

```

8520 POKE USR "f"+i,PEEK (USR "b
+i)
8530 POKE USR "g"+i,PEEK (USR "c
+i)
8540 POKE USR "h"+i,PEEK (USR "d
+i)
8550 NEXT i
8990 RETURN

```

```

9000 CLS
9020 PRINT AT 10,5; FLASH 1;"Te
arruinaste!!"

```

```

♦♦♦ gana 50p
♦♦? gana 25p
?♦♦ gana 10p
♦?? gana 5p

```

■ ■ ■

Tiene 105p
Se lleva 105p



Habían pasado dos horas y la computadora central seguía sin funcionar. Aquel campo de energía había afectado a la estabilidad de la nave más de lo que creíamos. Seguíamos descendiendo y, llegado el momento, tendríamos que hacer uso de los controles manuales. Todos estábamos aterrorizados: el aterrizaje siempre lo había hecho el ordenador central, y ahora no era sino un montón de chatarra estéril. En Rocket Mountains habíamos aprendido a manejar la cápsula con relativa facilidad: el mando "7" la propulsaba hacia arriba, y el "5", hacia la izquierda. Todo era sencillo en el simulador de vuelo. Pero ahora no había simulación. En algún lugar de aquel planeta se encontraba la base de aterrizaje y, posiblemente, la muerte.

George fue el encargado de efectuar la maniobra. Con manos sudorosas y torpes conectó los retropropulsores: ¡había comenzado la cuenta atrás! Lentamente, la

ALUNIZAJE



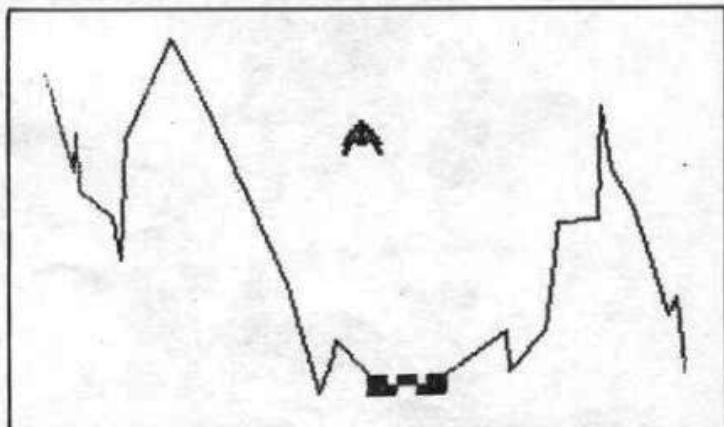
Notas gráficas

1040	Graphic	8 (Shift)
"	"	6 (Shift)
"	"	6 (Inverse video, Shift)
"	"	8 (True, video, Shift)
2040	"	AB
"	"	CD
2070	"	IJ
"	"	KL
2100	"	MN
"	"	OP
2140	"	DC
"	"	AB
8080	"	AB
"	"	CD
8180	"	EF
"	"	GH

nave descendió, hasta que un pequeño descuido provocó una enorme explosión. Esta vez no habíamos tenido suerte. Desconectamos el Spectrum y decidimos volver a intentarlo al día siguiente. (16K Spectrum).

```
100 REM Aterrizaje en la luna.
120 RESTORE : FOR i=1 TO 16: RE
130 LET s=USR (CHR$ (144+i-1))
140 FOR j=1 TO 8
150 READ g: POKE (s+j-1),g
160 NEXT j
170 NEXT i
180 DATA 0,0,1,1,7,7,31,25,0,0,
128,128,224,224,248,152,53,107,1
27,243,96,96,192,192,172,214,254
,207,6,6,3,3
190 DATA 1,1,1,1,1,1,1,1,224,17
5,140,134,129,135,140,176,1,1,7,
1,7,7,31,25,224,128,224,224,248,
152
200 DATA 0,0,0,4,5,15,5,15,0,0,
0,0,96,192,224,248,27,7,3,6,0,0,
0,0,120,168,112,32,0,0,0,0,0
210 DATA 150,67,38,60,24,48,96,
37,4,154,244,124,6,4,4,6,194,96,
32,36,104,120,79,193,130,14,12,3
5,124,78,195,2,0,0,0,0,0,0,0,0
220 LET ld=0
230 GO TO 8000
1000 REM Base.
1020 LET bx=INT (RND*21): IF bx<
11 THEN GO TO 1020
1030 LET lt=bx*8: LET rt=(bx+4)*
8
1040 PRINT AT 21,bx;"■": PRINT A
T 21,bx+1; PAPER 5; INK 1;"■":
PRINT AT 21,bx+3;"■"
1050 LET di=-1: GO SUB 1500
1060 LET di=1: GO SUB 1500
1070 RETURN
1500 REM Modulom
1530 LET ly=8+(di=1)*16
1540 LET lx=((rt+24)*(di=1))+((lt
*(di=-1))
1550 LET up=255-(di=-1)*255
1560 IF di=1 THEN PLOT rt,8: DRA
W INK 2;24,16
1570 PLOT lx,ly
1580 LET a=RND: LET ry=((a<=.6)-
(a>.6))*(INT (RND*48)+1)
1590 LET rx=di*(INT (RND*16)+1)
1600 IF di*(lx+rx)>up*di THEN LE
T rx=up-lx
1610 IF ly+ry>144 OR ly+ry<0 THE
N LET ry=0-ry
1620 LET lx=lx+rx: LET ly=ly+ry
1630 DRAW INK 2;rx,ry
```

```
1640 IF lx<>up THEN GO TO 1550
1650 RETURN
2000 REM Explosion.
2020 PRINT AT h,x;" "
2030 FOR j=1 TO 5
2040 PRINT AT h+1,x;"▲": PRINT
AT h+2,x;"▲"
2050 BEEP .05, -(RND*48)
2060 FOR i=1 TO 10: NEXT i
2070 PRINT AT h+1,x;"▲": PRINT
AT h+2,x;"▲"
2080 BEEP .05, -(RND*48)
2090 FOR i=1 TO 10: NEXT i
2100 PRINT AT h+1,x;"▲": PRINT
AT h+2,x;"▲"
2110 BEEP .05, -(RND*48)
2120 FOR i=1 TO 10: NEXT i
2130 NEXT j
2140 FOR i=h+1 TO 20: PRINT AT i
-1,x;" "; AT i,x;"▲"; AT i+1,x;"
"; NEXT i
2150 RETURN
8000 REM Bucle central.
8020 BORDER 3: INK 0: PAPER 7: O
VER 0: FLASH 0: CLS
8030 GO SUB 1000
8040 LET x=0: LET h=x
8050 LET oh=h: LET ox=x
8060 LET x=x+.5*(x<30)-(INKEY$="
S")
8070 LET h=h+.5-(INKEY$="7")*(h>
0)
8080 PRINT AT oh,ox;" " : PRINT
AT oh+1,ox;" "
8090 PRINT AT h,x;"▲": PRINT AT
h+1,x;"▲"
8100 LET cr=ATTR (h+2,x)+ATTR (h
+2,x+1)
8110 IF cr<58 OR cr=82 OR cr=112
THEN GO TO 8140
8120 GO SUB 2000
8130 BEEP .5,17: BEEP .5,15: BEE
P .25,13: BEEP .25,12: BEEP .5,1
0: BEEP .25,13: BEEP .25,12: GO
TO 8210
8140 IF h>=20 THEN GO TO 8120
8150 IF cr=82 THEN LET ld=1: GO
TO 8170
8160 IF ld=0 THEN GO TO 8050
8170 FOR i=1 TO 6: PAUSE 25: BOR
DER i: NEXT i: BORDER 3
8180 PRINT AT 18,bx+1;"▷": PRIN
T AT 19,bx+1;"▲"
8190 BEEP .125,12: BEEP .25,19:
PAUSE 10: BEEP .125,12: BEEP .25
,19: PAUSE 10: BEEP .125,12: BEE
P .25,19: PAUSE 10: BEEP .125,12
: BEEP .25,19: PAUSE 10
8200 BEEP .25,12: BEEP .125,14:
BEEP .125,15: BEEP .125,22: BEEP
.125,22: BEEP .125,15: BEEP .12
5,14: BEEP .125,19: BEEP .125,14
: BEEP .25,12
8210 FOR i=1 TO 100: NEXT i: CLS
: GO TO 220
```



CODE 4

CODE 4 es uno de esos juegos clásicos en los cuales la capacidad lógica del jugador es el arma decisiva para lograr el objetivo: descubrir un código secreto que el ordenador guarda en su memoria.

Primero, el ordenador creará de forma aleatoria un código de cuatro letras (de entre la "a" a la "f" cada una), que evidentemente no aparecerá en pantalla. Dado el carácter aleatorio del código, éste podría incluso estar formado por cuatro letras iguales.

El jugador dispone de seis oportunidades para lograr descubrir el código, teniendo, para eso, que introducir cada vez otro código también de 4 letras de entre la "a" y la "f" cada una.

Cada código introducido por el jugador será comparado con el código del ordenador y de esta comparación saldrá una información que el jugador deberá tener muy en cuenta antes de introducir su próximo código.

Esta información consistirá en una serie de asteriscos (de 0 a 4) que podrán aparecer de color verde o blanco (siempre en este orden si es que han de aparecer).

Un asterisco de color verde significará que una de las cuatro letras del código del jugador es idéntico a la que está en la misma posición en el código del ordenador. Por ejemplo:

código	ordenador	acde
código	jugador	abdf

en este caso aparecerían dos asteriscos verdes por coincidir la "a" y la "d" en los dos códigos y en la misma posición. Por tanto, si aparecen cuatro asteriscos verdes el

código del ordenador habrá sido descubierto y la partida habrá terminado. Un asterisco de color blanco significará que una de las cuatro letras del código del jugador aparece en el código del ordenador pero en distinta posición. Por ejemplo:

código	ordenador	abdc
código	jugador	defc

en este caso aparecerían dos asteriscos blancos, puesto que "d" y "c" se repiten en los códigos pero en distintas posiciones. Si aparecen cuatro asteriscos blancos querrá decir que las cuatro letras introducidas por el jugador son las que utilizan el ordenador en su código pero en distinta posición todas. Si no

PROGRAMA GANADOR DE 5.000 PTAS

Este juego, junto con las explicaciones sobre su utilización, nos ha sido remitido por el lector Toni Torreta, de Terrassa, Barcelona, quien se ha hecho acreedor al premio de 5.000 pesetas. Bien merecido, por cierto.

aparece ningún asterisco es porque ninguna letra del código del jugador aparece en el del ordenador.

En los casos de una letra que tiene varias similares en el otro código es muy importante recordar que dicha letra sólo se iguala a una de sus similares y no a todas. Por ejemplo:

código	ordenador	aabb
código	jugador	efab

produce un asterisco verde y un asterisco blanco.

PROGRAMAS

Si antes de seis oportunidades no se ha logrado descubrir el código del ordenador, éste lo sacará en pantalla y terminará la partida.

Explicación al programa

Primero se prepara la pantalla (5-14), se crea el código del ordenador (15-30), y a continuación, se empiezan a pedir los códigos del jugador (40). Después se compara el código del juga-

dor y el del ordenador para ver qué letras ocupan las mismas posiciones (45-80), y qué letras son iguales a los dos códigos sin estar en las mismas posiciones (90-220). Al final se imprime un mensaje según el resultado de la partida, contando las partidas realizadas, las ganadas y las perdidas y preguntando si se quiere volver a jugar (250-370).

(16 K-Spectrum).
¡Mide la capacidad lógica con este juego!

```

5 BORDER 0: PAPER 0: INK 7: C
LS
7 PRINT PAPER 2; AT 11,9; " C
Q D E 4 " : PAUSE 150: CLS
10 LET ap=0: LET ag=0: LET ao=
0
11 PLOT 83,163: DRAW 80,0: DRA
W 0,-56: DRAW -80,0: DRAW 0,56:
PLOT 124,163: DRAW 0,-56
12 PRINT INK 4; AT 10,1; "*": PR
INT AT 10,2; "en posición : *="UT
izable"
13 PRINT AT 12,11; "PARTIDAS ":
AT 13,11; "GANADAS "; AT 14,11; "PE
RDIDAS "
14 PRINT INK 6; AT 12,20; ap; AT
13,20; ag; AT 14,20; ao
15 DIM e$(4)
17 LET a$=""
20 FOR f=1 TO 4
25 LET e$(f)=CHR$(97+INT (RND
#6)): LET a$=a$+e$(f)
30 NEXT f
35 FOR g=1 TO 6: LET a$=""
36 FOR u=1 TO 4: LET a$=a$+e$(
u): NEXT u
37 LET r=0
40 INPUT (g); "code 4(a-f) "; b
$: IF LEN b$<4 THEN GO TO 40
41 FOR x=1 TO 4: IF CODE b$(x)
<97 OR CODE b$(x)>102 THEN GO TO
40
42 NEXT x
44 PRINT PAPER 5; INK 0; AT 9+1
,11;b$
45 LET c$="": LET d$=""
46 LET t=0
50 FOR f=1 TO 4
55 IF b$(f)=a$(f) THEN LET t=t
+1: PRINT INK 4; AT 9+1,15+t; "*":
GO TO 70
60 LET c$=c$+a$(f): LET d$=d$+
b$(f)
70 IF t=4 THEN LET ag=ag+1: GO
TO 250
80 NEXT f
90 LET r=r+1: LET a$=c$: LET b
$=d$
100 LET c$="": LET d$=""
110 FOR i=1 TO LEN a$
120 FOR j=1 TO LEN a$
132 IF b$(i)=a$(j) THEN GO TO 1
50
140 NEXT j
150 NEXT i
151 IF g=6 THEN LET ao=ao+1: GO
TO 250
155 NEXT g
160 FOR k=1 TO LEN a$
170 IF k=j THEN GO TO 180
175 LET c$=c$+a$(k)
180 NEXT k
190 FOR l=1 TO LEN a$
200 IF l=i THEN GO TO 210

```



```

205 LET d$=d$+b$(1)
210 NEXT I
220 PRINT AT 9+1,15+t+r;"*": GO
TO 90
250 PRINT AT 18,5;"LO SIENTO PE
RDISTE!"; FOR Z=1 TO 7: BEEP .3,
0: NEXT Z: GO TO 300
260 PRINT FLASH 1; AT 18,10;" AC
ERTASTE !"; FOR Z=15 TO 30: BEEP
.05,Z: NEXT Z
300 LET a$="": FOR f=1 TO 4: LE
T a$=a$+e$(f): NEXT f
310 PRINT PAPER 6; INK 0; AT 0,1
1,a$
320 LET ap=ap+1
330 PRINT INK 6; AT 12,20; ap; AT
13,20; ag; AT 14,20; ao
340 PRINT AT 21,0;"quieres volv
er a jugar ?(s/n)"
350 IF INKEY$="s" THEN CLS : GO
TO 11
360 IF INKEY$="n" THEN STOP
370 GO TO 350

```

dcde

aaaa	****
dcde	

dcde

*=en posicion ; *=utilizable

PARTIDAS 2
GANADAS 1
PERDIDAS 1

ACERTASTE !

quieres volver a jugar ?(s/n)

VARIACIONES GRAFICAS

El siguiente programa añade su granito de arena a la teoría de curvas, interpretaciones oscilográficas de Lissajous y la impresión de billetes de banco. Esto último se hace más patente cambiando los argumentos trigonométricos de las líneas 60 y 70 por números más altos y primos entre si. Y por supuesto, ensayando

nuevos valores de Q.
(16 K-Spectrum).

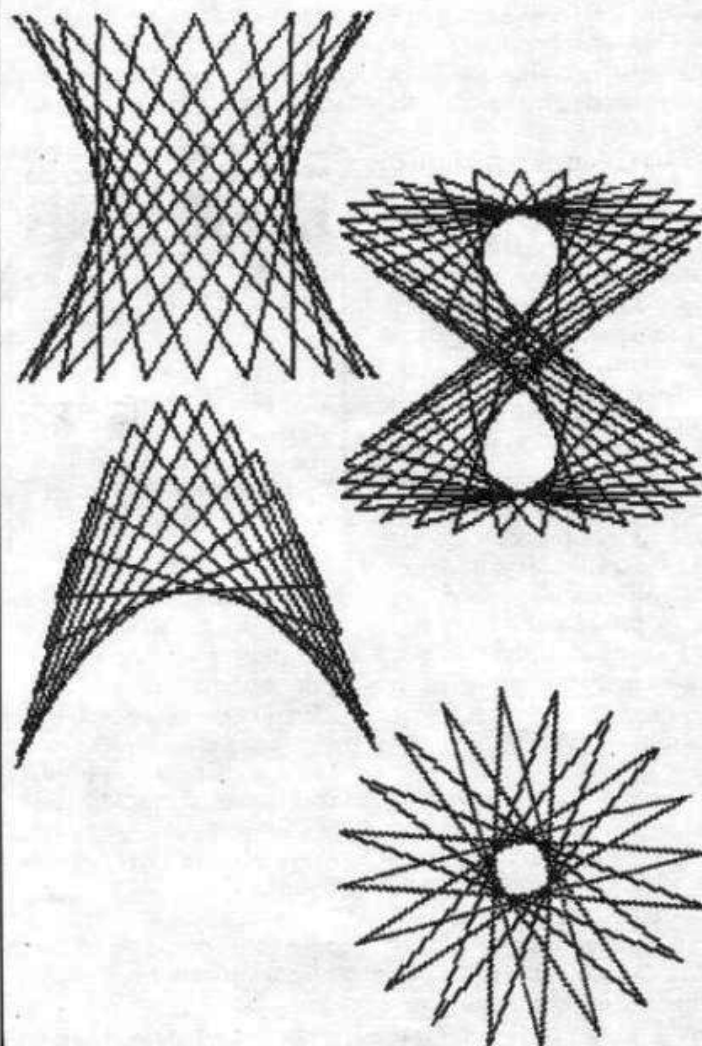
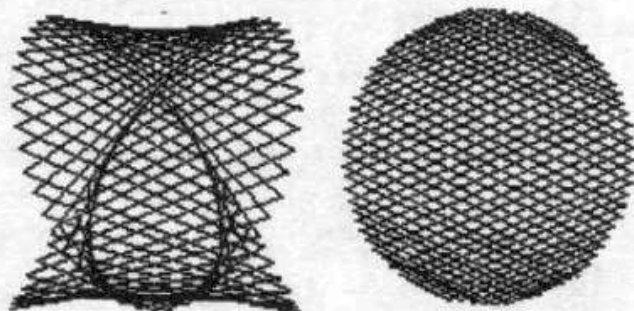
PROGRAMA GANADOR DE 5.000 PTAS

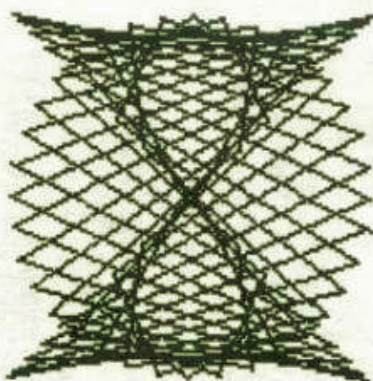
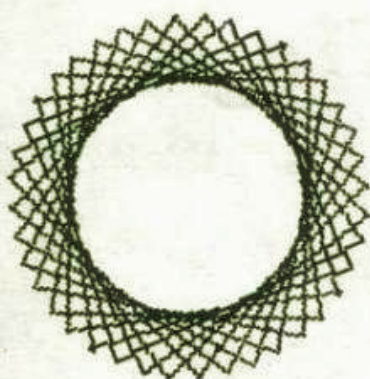
Programa remitido por Angel del Valle Tomillo, de Mairena del Aljarafe. Sevilla. Ha ganado una buena sus 5.000 pesetas.

```

10 READ q: IF NOT q THEN STOP
20 BORDER 1: PAPER 5: INK 1
30 CLS
40 LET f=0
50 FOR n=0 TO 2*PI+.01 STEP 2*
PI/q
60 LET x=(SIN (29*n)+1)*60+60
70 LET y=(COS (11*n)+1)*60+20
80 IF f=1 THEN DRAW x-a,y-b
90 LET f=1: LET a=x: LET b=y
100 PLOT x,y: NEXT n
110 PAUSE 200
120 DATA 20,22,23,40,47,51,69,7
2,80,83,0
130 GO TO 10

```





JUEGO DEL AHORCADO

Se trata del clásico juego de ahorcado y puede correr tanto en el Sinclair ZX81 con 1 K como en el Spectrum. El modo en que trabaja es muy sencillo, han de jugar dos personas, una de ellas le dirá al ordenador una palabra y el otro tendrá que adivinarla. En primer lugar el ordenador nos pedirá que le introduzcamos la palabra a descubrir mediante new line o enter (ZX81 o Spectrum), seguidamente el ordenador nos mostrará la primera y la última de las letras de la palabra a descubrir y, entre

ellas dos, tantos espacios con puntos como letras falten y nos indicará el número de intentos de que disponemos, cada letra tanto si es cierta como falsa restará un intento, finalizando la partida al acertarse la palabra o bien al agotar los intentos, en cuyo caso el ordenador nos mostrará cual era la palabra.

**PROGRAMA GANADOR DE
5.000 PTAS**

Este juego nos ha sido remitido por uno de nuestros numerosos lectores jóvenes Pedro Julio Abad, que vive en Barcelona, se ha ganado las 5.000 pesetas por la publicación de su programa. Y, como prometíamos en nuestro número anterior, hemos sorteado entre todos los programas recibidos durante el mes un ZX Microdrive. La suerte le ha correspondido

también a Pedro Julio Abad. Aunque no podrá conectarlo al ZX81 que, según nos dice, es el ordenador que posee, no deja de ser un aliciente para comprarse un Spectrum. Eso sí, para gozar del *microdrive*, Pedro tendrá que esperar hasta que esté disponible en España (nos dicen que un par de meses más). A disfrutarlo y gracias por la colaboración.

GANADOR
DEL **ZX** DE
ESTE MES
MICRODRIVE

```
10 REM PEDRO ABAD.
20 CLS
30 PRINT "Introduzca la palabra"
40 INPUT a$
50 CLS
60 LET x=LEN a$
70 PRINT AT 8,9;a$(1);
80 FOR n=2 TO x-1
90 PRINT " ";
100 NEXT n
110 PRINT a$(x)
120 LET k=-1
130 LET k=k+1
140 LET y=2*(x-2)-k
150 PRINT AT 16,0;"Tienes derecho a ";y;" intentos"
160 IF NOT y THEN PRINT AT 17,0;"Ha perdido, la palabra era ";a$
170 INPUT b$
180 FOR n=2 TO x-1
190 IF b$(n)=a$(n) THEN PRINT AT 8,6+n;b$
200 NEXT n
210 GO TO 130
```

n.b...o.o.o.or

Tienes derecho a 17 intentos

nabucodonosor

Tienes derecho a 9 intentos

JUEGO DE PALABRAS

Seguro que este juego no le es desconocido. El objetivo es mover determinada pieza para obtener la combinación deseada de letras o números. Para ello, se dispone de un tablero de 4 por 4 donde hay una posición libre que permite la movilidad de las fichas. Lógicamente, hay un número limitado de movimientos (dos, tres o cuatro, según el lugar donde se encuentra la posición libre) y no es posible mover en diagonal. No sea tramposo y no intente engañar a su Spectrum. ¡Seguro que no se deja!

Para efectuar su jugada introduzca el número de la

casilla que desea mover. Si desconoce dicho dato, presione cualquier letra, con lo que éste se sob reimprime en cada casilla.

(16K-Spectrum).

Notas gráficas:

Línea 2020 - Graphic 4
Línea 2050 - Graphic 4
Línea 2060 - Graphic ccb
Línea 2120 - Graphic 3
Línea 2130 - Graphic 3
Línea 2140 - Graphic 5
Línea 2150 - Graphic 5
Línea 2170 - Graphic 4
Línea 2180 - Graphic 2 (Shift)
Línea 2190 - Graphic 7 (Shift)
Línea 2200 - Graphic 1

```
10 DIM b(4,4)
20 DIM b$(16)
30 DIM w$(16)
40 GO SUB 5500
50 GO SUB 1000
60 IF wr=1 THEN GO SUB 6500
70 LET move=0: LET error=0
80 GO SUB 9000
90 GO SUB 2000
100 GO SUB 5000
110 GO SUB 2500
120 GO SUB 4000
130 GO SUB 500
140 LET move=move+1
150 IF end<>0 THEN GO TO 120
160 PRINT AT 19,0: INK 0; "Lo hi
ciste en ";move; " jugadas."
170 INPUT "Otro (s/n)";a$
180 IF a$(1)="s" OR a$(1)="S" T
HEN RUN
190 INK 0: PAPER 7
200 CLS
210 STOP
500 LET end=0: LET k=0
510 FOR i=1 TO 4
520 FOR j=1 TO 4
525 LET k=k+1
530 IF b$(b(i,j))<>w$(k) THEN L
ET end=1
540 NEXT j
550 NEXT i
560 RETURN
650 LET w$(5 TO 8)=a$
1000 LET k=0
1010 FOR i=1 TO 4
1020 FOR j=1 TO 4
1030 LET k=k+1
1040 LET b$(k)=CHR$(64+k)
1050 LET b(i,j)=k
1060 NEXT j
1070 NEXT i
1080 LET b$(16)=""
1090 LET w$=b$
```

```
1100 RETURN
2000 FOR i=0 TO 3
2010 FOR j=0 TO 3
2020 PRINT AT 4+i*3,10+j*3: INK
0: PAPER 6; " "
2030 NEXT j
2040 FOR j=0 TO 3
2050 PRINT AT 5+i*3,10+j*3: INK
0: PAPER 6; " " INK 0; b$(b(i+1,j
+1)); INK 2; " "
2060 NEXT j
2070 FOR j=0 TO 3
2080 PRINT AT 6+i*3,10+j*3: INK
2: PAPER 6; " "
2090 NEXT j
2100 NEXT i
2110 FOR i=0 TO 11
2120 PRINT AT 3,10+i: INK 7; PAP
ER 2; " "
2130 PRINT AT 16,10+i: INK 2; PA
PER 7; " "
2140 PRINT AT 4+i,9: INK 2; PAPE
R 7; " "
2150 PRINT AT 4+i,22: INK 7; PAP
ER 2; " "
2160 NEXT i
2170 PRINT AT 3,9: INK 2; PAPER
7; " "
2180 PRINT AT 16,22: INK 7; PAPE
R 2; " "
2190 PRINT AT 3,22: INK 2; PAPER
7; " "
2200 PRINT AT 16,9: INK 2; PAPER
7; " "
```

1	1	1	1
2	2	2	2
3	3	3	3
4	4		4

```
2210 RETURN
2500 LET is=4: LET js=4: LET io=
0: LET jo=0
2510 FOR d=1 TO 3
2520 LET i=is: LET j=js
2530 IF AND>.5 THEN GO TO 2570
2540 LET i=is+INT (RND*2)*2-1
2550 IF i>4 OR i<1 THEN LET i=is
GO TO 2570
2560 GO TO 2590
2570 LET j=js+INT (RND*2)*2-1
2580 IF j>4 OR j<1 THEN LET j=js
GO TO 2520
2590 IF i=io AND j=jo THEN GO TO
2520
2600 LET io=is: LET jo=js
2610 GO SUB 6000
2620 NEXT d
2630 RETURN
3000 LET k=0
3010 FOR i=0 TO 3
3020 FOR j=0 TO 3
3030 LET k=k+1
3040 PRINT AT 4+i*3,10+j*3: INK
4: PAPER 6; k
3050 NEXT j
3060 NEXT i
```


PROGRAMAS

```

3065 PAPER 4
3070 RETURN
3500 FOR l=0 TO 3
3510 FOR p=0 TO 3
3520 PRINT AT 4+l*3,10+p*3; PAPER 4;
3530 NEXT p
3540 NEXT l
3550 INK 0; PAPER 5
3555 IF error=0 THEN RETURN
3560 PRINT AT 20,0;
3570 FOR l=1 TO 64
3580 PRINT PAPER 4; " ";
3590 NEXT l
3600 LET error=0
3610 RETURN
4000 PAPER 4
4004 INPUT "Su movimiento"; LINE
4005 IF m$="" THEN GO TO 4000
4006 IF LEN m$<2 THEN LET m$="0"
4007 +m$
4010 IF m$(1)<"0" OR m$(1)>"9" OR m$(2)<"0" OR m$(2)>"9" THEN GO TO 3000
4020 LET m=VAL m$
4030 IF m>0 AND m<17 THEN GO TO 4070
4035 LET error=1
4040 GO SUB 3000
4050 PRINT AT 21,0; INK 0; "1 y 1 6."
4060 GO SUB 3000
4065 GO TO 4000
4070 LET i=INT ((m-1)/4)
4080 LET j=m-i*4
4090 LET i=i+1
4100 IF ABS (i-is)+ABS (j-js)=1 THEN GO TO 6000
4110 BEEP .1,15
4115 LET error=1
4120 PRINT AT 20,0; INK 0; "JUGADOR A NO VALIDA"
4130 PRINT AT 21,0; INK 0; "
: REM 9 espacios
4150 GO TO 4000
5000 LET k=0
5005 FOR l=1 TO 4
5010 FOR p=1 TO 4
5015 LET k=k+1
5020 IF b(p,l)=16 THEN LET is=l: LET js=p: LET ms=k
5030 NEXT p
5040 NEXT l
5050 RETURN
5500 INK 0; PAPER 7
5510 CLS
5520 PRINT TAB 7; "P U Z L E"
5530 PRINT AT 10,0; "Quiere introducir las letras"
5550 INPUT a$
5555 IF a$="" THEN GO TO 5550
5560 IF a$(1)="s" OR a$(1)="S" THEN LET wr=1: GO TO 5590
5570 IF a$(1)="n" OR a$(1)="N" THEN LET wr=0: GO TO 5590
5590 PRINT a$
5600 PRINT AT 15,0; "Cuantos arreglos"
5610 INPUT s
5620 IF s<1 THEN GO TO 5600
5630 PRINT s
5640 RETURN
6000 GO SUB 3500
6002 INK 0
6005 LET f1=(js-1)*3+(is-1)*3*32+5*32+22526+11
6020 LET f=f1: LET c=128
6030 GO SUB 7000
6040 LET f2=(j-1)*3+(i-1)*3*32+5*32+22526+11
6050 LET f=f2
6060 GO SUB 7000
6070 GO SUB 8000
6080 PRINT AT 5+(i-1)*3,11+(j-1)

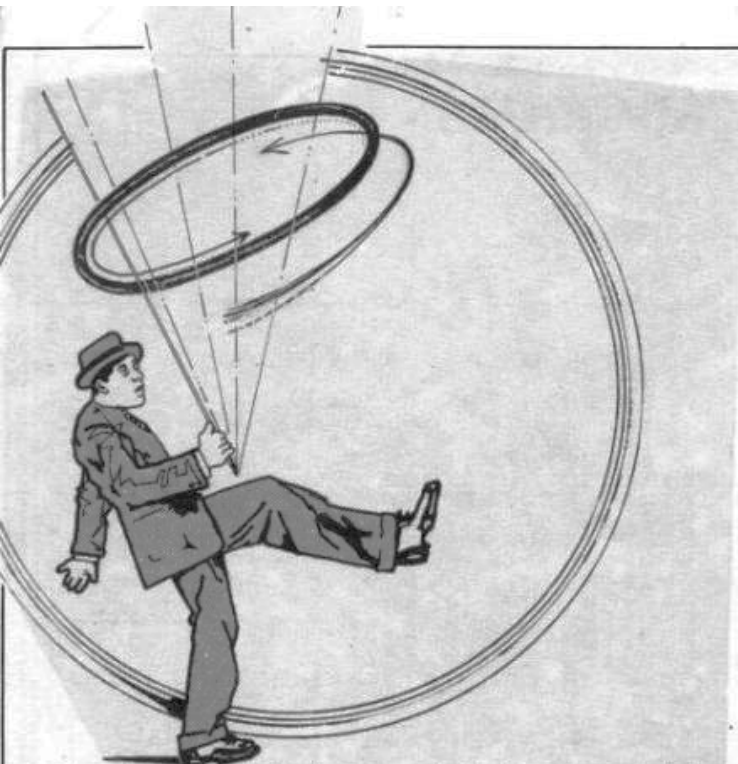
```

```

*3; INK 0; PAPER 6; FLASH 1; b$(b(i,j))
6090 PRINT AT 5+(is-1)*3,11+(js-1)*3; INK 0; PAPER 6; FLASH 1; b$(b(is,js))
6100 LET f=f1: LET c=-128
6110 GO SUB 7000
6120 LET f=f2
6130 GO SUB 7000
6140 RETURN
6500 CLS
6510 PRINT AT 5,0; "Eliga 3 palabras de cuatro letras"
6520 PRINT "y 1 de tres letras"
6530 INPUT "Escriba las primeras cuatro le-tras"; a$
6540 IF LEN a$<>4 THEN GO TO 6530
6550 LET w$(1 TO 4)=a$
6560 PRINT "Primera palabra=" ; a$
6570 INPUT "Escriba la segunda palabra de cuatro letras"; a$
6580 IF LEN a$<>4 THEN GO TO 6570
6590 LET w$(5 TO 8)=a$
6600 PRINT "Segunda palabra=" ; a$
6610 INPUT "Escriba la tercera palabra de cuatro letras"; a$
6620 IF LEN a$<>4 THEN GO TO 6610
6630 LET w$(9 TO 12)=a$
6640 PRINT "Tercera palabra=" ; a$
6650 INPUT "Escriba la palabra de tres le-tras"; a$
6660 IF LEN a$<>3 THEN GO TO 6650
6670 LET w$(13 TO 15)=a$
6680 PRINT "Cuarta palabra=" ; a$
6690 PAUSE 20
6700 LET w$(16)=" "
6710 LET b$=w$
6720 RETURN
7000 POKE f,PEEK f+c
7050 RETURN
8000 LET b(is,js)=b(i,j)
8030 LET b(i,j)=16
8040 LET t=is: LET is=i: LET i=t
8050 LET t=j: LET j=js: LET js=t
8060 RETURN
9000 FOR i=0 TO 7
9010 POKE USR "a"+i,BIN 00000001
9020 POKE USR "b"+i,BIN 00000001
9030 POKE USR "c"+i,0
9040 NEXT i
9050 POKE USR "b"+7,BIN 11111111
9060 POKE USR "c"+7,BIN 11111111
9070 PAPER 4
9080 CLS
9090 RETURN

```

A		B	K
E	F	D	C
9	10	11	12
I	J	L	G
13	14	15	16
M	N	H	O



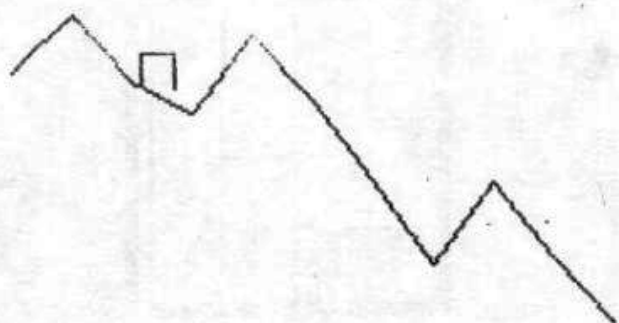
¿ES UD. HABIL CON EL ALAMBRE?

Este es un original programa que pretende poner a prueba su habilidad. Se basa en la antigua versión de un alambre y un aro: dentro del aro se encontraba el alambre y había que recorrerle sin tocarle con lo que la dificultad dependía de las irregularidades en el trazado del alambre y del diámetro del aro. Al igual que en el antiguo juego mecánico, con su Spectrum puede deslizar su aro (representado por un cuadrado de distinta dimen-

sión según el nivel de dificultad elegido) a través de un alambre especial que le dibuja su Spectrum en su pantalla. El movimiento de izquierda a derecha es automático y su misión consiste en coordinar los movimientos verticales ayudado de las teclas 6 y 7. ¡Intente llegar al final sin salirse! Al final se muestra, en porcentaje, el tiempo que ha estado "dentro del camino".

(16K-Spectrum).

Adelante



```

10 REM Seguimiento.
20 GO SUB 1000
30 GO SUB 2000
40 PAUSE 100
50 PRINT AT 1,0; BRIGHT 1;"Ade
lante"
60 BEEP .1,5
70 LET y=100: LET x=10
80 LET b=y: LET r=12-df
90 LET r2=r/2: LET v=2
100 GO SUB 5000
105 OVER 0
110 PRINT AT 1,0;"O.K. ";AT
2,10;INT (hit/(hit+miss)*10000)
/100;"% de tiempo"
120 INPUT "Otra vez (s/n) ";a$
130 IF a$="s" OR a$="S" THEN RU
N
140 IF a$<>"n" AND a$<>"N" THEN
GO TO 120
150 BORDER 7
160 CLS
170 STOP
1000 LET x=10
1010 LET y=100
1020 LET d=30
1030 LET p=0
1040 INK 0
1050 PAPER 7
1060 BORDER 4
1070 CLS
1150 INPUT "Seleccione el nivel
de dificu- tad"-" -1 (Facil)...
5 (Dificil) ";df
1160 IF df<1 OR df>5 THEN GO TO
1150
1170 CLS
1180 OVER 1
1190 RETURN
2000 PLOT x,y
2010 FOR i=1 TO 10
2020 LET r=d-INT (RND*2*d)
2030 LET y=y+r
2040 IF y>160 THEN LET y=y-r: LE
T r=-r
2050 IF y<10 THEN LET y=y+r: LET
r=-r
2060 DRAW 20,r
2070 NEXT i
2080 LET hit=0
2090 LET miss=0
2100 RETURN
5000 PLOT x-r2,y-r2
5010 DRAW 0,r
5020 DRAW r,0
5030 DRAW 0,-r
5040 DRAW -r,0
5100 LET a$=INKEY$
5110 IF a$="6" AND y-r2>2*v THEN
LET b=b-2*v
5120 IF a$="7" AND y+r<175-2*v T
HEN LET b=b+2*v
5510 PLOT x-r2,y-r2
5520 DRAW 0,r
5530 DRAW r,0
5540 DRAW 0,-r
5550 DRAW -r,0
5560 LET x=x+v
5565 IF x>210 THEN RETURN
5570 LET y=b
5580 FOR i=-r+1 TO r-1
5590 IF POINT (x,y+i) THEN LET h
it=hit+1: PAUSE 2: GO TO 5000
5600 NEXT i
5610 LET miss=miss+1
5700 BEEP .01,3
5710 GO TO 5000

```


TRES EN RAYA

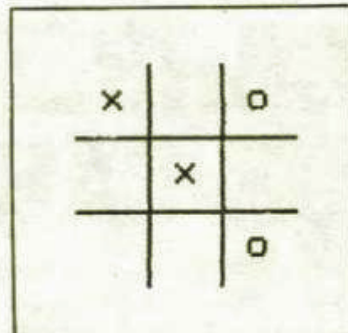
Las tres en raya es un juego muy antiguo. Se jugaba con monedas, maderas, piedras... o ceros y equis si quiere jugar con Spectrum.

Usted empieza (sus fichas son las "equis"). Para efec-

tuar su jugada ha de indicar la fila y columna donde desea colocar su ficha; así, 12 es la fila 1, columna 2. Después el Spectrum le dará su jugada.

Hay que confesar que es un juego bastante sencillo, ya

que no permite la movilidad de fichas, es decir, acaba al darle 9 posiciones. Le retamos a que lo mejore y nos lo mande a nuestro concurso de programas. ¿Se atreve? (16K Spectrum).



```

10 REM Las tres en raya.
20 GO SUB 8000
30 GO SUB 6000
40 GO SUB 7000
50 GO SUB 5000
60 IF end=1 THEN GO TO 9500
70 IF end=2 THEN GO SUB 7000:
GO TO 9000
75 IF dr=1 THEN GO TO 9100
80 GO SUB 7000
90 GO TO 30
99 STOP
4000 DIM x(4): DIM y(4)
4020 FOR l=1 TO 3
4030 LET s=0
4040 LET t=0
4050 FOR k=1 TO 3
4060 IF a(l,k)=1 THEN LET s=s+1
4070 IF b(l,k)=1 THEN LET t=t+1
4080 NEXT k
4090 IF s=0 THEN LET y(t+1)=y(t+
1)+1
4100 IF t=0 THEN LET x(s+1)=x(s+
1)+1
4105 NEXT l
4110 FOR l=1 TO 3
4120 LET t=0
4125 LET s=0
4130 FOR k=1 TO 3
4140 IF a(k,l)=1 THEN LET s=s+1
4150 IF b(k,l)=1 THEN LET t=t+1
4160 NEXT k
4170 IF s=0 THEN LET y(t+1)=y(t+
1)+1
4180 IF t=0 THEN LET x(s+1)=x(s+
1)+1
4185 NEXT l
4190 GO SUB 4300
4200 GO SUB 4400
4210 IF x(4)=1 THEN LET end=1: R
ETURN
4215 IF y(4)=1 THEN LET end=2
4220 LET e=128*y(4)-63*x(3)+31*y
(3)-15*x(2)+7*y(2)
4230 RETURN
4300 LET t=0
4310 LET s=0
4320 FOR k=1 TO 3
4330 LET t=t+a(k,k)
4340 LET s=s+b(k,k)
4350 NEXT k
4360 IF s=0 THEN LET x(t+1)=x(t+
1)+1
4370 IF t=0 THEN LET y(s+1)=y(s+
1)+1
4380 RETURN
4400 LET t=0
4410 LET s=0
4420 FOR k=1 TO 3
4430 LET t=t+a(4-k,k)
4440 LET s=s+b(4-k,k)
4450 NEXT k
4460 IF s=0 THEN LET x(t+1)=x(t+
1)+1
4470 IF t=0 THEN LET y(s+1)=y(s+
1)+1
4480 RETURN

```

```

5000 LET m=-256: LET dr=1
5005 FOR j=1 TO 3
5010 FOR i=1 TO 3
5015 IF a(i,j)=1 OR b(i,j)=1 THE
N GO TO 5040
5016 LET dr=0: LET b(i,j)=1
5020 GO SUB 4000
5025 IF end=1 THEN RETURN
5030 IF e>m THEN LET m=e: LET a=
i: LET b=j
5035 LET b(i,j)=0
5040 NEXT j
5050 NEXT i
5060 LET b(a,b)=1
5070 RETURN
6000 INPUT "Movimiento (fila col
umna) ";a$
5005 IF LEN a$<>2 THEN BEEP .5,-
5: GO TO 6000
6010 LET j=VAL a$(1): LET i=VAL
a$(2)
6020 IF i<1 OR i>3 THEN BEEP .5,
5: GO TO 6000
6030 IF j<1 OR j>3 THEN BEEP .5,
5: GO TO 6000
6040 IF a(i,j) THEN GO TO 6100
6050 IF b(i,j)=1 THEN GO TO 6100
6060 LET a(i,j)=1
6070 PRINT AT 21,0;"
6080 RETURN " : REM 25 espacios
6100 PRINT AT 21,0;"Posicion ocu
pada";
6110 BEEP .5,10
6120 GO TO 6000
7000 FOR j=1 TO 3
7010 FOR i=1 TO 3
7020 IF a(i,j)=1 THEN PRINT AT j
*3+3,i*3+8;"X";
7030 IF b(i,j)=1 THEN PRINT AT j
*3+3,i*3+8;"O";
7040 IF a(i,j)+b(i,j)=0 THEN PRI
NT AT j*3+3,i*3+8;" ";
7050 NEXT i
7060 PRINT
7070 NEXT j
7080 RETURN
8000 DIM a(3,3)
8004 PLOT 104,135
8005 DIM b(3,3)
8010 GO SUB 7000
8050 DRAW 0,-72
8060 PLOT 128,135
8070 DRAW 0,-72
8080 PLOT 80,111
8090 DRAW 72,0
8100 PLOT 80,87
8110 DRAW 72,0
8120 LET end=0
8130 LET dr=0
8200 RETURN
9000 PRINT "Yo gano!!"
9010 GO TO 9600
9100 PRINT "Tablas!!"
9110 GO TO 9600
9500 PRINT "Tu ganas!!"
9600 INPUT "Otra (s/n) ";a$
9610 IF a$="s" THEN RUN
9620 CLS

```

8040 PLOT 104,135

CARRERA DE CABALLOS

Póngase su mejor traje: ¡Hoy vamos a las carreras! Las apuestas son cinco a uno, es decir, si apuesta 20 y gana recibirá 100. Su Spectrum le llevará las cuentas y no le engañará. Al apostar por un caballo, éste se torna blanco para reconocerle mejor. La carrera es un juego de azar y no hay nada que Vd. pueda hacer para que su caballo gane: ¡Es sólo una cuestión de suerte!

Su presentación es bastante buena. Es una lástima que no permita jugar con más de una persona, con lo que podría apostar con sus amigos.

(16K-Spectrum).

Notas gráficas:
Línea 1310 - GraphicShift 6.
Línea 1540 - Graphic H.
Línea 4120 - Graphic H.

```

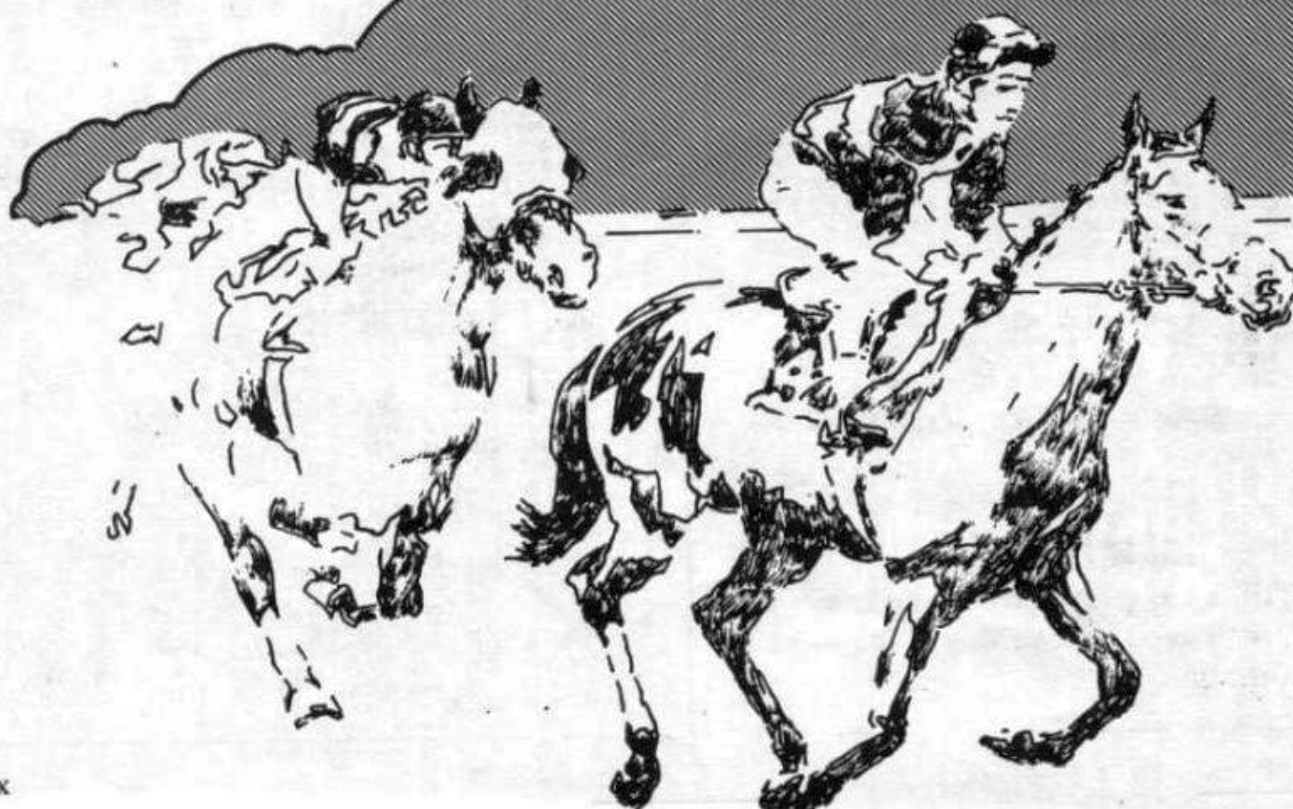
10 REM Carreras.
5000 POKE USR "h"+0,BIN 000000000
5100 POKE USR "h"+1,BIN 000001100
5200 POKE USR "h"+2,BIN 000011010
5300 POKE USR "h"+3,BIN 111111111
5400 POKE USR "h"+4,BIN 011111101
5500 POKE USR "h"+5,BIN 01000010
5600 POKE USR "h"+6,BIN 100000001
5700 POKE USR "h"+7,BIN 000000000
600 GO SUB 5000
610 LET total=100
1000 PAPER 4: INK 0
1100 CLS
1130 DIM x(5)
1140 DIM y(5)
1200 FOR x=0 TO 31
1220 PRINT AT 1,x; INK 0;"#";AT
11,x;"#"
1230 NEXT x
1300 FOR y=2 TO 10
1310 PRINT AT y,31; INK 0;"
"
1320 NEXT y
1500 FOR y=1 TO 5
1520 LET x(y)=2
1530 LET y(y)=y*2
1540 PRINT AT y(y),x(y)-1; INK 7
;y; INK 0;"A"
1550 NEXT y
1560 GO SUB 3000
1570 LET tempo=8
1710 GO SUB 4000
1720 GO TO 1710

```

```

#####
1          *
2          *
3          *
4          *
5          *
#####

```



PROGRAMAS

Insertar:

5030 PRINT AT 15,1;"□h" (□=espacio)

5035 PAUSE 2

```

2000 INK 0
2010 PAPER 7
2020 CLS
2040 BORDER 1
2050 PRINT AT 12,6;"C A R R E R
A S"
2060 RETURN
3000 INPUT "Por que caballo apue
sta";b
3010 IF b<1 OR b>5 THEN PRINT AT
18,0;"Ese caballo no corre."; G
O TO 3000
3020 INPUT "Tiene ";(total)
"Cuanto apuesta";m
3030 IF total-m<0 THEN PRINT AT
18,0;"No tiene tanto dinero!!";
BEEP .1,10; GO TO 3020
3040 LET total=total-m
3050 PRINT AT 18,0;" REM 30 espac
ios.
4000 REM Movimiento de los cabal
los.
4010 FOR z=1 TO 5
4050 PRINT AT y(z),x(z); INK 4;"
4110 LET x(z)=x(z)+(1+RND*.6)
4120 PRINT AT y(z),x(z); INK (z=
b)*7;"A"
4130 IF x(z)>30 THEN GO TO 8000
4135 GO SUB 6000
4138 IF t=99 THEN RESTORE
4140 NEXT z
4150 RETURN
5000 GO SUB 2000
5005 LET tempo=4
5010 LET i=1
5015 GO SUB 6000
5020 IF t=99 THEN PRINT AT 15,i+
1;" "; RESTORE : RETURN
5030 PAUSE 2
5040 LET i=i+.4
5050 GO TO 5015
6000 DATA 9,.5,9,.5,9,.5,6,.5,9,
.5,11,.5,9,.5,6,.5,-99,.5,6,.5,4
.1,5,6,.5,4,1
6010 DATA 9,.5,9,.5,6,.5,9,.5,11
.5,9,.5,6,.5
6020 DATA -99,.5,6,.25,4,.25,2,.
25,4,.25,6,.5,4,.5,2,1,5
6030 DATA -99,1,2,.75,2,.25,6,.5
.9,.5,14,1,5
6040 DATA -99,.5,11,.75,11,.25,1
4,.5,11,.5,9,1,5
6050 DATA 6,.25,7,.25,9,.5,9,.5,
6,.25,6,.25
6060 DATA 9,.25,9,.25,11,.5,9,.5
6,1
6070 DATA 4,.5,6,.5,7,.25,6,.5,4
,.25,4,.25,2,1,5
6080 DATA 99,99
6090 READ p,t
6100 IF t=99 THEN RETURN
6110 LET t=t/tempo
6120 IF p=-99 THEN PAUSE INT (t*
50); RETURN
6130 BEEP t,p
6140 RETURN
8000 IF z=b THEN PRINT AT 18,0;"
Ganaste "; LET total=total
+(5*m); PRINT INT (5*m)
8010 IF z<>b THEN PRINT AT 18,0;"
Perdiste ";m
8020 IF total<=0 THEN PRINT AT 1
8,0; FLASH 1;"Te arruinaste!!";
PAUSE 500; GO TO 9020
9000 INPUT "Tienes ";(total) "Ot
ra carrera";a$
9005 RESTORE
9010 IF a$(1)="s" OR a$(1)="5" T
HEN GO TO 1000
9020 INK 0
9030 PAPER 7
9040 BORDER 7
9050 CLS

```

GREENPEACE



Aunque éste no sea el año de la ballena, tampoco hay que olvidarlas. Muchos van en su búsqueda a través de los *iceberg*. Lo que Vd. debe hacer es guiar a la ballena a través de escondidos pasajes entre los *iceberg*. Con ello ha de intentar lograr un doble objetivo: que los cazadores choquen con los bloques de hielo o que choquen entre ellos. Tiene distintos niveles de dificultad que afectan al número de *icebergs*. Para mover la ballena utilice las teclas conocidas (5, 6, 7 y 8). El juego finaliza cuando los

cazadores se han hundido (cosa no probable) o cuando la ballena es alcanzada por un arpón (bastante más probable).

(16K-Spectrum).

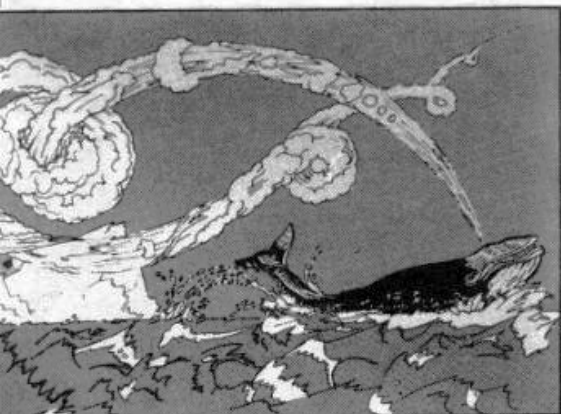
Notas gráficas:

Línea 1030 - Graphic e
Línea 1040 - Graphic u
Línea 1070 - Graphic i
Línea 2050 - Graphic e
Línea 2150 - Graphic i
Línea 2240 - Graphic u
Línea 3110 - Graphic u
Línea 4130 - Graphic e

JUGADA NO VALIDA



La ballena murio.



```

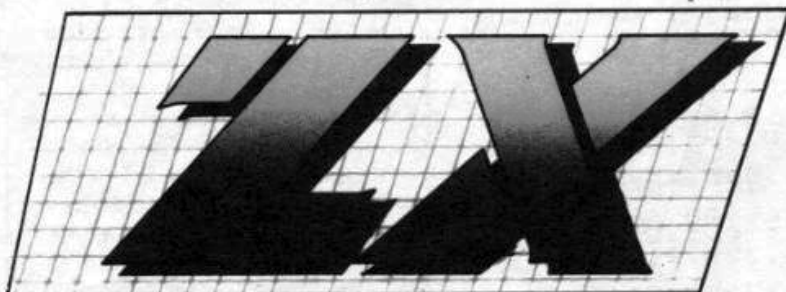
5 REM Salvar la ballena.
10 DIM X(20)
20 DIM Y(20)
300 POKE USR "u"+0,BIN 00000000
310 POKE USR "u"+1,BIN 00000000
320 POKE USR "u"+2,BIN 00110000
330 POKE USR "u"+3,BIN 01111000
340 POKE USR "u"+4,BIN 11111001
350 POKE USR "u"+5,BIN 11111111
360 POKE USR "u"+6,BIN 11111001
370 POKE USR "u"+7,BIN 00000000
380 POKE USR "i"+0,BIN 00000000
390 POKE USR "i"+1,BIN 00100000

```

```

620 POKE USR "i"+2,BIN 01110010
630 POKE USR "i"+3,BIN 01110110
640 POKE USR "i"+4,BIN 01110110
650 POKE USR "i"+5,BIN 11111111
660 POKE USR "i"+6,BIN 11111111
670 POKE USR "i"+7,BIN 11111111
700 POKE USR "e"+0,BIN 00000000
710 POKE USR "e"+1,BIN 11001000
720 POKE USR "e"+2,BIN 01011000
730 POKE USR "e"+3,BIN 00111000
740 POKE USR "e"+4,BIN 11111111
750 POKE USR "e"+5,BIN 00111100
760 POKE USR "e"+6,BIN 00001000
770 POKE USR "e"+7,BIN 00000110
1000 INK 0: PAPER 5
1010 CLS
1020 PRINT AT 2,2;"Salvar la bal
lena."
1030 PRINT AT 5,2;"Esquimales ";
INK 2;" "
1040 PRINT "quieren cazar una bl
lena."
1050 PRINT "Para salvarla ha de
romper sus kayaks en el iceber
g."; INK 7;" "
1080 PRINT AT 15,2;"3 niveles de
dificultad:"
1090 PRINT "Elija: ";
1110 INPUT "1=Difícil, 2=Norma
l, 3=Facil.";d
1120 IF d<1 OR d>3 THEN GO TO 11
10
1500 PAPER 5: INK 5
1550 CLS
2000 REM Impresion kayaks

```



TARIFA DE SUSCRIPCION

	CORREO ORDINARIO		CORREO CERTIFICADO		CORREO AEREO		CORREO AEREO-CERTIF.	
	PTAS.	\$	PTAS.	\$	PTAS.	\$	PTAS.	\$
ESPAÑA	2.200	16	2.453	18	2.244	16	2.497	18
EUROPA, MARRUECOS, TUNEZ, TURQUIA, ARGELIA Y CHIPRE. COSTA RICA, CUBA, CHILE, PA- RAGUAY Y REP. DOMINICANA. GIBRALTAR Y PORTUGAL	2.618	19	3.366	24	2.717	19	3.465	25
FILIPINAS	2.563	18	3.311	24	3.091	22	3.839	27
RESTO DEL MUNDO	2.442	17	3.190	23	2.305	16	3.053	22
	2.442	17	2.695	19	2.734	20	2.987	21
	2.618	19	3.366	24	3.146	22	3.894	28

CUPON DE PEDIDO

Recorte y envíe este CUPON DE PEDIDO a: **ZX** Jerez, 3 Madrid 16

NOMBRE _____

CALLE _____ N.º _____

CIUDAD _____ D.P. _____ PROVINCIA _____

PROGRAMAS

```

2010 FOR c=1 TO 20
2020 LET x=((SGN (RND-.5))*((RND
*4)+11))+15
2030 LET y=((SGN (RND-.5))*((RND
*4)+6))+9
2040 INK 2
2050 PRINT AT y,x;"X"
2055 LET x(c)=x: LET y(c)=y
2060 NEXT c
2070 REM Impresion icebergs
2080 FOR c=1 TO 20
2090 LET x=((SGN (RND-.5))*((RND
*4)+5-d))+15
2100 LET y=((SGN (RND-.5))*((RND
*4)+4-d))+9
2110 INK 7
2120 PRINT AT y,x;"A"
2130 NEXT c
2140 REM Impresion ballena
2150 INK 0
2160 LET x=INT ((RND*2)+10)
2170 LET y=INT ((RND*2)+10)
2180 PRINT AT y,x;"A"
2190 GO SUB 3000
2200 LET f=0
2210 FOR c=1 TO 20
2220 IF x(c)=0 THEN GO TO 2340
2230 LET f=1
2240 GO SUB 4000
2250 NEXT c
2260 IF f=0 THEN GO TO 8000
2270 GO TO 2300
2280 REM Movimiento ballena
2290 BEEP .1,10
2300 LET z=x: LET v=y

```

```

3040 LET a$=INKEY$
3050 IF a$="" THEN GO TO 3040
3060 IF a$="5" THEN LET x=x-1
3070 IF a$="8" THEN LET x=x+1
3080 IF a$="7" THEN LET y=y-1
3090 IF a$="6" THEN LET y=y+1
3100 PRINT AT y,z; INK 5;" "
3110 PRINT AT y,x; INK 0;"A"
3120 RETURN
4000 REM Movimiento kayaks
4010 PRINT AT y(c),x(c); INK 5;"
"
4030 LET e=0: LET d=0
4040 LET d=SGN (x(c)-x)
4050 LET x(c)=INT (x(c)-d)
4060 LET e=SGN (y(c)-y)
4070 LET y(c)=INT (y(c)-e)
4080 IF ATTR (y(c),x(c))=47 THEN
LET x(c)=0: GO TO 4500
4110 IF ATTR (y(c),x(c))=40 THEN
GO TO 7000
4120 INK 2
4130 PRINT AT y(c),x(c); "X"
4500 RETURN
7000 PRINT AT 20,1; PAPER 7;"La
ballena murio."
7005 PRINT AT y(c),x(c); " "
7010 GO TO 9000
8000 PRINT AT 20,1; PAPER 7;"Es c
apo por esta vez."
9000 INPUT "Otra vez (s/n)";g$
9010 IF g$(1)="s" OR g$(1)="S" T
HEN RUN
9020 STOP

```

FICHAS SALTARINAS

■ ■ ■ ■ ■ ■ ■ ■ ■
1 2 3 4 5 6 7 8 9

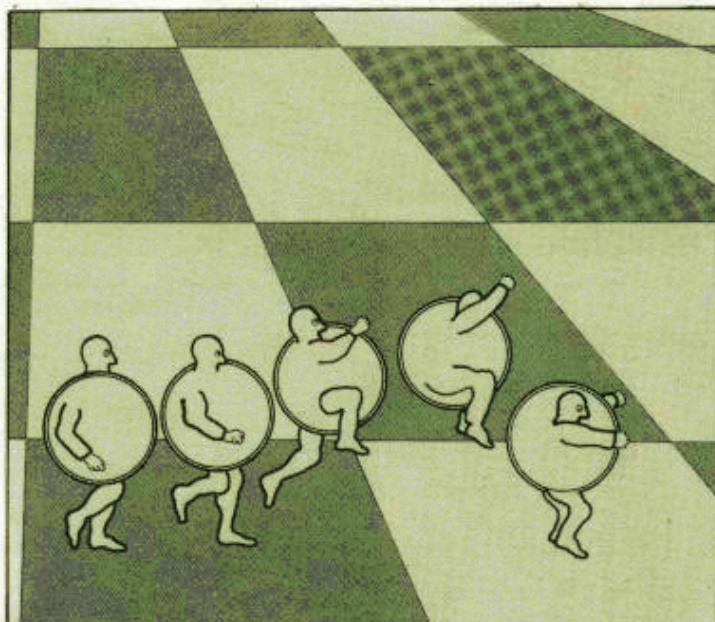
```

100 LET a$="■ ■ ■ ■ ■ ■ ■ ■ ■"
110 LET b$=a$
120 LET count=0
130 CLS
140 PRINT AT 5,3;b$;AT 6,3;"1 2
3 4 5 6 7 8 9"
150 IF b$(1 TO 7)=a$(11 TO 17)
AND b$(11 TO 17)=a$(1 TO 7) THEN
GO TO 250
160 INPUT "Movimiento ";k$
170 IF LEN k$<2 THEN GO TO 160
180 LET from=2*(CODE k$(1)-48)-
1
190 LET to=2*(CODE k$(2)-48)-1
200 IF b$(to)<>" " OR ABS (to-f
rom)>4 THEN GO TO 160
210 LET count=count+1
220 LET b$(to)=b$(from)
230 LET b$(from)=" "
240 GO TO 140
250 PRINT "Lo lograste en ";cou
nt;" movimientos."
260 INPUT "Repetimos ";k$
270 IF CODE k$=115 THEN RUN

```

Hay programas pequeños y simples dotados de cierto atractivo. Este es uno de ellos en el que tiene cuatro fichas a la izquierda y cuatro a la derecha separadas por un espacio en blanco. Cada pieza sólo puede moverse hacia un espacio vacío o saltar sobre otra ficha para acceder a dicho espacio vacío siempre que no salte más de

una ficha. Se ha de introducir por tanto, dos números: el que indica la pieza a mover y el de la posición a ocupar. De acuerdo con las reglas descritas el primer movimiento sólo puede ser 35, 45, 65 ó 75. ¿Fácil no? El objetivo es llegar a la posición opuesta con el menor número de movimientos. ¡Suerte! (16K-Spectrum).



SOBRE EL ABISMO

Quedan Cruzaron Perdidos
4 0 1

†



En este juego en el que se combina el gráfico de alta resolución con el de baja resolución, ha de saltar a través de un inmenso barranco. Pero su salto no lo es en el vacío si logra ayudarse de la "liana" que oscila sobre la inmensidad del abismo que tiene a sus pies. ¿Tiene el valor suficiente para conducir a sus cinco exploradores favoritos? Lo único que tiene que hacer es decidir cuándo empezar a correr para llegar a tiempo de engancharse de la cuerda (presionando cualquier tecla) empieza a correr.

(16K-Spectrum).

Notas gráficas:
Línea 3070 - GraphicShift
8.

```
10 REM Barranco.
20 DIM x(16)
30 DIM y(16)
40 GO SUB 3000
50 GO SUB 2000
60 GO SUB 7500
70 GO SUB 6000
80 GO SUB 1500
90 IF men=0 THEN GO TO 9000
100 GO TO 80
1500 FOR t=1+r TO n-r
1510 PLOT 127,150
1520 DRAW x(t),y(t)
1525 LET s=1: GO SUB 4000
1530 PLOT 127,150
1540 DRAW OVER 1;x(t),y(t)
```

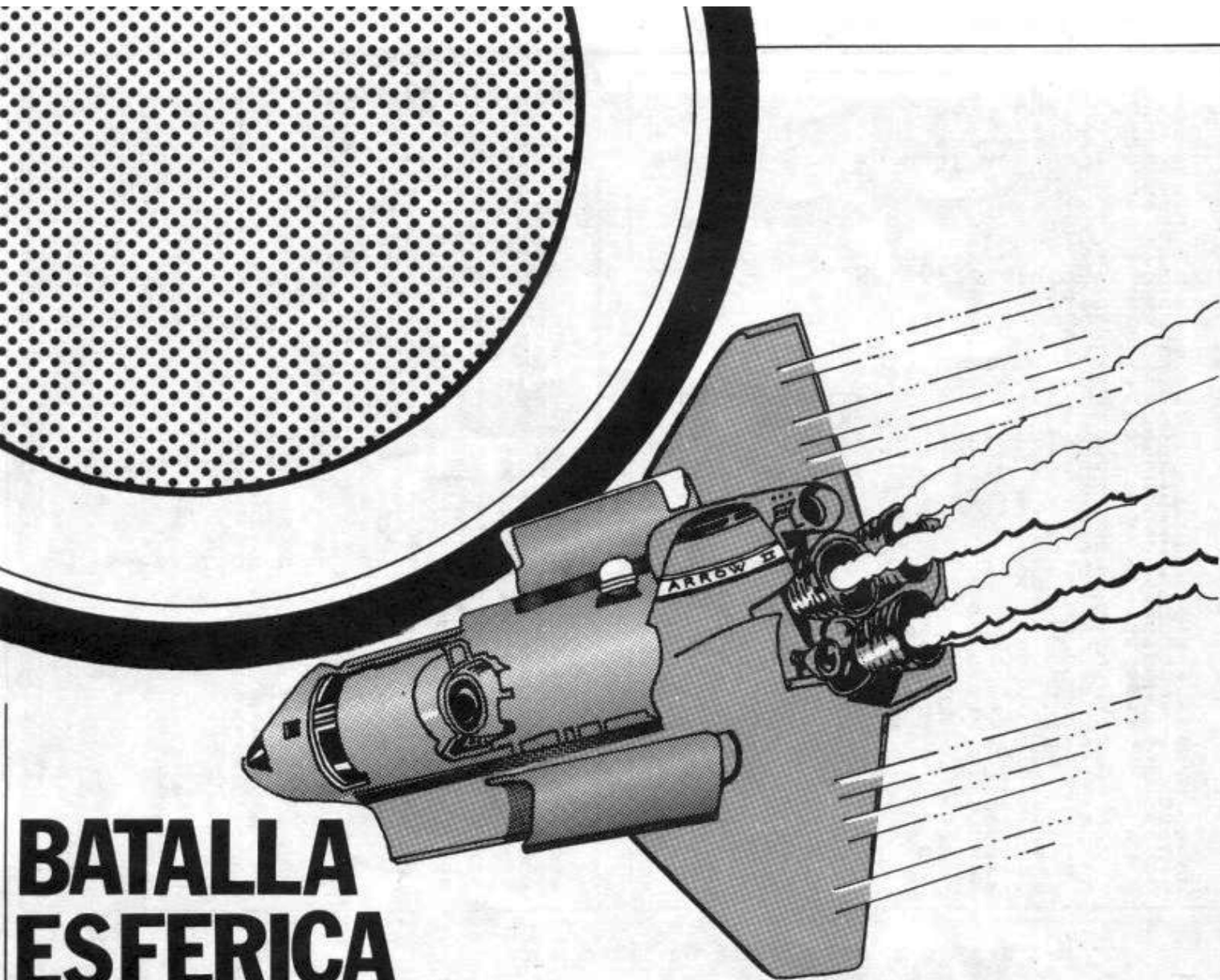
```
1550 IF j=1 THEN LET s=2: GO SUB
4000
1560 NEXT t
1570 IF c=1 THEN GO TO 1680
1600 FOR t=n-r TO 1+r STEP -1
1610 PLOT 127,150
1620 DRAW x(t),y(t)
1625 LET s=3: GO SUB 4000
1630 PLOT 127,150
1640 DRAW OVER 1;x(t),y(t)
1650 IF j=1 THEN LET s=4: GO SUB
4000
1660 NEXT t
1670 RETURN
1680 LET across=across+1
1690 LET dx=(across-1)*10+5
```



```

5050 GO SUB 4020
5090 IF ABS (MX-127-X(t)) <10 AND
S=2 THEN LET C=1: GO TO 4020
5200 IF MX<160 THEN GO TO 7000
5210 RETURN
6000 LET MY=50
6010 LET MX=240
6020 LET DY=MY
6030 LET DX=MX
6040 LET J=0
6050 GO SUB 4020
6060 PRINT AT 0,1;
"Quedan Cruzaron Perdidos
"
6070 PRINT AT 1,4;MEN;TAB 12;ACR
OSS;TAB 22;LOST
6080 RETURN
7000 GO SUB 4020
7010 LET MY=MY-2
7020 LET DY=MY
7030 GO SUB 4020
7040 BEEP .01,MY/2
7050 GO SUB 4020
7060 IF MY>30 THEN GO TO 7010
7070 BEEP 1,-10
7080 LET LOST=LOST+1
7090 LET C=0
7100 LET J=0
7110 LET MEN=MEN-1
7120 IF MEN=0 THEN GO TO 6060
7130 GO SUB 6000
7140 RETURN
7500 LET MEN=5
7510 LET LOST=0
7520 LET ACROSS=0
7530 LET J=0
7540 LET C=0
7550 LET R=INT (RND*4)
7560 PRINT AT 15,0;"
REM 10 espacios.
7570 PRINT AT 16,0;"
REM 10 espacios.
7580 RETURN
9000 INPUT "Perdio ";(LOST)"Otr
a vez ";A$
9010 IF A$="S" OR A$="s" THEN GO
TO 60
9020 INK 0
9030 PAPER 7
9040 CLS

```

BATALLA ESFERICA

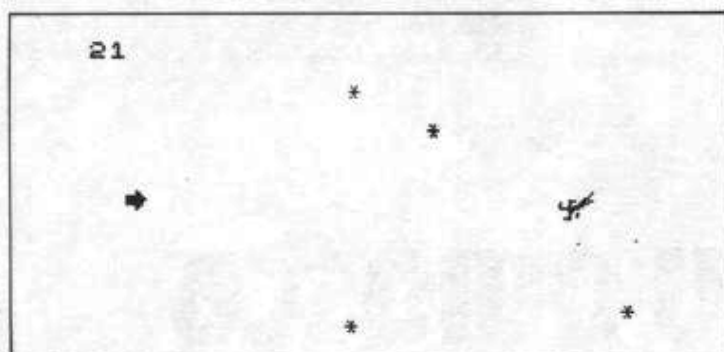
Este es un original juego de guerras galácticas. Su nave tiene la forma de flecha que le indica el sentido del avance. La pantalla simula un mundo esférico: si sale por arriba aparece por abajo y si sale por la izquierda aparece por la derecha. El tiempo es su mayor enemigo: dispone de 200 segundos para destruir al enemigo.

Tiene distintos niveles de dificultad que afectan al curso de la nave enemiga y al número de estrellas que

aparecen y que son un obstáculo. Su nave se mueve continuamente y cada vez que presione una tecla varia su curso en 45 grados en sentido de las agujas del reloj. Para efectuar disparos presione la tecla del "7".

(16K-Spectrum).

Notas gráficas:
Línea 8500 - Graphic chbfdeag
Línea 8760 - Graphic ij



```

10 REM laser
20 GO SUB 7000
30 GO SUB 8000
40 GO SUB 7500
50 IF INT t>1000 THEN GO TO 90
00
60 GO SUB 1000
70 GO SUB 3000
80 IF h=1 THEN GO TO 9000
90 GO SUB 4000
100 GO TO 40
1000 LET a$=INKEY$
1010 IF a$="" THEN RETURN
1020 IF a$="7" THEN GO TO 2000
1030 LET k=k+1
1040 IF k>8 THEN LET k=1
1050 RETURN
2000 LET xl=x*8+4
2010 LET yl=175-y*8-4
2015 GO SUB 2500
2050 PLOT xl,yl
2060 LET dx=0
2070 IF v(k)=1 THEN LET dx=255-x
2080 IF v(k)=-1 THEN LET dx=-xl
2090 LET dy=0
2100 IF w(k)=1 THEN LET dy=-yl
2120 IF w(k)=-1 THEN LET dy=175-yl
2130 IF v(k)*w(k)=0 THEN GO TO 2000
2140 IF ABS dx<ABS dy THEN LET d=ABS dx*SGN dy: GO TO 2200
2150 LET dx=ABS dy*SGN dx
2200 DRAW dx,dy
2210 PLOT xl,yl
2215 BEEP .01,10

```


PROGRAMAS

```

2220 DRAW OVER 1;dx,dy
2230 IF h=0 THEN RETURN
2235 LET mx=b*8+4: LET my=175-a*
0-4
2240 FOR i=1 TO RND*5+20
2250 PLOT mx,my
2260 LET dx=10-RND*20
2270 IF mx+dx>255 OR mx+dx<0 THE
N GO TO 2330
2280 LET dy=10-RND*20
2290 IF my+dy>175 OR my+dy<0 THE
N GO TO 2330
2300 PLOT mx,my
2310 DRAW dx,dy
2320 BEEP .01,-10
2330 NEXT i
2340 RETURN
2500 LET h=0
2510 LET dy=a-y
2520 LET dx=b-x
2530 IF w(k)*dx<>v(k)*dy THEN RE
TURN
2540 IF ABS v(k)*SGN dx<>v(k) OR
ABS w(k)*SGN dy<>w(k) THEN RETU
RN
2550 LET h=1
2560 RETURN
3000 IF nb1=0 THEN PRINT AT y,x;
" "
3010 LET x=x+v(k)
3020 LET y=y+w(k)
3030 IF x<0 THEN LET x=31
3040 IF x>31 THEN LET x=0
3050 IF y<0 THEN LET y=21
3060 IF y>21 THEN LET y=0
3065 IF ATTR (y,x)<>15 THEN BEEP
.1,-10: LET nb1=1: GO TO 1030
3070 PRINT AT y,x;m$(k)
3080 LET nb1=0
3090 RETURN
4000 IF RND>1.05-df/20 THEN LET
z=z+1
4010 IF z>8 THEN LET z=1
4020 IF nb2=0 THEN PRINT AT a,b;
" "
4030 LET a=a+v(z)
4040 LET b=b+w(z)
4050 IF b<0 THEN LET b=31
4060 IF b>31 THEN LET b=0
4070 IF a<0 THEN LET a=21
4080 IF a>21 THEN LET a=0
4085 IF ATTR (a,b)<>15 THEN BEEP
.1,-10: LET nb2=1: LET z=z+1: G
O TO 4010
4090 PRINT AT a,b;w$(r+1)
4100 LET r=NOT r
4110 LET nb2=0
4120 RETURN
7000 PAPER 1
7010 BORDER 5
7020 INK 7
7030 CLS
7040 PRINT AT 2,5;"Ataque galact
ico"
7100 INPUT "Selecione el nivel
de dificultad: 1 (Facil)... 1
0 (Dificil) ";df
7110 IF df<1 OR df>10 THEN GO TO
7100
7120 CLS
7130 RETURN
7500 LET t=(PEEK 23672+256*PEEK
23673+65536*PEEK 23674)/5
7510 PRINT AT 0,0;INT t/10;" "
7520 RETURN
7900 IF RND>.1+df/50 THEN RETURN
7910 PRINT AT RND*21,RND*31; INK
6;" "
7920 RETURN
8000 DIM w(8): DIM v(8)
8010 DIM s$(8,8): DIM r$(8,8)
8020 DATA 0,1,1,1,1,0,1,-1,0,-1,
-1,-1,-1,0,-1,1
8030 FOR i=1 TO 8
8040 READ w(i),v(i)

```

```

8050 NEXT i
8060 LET k=1
8070 LET x=20
8080 LET y=10
8090 LET v=v(k)
8100 LET w=w(k)
8110 LET s$(1)="00011000"
8120 LET s$(2)="00111100"
8130 LET s$(3)="01111110"
8140 LET s$(4)="11111111"
8150 LET s$(5)="00111100"
8160 LET s$(6)="00111100"
8170 LET s$(7)="00111100"
8180 LET s$(8)="00111100"
8190 LET r$(1)="11111000"
8200 LET r$(2)="11110000"
8210 LET r$(3)="11110000"
8220 LET r$(4)="11111100"
8225 LET r$(5)="10111110"
8230 LET r$(6)="00011111"
8240 LET r$(7)="00001110"
8250 LET r$(8)="00000100"
8260 FOR i=1 TO 8
8270 LET a=0: LET b=0: LET c=0
8275 LET d=0: LET e=0: LET f=0
8280 FOR j=1 TO 8
8290 LET a=a*2+VAL (s$(i,j))
8300 LET b=b*2+VAL (s$(9-j,i))
8310 LET c=c*2+VAL (s$(j,i))
8320 LET d=d*2+VAL (r$(i,j))
8330 LET e=e*2+VAL (r$(9-j,i))
8340 LET f=f*2+VAL (r$(9-i,9-j))
8345 GO SUB 7900
8380 NEXT j
8390 POKE USR "a"+i-1,a
8400 POKE USR "b"+8-i,a
8410 POKE USR "c"+i-1,b
8420 POKE USR "d"+i-1,c
8430 POKE USR "e"+i-1,d
8440 POKE USR "f"+8-i,d
8450 POKE USR "g"+i-1,e
8460 POKE USR "h"+i-1,f
8490 NEXT i
8500 LET m$="*****"
8510 LET a=10
8520 LET b=10
8530 LET z=INT (RND*8)+1
8540 POKE USR "i"+0,BIN 00100000
8550 POKE USR "i"+1,BIN 01000010
8560 POKE USR "i"+2,BIN 00100101
8570 POKE USR "i"+3,BIN 00011000
8580 POKE USR "i"+4,BIN 00011000
8590 POKE USR "i"+5,BIN 10100100
8600 POKE USR "i"+6,BIN 01000010
8610 POKE USR "i"+7,BIN 00000100
8620 POKE USR "j"+0,BIN 00001110
8630 POKE USR "j"+1,BIN 10001000
8640 POKE USR "j"+2,BIN 10001000
8650 POKE USR "j"+3,BIN 11111000
8660 POKE USR "j"+4,BIN 00011111
8670 POKE USR "j"+5,BIN 00001001
8680 POKE USR "j"+6,BIN 00001001
8690 POKE USR "j"+7,BIN 00111000
8700 LET w$="x5"
8710 LET r=0
8720 INK 7
8730 LET nb1=0
8740 LET nb2=0
8750 POKE 23674,0
8760 POKE 23673,0
8770 POKE 23672,0
8780 LET h=0
8790 RETURN
8800 IF h<>1 THEN GO TO 9500
8810 PRINT AT 21,0;"Le diste!!"
8820 GO TO 9600
8830 PRINT AT 21,0;"Se acabo el
tiempo"
8840 INPUT "Otra vez (s/n) ";a$
8850 IF a$(1)="s" OR a$="S" THEN
RUN
8860 PAPER 7
8870 INK 0
8880 BORDER 7
8890 CLS

```


¡QUE TE PILLA EL GUARDIA!



Este es un sencillo juego en que tendrá que correr bastante si desea escapar y no ser capturado por el guardia que incansablemente le sigue los pasos. Como habrá adivinado Vd. es un ladrón y en sus manos tiene una considerable suma. Dispone de 9 niveles de juego con los que regular la veloci-

dad del juego (especialmente la del representante de la ley). No utiliza alta resolución gráfica por lo que no resulta especialmente vistoso, representando a la ley por "E" y al "fuera de la ley" por "*". Para moverse utilice las teclas 5, 6, 7 y 8.

(16K-Spectrum).

```

1 LET q#0: GO SUB 8000: FOR
j=1 TO 100: NEXT j
2 GO SUB 2000
3 LET s=400
4 PRINT "Ud. es: "; FLASH 1; "*"
FLASH 0; "El guardia es "; FLA
5H 1; "E"
5 FOR f=1 TO 500: NEXT f: CLS

7 LET p=0
10 DIM q(22,32)
20 FOR f=1 TO (z*20)+50
30 LET a=INT (RND*18)+3
40 LET b=INT (RND*28)+3
50 PRINT AT a,b; ""
60 LET q(a,b)=1
70 NEXT f
75 GO SUB 1000
80 LET c=INT (RND*17)+2
90 LET d=INT (RND*27)+2
100 LET q(c,d)=0
110 PRINT FLASH 1; AT c,d; "*"
115 LET q(c,d)=0
120 PRINT AT a,b; ""
130 LET b=INT (RND*27)+2
135 LET q(a,b)=0
140 IF a=c OR b=d THEN GO TO 12

```

```

150 PRINT AT a,b; FLASH 1; "E"
152 PRINT PAPER 4; AT 3,3; " "; AT
3,30; " "; AT 20,3; " "; AT 20,30; "
153 LET q(3,3)=0: LET q(3,30)=0
LET q(20,3)=0: LET q(20,30)=0
154 PAUSE 200
155 LET a1=a: LET b1=b: LET c1=
c: LET d1=d
156 LET s=s-1
157 PRINT AT 0,0; ""
158 IF s=0 THEN GO TO 4000
160 LET c=c+(INKEY$="6" AND q(c
+1,d) <> 1 AND c<20)-(INKEY$="7" A
ND q(c-1,d) <> 1 AND c>3)
170 LET d=d+(INKEY$="8" AND q(c
,d+1) <> 1 AND d<30)-(INKEY$="5" A
ND q(c,d-1) <> 1 AND d>3)
180 LET a=a+(c>a AND q(a+1,b) <>
1)-(c<a AND q(a-1,b) <> 1)
190 LET b=b+(d>b AND q(a,b+1) <>
1)-(d<b AND q(a,b-1) <> 1)
200 PRINT AT a1,b1; " "; AT c1,d1
"; FLASH 1; AT a,b; "E"; AT c,d;
"
210 IF a=c AND d=b THEN GO TO 5
20

```


PROGRAMAS

```

220 IF q(c,d)=2 THEN LET P=P+IN
T (RND*10)*10+10: LET q(c,d)=0
230 IF (c=3 AND d=3) OR (c=20 A
ND d=3) OR (c=3 AND d=30) OR (c=
20 AND d=30) THEN GO TO 1500
235 IF (INKEY$="t" OR INKEY$="T
") AND qw=0 THEN LET qw=1: GO SU
B 3000
240 GO TO 155
500 PRINT AT 0,0;"Te cogieron c
on ";P;AT 1,0;"en las manos...";
AT 2,0;"Repetimos (s/n)?"
510 IF INKEY$<>"s" AND INKEY$<>
"s" THEN GO TO 510
520 RUN
999 STOP
1000 FOR f=1 TO 25
1010 LET a=INT (RND*18)+3
1015 LET b=INT (RND*28)+3
1030 LET q(a,b)=2
1040 NEXT f
1050 RETURN
1500 PRINT AT 0,0;"Logro escapar
con ";P;AT 1,0;"Repetimos (s/n)
?"
1510 IF INKEY$<>"s" AND INKEY$<>
"s" THEN GO TO 1510
1520 RUN
2000 PRINT "Nivel de dificultad.
.(1=Difícil..... 9=Fácil)"
2010 IF INKEY$="" THEN GO TO 201
0
2020 LET z=CODE INKEY$-48
2023 IF z<1 OR z>9 THEN GO TO 20
10
2030 CLS
2040 RETURN
3000 PRINT AT c+1,d;" ";AT c-1,d

```

```

;" ";AT c,d+1;" ";AT c,d-1;" "
3010 LET q(c+1,d)=0
3020 LET q(c-1,d)=0
3030 LET q(c,d+1)=0
3040 LET q(c,d-1)=0
3050 LET s=s-20
3060 RETURN
4000 PRINT AT 0,0;"Se acabo el t
iempo!!";AT 1,0;"Repetimos (s/n)
?"
4010 GO TO 1510
5000 BORDER 6: PAPER 6: INK 0: C
LS
5020 CLS : PAPER 1: INK 7
5050 RETURN

```

Logro escapar con 150
Repetimos (s/n)?

£

*



SUSCRIBASE POR TELEFONO

- * más fácil,
- * más cómodo,
- * más rápido

(91) 250 15 93

Servicio permanente durante las 24 horas del día
"CONTESTADOR AUTOMATICO"

SUSCRIBASE A

ZX

SERVICIO DE EJEMP

Estos son todos los ejemplares
de ORDENADOR POPULAR aparecidos
hasta ahora en el mercado,
con un resumen de sus contenidos.



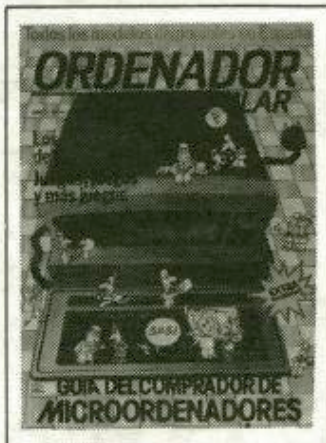
Núm. 1
Marzo 1983
IBM PC. pisando fuerte /
Aprenda Basic con
Sherlock
Holmes / Software /
Juegos / Suplemento
Byte. Imágenes
TRONicas en el cine /
Silicon Valley no es un
mito.



Núm. 2
Abril 1983
Apple. Lisa no es una
chica / Aprenda Basic
con Sherlock Holmes /
Juegos / Suplemento
Byte. El confuso mundo
de las conexiones /
Hardware / Educación /
Chips: La tecnología de
nunca acabar / Tiendas
de Ordenadores.



Núm. 3
Mayo 1983
Actualidad / Crónica de
dos Salones / Sinclair ZX
Spectrum / Aprenda
Basic con Sherlock
Holmes / Juegos /
Suplemento Byte.
Gráficos / El Robot
personal / Espionaje / El
Ordenador del futuro.



Núm. 8 - EXTRA
Noviembre 1983
Cara a cara con los
lenguajes (2.ª parte) / Locos
por el Forth / Suplemento
Byte. El futuro del diseño de
Software / Guía del
comprador de
Microordenadores / Juegos
Pánico en el Pentágono /
Como "Penetrar" un
ordenador / Entrevista.



Núm. 9
Diciembre 1983
Especial juegos / SIMO 83:
balance de tendencias y
novedades / Resolución
gráfica ampliada (2.ª parte)
Hardware / Suplemento
Byte / Microinformática /
Mánager y Ordenadores:
revolución informática.



Para hacer su pedido,
rellene el cupón adjunto, córtelo
y envíelo HOY MISMO a
ORDENADOR POPULAR,
C/ Jerez, 3, Madrid-16.

LARES ATRASADOS



Núm. 4.
Junio 1983
Commodore 64 / Aprenda Basic con Sherlock Holmes / Software / Suplemento Byte. LOGO / Hardware / Así diseño mis juegos / El Zodíaco en 8 Bits.

Núm. 5
Julio / Agosto 1983
Rainbow 100 / Aprenda Basic con Sherlock Holmes / Software / Suplemento Byte. Discos y Diskettes / Hardware / Educación / Videodisco Interactivo.

Núm. 6
Septiembre 1983
Texas Instruments juega dos bazas / Aprenda Basic con Sherlock Holmes / Software / Juegos / Suplemento Byte / Los Nuevos Chips / Hardware / Educación / Tecnología / De la Informática como una de las Bellas Artes.

Núm. 7
Octubre 1983
Cara a cara con los lenguajes (1.ª parte): Cobol-Pascal-Fortran-Basic / Suplemento Byte. Videotex / Hardware / Juegos / Educación / Confesiones de un científico.

Disponemos de tapas
para la
encuadernación de
sus ejemplares.



PRECIO POR UNIDAD:
275 ptas.

En cada tapa se puede
encuadernar 6 números.

EJEMPLARES ATRASADOS Y TAPAS

Los ejemplares atrasados de Ordenador Popular serán una fuente constante de conocimientos, ideas, soluciones, y entretenimientos para el futuro. Todo lo anterior hace recomendable que los guarde ordenadamente en una de las tapas especiales para Ordenador Popular. Cada tapa puede contener 6 ejemplares y cuesta solamente 275 ptas.

Por favor envíe los siguientes ejemplares: 1 2 3 4 5 6 7 8 9 (rodee con un círculo el número del ejemplar que quiera) que le serán facturados al precio de 300 ptas. cada uno, excepto el número 8 cuyo precio es de 475 ptas.

Por favor envíe ____ tapa(s) al precio de 275 ptas. cada una (+ gastos de envío).

El importe lo abonaré:

☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO

American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta: _____

Fecha de caducidad: _____ Firma: _____

NOMBRE _____

DIRECCION _____

CIUDAD _____

PROVINCIA _____



LA SOLDADURA BIEN HECHA

Intentaremos que, habitualmente, aparezca en estas páginas un montaje electrónico destinado a los ordenadores ZX 81, ZX Spectrum o cualquier otro que salga en el futuro de la fábrica de ideas de Clive Sinclair. El objetivo de esta sección es permitir a quienes deseen "cacharrear" con su micro, que puedan dar rienda suelta a su afición. Algunos lectores, interesados en el tema pero no suficientemente formados, nos han hecho notar que estábamos apuntando demasiado alto. ¿Por qué no habrían de aprovechar esta sección los lectores que no presumen de manitas?

Por tanto, hemos decidido publicar periódicamente, entre montaje y montaje, artículos de iniciación.

A no ser que uno utilice esas caras tarjetas que facilitan los montajes provisionales, se impone aprender a soldar con propiedad. Las *meigas*, que tienen la virtud de fastidiarnos los montajes, desaparecerán con el simple encanto de la soldadura bien hecha.

No se confíe usted creyendo que soldar es un juego de niños, que no consiste más que en aplicar los grados de calor adecuados al estaño y ya está.

Fundir el estaño no es más que una parte del proceso total.

Una buena forma de aprender a soldar es observar a alguien que haya conseguido una cierta maestría en el tema. Después se hará lo propio bajo la atenta mirada del maestro.

Como no todos tenemos ese amigo dispuesto a enseñarnos, veremos unas cuantas líneas maestras prácticas para hacer soldaduras fiables, sin por ello freír los componentes, ni los dedos.

Una norma, aunque no tiene por que cumplirse siempre, es que la soldadura

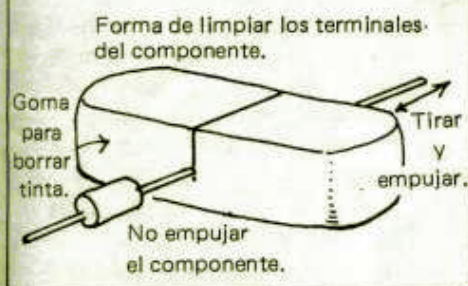
que tiene un aspecto bonito, suele ser correcta.

Es conveniente, como primer paso, hacernos con una serie de útiles y herramientas, de las cuales lo más importante son el soldador y el estaño.

El tipo de circuitería que vamos a soldar sobre placa de circuito impreso, o las tarjetas Veroboard, que ya vienen de fábrica con pistas de cobre y paralelas e hileras de taladros distribuidos uniformemente, requieren un soldador de baja potencia, normalmente de 15 vatios. Por la limitación de esta potencia, el calor que genera se expande con lentitud, y por tanto sirve mejor para los fines de soldaduras puntuales. No obstante, puede ser conveniente proveernos de un soldador de mayor potencia, de unos 30 a 45 vatios, que nos facilitará la soldadura cuando existan grandes masas metálicas y el calor se disipe con rapidez.

Algo recomendable es comprarnos un modelo de puntas intercambiables, de tal manera que podamos utilizar la punta más adecuada para cada trabajo. Cuanto menor es el tamaño de la punta, menor es también el calor que almacena. Las puntas pequeñas, para soldadu-

Figura 1.



ras pequeñas y las grandes, para soldar conmutadores o terminales. Algunos soldadores de 30 vatios disponen de un tipo de punta en forma de H, que sirve para desoldar de una sola vez las 14 ó 16 patillas de un circuito integrado DIL, en lugar de una a una, operación más trabajosa y arriesgada, por la cantidad de tiempo que se tiene caliente el chip.

Aspecto fundamental para la buena soldadura es también el estaño. En los comercios del ramo nos venderán un tipo de soldadura adecuada para fines electrónicos. Su composición no sólo es el estaño; lleva aleado plomo en pro-

Figura 2.

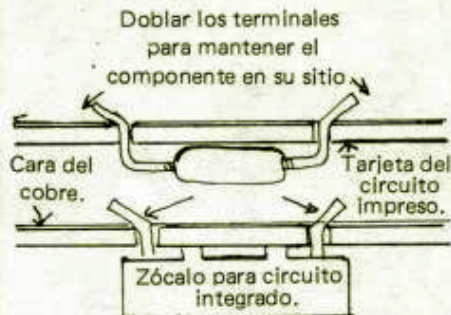
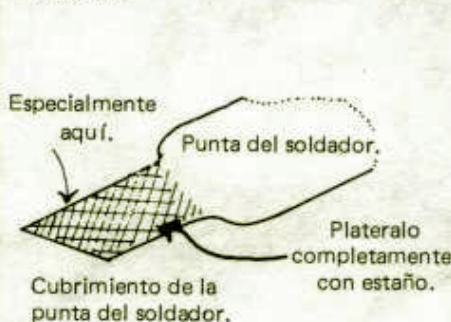


Figura 3.



porciones centesimales 60/40. Para facilitar que el estaño fundido corra con mayor facilidad, lleva un alma de resina, que desprende un olor característico cuando funde. También protege a los terminales del componente contra la oxidación durante el proceso de la soldadura. Cuando se enfría, la resina solidifica, tomando un aspecto pardusco, que aunque afea el acabado, no contiene sustancias químicas corrosivas. Para obtener un mejor acabado, se puede utilizar disolvente en cualquier droguería y, con ayuda de un pincel grueso, eliminarla de la placa.

El estaño se vende con distintos grosores; para los trabajos de microelectrónica es aconsejable uno de pequeño diámetro.

Después de concluir el trabajo, es importante limpiar la resina depositada en la punta del soldador antes de que se enfríe. Para ello se podría utilizar algún papel de tipo grueso o, mejor, un cepillo de cerdas metálicas, procurando utilizar la mínima energía al frotarlo.

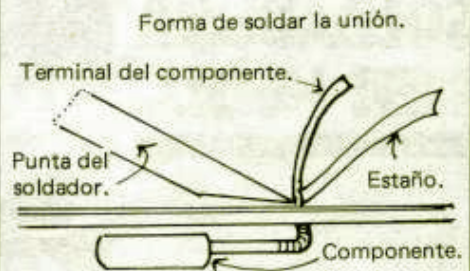
Algunas bases a la venta para ser utilizadas como soporte para el soldador llevan una pequeña esponja destinada a

esta tarea de limpieza. Personalmente no es mi gusto.

Los componentes electrónicos, normalmente soportan el calor generado durante un tiempo entre los 3 y los 10 segundos, sin sufrir daño. Si por cualquier causa fuera necesario más tiempo para asegurar la soldadura, es conveniente dejar que el calor se disipe durante un rato y volver a insistir posteriormente.

Muchos aficionados doblan los terminales que salen de los componentes con los dedos. Para obtener un acabado más atractivo es aconsejable tomar cada

Figura 4.



uno de los terminales con unas pinzas de punta plana, calcular la distancia del terminal al taladro del circuito impreso y, por ese punto, doblar el resto del terminal en 90°. Se doblará siempre por el extremo que queda libre, no por el lado del terminal.

Los componentes pueden, a veces, llevar una fina y transparente capa de barniz, grasa u óxido o cualquier otra sustancia. Por ello es conveniente limpiar los terminales; deberían limpiarse antes de la soldadura. Una goma de borrar para tinta se mostrará particularmente útil.

Haciéndole un corte perpendicular, como muestra la figura, se introducirá el terminal por él, tirando y empujando hasta asegurar que está limpio. Un terminal sucio puede producir un falso contacto. Muchas veces no es realmente necesaria esta limpieza, pero vale más prevenir.

Por su lado, conviene lavar con un jabón desengrasante la tarjeta de circuito impreso por el lado de las pistas.

Con un trozo de tarjeta Veroboard practicaremos la soldadura de distintos componentes, siguiendo las normas que veremos a continuación.

MONTAJES

Si trabaja en una mesa que pueda dañarse, no olvide protegerla con algunos periódicos o algo similar.

Conecte el soldador y deje que pase un tiempo prudencial para que la punta adquiera suficiente temperatura. Si intentamos utilizarlo apenas pueda derretir el estaño, las probabilidades de un falso contacto se multiplican. Unos cinco minutos suele ser suficiente.

En primer lugar, se doblarán los terminales del componente como describimos anteriormente. Una vez hecho, se inserta en los taladros correspondientes de la tarjeta de circuito

LA SOLDADURA BIEN HECHA



impreso, doblándolos ligeramente, lo suficiente para que no se caigan al voltear la tarjeta.

En el mercado existen algunos artefactos que facilitan el manejo y sujeción del circuito integrado durante la soldadura, pero aparte de tener un precio algo elevado, no son especialmente relevantes sino se trabaja masivamente. Antes de utilizar el soldador, es necesario asegurarse de que la punta está correctamente plateada con estaño, sobre todo la parte que estará en contacto con la unión componente-pista de cobre. En un soldador nuevo esto se consigue mediante la aplicación de pequeñas cantidades de hilo de estaño cuando el soldador tiene una temperatura elevada, según ilustra la figura 3.

La punta del soldador se colocará de tal forma que toque tanto en la pista como en la parte del terminal más próxima a ella. Se deja que transmita su calor a ambos durante aproximadamente un par de segundos y, a continuación, se acerca el hilo de estaño (figura 4). Al fundirse, irá fluyendo como si de un líquido se tratase, hasta "mojar" ambas partes a soldar.

En el punto de unión, se formará una especie de montañita, cuya base se sitúa

Figura 5.

Unión ya soldada.

La gota del estaño no debe tener forma de gota.



en la pista de cobre y va creciendo en torno al soldador (figura 5).

Podría ocurrir que la forma tomada por el estaño parezca una gota, con bordes redondeados. Esto se produce por dos causas. Bien la soldadura es correcta, pero hemos aplicado demasiada cantidad de estaño, y ese sobrante ha tomado dicha forma; o estamos ante una soldadura fría. Esto quiere decir que, si por ejemplo la pista no estaba lo suficientemente caliente antes de aplicar el estaño, cuando se ha puesto en contacto con ella ha absorbido el calor de la pista con intención de fundirse —el segundo principio de la termodinámica dice que el calor fluye de los cuerpos más calientes a los más fríos—, y se han acabado enfriando los dos, sin conseguir la perfecta unión.

El soldador seguirá aplicado durante algunos segundos más, hasta que veamos que el estaño quede en una posición más o menos estable. Una recomendación: no soplar nunca para que enfríe antes la unión, el riesgo de soldadura fría vuelve a aparecer.

Varias veces hemos mencionado la "soldadura fría", pero nos gustaría saber cómo se detecta. Bueno, pues en primer lugar mediante la observación visual

antes aludida. A continuación, tomando la parte del terminal que sobra, se tirará en sentido vertical hacia arriba. En muchos casos, si la soldadura es fría, termina cediendo la unión. Pero, ¡mucho cuidado! No tire demasiado fuerte, se podría llevar parte de la pista con la soldadura correspondiente.

Una vez enfriada la unión, se podrán cortar los rabos sobrantes con un alicate de corte. No es aconsejable apurar demasiado.

Haga lo mismo con todos los componentes. Es recomendable seguir un orden durante la soldadura, dejando los

Figura 6.

Corte de los terminales.

No apurar mucho la distancia.



componentes semiconductores, más propensos a sucumbir bajo la influencia del calor, para el último lugar.

Cuando hayamos soldado todos, la inspección ocular tomará el relevo. Se comprobará si cada componente ha sido colocado en su lugar correspondiente, si la disposición de las patillas de los componentes activos (transistores, diodos y circuitos integrados) es la que debiera.

Por último se puede limpiar toda la tarjeta, por el lado del cobre, con el disolvente y el pincel, como decíamos antes. Mucho cuidado, esta sustancia podría disolver determinadas partes de algunos componentes.

Una última revisión ocular siempre es recomendable. Esta vez conviene comprobar que las pistas no se tocan por efecto del estaño, etc.

Siguiendo estas observaciones, es prácticamente seguro que el montaje funcionará a la primera, siempre que los componentes no sean defectuosos.

Alejandro Diges

Sinclair ZX Spectrum



- 16K: 34.950 ptas.
- 48K: 43.950 ptas.

■ EL PRECIO INCLUYE: ALIMENTADOR, CABLES PARA CASSETTE NORMAL Y TV (COLOR O B/N), CASSETTE DE DEMOSTRACIÓN, MANUALES EN INGLÉS, MANUAL EN CASTELLANO Y CASSETTE DE PROGRAMAS.

- MICROPROCESADOR Z80A ■ 8 COLORES ■ 2 INTENSIDADES ■ SONIDO POR ALTA VOZ INTERNO ■ 40 TECLAS MÓVILES CON AUTO-REPETICIÓN Y SONIDO ■ MAYÚSCULAS, MINÚSCULAS, CARACTERES GRÁFICOS, INVERSO Y DEFINIBLES ■ CÓDIGO ASCII ■ PANTALLA DE 24x32 CARACTERES ■ GRÁFICOS DE ALTA RESOLUCIÓN (256x192 PUNTOS) ■ BASIC SINCLAIR AMPLIADO EN 16K ROM ■ ALMACENAMIENTO DE DATOS Y PROGRAMAS EN CASSETTE (1.500 BAUDIOS) ■ CONECTOR DE EXPANSIONES.

- KIT AMPLIACIÓN A 48K RAM: 7.950 ptas.
- ADAPTADOR MEMORIAS 16K ZX81: 2.250 ptas.
- AMPLIFICADOR DE SONIDO+RESET: 4.990 ptas.
- IMPRESORA ZX: 13.950 ptas.
- CAJA 5 ROLLOS PAPEL: 2.625 ptas.
- INTERFACE CENTRONICS+CABLE: 11.450 ptas.
- INTERFACE PALANCA JUEGOS: 3.550 ptas.
- PALANCA JUEGOS "QUICK-SHOT": 2.900 ptas.
- CONECTOR HEMBRA: 990 ptas.

PRONTO DISPONIBLE: ■ ZX INTERFACE 1 ■ ZX INTERFACE 2 ■ ZX MICRO-DRIVE ■ MODEM ■ TRAZADOR DIGITAL ■ TECLADO PROFESIONAL ■ INTERFACE PROGRAMABLE PALANCA JUEGOS ■ PALANCA JUEGOS ALTA SENSIBILIDAD ■ FLOPPY DISK

Superprogramas ZX-Spectrum

EN ESPAÑOL

VIDEO-JUEGOS 16K/48K

- SUPER-COMECCOS: 1.190,-
- CYBOTRON: 1.190,-
- GULPMAN: 1.390,-
- PHENIX: 1.390,-
- WRECKAGE (DESTRUCCIÓN): 1.390,-
- LA RANA + Z-MAN: 1.690,-
- ANDROIDE UNO: 1.390,-
- BEDLAM: 1.390,-
- GALAXIANS + SPYNADS: 1.490,-
- CIENPIES + STORM-FIGHTERS: 1.490,-
- ARMAGGEDON + AMENAZA: 1.690,-

VIDEO-JUEGOS 48K

- EL DETECTIVE: 1.390,-
- TÚNELES MARCIANOS: 1.390,-
- ESCALADOR LOCO: 1.390,-
- GRAND PRIZ: 1.490,-
- STOMPING STAN: 1.390,-
- KILLER KONG: 1.390,-
- BARMY BURGER'S: 1.390,-
- MINA MALDITA: 1.390,-
- AUTOSTOPISTA GALÁCTICO: 1.390,-
- ÚLTIMO AMANECER EN LATTICA: 1.390,-

AVENTURAS (48K)

- THE QUILL (DISEÑO DE AVENTURAS) + EJEMPLO: 2.490,-

JUEGOS INTELIGENTES 48K

- AJEDREZ 2002: 2.490,-
- DALLAS: 1.390,-

JUEGOS INTELIGENTES 16K

- GUERRA DE BARCOS: 1.390,-

EDUCATIVOS 16K

- TUTOR MORSE: 1.390,-

GESTIÓN 48K

- MASTERFILE (BASE DE DATOS): 2.990,-
- CONTABILIDAD PERSONAL: 2.500,-
- SIMPLEX (PROGRAMACIÓN LINEAL): 2.500,-
- CONTEXT (PROCESADOR TEXTOS 64 COLUMNAS): 2.990,-

UTILIDADES 16K/48K

- ENSAMBL/DESENSAMBL: 2.490,-
- COMPIADOR: 1.990,-
- 3D VISIÓN: 1.990,-
- VIDEO-DISPLAY: 1.990,-
- FORTH: 2.990,-
- SUPERGRÁFICOS + RUIDO: 1.990,-
- 64 PRINT: 1.490,-
- ADAPT. PROGRAMAS BASIC ZX81: 1.490,-

UTILIDADES 48K

- MASTER-DISEÑO: 1.990,-

AMPLIA SELECCIÓN DE LOS MEJORES PROGRAMAS DE IMPORTACIÓN EN INGLÉS: SINCLAIR, ULTIMATE, IMAGINE, BUG-BYTE, ETC. DISPONIBLES

ORIC-1

48.950 ptas.

Ventamatic micro-informática

Avda. de Rhode, 253 - Apartado 168
ROSAS (GERONA) - Tel. (972) 255616

ESPECIALISTAS EN VENTA POR CORREO - ENVÍOS INMEDIATOS A TODA ESPAÑA - TODO EN STOCK - 6 MESES GARANTÍA

Sinclair ZX81



NUEVOS MICRO-PRECIOS AHORA SÓLO: 13.450 ptas.

- EL PRECIO INCLUYE: ALIMENTADOR, CABLES PARA CASSETTE NORMAL Y TV, MANUAL EN INGLÉS, MANUAL AMPLIADO EN CASTELLANO Y CASSETTE DEMOSTRACIÓN ■ IDEAL PARA INICIACIÓN A LA MICRO-INFORMÁTICA Y PROGRAMACIÓN, JUEGOS, GESTIÓN DOMÉSTICA Y PERSONAL, EDUCACIÓN, ETC. ■ 1K RAM ■ BASIC EN 8K ROM ■ MICROPROCESADOR Z80 A ■ ALMACENAMIENTO DE DATOS Y PROGRAMAS EN CASSETTE (250 BAUDIOS) ■ GRÁFICOS DE 44x64 PUNTOS ■ PANTALLA DE 24x32 CARACTERES ■ CONECTOR DE EXPANSIONES ■ 40 TECLAS SENSITIVAS.

SUPER OFERTA ESPECIAL: ZX81 + 16K RAM PACK SÓLO 17.950 ptas.

- CONECTOR MACHO: 300 ptas.
- INVERSOR DE VIDEO: 1.790 ptas.

MEMOTECH + ZX81 = LA ESTÉTICA DEL CONJUNTO

NO MÁS BORRADOS ACCIDENTALES DE MEMORIA



- MEMOPAK 16K (AMPLIABLE): 7.950 ptas.
- MEMOPAK 32K (AMPLIABLE): 14.950 ptas.
- MEMOPAK 64K (56K ÚTILES): 17.950 ptas.

- MEMOPAK INTERFACE RS232: 12.950 ptas.
- MEMOPAK INTERFACE CENTRONICS+CABLE PARA IMPRESORA NORMAL 80 COLUMNAS (MAYÚSCULAS Y MINÚSCULAS): 13.950 ptas.
- MEMOPAK ALTA RESOLUCIÓN GRÁFICA (192x256 PUNTOS) CON GRAN NÚMERO DE INSTRUCCIONES GRÁFICAS INCORPORADAS: 11.950 ptas.
- TECLADO PROFESIONAL MEMOTECH CON BUFFER: 14.950 ptas.



IMPORTADOR EXCLUSIVO PARA ESPAÑA

- MEMOPAK EPROM: ENSAMBLADOR Z80: 8.950 ptas.
- MEMOPAK EPROM: MEMOCALC (HOJA DE CÁLCULO): 8.950 ptas.
- MEMOPAK EPROM: MEMOTEXT (PROCESADO TEXTOS): 8.950 ptas.

Superprogramas ZX81

VIDEO JUEGOS

- SUPER-COMECCOS: 1.190,-
- SUPER GULP: 990,-
- FROGGER: 1.190,-
- ALUNIZAJE: 1.190,-
- BATALLA ESPACIAL 3D: 1.190,-
- ASTEROIDES: 990,-
- DANGER TRACK: 990,-
- SCRAMBLE: 990,-
- CRASHBOOT + COMECOCOS: 990,-
- SUPER DEFENDER: 990,-
- SUPER JUEGOS (9 DE 1K): 990,-
- CASSETTE UNO (11 DE 1K): 990,-
- CASSETTE 2 (9 DE 1K): 1.590,-
- EL ACORRALADO: 990,-

MÚSICA

- ORQUESTA: 990,-

JUEGOS INTELIGENTES

- ZX AJEDREZ II: 1.990,-
- GUERRA DE BARCOS: 990,-
- MISIÓN GALÁCTICA: 990,-

EDUCATIVOS

- GEOGRAFÍA ESPAÑA: 1.390,-

UTILIDADES

- SUPERGRÁFICOS: 1.490,-
- VIDEOGRÁFIC: 1.890,-
- ESCAPARATES: 990,-
- COMPIADOR: 1.890,-
- ENSAMB/DESENSAMB: 1.890,-
- RAPID SAVER: 1.490,-
- ALTA RESOLUCIÓN: 1.490,-

GESTIÓN

- BASE DE DATOS: 2.790,-
- S. CONTROL STOCKS: 2.790,-
- VISI-PLAN: 1.890,-

ACCESORIOS

- CAJA 15 CINTAS VÍRGENES C-15: 1.350 ptas.
- CAJA 15 CINTAS VÍRGENES C-30: 1.800 ptas.
- MONITOR FÓSFORO VERDE 12": 24.950 ptas.
- MONITOR FÓSFORO VERDE 9": 20.450 ptas.
- MONITOR COLOR RGB 14": 69.950 ptas.

SEIKOSHA

IMPRESORAS GRÁFICAS

SIMPLEMENTE LA MEJOR RELACIÓN CALIDAD/PRECIO



49.900 ptas.

- INTERFACE CENTRONICS DE ORIGEN ■ IMPRESIÓN AGUJAS UNIHAMMER
- GP100 ■ 80 COLUMNAS ■ 30 CARACT/SEG ■ MAYÚSCULAS, MINÚSCULAS (CÓDIGO ASCII) ■ CARACTERES EXPANDIDOS ■ PAPEL 10"
- GP250 ■ 50 CARACT./SEG. ■ INTERFACE RS232 INCORPORADO ■ CARACTERES DOBLE ALTO/DOBLE ANCHO ■ RESTO COMO GP100: 59.900 ptas.
- GP700 ■ TODOS LOS COLORES ■ ARRASTRE FRICCIÓN/TRACCIÓN ■ PAPEL HASTA 10": 98.500 ptas.

LIBROS

- 20 SIMPLE ELECTRONIC PROJECTS FOR THE ZX81: 1.590,-
- THE ZX81 POCKET BOOK: 1.660,-
- 49 EXPLOSIVE GAMES FOR THE ZX81: 1.490,-
- MASTERING MACHINE CODE ON YOUR ZX81: 1.890,-
- GAMES ZX COMPUTERS PLAY (30 GAMES FOR ZX81 & SPECTRUM): 990,-
- 60 GAMES & APPLICATIONS FOR SPECTRUM: 1.490,-
- CREATING ARCADE GAMES ON SPECTRUM: 1.190,-
- SPECTRUM GRAPHICS: 1.890,-
- SPECTRUM PROGRAMMES: 1.890,-
- GUÍA PRINCIPIANTE NEWBRAIN (C/CASSETTE): 1.000,-
- NEWBRAIN TECHNICAL MANUAL: 6.000,-
- BOLETINES CLUB NACIONAL USUARIOS ZX 1982 (1-4): 1.200,-
- BOLETINES CLUB NACIONAL USUARIOS ZX 1983 (5-10): 2.500,-
- MANUAL AMPLIADO ZX81
- MANUAL AMPLIADO ZX-SPECTRUM
- MANUAL CÓDIGO MÁQUINA ZX81
- MANUAL CÓDIGO MÁQUINA ZX-SPECTRUM
- LIBROS PROGRAMAS ZX

NewBrain

74.950 ptas.

- MÓDULO BATERÍAS: 18.950 ptas.
- PROGRAMAS: 1.000 ptas. c/u

- BASE DE DATOS ■ CONTABILIDAD PERSONAL ■ ENTRETENIMIENTOS I ■ ENTRETENIMIENTOS II ■ UTILIDADES I ■ UTILIDADES II ■ VOLPLOT ■ FUENTES

SORD M-5 54.950 ptas.

ENVÍENME:

FECHA

ENVÍO GIRO/TALÓN CONFORMADO PTAS. PARA ENVÍOS C/REEMBOLSO MANDAR 20% A CTA.

NOMBRE

APELLIDOS

DOMICILIO

POBLACIÓN D.P.

PROVINCIA

■ TARJETA VISA/MASTERCARD N.º

CADUCA FIRMA

■ GIRO POSTAL N.º FECHA

GASTOS ENVÍO: 400 PTAS. REEMBOLSO O TARJETA DE CRÉDITO 200 PTAS. CUALQUIER OTRA FORMA

APRENDIENDO EL

PRIMERA PARTE: ORGANIZACION DE LA MEMORIA

Comenzamos en este número la publicación de una serie de artículos destinados a explicar a los lectores el tema, generalmente árido, del código de máquina del Spectrum. Se trata de un texto escrito por Juan Martínez Velarde que procura dar una primera visión, susceptible de incitar al aprendizaje de esta materia que, una vez dominada, lleva al usuario a sacar todo el partido posible al ordenador.

1

Los sistemas
binario
y hexadecimal

A. En nuestro mundo, los seres humanos contamos los números y cantidades en base 10. Para ello utilizamos diez cifras, que van del 0 al 9. Con estos 10 dígitos podremos formar cualquier número.

B. Otro sistema, el de base 2, utiliza solamente dos cifras, el 0 y el 1. También se podrá definir cualquier número en este sistema 10010101.

Para pasarlo a base decimal y poder comprenderlo, debemos multiplicar el dígito situado más a la derecha por 2 elevado a 0 (=1), el siguiente dígito por la izquierda por 2 elevado a 1 (=2), el tercero por 2 elevado a 2 (=4), y así sucesivamente. La suma de todos estos resultados nos dará el equivalente de ese número binario en base 10 (149).

Con 8 dígitos (8 ceros y unos), el número más elevado que podremos definir será el 11111111, cuyo equivalente decimal será el 255.

Con 8 dígitos podemos definir, en base 2, hasta 256 números diferentes, que van desde el 0 hasta el 255, en base 10.

C. Un cuadrado con 16 casillas en cada lado contendría 256 casillas en su interior.

Eligiendo otro sistema de numeración, cuya base sea 16, podremos definir, solamente con dos dígitos, estos 256 números decimales.

Si contamos en base 16, necesitaremos de dieciséis cifras o dígitos. Las primeras diez cifras corresponden con las de base 10 (0 a 9). Las restantes 6 son las letras que van de la A a la F; así tendremos: 0 1 2 3 4 5 6 7 8 9 A B C D E F.

Con este sistema cualquier número expresable con 3 cifras (entre 0 y 255) en base 10, o expresable con ocho cifras en base 2, puede ser compuesto con solamente dos dígitos, p. e. 59. El primer dígito lo multiplicamos por 16 elev. a 0 (=1), el segundo por 16 elev. a 1 (=16), así:

$$9 \cdot 16^0 + 5 \cdot 16^1 = 89$$

Los dígitos A, B, C, D, E y F corresponden respectivamente a los números decimales 10, 11, 12, 13, 14, 15.

Así el número en base 16, C9 será en decimal;

$$9 \cdot 16^0 + 12 \cdot 16^1 = 201$$

(Ver apéndice E del manual)

La memoria de un ordenador está organizada por direcciones. En los ordenadores Sinclair cada una de las direcciones de memoria contiene ocho instrucciones elementales. Una instrucción elemental es la que responde a una determinada pregunta con un "sí" o con un "no". En una dirección de memoria, el "sí" está representado por un 1 y el "no" está representado por un 0. Cada uno de estos unos y ceros ("sies" o "noes") recibe el nombre de BIT. Una dirección de memoria contiene 8 de estos BITS, 8 "sies" o "noes". 8 bits se agrupan en una unidad mayor, un byte. Cada dirección de memoria contiene un byte, en forma de unos y ceros.

2

La dirección de memoria.
BIT y BYTE

CODIGO MAQUINA

10010101

El contenido de un byte se puede expresar en el sistema decimal, hexadecimal, o como hemos visto, binario.

Cada byte, en cada dirección de memoria significará una información determinada por los bits. El ordenador hará uso de esa información para realizar, tanto programas, como para almacenar información.

Podemos introducir información en una dirección de memoria, introduciendo un número en esa dirección de memoria. Cambiando ese número, esa información al introducir una nueva información en la misma dirección de memoria.

Dentro del *hardware* de un ordenador la parte más importante es la CPU (*CENTRAL PROCESSING UNIT* O UNIDAD DE PROCESO CENTRAL) o microprocesador. La función del microprocesador es procesar, trabajar con una serie de datos e información almacenadas en las direcciones de memoria, para que el ordenador pueda empezar a funcionar. En el Spectrum el microprocesador es el Z80A, una variación más rápida de otro microprocesador, el Z80 construido en California por Zilog.

La velocidad del Z80A es increíble: es capaz de procesar 875.000 instrucciones simples en un segundo. El máximo de direcciones de memoria que este microprocesador puede generar y controlar es de 65536 (64×1024).

El Spectrum 16 K contiene 32.767, el Spectrum 48 K 65.536 direcciones de memoria.

La CPU Z80A traslada datos y direcciones de memoria mediante unos "buses" o canales de información.

El microprocesador controla, dirige y ordena sobre la memoria y sus direcciones.

El "cerebro" de la computadora necesita de una serie de datos, de información para que pueda realizar su función, y para que se pueda comunicar con el usuario del ordenador. De esta manera toma parte de la memoria para sí y en esas direcciones guarda su información vital. A esta parte de la memoria la llamamos ROM (*READ ONLY MEMORY* O MEMORIA DE SOLO LECTURA). La parte restante de la memoria de la memoria es llamada RAM (*RANDOM ACCESS MEMORY* O MEMORIA DE ACCESO ALEATORIO). Las direcciones de esta parte de la memoria sirven para almacenar información de los programas que se realicen.

Introducimos un dato en una dirección de memoria con el comando POKE dirección, dato.

Una dirección de memoria p. e. 22540, es ocupada por un byte, p. e. 149 d, 10010101 bin.

A partir de este momento aquella dirección contendrá ese byte. Lo comprobaremos dando el comando PRINT PEEK dirección y veremos que continúa teniendo ese valor. Este byte no será borrado de la memoria hasta que no introduzcamos otro valor de byte en la misma dirección de memoria, el nuevo dato pasaría a ocupar esa dirección y el anterior dejaría de existir.

El CPU protege la zona de la memoria que necesita, la ROM (memoria de solo lectura), no podemos cambiar ningún byte de las direcciones de memoria que pertenecen a la ROM. Por ello es llamada MEMORIA DE SOLO LECTURA, pues solamente podremos "leer" o preguntar por los bytes de la ROM, pero nunca modificarlos. La otra parte de la memoria, la RAM queda totalmente libre para el usuario. La RAM (memoria de acceso aleatorio), también llamada memoria de lectura/escritura, nos permite tanto introducir datos en sus posiciones de memoria como recuperarlos o "leerlos" más adelante.

3

Las memorias ROM, RAM
y la CPU Z80A.
PEEK y POKE

4

Estructura de la ROM

La ROM es esa zona de la memoria en cuyas direcciones está almacenada cierta información que el microprocesador necesita para comenzar a funcionar.

En el primer ordenador fabricado por la casa Sinclair, el ZX80, la ROM sólo ocupaba 4096 posiciones de memoria (4K), ocupadas por 4986 bytes. En la segunda computadora de la misma casa, el ZX81, el número de direcciones de memoria de la ROM era de 8192, ocupadas por el mismo número de bytes. Las dos versiones del Spectrum, el último lanzamiento de Sinclair, amplían la ROM a 16384 posiciones de memoria (16K). Los chips que almacenan la información de la ROM están "sellados". No se puede alterar esta información.

Las direcciones de memoria que ocupa la ROM van de la 0 a la 16383 (hex. 0000-3FFF).

Los 16K de la memoria ROM se dividen de la siguiente manera:

- 7K sistema operativo
- 8K traductor BASIC
- 1K juego de caracteres

En los 7K del sistema operativo están almacenadas todas aquellas rutinas o grupos de órdenes que el microprocesador utiliza para inicializar registros y datos, entre otros.

En estos 7K podemos encontrar las rutinas para el altavoz, rutinas de grabación, verificación, carga y fusión de programas o datos, rutinas de INPUT/OUTPUT (entrada/salida) de información, entre otras.

Todas estas rutinas están escritas en la propia lengua del microprocesador; cada código binario compuesto de 8 bits, forma un byte con un significado para él.

BIN	HEX	MNEMONICO	DEC
10010101	95	SUB L	149

La instrucción correspondiente a cada número se conoce como MNEMONICO. El conjunto de instrucciones mnemónicas, forman el lenguaje ASSEMBLER o lenguaje de CODIGO MAQUINA.

El BASIC es un lenguaje que permite programar sin excesiva dificultad. Las órdenes del lenguaje BASIC no pueden ser comprendidas por el microprocesador, y por ello necesita de un TRADUCTOR. Con la ayuda del traductor, el microprocesador podrá reconocer, traducir y ejecutar las órdenes escritas en lenguaje BASIC. Para poder interpretar los comandos BASIC, el microprocesador necesita de un cierto tiempo. Es por esto, por lo que el BASIC, comparado con el ASSEMBLER, es muchísimo más lento, ocupando además más memoria.

10 LET A = 5 : 21 bytes — BASIC
LD A,5 : 2 bytes — ASSEMBLER

El intérprete o traductor BASIC ocupa en memoria ROM aproximadamente 8K. Aparte de estar almacenados los comandos BASIC, también están las tablas para cálculos de senos, cosenos, tangentes.

Un poco menos del K restante, sirve para almacenar el juego de caracteres. La forma de cada letra, cada número que se representa en pantalla o impresora está definido en el ordenador.

Programa 1:

```
10 REM Programa de demostracio
n area BASIC
100 LET f=PEEK 23635+256*PEEK 2
3636
110 PRINT f;TAB 10;PEEK f;TAB 2
0;CHR$ PEEK f AND PEEK f>32
120 LET f=f+1: GO TO 110
```

```
23755      0
23756      10
23757      37      X
23758      20      REM
23759      34      P
23760      80
```

```
23761      114      r
23762      111      o
23763      103      g
23764      114      r
23765      97      a
23766      109      m
23767      97      a
23768      32      a
23769      100      d
23770      101      e
23771      32      e
23772      100      e
23773      101      e
23774      109      m
23775      111      o
23776      115      s
```


5

Estructura de la RAM

La forma de representación de un número o una letra, ya sea en la pantalla o en la impresora, recibe el nombre de carácter. El Spectrum consta de 96 caracteres, que van (ver Apéndice A del manual) desde el carácter de "espacio" hasta el carácter de *copyright* (códigos 32 a 127). Cada carácter está constituido por 8 bytes. Cada byte consta de 8 bits, ocho "ceros" o "unos". Un carácter está formado por tanto por 64 bits. Si el bit es cero, representará un punto en blanco, si el bit es uno, tomará la forma de punto en negro, o a la inversa.

El juego de caracteres ocupa $96 \times 8 \text{ bytes/carácter} = 768 \text{ bytes}$. Las direcciones de memoria ocupadas por el juego de caracteres van de la 15616 a la 16383, donde acaba la ROM.

La RAM es la segunda división de la memoria del Spectrum. Existen partes de la RAM donde ya hay información, otras donde no la hay. Los *chips* que almacenan la información de la RAM no están "sellados" y por eso podremos alterar o conservar esa información.

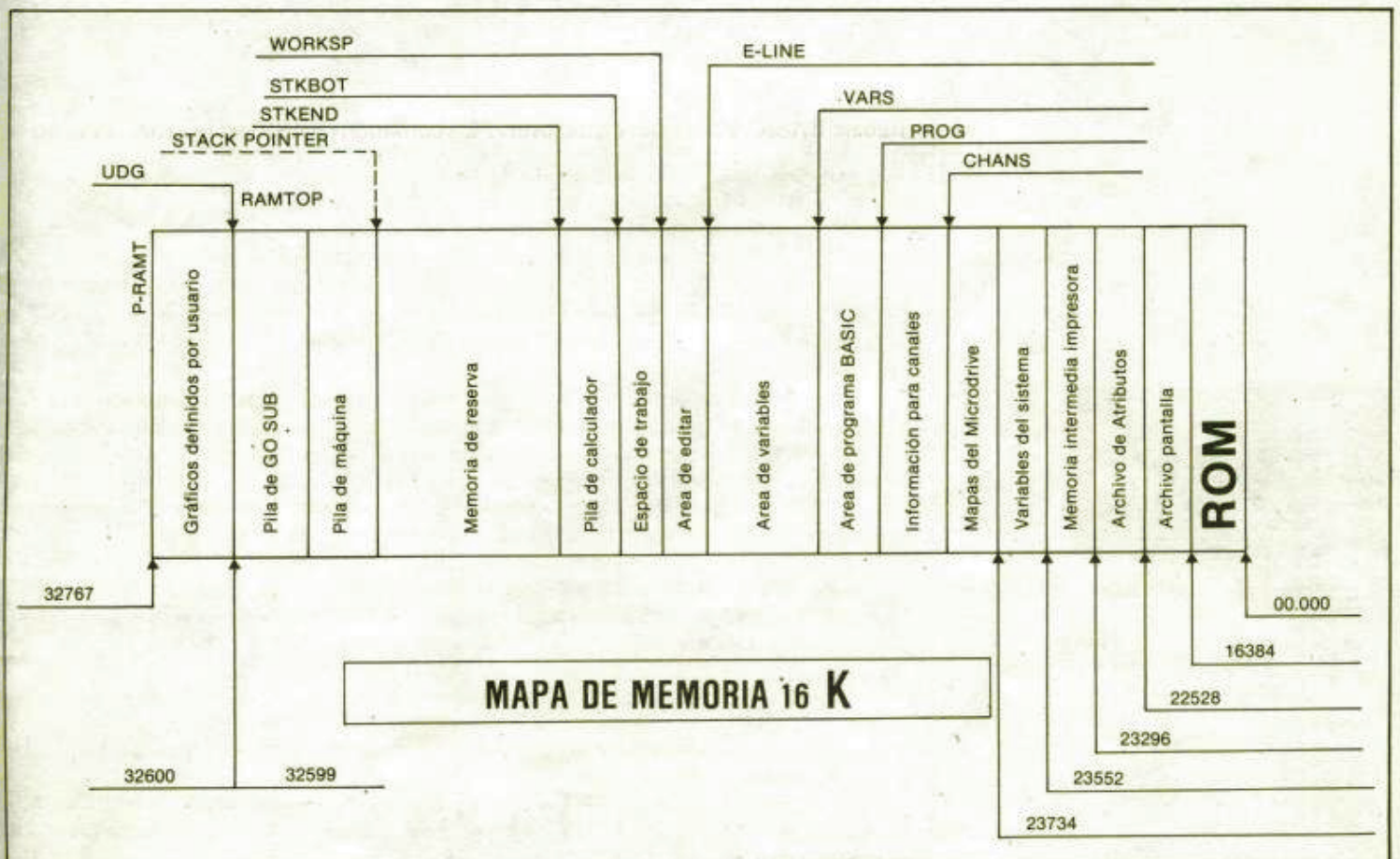
En la RAM quedarán almacenados los programas, las variables. Con la pérdida de tensión, la información contenida en la RAM se pierde, quedando "vacía". Por ello grabamos en cinta magnética el contenido de la RAM. Es aquí donde se diferencian las dos versiones del Spectrum. En la versión *standard* la de 16K, la RAM contiene 16384 direcciones de memoria (16K). En el Spectrum extendido, el de 48K, la RAM contiene 49152 direcciones de memoria (48K).

16K:16384-32767)
48K:16384-65535)

Extensión de la memoria RAM
en el Spectrum 16K y 48K.

El ZX81 tenía como base una memoria RAM de sólo 1K, ampliable a 16,32 ó 64K (entre RAM y ROM).

La RAM contiene las áreas siguientes:



1. archivo de pantalla
2. archivo de atributos
3. memoria intermedia de la impresora
4. variables del sistema
5. mapas de *microdrive*
6. información para canales
7. área de programa BASIC
8. área de variables
9. área de editar
10. espacio de trabajo
11. pila del calculador
12. memoria de reserva
13. pila de máquina
14. pila de GO SUB
15. gráficos definidos por el usuario.

El archivo de pantalla, también llamado *DISPLAY AREA*, es la primera subdivisión de la RAM.

La pantalla se divide en 24 filas $\times$ 32 columnas, lo que permite la posibilidad de imprimir $24 \times 32 = 768$ caracteres en la pantalla.

Cada carácter consta de 8 bytes, así el resultado total en bytes será de $768 \times 8 = 6.144$ bytes ó 6K. Cada byte consta de 8 bits, también llamados puntos o *pixels*. El resultado global de *pixels* en pantalla será de $6.144 \times 8 = 49.152$ *pixels*. Esto nos ofrece la así llamada "*high*" o "alta" resolución.

Las direcciones ocupadas por el archivo de pantalla van de la 16384 a la 22527 (hex. 4000-57FF), es decir 6.144 bytes ó 6K.

Imaginemos la pantalla del televisor dividida en 3 partes. El primer tercio (líneas 0 a 7) corresponderá a 8 líneas $\times$ 32 caracteres/línea $\times$ 8 bytes/carácter = 2.048 bytes ó 2K. Los otros dos tercios corresponden a las líneas 8 a 15 y a las líneas 16 a 23, ocupando ambos 2.048 bytes ó 2K de memoria.

1. tercio : 16384 a 18431 (hex. 4000-47 FF)
2. tercio : 18432 a 20479 (hex. 48000-4FFF)
3. tercio : 20480 a 22527 (hex. 5000-57FF)

Consideremos que cada uno de estos tres tercios de 2 K cada uno, divididos en ocho partes, cada una de 1/4 K ó 256 bytes de memoria.

Los 256 bytes corresponderán al primer byte de cada uno de los caracteres del primer tercio. El primer tercio contiene 256 caracteres.

El segundo grupo de 256 bytes contiene la información, contienen los bits, para el 2.º bytes de cada uno de esos 256 caracteres del primer tercio.

Una idea más clara de la distribución de la información en el archivo de pantalla, nos lo dará al cargar una imagen del televisor con la instrucción *LOAD* "SCREEN\$ (ver lo que ocurre con la *cassette* HORIZONTES y esta instrucción).

Tras la última dirección de memoria del archivo de pantalla, comienza en la RAM el archivo de atributos o archivo de colores. Aquí se irán almacenando todos los colores que van apareciendo en la pantalla. Como "colores" entendemos el aspecto externo de un carácter, es decir su *PAPER* o fondo, su *INK* o tinta, su *BRIGHT* y su *FLASH*. Dependiendo de los valores que cada una de estas funciones tengan, aparecerá el dibujo coloreado de una u otra forma.

Al contrario que los caracteres (Gráficos y dibujos), el archivo de atributos carece de alta resolución. Nos ofrece poder definir el *INK*, *PAPER*, *BRIGHT* y *FLASH* = atributos para 32 caracteres por línea y 24 líneas, en total 768 caracteres por pantalla. Los atributos de un conjunto de 64 *pixels* (1 carácter) podrán ser definidos.

Las direcciones de memoria ocupadas por el archivo de atributos van de la 22528 a la 23295.

Si queremos introducirnos en el archivo para variar los atributos, debemos permanecer en este intervalo.

Escogida una dirección de memoria perteneciente al archivo de atributos y un byte, un valor en el rango 0-255 (v. prog. 6), podemos comprobar sus efectos en la pantalla.



El archivo de pantalla



El archivo de atributos

★ ★ **GANE** ★ ★ **5.000 PESETAS**

**MENSUALMENTE
PARTICIPANDO EN NUESTRO CONCURSO**

A partir del próximo número, ZX premiará mensualmente los programas que hagan llegar los lectores.

Para participar en este concurso abierto, todo aficionado a los ordenadores ZX81 y ZX Spectrum, deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.

Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:

- Originalidad de la aplicación.
- Simplicidad del método de programación.

La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.



Y TAMBIEN...

UN ZX MICRODRIVE *
**será sorteado cada mes entre todos
los programas que recibamos,
con independencia de que sean
publicados o no.**

* El ZX MICRODRIVE estará disponible en España, en principio, a partir de marzo de este año



Observamos parpadear un cuadrado en la pantalla, cambiando alternativamente los colores rojo y azul celeste.

Para comprender este atributo representado, debemos analizar el programa. Este realiza un POKE, es decir una alteración de una dirección de memoria (la 22540), introduciendo un nuevo valor expresado en decimal tras la coma (149).

Cada dirección de memoria almacena los atributos para un carácter. La 1.ª, la 22528, se ocupa del carácter de la línea n.º 0, columna n.º 0. La siguiente dirección, la 22529, lo hará para el carácter de la línea n.º 0, columna n.º 1 y así sucesivamente hasta el carácter de la línea n.º 23, columna n.º 31, en la dirección de memoria n.º 23295.

Direcciones: 22529 — 23295 (hex. 5800 — 5AFF)

Longitud: 24 líneas × 32 columnas = 768 caracteres = 768 bytes

El byte, el valor introducido tras la coma, representa un atributo concreto.

Cada n.º en el rango 0-255, cada byte, representa un atributo diferente:

BYTE / BIT N.º 7 6 5-3 2-0

149 decimal 1 0 010 101

Transformemos el byte 149 en base 10 a una base que la máquina lo entienda mejor, la base 2.

Cada byte se compone de 8 bits y uno o varios bits definen una función del atributo.

Comenzando a contar por la derecha, nos encontraremos al primer conjunto de bits, los bits n.º 0, n.º 1, n.º 2.

Estos tres bits definen el estado de tinta o INK, dependiendo del valor que sean, desde el 000 = 0 decimal = negro, hasta el 111 = 7 decimal = blanco.

Podemos transformar rápidamente un valor binario a uno decimal con la función BIN, y leer después el color correspondiente en la fila superior de teclas.

Los tres bits siguientes, los bits n.º 3, n.º 4, n.º 5, definen el estado de PAPER o fondo de un carácter.

Estos 3 bits pueden tomar el valor de 0 a 7 (en dec.); el funcionamiento es análogo al grupo de bits 0 a 2, que definen el estado de INK.

El bit n.º 6 define el estado de brillo o BRIGHT.

Si el bit tiene el valor 1 ("on") el carácter cuyos atributos se almacenan en esa dirección de memoria tornará a brillante. En caso que el bit tenga el valor 0 ("off") este carácter carecería de brillo.

El bit n.º 7, el último, define el estado de parpadeo o FLASH. Si el bit tiene el valor 1 ("on"), el carácter cuyos atributos se almacenan en aquella dirección de memoria parpadeará cambiando alternativamente el valor del fondo o PAPER por el de la tinta o INK.

Los caracteres permanecerán estables si el bit n.º 7 tiene el valor 0.

Al conectar el aparato, los atributos, los bytes de las direcciones de memoria del archivo de atributos, tienen siempre valor 56 (binario 00111000 = PAPER 7, INK 0).

Si damos la orden directa FLASH 1: CLS toda la pantalla parpadeará: el 7.º bit de todas las direcciones de memoria del archivo de atributos tornará a 1.

Podemos observar tanto en el archivo de pantalla como en el de atributos, que dando un comando directo, un POKE, cuya dirección de memoria corresponda

Programa 2:

Direcciones:

23937 código carácter "a" (97)
23938-23942 contenido variable "a"
23943 código carácter "b" (98)
23944-23948 contenido variable b
etc.

```
10 LET a=1: LET B=3: LET c$="U
ARS"
20 LET d$="vars": LET d=4096
100 LET f=PEEK 23627+256*B+PEEK 2
3628
110 PRINT f;TAB 10;PEEK f;TAB 2
0;CHR$ PEEK f AND PEEK f>32
120 LET f=f+1: GO TO 110
```

23938	97	a
23939	00	
23940	0	
23941	1	
23942	0	
23943	0	

23944	98	b
23945	00	
23946	00	
23947	3	
23948	00	
23949	00	
23950	67	c
23951	4	
23952	00	
23953	006	
23954	006	
23955	006	
23956	003	
23957	40	
23958	4	
23959	0	

YA ESTA
A LA VENTA
EL NUMERO **50**

circuito IMPRESO

Sensórgano

**Calentador
regulable
para revelador
papel**

Sinclair
ZX Spectrum



**ZX-81
y SPECTRUM**
Un Display
para el ZX-81
Programas,
Programas,
Programas...

**Generador
de efecto vocal**

**PIDALA
EN SU KIOSCO**

MEMORIA ROM

16 K

CARACTERES 1K	TRADUCTOR	SISTEMA OPERATIVO
	8K	7K

ROM

22540

DIRECCION DE
MEMORIA

BIT

1 0 0 1 0 1 0 1

149 D

BYTE

SOFTWARE

a un byte o un caracter de las 2 últimas líneas, no observaremos impresión alguna en la pantalla. Esto se debe a que el Spectrum imprime, una vez acabada una orden, el informe correspondiente, en la mayoría de los casos el "O OK", utilizando y borrando todo aquello que estaba en dos últimas líneas.

Estos dos programas afectan de igual manera al archivo de atributos, realizando la misma función:

```
10 PRINT AT 0,12, "J"
20 POKE 22540,149
```

```
10 PRINT AT 0,12; FLASH 1; BRIGHT 0; PAPER 2; INK 5; "J"
```

Para transformar un número binario en decimal, debemos multiplicar el bit n.º 0 por 1(=2⁰); el bit n.º 1 por 2(=2¹); el bit n.º 2 por 4(=2²); el bit n.º 3 por 8(=2³); el bit n.º 4 por 16(=2⁴); el bit n.º 5 por 32(=2⁵); el bit n.º 6 por 64(=2⁶); el bit n.º 7, el último, por 128(=2⁷).

La memoria intermedia de la impresora, o usando un término de la jerga informática el *printer buffer*, traspasa información del ordenador a uno de sus más directos colaboradores, la impresora. La información a traspasar son los caracteres correspondientes a un listado, a una palabra en concreto, etc. Se traspasan los bytes que forman aquellos caracteres, para imprimirlo en papel. La instrucción COPY reproduce el contenido del archivo de memoria a una impresora sin colores, traspasando los atributos de los caracteres si estuviera conectada una impresora en color, aunque esto último más depende del tipo de impresora y su conector o *interface* que del Spectrum.

Si no está conectada una impresora al Spectrum, una orden que afecte a este periférico no tendrá validez. Sin embargo esté o no esté conectada una impresora al ZX Spectrum, siempre se utilizará un espacio determinado de la RAM para este fin. Las direcciones de memoria ocupadas por el *primer buffer* van de la 23296 a la 23551 (hex. 5B00-5BFF).

Solamente 256 bytes (1/4 K) son necesarios para la compleja transmisión de información.

En esta zona de la memoria de lectura/escritura, se almacenan todos aquellos datos que utiliza el microprocesador para su propio funcionamiento, datos para el uso específico del sistema. No son variables BASIC y no serán reconocidas por el microprocesador.

Variable	Dirección	Uso
PROG	23635 23636	Dirección de memoria que indica el lugar de comienzo del programa BASIC
P-RAMT	23732 23733	N.º de bytes de la memoria (ROM + RAM) 16K = 32767; 48K = 65535
UDG	23675 23676	Dirección de memoria que indica el comienzo de los UDG (Gráficos definidos por el usuario)
ETC... (Ver Cap. 25 manual)		

Las variables del sistema ocupan 182 bytes de memoria. Este área comienza en la dirección n.º 23552, extendiéndose hasta la n.º 27333 (hex. 5C00-5CB5).

Una variable del sistema que ocupa dos o más direcciones de memoria, almacena sus valores en base hexadecimal, en base 16, normalmente éstos son una dirección.

Para poder comprender estos valores, debemos cambiarlos a base decimal, multiplicando el byte "high" o "alto" por 256 y sumar ese resultado a byte "low" o "bajo".

Para poder comprender la razón de esta operación bastaría con que nos



La memoria intermedia de la impresora



Las variables del sistema

fijáramos en cómo tendríamos que operar caso de tratarse de cifras expresadas en base 10. Supongamos el número decimal 124.349.

base 10. Supongamos el número decimal 124.349. Supongamos que la información sobre este valor la almacenamos en dos direcciones contiguas. En la alta el 124 y en la baja el 349. Con tres dígitos y siempre en base 10, el número más alto sería el 999 que más 1 (el n.º 0 también cuenta), es 1.000. Tomaremos el alto y lo multiplicaremos por 1.000 a cuyo resultado sumaremos el bajo $124 \times 1.000 + 349 = 124349$.

P-RAMT	byte "high"	byte "low"
23732	—	255
23733	255	—

La dirección 23732 almacena el byte "low", la 23733 el "high". Multiplicando el byte "high", el que almacena los valores altos por 256 y sumándolo al byte "low" daremos con el valor decimal correspondiente a aquellos dos bytes.

Programa 3:

```

20 DIM b$(3,4): DIM a(2,3)
25 LET x=125
30 FOR g=1 TO 2: FOR f=1 TO 3:
LET a(g,f)=65535: NEXT f: NEXT
9 100 LET f=PEEK 23627+256*PEEK 2
3528 110 PRINT f;TAB 10;PEEK f;TAB 2
0;CHR$ PEEK f AND PEEK f>32
120 LET f=f+1: GO TO 110
24002      194      CHR$
24003      17
24004      0
24005      2
24006      3

```

```

24007      0
24008      4
24009      0
24010      3000
24011      3000
24012      3000
24013      3000
24014      3000
24015      3000
24016      3000
24017      3000
24018      3000
24019      3000
24020      3000
24021      3000
24022      129
24023      35

```

De las variables del sistema podemos sacar información muy valiosa (direcciones de comienzo de diferentes zonas, valores necesarios para realizar determinadas funciones), que modificándola nos puede dar buenos resultados.

Ejemplo:

Protección de un programa BASIC:

POKE 23659, 0

La dirección N.º 23659 almacena el número de líneas, incluida una en blanco, de la mitad inferior de la pantalla.

La parte inferior de la pantalla es utilizada por el ordenador para transmitirnos información mediante RAPORTS (ver Apéndice B del manual). Se utiliza normalmente dos líneas para este fin. El valor de las líneas que utiliza el ordenador está almacenado en esta dirección, en una variable del sistema llamada DF SZ, y normalmente siempre es dos.

Si alteramos el valor de esta variable, DF SZ, introduciendo el valor 0, el ordenador no dispondrá de ninguna línea para comunicarnos sus informes y todo el sistema se parará, se quebrará, en un "CRASH", en el caso que nos tenga que imprimir un error.

Al dar 0 a DF SZ se protege un programa BASIC, pues si éste se intenta parar mediante un "BREAK" ante la imposibilidad de aparecer el informe correspondiente (informe L BREAK into program), todo el sistema se parará. Habría que volver a restablecer la variable DF SZ (dirección 23659) a su valor original, valor 2, para poder "introducirse" en el programa en cuestión.

Mapas de *microdrive*

Información para canales

Esta zona de la memoria, cuyo comienzo está fijado en la dirección de memoria n.º 23734 tiene sólo una existencia, por el momento teórica, en el *suo* del ZX Spectrum con *microdrives* (periféricos para acceder a datos y programas a alta velocidad). Sólo se usa el ZX Spectrum sin estos periféricos, este área de la memoria será utilizada por el siguiente bloque de la RAM.

Al conectar los *microdrives*, todo aquello que quede por encima de los mapas de *microdrives* en la memoria será desplazado hacia arriba tantos bytes como ocupen los mapas de *microdrives*.

Por esta misma causa, el comienzo de la mayoría de las siguientes zonas de la RAM, no es fijo, y será almacenado en una variable del sistema o en la ROM misma.

La primera dirección de memoria correspondiente al área de información para canales está almacenada en la variable del sistema CHANS (23631 y 23632. Hex. 5C 4F — 5C50). En el Spectrum, sin *microdrives*, esta primera dirección será la 23734.

Existen 4 canales de información:

CANAL "K": Por ese canal pasa información proveniente del teclado y en dirección a la parte inferior de la pantalla.

CANAL "S": Este canal de información sólo permite salida de información hacia la parte superior de la pantalla. Ninguna entrada de información está permitida.

CANAL "R": Tampoco permite entrada de información alguna, sólo permite su salida hacia otra zona de la RAM llamada "espacio de trabajo".

CANAL "P": Otro canal que sólo permitirá salida de información hacia un periférico, la impresora.

Debemos imaginarnos estos 4 canales de entrada/salida de información, como 4 "puertas" por las cuales irán entrando o saliendo datos según cada orden.

Cada canal de información utiliza 5 bytes. En dos de ellos son almacenadas las direcciones de la rutina de salida, en otros dos, las direcciones de la rutina de entrada de información. El último byte indica el nombre de cada canal.

En el Spectrum con *microdrives* existirán canales o "puertas" suplementarias abiertas; éstas permitirán una mayor entrada o salida de información.

En el Spectrum estándar se utilizan en total 21 bytes. El byte n.º 21 responde a una demarcación del final de esta zona; mediante un código determinado, el 128 decimal (80 hex), el microprocesador podrá localizar el final de este área.

Ejemplo con impresora:

Escribiendo como primera línea de su programa:

10 OPEN#2, "p" o en un comando directo, y si tiene conectada su impresora, todos los comandos PRINT o LIST funcionarán como LPRINT o LLIST.

Podemos utilizar los canales de información para más usos: Dando la orden PRINT # 0; "Introducción de texto en la parte inferior de la pantalla":
PAUSE 0

Podremos utilizar las dos últimas líneas (n.º 22 y 23), que no se pueden alcanzar con la instrucción PRINT AT 22, 0; " "

Es importante detrás de la orden PRINT # 0, que el ordenador no ocupe inmediatamente estas dos últimas líneas, por que no da tiempo a ver los resultados.

La instrucción PAUSE 0, espera a que se pulse una tecla para continuar.

Con la orden PRINT # 3; "Salida de información por impresora sin usar LPRINT", conseguimos la impresión en papel accediendo directamente al canal "p" o "puerta hacia impresora".

7

El área de programa BASIC

Programa 4:

Observación: Desajuste: el Spectrum tuvo que ampliar la zona de variables, al conocer la variable *f*.

Dirección 23878: demarcación FIN zona variables.

Dirección 23879: código perteneciente al carácter de la orden introducida.

Dirección 23880: código de ENTER.

Dirección 23881: demarcación FIN zona de editar.

Dando RUN, parando el programa con la pregunta "¿scroll?", e introduciendo GOTO 1, la segunda vez no existirá desajuste, pues la variable *f* ya es conocida.

Este área de la memoria RAM almacena el programa BASIC, en caso de que exista.

El comienzo de esta zona está almacenado en la variable del sistema PROG, que ocupa las direcciones de memoria 23635 y 23636.

Si no hay ningún microdrive conectado, el área de programa BASIC comenzará en la dirección de memoria n.º 23755 (hex. 5CCB).

A partir de esta dirección de memoria, las líneas del programa se almacenan de la siguiente manera:

Los dos primeros bytes almacenan el n.º de la línea.

El 1.º será el byte "high", y el 2.º será el byte "low".

Los dos bytes siguientes almacenan la longitud de aquella línea en bytes, a partir del byte n.º 5 hasta el final de la línea correspondiente. En este caso, los bytes 3.º y 4.º, almacenan primero el byte "low" y después el "high", es decir, en orden inverso a como se almacenan el n.º de línea.

El fin de cada línea se marca con un código concreto: el código 13 (ENTER). Ver programa 1.

Después de este código, se procederá a almacenar la siguiente línea, comenzando por el n.º de línea y siguiendo con su longitud.

Entre el 5.º byte y el código 13, son almacenados los códigos que corresponden a los caracteres de aquella línea.

Si en una misma línea se encuentran dos órdenes, se separan por el signo de dos puntos (:), código 58 (hex. 3A), sin haber otro tipo de marcas o señas para estas sentencias múltiples.

Los números son tratados de manera especial.

Si aparece un número en una sentencia, después de los códigos correspondientes a sus caracteres, aparece el código 14 (número) y la representación de aquel número en coma flotante o en forma de números enteros para enteros en el intervalo -65535 a 65535. En cualquiera de los dos casos, tras aparecer un número en una sentencia, serán necesarios 6 bytes extra, 6 direcciones de memoria para almacenarlo.

Veamos ahora como se puede ahorrar memoria utilizando diversos métodos:

```
10 LET A=0: LET B=0: LET C=1
10 LET A=PI-PI: LET B=A: LET C
=PI/PI
10 LET A=23755: LET B=2: LET C
=23757
10 LET A=VAL "23755": LET B=VA
"2": LET C=A+B
10 PRINT "abcde"
20 PRINT AT 10,10;"f"
30 PRINT AT 12,9;"g"
10 PRINT "abcde";AT 10,10;"f",
AT 12,9;"g"
```

```
10 PRINT AT 1,0;"p"
10 PRINT "p"
```

```
PRINT 65536-USR 7962
PRINT (PEEK 23627+256*PEEK 2362
8)-(PEEK 23635+256*PEEK 23636)
PRINT (PEEK 23641+256*PEEK 2364
2)-(PEEK 23635+256*PEEK 23636)
```

8

La zona de las variables

No se debe confundir esta zona con el área de variables del sistema. La función de este área es el almacenar todas aquellas variables, ya sean numéricas o alfanuméricas, y conjuntos, numéricos o alfanuméricos.

El comienzo de esta zona está definido por la variable del sistema VARS (23627 y 23628). El área o zona de las variables se encuentra, en la memoria, por encima de el área de programa BASIC; a medida que el programa BASIC va aumentando en su número de líneas, el área de variables irá quedando más por encima en memoria. El último byte de la zona de variables tendrá el valor 128 decimal ó 80 hex.; de este modo se separa el área de variables de la siguiente zona de la RAM.

Variables numéricas y alfanuméricas:

Las variables numéricas se almacenan de la siguiente manera: como primer

SOFTWARE

byte irá el código del carácter que corresponde a la variable en cuestión. Ver programa 2.

Para almacenar la variable "a", el primer byte será el código n.º 97, que es el código que corresponde al nombre de la variable. Existen 5 bytes para almacenar el valor de una variable numérica. Estos 5 bytes se encuentran tras el byte que define el código de aquella variable. Con estos 5 bytes se pueden expresar desde el $\pm 1E + 38$ (un 1 seguido de 38 ceros).

El almacén de variables alfanuméricas es análogo. Como primer byte está el código del carácter de la variable. Si la variable es d\$, el código será el 68, perteneciente al "D". La longitud de una variable alfanumérica no tiene límites, si no es el de la memoria. Después del código que define el nombre de la variable, el código del carácter, encontramos dos códigos que definen la longitud de la variable alfanumérica.

Tras estos dos códigos, encontramos el valor en sí de la variable alfanumérica, los códigos de los caracteres que la forman.

El microprocesador diferencia las variables numéricas de las alfanuméricas por este método:

El código que define el nombre de las variables numéricas pertenecerá a un carácter en minúscula, es decir códigos entre el n.º 97 y el n.º 122.

Los códigos que definen el nombre de las variables alfanuméricas, pertenecerá a un carácter en mayúscula, entre los códigos n.º 65 y n.º 90.

De esta manera el ordenador distingue y reconoce unas variables de otras.

Conociendo las direcciones de memoria que almacenan el valor de una variable, podemos modificarlo mediante POKES, alteraciones de aquellas direcciones de memoria, que variarán el valor de la variable en cuestión.



SUSCRIBASE POR TELEFONO

- * más fácil,
- * más cómodo,
- * más rápido

(91) 250 15 93

7 días por semana, 24 horas a su servicio

SUSCRIBASE A

**CIRCUITO
IMPRESO**

9

El área de editar

10

El espacio de trabajo ("Espacio para tareas eventuales")

11

La pila del calculador

12

La memoria de reserva

Si se conoce la dirección de memoria que almacena el código del carácter de aquella variable, el nombre de aquella variable, también podemos modificarlo, introduciendo otro código perteneciente a otro carácter. De esta manera el ordenador no podrá reconocer la variable antigua, asignando el valor de aquella a la recién incorporada, cuyo nombre coincidirá con el carácter del código que introdujimos.

Conjuntos numéricos y alfanuméricos:

Los conjuntos numéricos o alfanuméricos se consiguen mediante la orden DIM. Ver programa 3.

Al DIMensionar un conjunto alfanumérico A\$ (3,4), quedan en la zona de variables $3 \times 4 = 12$ direcciones de memoria "libres". A medida se van definiendo los elementos de este conjunto, los espacios de memoria que quedaron "libres" al DIMensionar el conjunto, son "ocupados" por los códigos de los caracteres del elemento del conjunto.

Al DIMensionar un conjunto numérico a (3,4), quedan en la zona de variables $3 \times 4 \times 5 = 60$ direcciones de memoria "libre", pues cada número, al igual que en la zona del programa BASIC, o al almacenar variables numéricas en la zona de variables, necesita de 5 bytes para ser definido.

Esta zona de la RAM tiene la función de almacenar o editar la línea de comando BASIC que en ese momento se esté escribiendo o se haya editado.

El comienzo del área de editar se almacena en la variable del sistema E-LINE (direcciones de memoria 23641, 23642).

Cuando en la parte inferior de la pantalla aparece un cursor parpadeante, el área de editar sólo ocupa 3 posiciones o direcciones de memoria: el código de ENTER (también almacenado en la variable del sistema K-CUR) y una demarcación final, representada aquí por el código decimal 128, hex. 80.

EDIT "baja" una línea BASIC a la parte inferior de la pantalla. Para realizar esto, primeramente se amplía la zona de editar para que la línea BASIC quepa en su totalidad; seguidamente el contenido de esta zona, se "copia" en la parte baja de la pantalla. Ver programa 4.

El espacio de memoria de este área está usado para tareas muy diversas: introducción de datos, concatevación de cadenas, etc. El comienzo de esta zona está definido por la variable del sistema WORKSP, que ocupa las direcciones de memoria 23649 y 23650. Cada vez que se necesita memoria en el "espacio de trabajo", este área es expandido.

Después del trabajo en esta zona, el "espacio de trabajo" es "vaciado". Siempre se trata de ahorrar al máximo de memoria, utilizando siempre y solamente el necesario.

La pila del calculador tiene la función de almacenar números en coma flotante, números enteros en 5 bytes y con tratamiento de cadenas.

La pila es manipulada como "último dentro, primero fuera", por ello, el valor almacenado en lo alto de la pila será considerado como el último introducido.

El microprocesador Z80 hace uso de la pila del calculador para cálculos internos matemáticos.

La dirección de comienzo de la pila del calculador es almacenada en la variable del sistema STKBOT (direcciones 23651 y 23652). El fin de la pila del calculador es almacenado en la variable del sistema STKEND (23653 y 23654).

Los bytes existentes entre la pila del calculador y la pila de máquina, representan la cantidad de memoria disponible para el usuario. Sólo en esto se diferencian un modelo del ZX Spectrum del otro.

En el modelo de 16K, la memoria de reserva se eleva a 8.839 posiciones de memoria. En el modelo de 48K, la memoria de reserva contiene 41.472 direcciones de memoria "libres".

SOFTWARE

```

10 LET f=PEEK 23641+256*PEEK 2
3642
100 PRINT f;TAB 10;PEEK f;TAB 2
0;CHR$ PEEK f AND PEEK f>32
110 LET f=f+1: GO TO 100

```

23873	0	
23874	0	
23875	67	C
23876	93	I
23877	0	
23878	128	
23879	247	RUN
23880	13	
23881	128	
23882	0	
23883	0	
23884	76	L
23885	93	J
23886	0	
23887	0	
23888	51	
23889	0	
23890	0	?
23891	0	?
23892	0	

```

23893 0
23894 0

```

```

10 GO SUB 20: STOP
20 GO SUB 30: RETURN
30 GO SUB 40: RETURN
40 FOR a=65365 TO 65367 STEP -
1: PRINT a,PEEK a: NEXT a: RETUR
N

```

65365	2
65364	0
65363	10
65362	20
65361	0
65360	20
65359	20
65358	0
65357	30

```

40 FOR f=32597 TO 32589 STEP -
1: PRINT a,PEEK a: NEXT a: RETUR
N

```

Programa 5:

Para 16K:

Cambiar línea 40 a 40 FOR f=32597 TO 32589 STEP 1; PRINT A, PEEK A; NEXT A; RETURN

El área de memoria usada por el microprocesador para su uso personal, es la pila de máquina. Instrucciones de código máquina (Assembler) como PUSH o POP, guardan en esta zona los valores de ciertos registros.

13

La pila de máquina

ZX pone sus páginas a disposición de los lectores que deseen publicar anuncios clasificados para compra venta de ordenadores y periféricos, intercambiar programas o simplemente tomar contacto con otros usuarios de Sinclair. La publicación de estos anuncios será gratuita. No serán aceptados los que tengan carácter publicitario. Enviar los textos, acompañados de dirección, identificación y teléfono, a:

Revista ZX. Tablero de anuncios. Jerez, 3 Madrid - 16.



14

La pila de GO SUB

La instrucción BASIC GO SUB, envía al programa a una subrutina, de la cual regresará al encontrarse la instrucción RETURN. El ordenador debe almacenar todas las instrucciones de retorno. Para cada GO SUB, se almacenan en la pila de GO SUB 3 datos: el primer byte señala el número de comando dentro de una misma línea a donde tiene que retornar de la subrutina. Los dos bytes restantes almacenan el número de líneas a donde debe volver el control del programa. Ver programa 5.

Primero se almacena el byte "high" o alto, seguido del byte "low". Multiplicando el primero por 256, y sumándose al segundo, dará como resultado el número de línea a donde debe volver el RETURN.

La casa Sinclair no ha definido todavía como se escribe la palabra "GO SUB". Como comando aparece escrito junto "GOSUB", pero cuando aparece el informe 7 (probar dando RETURN como orden directa), escribirá la palabra separada.

Encima de la pila del GO SUB, se encuentran dos códigos especiales: el 0 y 62 (hex. 0 y 3E). Estos dos códigos separan esta zona de la siguiente, delimitando el RAMTOP o tope máximo del sistema BASIC.

La dirección de memoria donde se encuentran estos dos códigos se almacena en la variable del sistema RAMTOP (23730 y 23731). Mediante la instrucción CLEAR, se varía la dirección del RAMTOP, alterándose la organización de la memoria.

15

Los gráficos definidos por el usuario

Por encima del RAMTOP, se encuentra la zona de gráficos definidos por el usuario. Sabiendo BASIC, el ZX Spectrum ofrece la posibilidad de definir hasta 21 gráficos.

Cada caracter consta de 8 bytes, cada byte de 8 bits.

Cada caracter consta en total de 64 bits/puntos/pixels.

Introduciremos como dato un conjunto de 8 bits, un byte.

El comienzo de esta zona está definido por la variable del sistema UDG (23675, 23676). Podemos cambiar el comienzo de esta para ahorrar memoria, a costa de tener menos gráficos. El comienzo de esta zona, es en un 48K, la 65368. En un 16K es la 32600. A partir de esta dirección se almacenan los bytes de los 21 caracteres definibles. En total $21 \times 8 = 168$ bytes o direcciones de memoria ocupadas por esta zona. Los caracteres definibles por el usuario se pueden conseguir con el cursor en modo gráfico, pulsando las teclas de la "A" a la "U".

Al principio, estos gráficos corresponderán con los caracteres de las letras mayúsculas de la "A" a la "U".

El fin de la RAM, la dirección de memoria más alta se encuentra almacenada en la variable del sistema P-RAMT (23732 y 23733).

Podremos distinguir un 16K de un 48K leyendo el contenido de esta variable:

PRINT PEEK 23732 + 256 * PEEK 23733.

En el modelo 16K dará como resultado 32767.

En el modelo 48K dará como resultado 65535.

¿Cuánta memoria me queda?

El ZX Spectrum dispone de una subrutina ROM que calcula el espacio de memoria libre:

PRINT 65536-USR 7962

¿Cuánta memoria ocupa este programa?

a) sin variables: VARS - PROG

PRINT (PEEK 23627+256* PEEK 23628) - (PEEK 23635+256* PEEK 23636)

b) con variables: E-LINE - PROG

PRINT (PEEK 23641+256* PEEK 23642) - (PEEK 23635+256* PEEK 23636).

Programa 6
10 POKE 22540, 149

Seguiremos en el próximo número.
Juan Martínez Velarde

MICROBYTE

EL PRINCIPIO DE ALGO NUEVO

SOFTWARE SPECTRUM



1 JET PAC

Debes ensamblar el cohete, con piezas que encontrarás por la galaxia y evitar las agresiones de otros zares. Varios niveles de juego, posibilidad de utilizar Joystick y 1 ó 2 jugadores. Gráficos y efectos espectaculares.

16 K

1.800 Pts.



2 PSSST

Como defensor de la naturaleza debes cuidar la "planta de la vida" de las agresiones de insectos, gusanos, etc., para ello utilizarás los distintos sprays de colores con habilidad y movimiento. 1 ó 2 jugadores. Posibilidad de Joystick. Efectos gráficos espectaculares.

16 K

1.800 Pts.



3 MASTERCHES

Juego de ajedrez con diferentes niveles de dificultad. Posibilidades de análisis, etc., excelentes gráficos.

48 K

1.800 Pts.



4 ESCAPE M.C.P.

El hombre en lucha contra máquinas cibernéticas a través del laberinto de circuitos de una gran computadora.

16 K

1.600 Pts.



5 MONSTER MADE 3D

(Tridimensional) El objetivo ha de consistir en encontrar la salida del laberinto de la muerte, que custodia el terrible monstruo.

16 K

1.600 Pts.



6 TOBOR

La ciudad ha sufrido una invasión de robots con el fin de adueñarse del mando del computador central; tu misión: evitar que esto suceda...

48 K

1.600 Pts.



7 FROGGER

Conduce con habilidad al mayor número de sapitos "frogger" hasta su refugio, atravesando la autopista y al río de la muerte.

16 K

1.500 Pts.



8 FIREBIRDS

El espacio sufre la invasión de los terribles plásmos del infierno. La nave Orbiter-1 es la encargada de interceptarlos... ¿lo conseguirá?

16 K

1.500 Pts.



9 PACKMAN

Versión mejorada del popular juego del conecoco, realizado en C.M.

16 K

1.500 Pts.



10 COMBAT 3D

Eres el piloto de un sofisticado tanque de guerra; el objetivo, combatir al enemigo en el espacio más hostil de 3 dimensiones, que con sus tanques y armas sofisticadas intentarán arrebatarlos.

16 K

1.800 Pts.

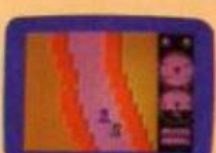


11 CONTROL AEREO

Simula las funciones del aeropuerto de Heathrow y el objetivo será controlar el aterrizaje del máximo número de aviones en el menor tiempo posible.

16 K

1.800 Pts.



12 RACE CARS

Recorrido apasionante pilotando un potente y veloz Fórmula 1, las curvas y los demás participantes no harán fácil al finalizar la carrera.

16 K

1.600 Pts.



13 AQUAPLANE

Atractivo juego en el que para conseguir alta puntuación debes manejar con acierto los movimientos de un esquiador en las cristalinas aguas de Honolulu. Extraordinarios gráficos.

16 K

1.600 Pts.



14 COLOUR CLASH

Juego de inteligencia y reflejos. En el menor tiempo posible se deben rellenar el máximo número de figuras sin ser interceptados por los "COLOUR CLASH".

16 K

1.600 Pts.



15 3D CUADRACUBE

Juego para intelectuales en 3 dimensiones, donde debes colocar 4 cubos del mismo color en una sola dirección. Posibilidad de 1 ó 2 jugadores.

16 K

1.600 Pts.

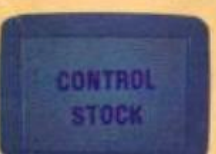


16 CIUDADES DEL MUNDO

Excepcional programa educativo de geografía universal. Poblaciones, idiomas y datos de interés general serán de ayuda en el conocimiento de esta materia.

16 - 48 K

1.600 Pts.



17 CONTROL STOCK

Ordena y proporciona información sobre un stock de 200 items.

48 K

1.800 Pts.



18 CONTABILIDAD

Contabilidad personal, permite controlar hasta 24 grupos de ingresos y gastos, balances mensuales, anuales, etc.

48 K

1.800 Pts.

Envíenos a MICROBYTE		San Gerardo, 59. MADRID-35	
Nombre		Cantidad	Nº Artículo
Apellidos		Precio	TOTAL
Dirección			
Población			
D.P.	Teléfono		
Incluyo talón bancario nominativo			
Contra - Reembolso			
Giro Postal			
ENVÍOS GRATIS		PRECIO TOTAL PESETAS	
		Pedidos por Teléfono 91-6565002	



Todo para sinclair

Te esperamos en ***sinclair store***, tu tienda informática, donde encontrarás la mas extensa variedad en hardware y software para **sinclair**.

€ SINCLAIR STORE · Bravo Murillo, 2 - Tél. - 4 46 62 31 - Madrid