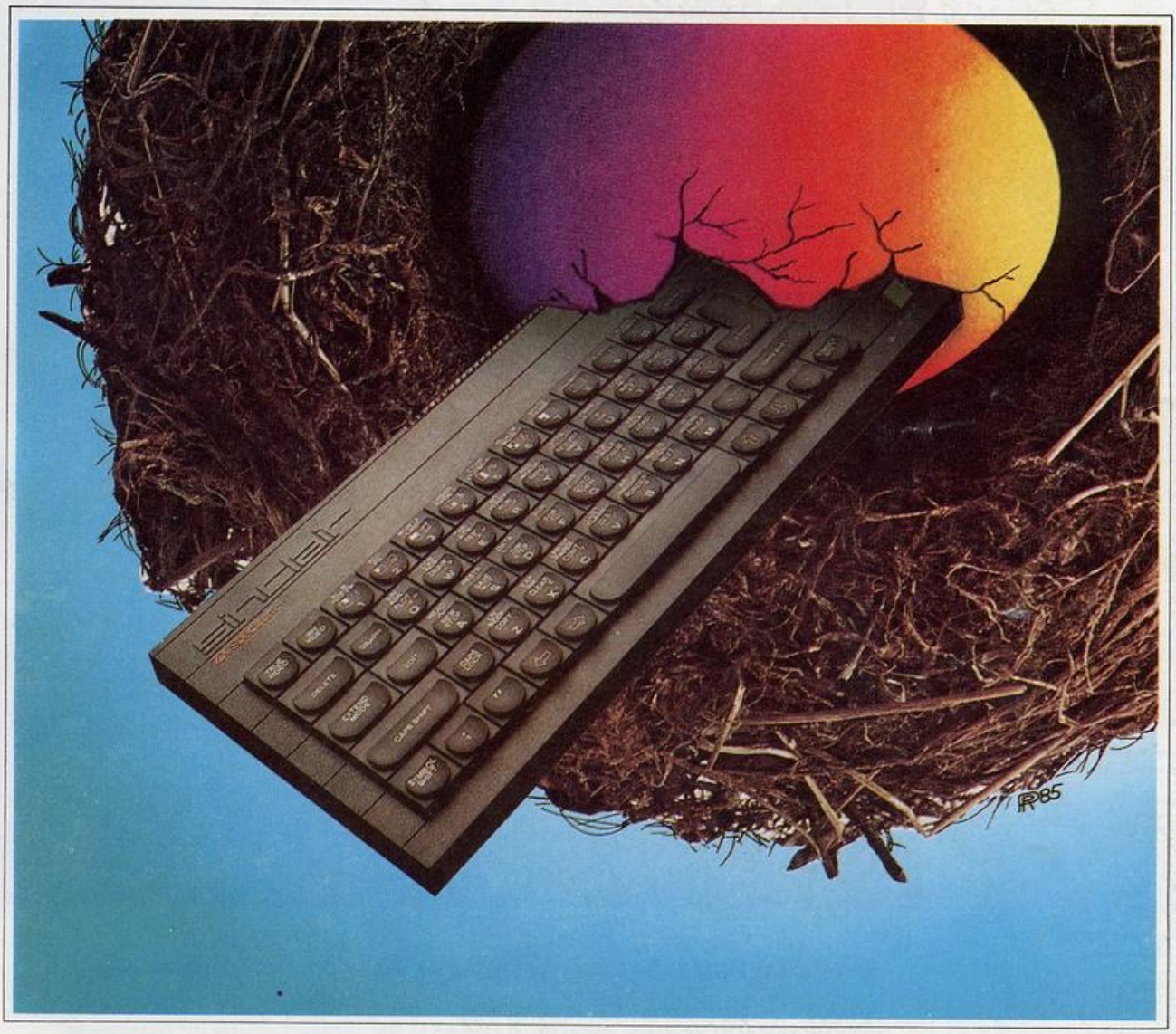
29
150 pts.

RUN

Enciclopedia Práctica del Spectrum



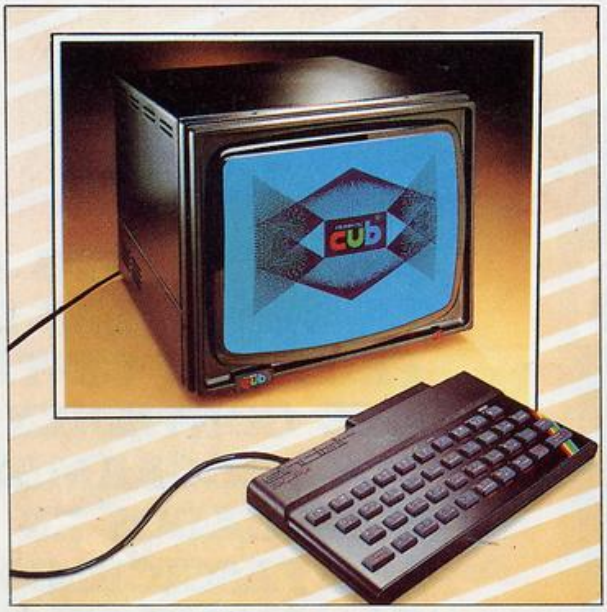
Nueva Lente/Ingelek

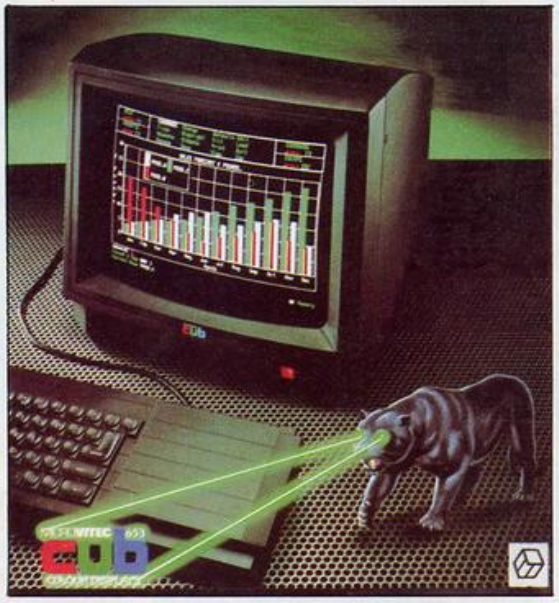
LA UNICA Y DEFINITIVA SOLUCION EN COLOR COMPATIBLE CON SU SISTEMA

Modelos	Pixels
14" Standard	452 x 585
14" Media	653 x 585
14" Alta	895 x 585
20" Standard	505 x 585
20" Alta	860 x 625

1431/AP DS
P.V.P. 86.200 ptas.

1431/MZ 3
P.V.P. 83.700 ptas.





1451/DQ P.V.P. 91.600 ptas.

¿QUIEN NECESITA UN MONITOR EN COLOR?

Por supuesto toda persona que tenga un ordenador.

Hasta ahora Vd. probablemente usaba su televisor doméstico con su ordenador y habrá notado bastantes interferencias, especialmente cuando visualiza textos. Los televisores no están básicamente diseñados para visualizar datos, ya que están construidos con circuitos de codificación y modulación para aceptar únicamente las ondas de televisión a través del aire.

La diferencia entre su televisor y un monitor CUB, es que éste último está especialmente diseñado para la visualización de textos y gráficos, ésto se evidencia inmediatamente en la imagen estable y clara que réduce notablemente el esfuerzo de la vista.

El monitor CUB está preparado para desarrollar las capacidades sofisticadas de visualización de los ordenadores de hoy y del mañana.

¿POR QUE ELEGIR UN MONITOR CUB?

Sólo la gama CUB de Microvitec, es suficientemente completa para cubrir la compatibilidad de casi todos los micro ordenadores del mercado.

Estos magníficos monitores británicos, son los únicos elegidos por el Gobierno inglés para usarlos en las escuelas primaria y secundaria de todo el país.

Nuestra gama de monitores de resoluciones standar, media y alta, más los modelos PAL/RGB, son compatibles totalmente con IBM PC/PCjr, APPLE II/IIe/IIi, SINCLAIR SPECTRUM/QL, COMMODORE 64/VIC 20, DRAGON 32/64, ORIC, BBC, ACORN ATOM, ATARI, ACT APRICOT, SHARP, ITT, TANDY, ADVANCE, CROMMENCO 501, LYNX, TEXAS INSTRUMENTS T 99/4A y muchos más.

Piense, cuando tome su decisión final, que sólo los CUB de Microvitec le pueden proporcionar la mayor calidad, rendimiento y fiabilidad al mejor precio.

TODOS LOS MONITORES CUB INCLUYEN:

- Garantía total por un año.
- Chasis aislado para máxima seguridad.
- Interruptor de potencia para un mejor rendimiento.
- Mínimo error de convergencia esencial para visualización de textos gráficos.
- Diseñados para introducir los standars reconocidos de seguridad (i.e. BS415).
- Chasis preparado para bajo consumo de potencia.
- Componentes de alta calidad para asegurar la máxima fiabilidad.
- Mando de conexión de potencia, plug y RGB.
- Diseño práctico, atractivo y moderno.
- Aprobación por la B.E.A.B. de nuestros más populares modelos.
- La mejor relación calidad-precio.

MICROVITEC CUB
COLOUR MONITORS

multilogic

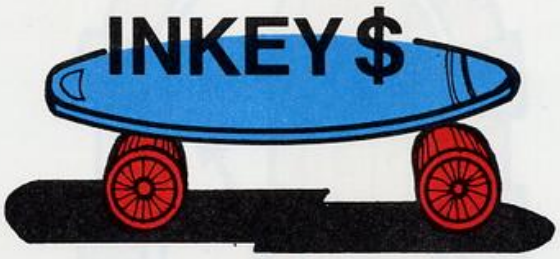
Paseo de la Habana, 145 - 28036 Madrid
Tel. 458 74 75 - Telex 44921 MLOG

MAS DE UN JUGADOR

La programación de aplicaciones en tiempo real (generalmente juegos de acción) impone la necesidad de contar con una rápida entrada de información vía teclado. Esta tarea es acometida por el BASIC mediante la función **INKEY$**, cuyo resultado es la obtención del código de la tecla pulsada en el instante de ejecutarse la referida sentencia, o la cadena nula en el caso de que no esté accionada tecla alguna.

A nadie se le oculta que el problema surge cuando se pulsan a la vez varias teclas. ¿Qué tecla es la que hay que considerar? La verdad es que el ordenador tiene una solución drástica para salir airoso de estas situaciones: hasta que no se libera la tecla accionada, ignora la pulsación de cualquier otra.

Sin embargo, tan contundente medida acarrea problemas insalvables a la hora de codificar programas en los que interviene más de un usuario simultáneamente.



La programación de juegos de acción precisa una rápida entrada de datos desde el teclado; para ello utilizamos la función **INKEY$**.

COMUNICACIONES DE NUESTRO ORDENADOR

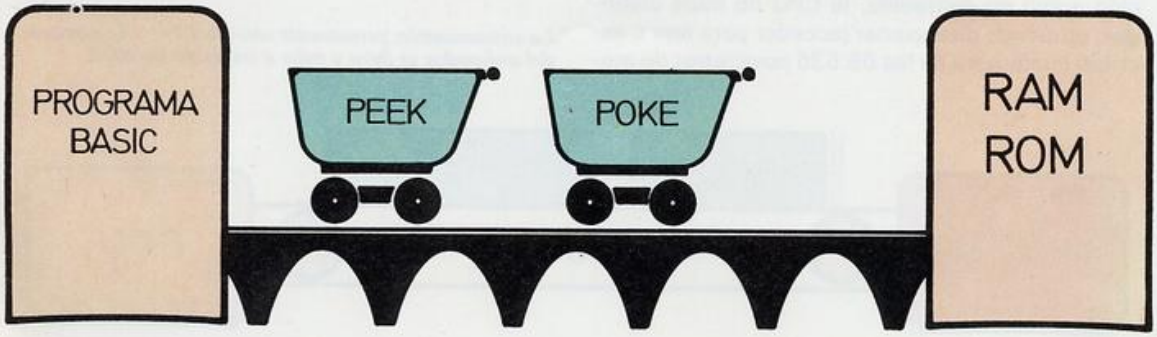
Ya conocemos, por capítulos anteriores, que el empleo de las funciones **PEEK** y **POKE** nos facilita un medio de intercambio de información entre nuestro programa BASIC y el contenido de determinadas posiciones de la memoria ROM y RAM del ordenador.

Recordemos ahora la diferencia fundamental de tratamiento entre la memoria del tipo ROM (*Read Only Memory,* memoria de sólo lectura) y RAM (*Random Access Memory,* memoria de acceso aleatorio).

Como su propio nombre indica, la memoria ROM permite únicamente su lectura y no la escritura, dado que la memoria de este tipo es la que constituye el *firmware* de nuestro ordenador, o, lo que es lo mismo, el conjunto formado por el sistema operativo y el lenguaje de programación, cuyos contenidos no deben alterarse en ningún caso.

Para asegurarse una protección eficaz, el sistema inhibe cualquier sentencia **POKE** que afecte

El empleo de **PEEK** *y* **POKE** *facilita el intercambio de información entre el BASIC y la memoria.*



¡!

Una vez conocido el byte que corresponde a cada tecla basta con observar su estado lógico: si el bit es cero, la tecla está pulsada; mientras que si es uno, significa que está libre.

*

La dirección del port 254 permite utilizar parte de sus bits para el control directo de las conexiones MIC y EAR, así como el color del contorno.

*

Las direcciones de los ports del teclado se obtienen por aplicación de la fórmula:

```
FOR I=0 TO 7:
PRINT 254+256*
(255-2↑I),:NEXT I
```

¡!

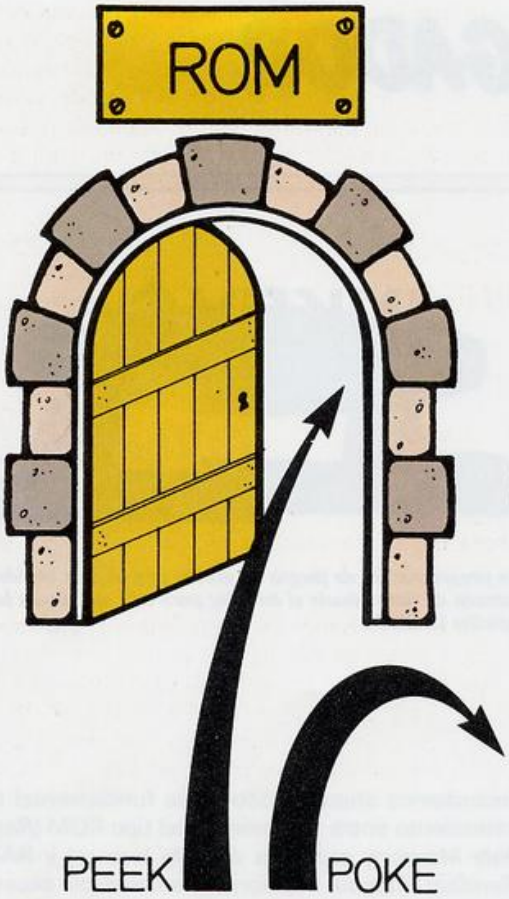
De hecho, aunque en la mayoría de las versiones los tres bits más significativos obtenidos del byte leído del port son tres unos, se da el caso de que, al menos en la versión 3 *(ISSUE 3)*, el sexto bit resulta ser un cero.

*

OUT puede emplearse para controlar el altavoz o la impresora del ordenador.

*

Para encontrar el significado del byte obtenido por medio de la función **IN**, es necesario decodificarlo en formato binario, puesto que **IN**, al igual que **PEEK**, nos brinda el resultado en base decimal.



Para asegurar una protección eficaz, la ROM inhibe el efecto de cualquier sentencia **POKE**, *aunque admite la función* **PEEK**.

a alguna de estas direcciones, permitiendo, sin embargo, el empleo de la función **PEEK**.

Por el contrario, la memoria RAM, de que se compone la mayor parte del total disponible en nuestro ordenador, no incorpora esta protección, por lo que cualquiera de sus posiciones nos puede resultar accesible, tanto para lectura como escritura, a través de las sentencias **PEEK** y **POKE**, respectivamente.

A pesar de la clara diferencia, tanto en protección como en contenido, la CPU no hace distingos, pudiendo direccionar (acceder para leer o escribir) cualquiera de las 65.536 posiciones de memoria, máximo número cuya expresión binaria utiliza dos bytes, para obtener el byte que ocupa la posición (lectura) o reemplazar por uno nuevo (escritura).

Esta comunicación permanente entre la memoria del ordenador y su CPU se produce a través de los BUS, uno de direcciones (16 bits) y otro de datos (8 bits). Sin embargo, con lo dicho hasta ahora sólo hemos comentado la forma de comunicación interna del ordenador o, lo que es lo mismo, el modo de llevar un byte de datos desde la CPU a una posición determinada de memoria y viceversa.

El ordenador no es una máquina cerrada al mundo exterior, y, por tanto, dispone de un sistema de comunicación bastante completo (ya sea con la impresora, teclado, etc.). Estos diálogos con el mundo exterior se realizan a través de los accesos de entrada/salida del Spectrum (*Port Input/Output* o *PIO*).

Existen 65.536 ports a los cuales se puede acceder para establecer una comunicación entre cualquier periférico y el ordenador, aunque realmente sólo se emplean unos pocos, estando regida esta comunicación por las sentencias BASIC **IN** y **OUT**, de gran similitud con las conocidas **PEEK** y **POKE**, respectivamente.

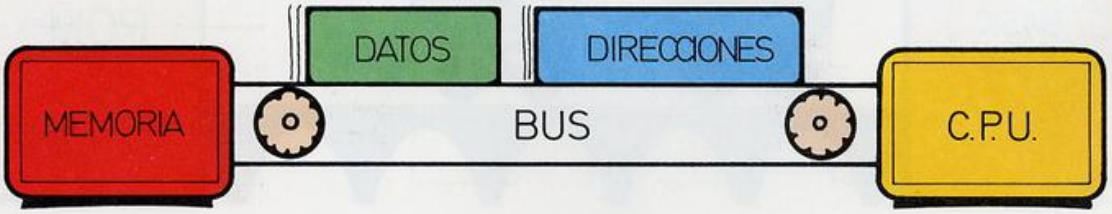
LAS SENTENCIAS IN Y OUT

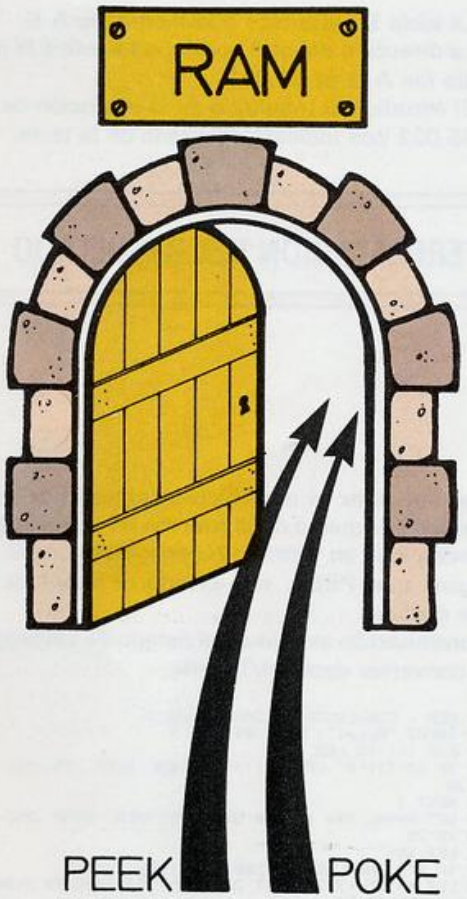
La función **IN** lleva como parámetro la dirección del port, siendo su resultado el byte de datos que ocupa éste en el momento de ejecutar la función.

La sentencia **OUT** lleva como parámetros la dirección del port al cual se desea acceder y el dato que por él se desea enviar, separados por una coma.

Para los aficionados a la electrónica y entendi-

La comunicación permanente entre la CPU y la memoria del ordenador se lleva a cabo a través de los BUS.





Al contrario que la ROM, la RAM permite operaciones tanto de lectura como de escritura.

dos en el lenguaje Ensamblador el empleo de los ports abre un mundo de posibilidades para la comunicación de nuestro ordenador con dispositivos exteriores muy variados: electrodomésticos, alarmas, robots, etcétera.
Dejando un poco de lado todo ello, nos centraremos en la forma en que puede establecerse una comunicación con los diferentes ports que controlan el teclado del ordenador, para conseguir de esta forma efectuar una lectura rápida del mismo que permita la acción simultánea de varios usuarios o, simplemente, la posibilidad de que en un programa BASIC pueda tenerse en cuenta la pulsación simultánea de varias teclas.

UN PEEK DE LAS TECLAS

Ahora que conocemos la posibilidad de leer del teclado a través de un *PIO* (*Port Input/Output*), por medio de la función **IN**, sólo nos queda saber en qué dirección debemos preguntar, según la situación de las teclas a investigar.
A efectos de direccionamiento para captación de datos de los ports, el teclado se encuentra dividido en ocho «medias filas» de cinco teclas, a cada una de las cuales corresponde una dirección, de la forma:

Media fila	Dirección
1-5	63.486
0-6	61.438
Q-T	64.510
P-Y	57.342
A-G	65.022
ENTER-H	49.150
CAPS SHIFT-V	65.278
SPACE-B	32.766

Mientras que* IN *precisa un solo parámetro, una dirección del port,* OUT *necesita, además, el dato a enviar.

¡!

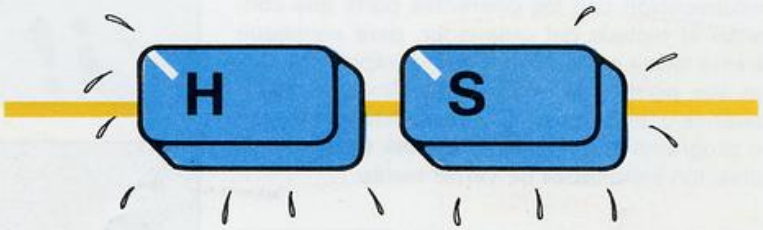
El empleo de los ports abre un mundo de posibilidades para la comunicación con dispositivos exteriores muy variados.

*

La sentencia **OUT** lleva como parámetros la dirección del port al cual se desea acceder y el dato que por él se desea enviar, separados por una coma.

*

A efectos de direccionamiento para captación de datos de los ports, el teclado se encuentra dividido en ocho «medias filas» de cinco teclas, a cada una de las cuales corresponde una dirección.



En algunos programas es necesario la pulsación simultánea de más de una tecla.

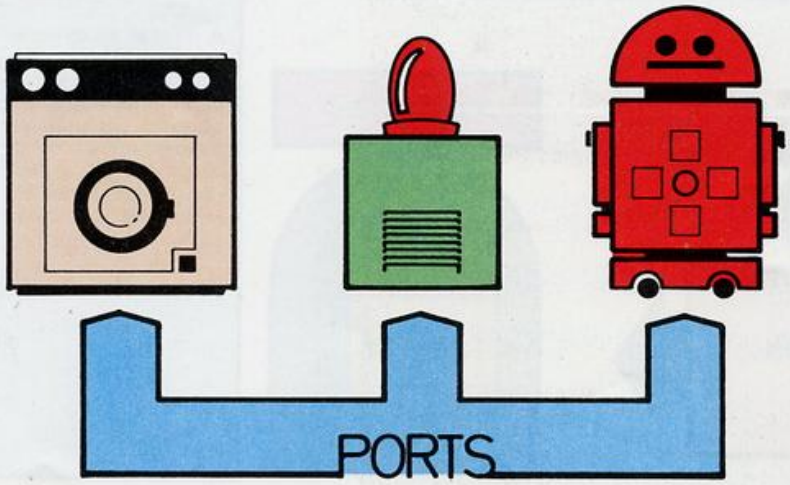
La decodificación del byte produce ocho bits. Eliminando los tres bits situados más a la izquierda (los más significativos), de los cinco bits que conservamos, el menos significativo (el situado más a la derecha) corresponde a la tecla más exterior de la media fila.

Estas direcciones de ports del teclado, aparentemente caprichosas, se obtienen por aplicación de una fórmula:

FOR I=0 TO 7:PRINT 254+256*(255-2↑I),;: NEXT I

Así pues, si deseamos conocer el estado de una tecla en concreto, primero hemos de determinar a qué zona pertenece; a continuación averiguar cuál es la dirección del port correspondiente a la zona en cuestión y, finalmente, efectuar el **IN** de dicha dirección.
Este último paso dará como resultado un byte, de cuyo análisis nos ocuparemos en el siguiente apartado. Pero primero resolvamos un ejemplo práctico: supongamos que deseamos averiguar si la tecla **S** está pulsada; el método a seguir se concreta en los siguientes pasos:

El empleo del port abre un mundo de posibilidades para la comunicación de nuestro ordenador con dispositivos exteriores muy variados.



1. La tecla **S** pertenece a la media fila A-G.
2. La dirección del port correspondiente a la media fila A-G es 65.022.
3. El estudio del resultado de la ejecución de **IN** 65.022 nos indicará el estado de la tecla.

INTERPRETACION DEL RESULTADO

Para encontrar el auténtico significado del byte obtenido por medio de la función **IN** es necesario decodificarlo en formato binario, puesto que **IN**, al igual que **PEEK**, nos brinda el resultado en base decimal.
A continuación incluimos el listado de un programa conversor decimal/binario:

```
  10 REM - CONVERSOR DECIMAL/BINARIO
  20 INPUT "Byte:"; LINE X$
  30 FOR I=1 TO LEN X$
  40 IF X$(I)<"0" OR X$(I)>"9" THEN  BEEP .25,-10: GO
 TO 20
  50 NEXT I
  60 LET X=VAL X$: IF X<0 OR X>255 THEN  BEEP .25,-10
: GO TO 20
  70 LET X$=""
  80 IF X<2 THEN  GO TO 100
  90 LET Y=INT (X/2): LET Z=X-2*Y: LET X$=STR$ Z+X$:
LET X=Y: GO TO 80
 100 LET X$="0000000"+STR$ X+X$: LET X$=X$(LEN X$-7 T
O ): PRINT X$,;: GO TO 20
```

En la línea 20 se incluye un **INPUT LINE**, por lo cual, al concluir con la introducción de datos que deseamos convertir a formato binario, es necesario pulsar **CAPS SHIFT 6 (CURSOR ABAJO)** para salir de programa.
En las líneas 30 a 50 se efectúa la depuración de que los datos introducidos sean numéricos, mediante un recorrido de la variable de cadena carácter a carácter. Si se produce un error, se emite un zumbido bifurcando a la línea 20, para repetir la entrada del dato.
Una vez superada la depuración anterior, la línea 60 efectúa una complementaria, consistente en controlar que el dato numérico introducido no exceda los límites de representación en formato binario en un byte. Como sabemos, la mayor cifra decimal representable en un byte es 255, en dos 65.535, etcétera.
La línea 70 restaura la cadena **X$** al valor nulo, para incorporarle, sucesivamente, los dígitos binarios obtenidos mediante la traducción del byte decimal original.
La línea 80 controla el final del bucle de decodificación, efectuando una bifurcación a la línea de impresión del resultado final.
La línea 90 realiza la función de traducción concretamente, añadiendo un nuevo dígito binario a la cadena **X$** por cada pasada.

Por último, la línea 100 gestiona la impresión en pantalla, a doble columna, del resultado binario obtenido como conclusión del proceso de traducción.

Terminada esta operación, nos encontraremos ante una fila de ocho bits (ceros y unos). Para empezar la interpretación, hemos de eliminar los tres bits situados más a la izquierda (los más significativos). De los cinco bits que conservamos, el menos significativo (el situado más a la derecha) corresponde a la tecla más exterior de la media fila.

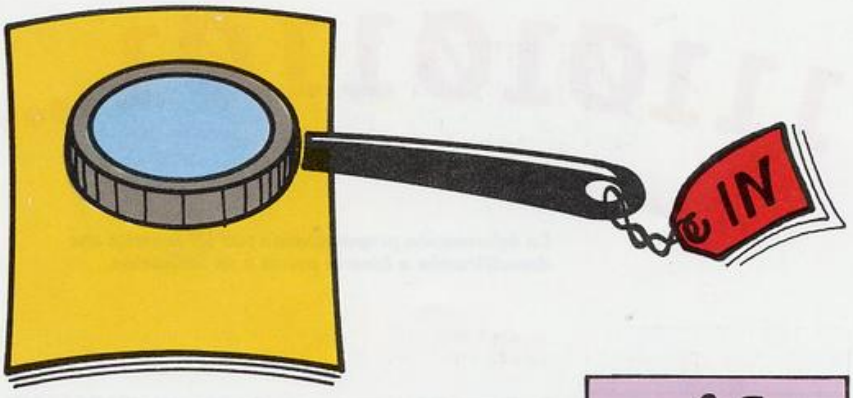
Llegados a este punto, es importante hacer una matización: podemos estar tentados de efectuar una pregunta mucho más simple, en el caso de que únicamente deseemos averiguar si se encuentra o no pulsada alguna de las teclas de una media fila concreta.

En teoría, bastaría con preguntar por el **PEEK** de la dirección del port correspondiente para saber si alguna tecla de las que deseamos investigar ha sido pulsada. En este caso, si recibimos un 255 como respuesta, no cabe duda de que se cumple esta condición.

Sin embargo, debemos ser en extremo cuidadosos, dado que algunas versiones de Spectrum no responderán de igual forma a esta pregunta (con un 255), a pesar de no estar pulsada tecla alguna en la media fila interrogada.

De hecho, aunque en la mayoría de las versiones los tres bits más significativos obtenidos del byte leído del port son tres unos, se da el caso de que, al menos en la versión 3 (*ISSUE 3*), el sexto bit resulta ser un cero.

En cualquier caso, este hecho no debe preocuparnos, puesto que al realizar nuestros programas para un ordenador concreto no nos importa que el **PEEK** del port de determinada media fila sea un 255 o un 191, ya que siempre la respuesta, cuando ninguna tecla se encuentre pulsada, será la misma.

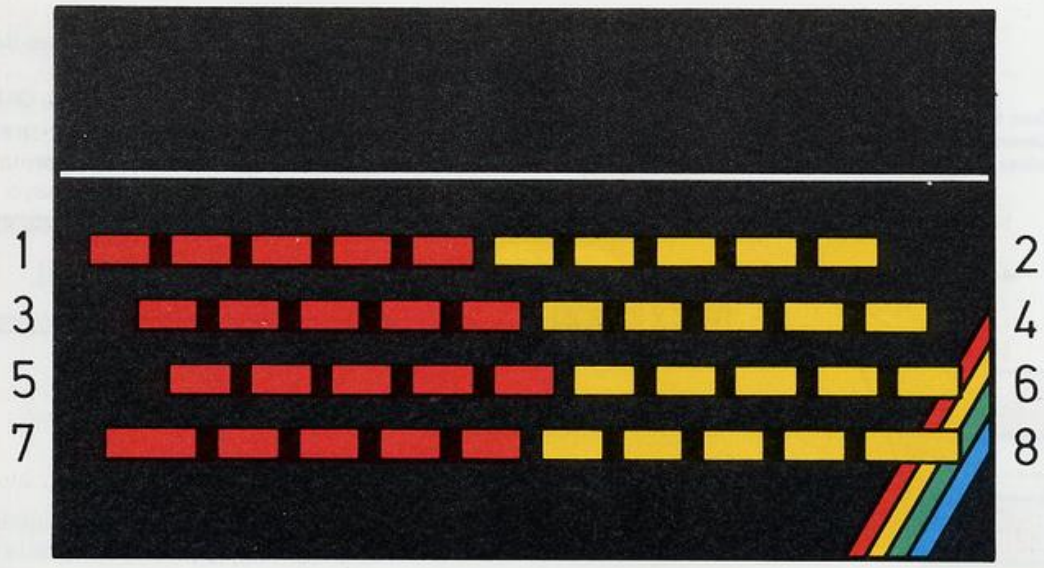


Mediante la función BASIC **IN** *podemos leer el teclado como cualquier dirección del port.*

Así pues, una vez que ya conocemos cuál es el byte que corresponde a cada tecla, basta con observar su estado lógico: si el bit es cero, la tecla está pulsada; mientras que si es uno, significa que la tecla está libre.

Avanzando con el ejemplo práctico, supongamos que el byte obtenido como resultado de la ejecución de **IN** 65022 es el número decimal 94. Recordemos que la secuencia de operaciones a realizar es:

A efectos de direccionamiento para captación de datos de los ports, el teclado se encuentra dividido en ocho medias filas de cinco teclas.



¡!

Los diálogos del ordenador con el mundo exterior se realizan a través de los accesos de entrada/salida *(Port Input/Output* o *PIO).*

*

La CPU puede direccionar (acceder para leer o escribir) cualquiera de las 65.536 posiciones de memoria.

11101011101...

La información proporcionada por **IN** *necesita una decodificación a binario previa a su utilización.*

¡!

La función **IN** lleva como parámetro la dirección del port, siendo su resultado el byte de datos que ocupa éste en el momento de ejecutar la función.

*

La comunicación entre la memoria del ordenador y su CPU se produce a través de los BUS: uno de direcciones (16 bits) y otro de datos (8 bits).

1. Pasar 94 a su expresión binaria (01011110).
2. Eliminar los tres bits más significativos (nos quedamos con: 11110).
3. Averiguar la posición de la tecla a estudiar en la media fila (**S** es la segunda tecla de su media fila, contando desde el exterior hacia el interior).
4. Averiguar el estado del bit correspondiente (uno, el segundo contando por la derecha).

Como conclusión, podemos decir que la tecla **S** no está pulsada, pero, por lo que hemos visto, la **A** sí lo está, y el resto de la media fila (**D**, **F** y **G**) están libres.

POSIBILIDADES DE OUT

Hemos podido comprobar la enorme utilidad de la función **IN** a la hora de hacer una exploración exhaustiva del teclado, en busca de más de una tecla pulsada simultáneamente.

Desgraciadamente, mucha menos utilidad es la que podemos encontrar en la sentencia inversa **OUT**. Esto es debido a que su misión principal es el control de periféricos externos, o la conexión del ordenador a dispositivos electrónicos diseñados por el propio usuario o adquiridos en el mercado.

Para estos fines, suele recurrirse a rutinas de manejo en lenguaje Ensamblador, puesto que, por sus características, permite una mayor velocidad de ejecución que el BASIC no puede dar. Por ello, no suele emplearse mucho esta sentencia, al menos como tal comando BASIC.

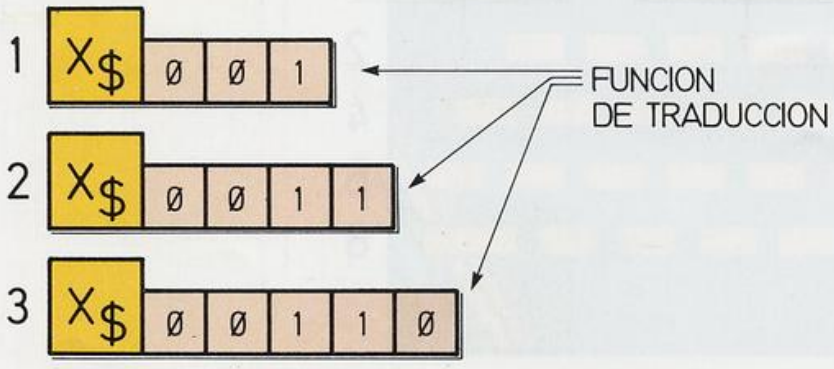
Sin embargo, **OUT** puede emplearse para controlar el altavoz o la impresora del ordenador. La dirección del port 254, por ejemplo, permite utilizar parte de sus bits para el control directo de las conexiones MIC y EAR, así como el color del contorno de la pantalla.

Otra posibilidad es el empleo de la dirección del port 251, que gestiona el manejo de la impresora en las operaciones de lectura y escritura. En cualquier caso, el control de la impresora es mucho más sencillo por medio de las sentencias BASIC convencionales.

OUT *puede emplearse para controlar el altavoz del ordenador.*

En la práctica, el empleo de la sentencia **OUT** se hace absolutamente innecesario, dado el gran repertorio de sentencias BASIC disponibles.

La línea 90 del programa decodificador es la que realiza propiamente la traducción, añadiendo un nuevo dígito a la cadena X$ por cada pasada.

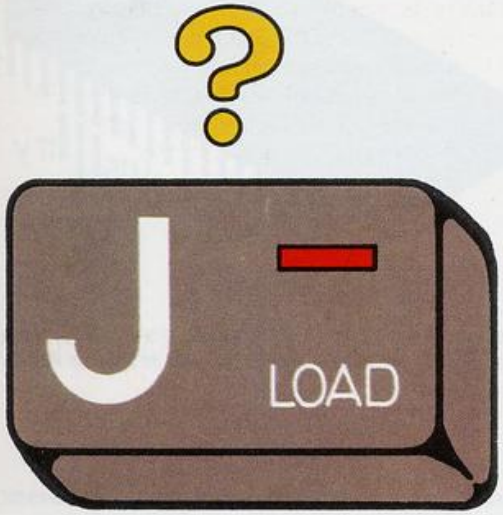


DESPLAZAMIENTO DE UN MOVIL

A modo de ejemplo, a continuación incluimos el listado de un programa, que desplaza por la pan-

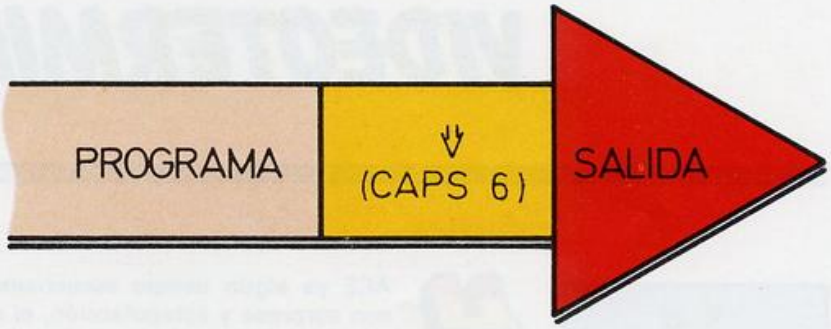
talla un círculo, teniendo en cuenta la pulsación de las teclas de la media fila 0-6. 6 se considera arriba, 7 abajo, 8 izquierda, 9 derecha y 0 fin de programa:

```
  10 REM - MOVIMIENTO
  20 LET F=11: LET C=16: GO SUB 180
  30 LET X=IN 61438: IF X=255 THEN  GO TO 30
  40 GO SUB 50: GO SUB 90: GO TO 30
  50 LET X$=""
  60 IF X<2 THEN  GO TO 80
  70 LET Y=INT (X/2): LET Z=X-2*Y: LET X$=STR$ Z+X$:
LET X=Y: GO TO 60
  80 LET X$="0000000"+STR$ X+X$: LET X$=X$(LEN X$-7 T
O ): RETURN
  90 IF X$(4)="0" THEN  LET F=F-1
 100 IF X$(5)="0" THEN  LET F=F+1
 110 IF X$(6)="0" THEN  LET C=C-1
 120 IF X$(7)="0" THEN  LET C=C+1
 130 IF X$(8)="0" THEN  GO TO 9999
 140 IF F<0 THEN  LET F=0
 150 IF F>21 THEN  LET F=21
 160 IF C>31 THEN  LET C=31
 170 IF C<0 THEN  LET C=0
 180 PRINT AT F,C;"O": RETURN
```



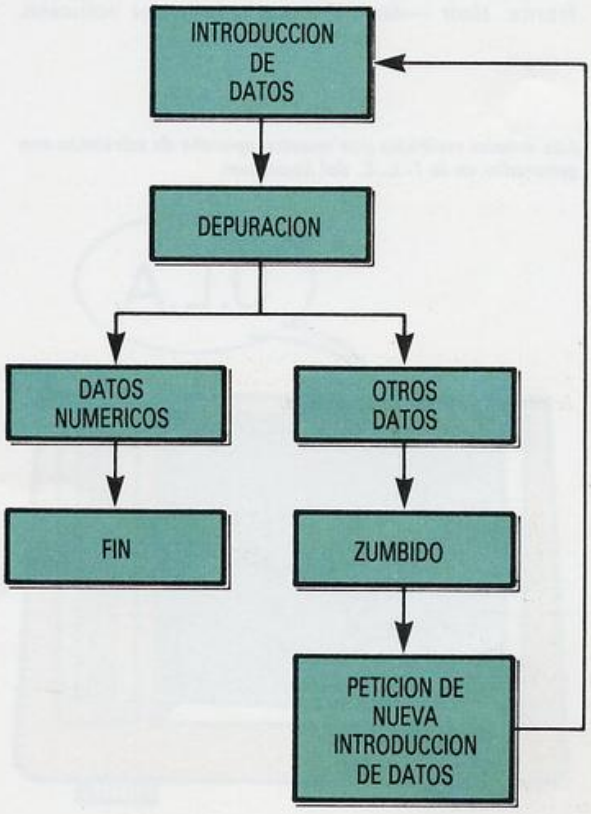
El estudio del resultado obtenido mediante **IN** *nos indicará el estado de una tecla.*

En la línea 20 se fijan los valores iniciales de las coordenadas del móvil (**F** y **C**) llamando a la subrutina de impresión en pantalla.
La línea 30 lee la media fila superior derecha del teclado a través de la dirección del port 61.438. Caso de no existir pulsación de ninguna tecla (**PEEK**=255), se efectúa un **GO TO** a la propia instrucción.
La línea 40 llama a la subrutina de decodificación del byte obtenido desde el port a binario, para continuar con la subrutina de evaluación de valores de fila y columna e impresión en pantalla. Hecho esto, se bifurca a la línea 30 para comenzar nuevamente el ciclo.
Las líneas 50 a 80 gestionan la decodificación del byte a binario, sobre la variable **X$**, de la forma descrita en el programa de ejemplo anterior.
Las líneas 90 a 130 efectúan el incremento o decremento de fila y columna, así como el control de fin de programa por la pulsación de la tecla **0**.
Las líneas 140 a 170 controlan que los valores adquiridos no superen los límites de la pantalla, en evitación de errores.
Por último, la línea 180 gestiona la impresión del móvil en pantalla, en las coordenadas especificadas por las variables **F** y **C**.



***Para abandonar el programa decodificador de* IN *utilizaremos la tecla* CURSOR ABAJO (CAPS SHIFT + 6).**

***Un sistema de depuración de errores emite un zumbido en las entradas erróneas al programa decodificador de* IN.**



¡!

La comunicación entre la CPU y los periféricos está regida por las sentencias BASIC **IN** y **OUT**, de gran similitud con las conocidas **PEEK** y **POKE**, respectivamente.

*

Existen 65.536 ports, a los cuales se puede acceder para establecer una comunicación entre cualquier periférico y el ordenador, aunque realmente sólo se emplean unos pocos.

VIDEOTERMINALES

¡!

La U.L.A. *(Uncommited Logic Array)* es el chip que se encarga, en el Spectrum, de la emisión de la señal de televisión.

*

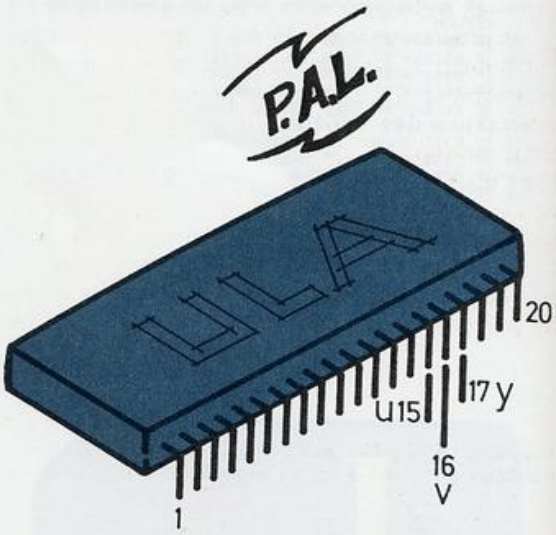
En el proceso de emisión de la imagen de televisión, la U.L.A. es auxiliada por otro componente del *hardware*, como, por ejemplo, el chip LM-1889 o el modulador de UHF.

HACE ya algún tiempo escuchamos, con sorpresa y estupefacción, el relato de una historia acaecida a un usuario del Spectrum, la cual nos aseguraron cierta por entero. Seguidamente os transcribimos, al pie de la letra, parte de la conversación mantenida con nuestro interlocutor:

«... irritado ante la intransigencia de su Spectrum, frenéticamente buscó un destornillador. Uno, dos..., cinco tornillos: ¡adiós a la garantía! No importa —pensó—, quería llegar, a toda costa, al fondo del asunto. Decenas de resistencias, transistores, chips, estaban ante sus ojos; sin resultado, intentó descubrir el posible camino seguido por la imagen de televisión, "pinchando" con la punta del destornillador cada uno de los contactos del circuito.»

Nuestro asombro iba en aumento; pero dejemos que acabe la narración: «... el daño —continuó relatando— ya estaba hecho. Su micro no respondía a las órdenes que le daba a través del teclado; los intentos por reanimarlo resultaron vanos. Repentinamente, un sentimiento de culpabilidad le invadió, mientras un sudor frío le cubría la frente. Huir —decició— sería la mejor solución.

Las señales recibidas por nuestro aparato de televisión son generadas en la U.L.A. del Spectrum.



A través de los pines 15, 16 y 17 (U, V e Y) se envían la totalidad de las señales requeridas por el sistema P.A.L.

Pero el agudo grito de las sirenas le hizo desistir: estaba rodeado; tan sólo le quedaba entregarse y cumplir condena por su horrible crimen.»

Y esto fue todo. Tras un laborioso proceso de investigación, dimos con el protagonista del relato en un oscuro centro de rehabilitación y prevención de microadictos, donde hoy en día aún permanece. Allí nos confirmaron la fidelidad de los hechos, mientras nos pedían, por favor, que obviáramos nombres y lugares antes de publicarla. Antes de recurrir a medios tan drásticos, como los seguidos por tan particular individuo, conviene leer las siguientes líneas.

UN LARGO CAMINO

Las señales que maneja nuestra pantalla son generadas en la U.L.A. Nuevamente aparece este

importante chip, el cual, recordaremos, se encarga de proporcionar también los impulsos correspondientes a los unos y ceros que finalmente acababan en el casete; o, en proceso inverso, los interpreta, además de activar el pequeño minialtavoz interno.

Pues bien, a través de los pines 15, 16 y 17, denominados por Sinclair U, V, Y, respectivamente, se envían la totalidad de las señales requeridas en el sistema P.A.L. *(Phase Alternating Line,* línea con alternancia de fase), utilizado en nuestro país y en la mayoría de los europeos para las emisiones de televisión. Este maneja dos tipos de señales compuestas de color: R-Y *(Red-Yellow,* rojo-amarillo) y B-Y *(Blue-Yellow,* azul-amarillo). El pin 15 (U) envía la señal R-Y, el 16 (V) la B-Y, y a través del 17 (Y) circula la información referente al nivel de luminosidad o luminancia de cada punto de la pantalla, además de los impulsos de sincronismo encargados de indicar el haz catódico del televisor cuando debe desplazarse de izquierda a derecha (sincronismo de línea) y cuando debe hacerlo de arriba a abajo (sincronismo de cuadro). La frecuencia de los primeros es de 15.625 Hz., con objeto de conformar las 625 líneas que componen la pantalla.

Las señales R-Y y B-Y emitidas por la U.L.A. pasan, seguidamente, al circuito integrado LM-1889. Este tiene como misión principal combinar las señales anteriores, convirtiéndolas en una única denominada subportadora de color. Esta precisa unas condiciones de estabilidad muy altas, pues cualquier desplazamiento provocaría interferencias en la zona visible de frecuencias. Para ello, un cristal de cuarzo genera la frecuencia de la subportadora, la cual, en el caso del sistema P.A.L., está normalizada a 4,43361875 MHz. Por el pin número 13 del LM-1889 sale ya esta señal, modulada en fase.

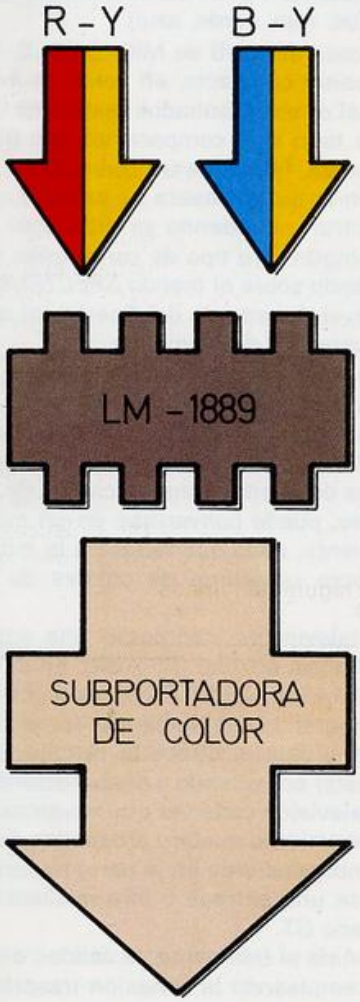
A partir de aquí, queda tan sólo mezclar la señal anterior con la procedente del pin 17 de la U.L.A. correspondiente a luminancia y sincronismos. Un transistor colocado al efecto es el encargado de efectuar la suma, y a la salida de éste tenemos ya la señal de vídeo completa y compuesta.

A continuación, otro transistor se encarga de adaptar impedancia de entrada/salida, y la señal se duplica, yendo a parar, por una parte, al modulador de vídeo y, por otro lado, a una de las patillas (15B) del conector trasero de expansión, denominada VIDEO.

En el caso de aparatos de televisión, y a diferencia con los de radio, cada emisora no se identifica por la frecuencia, si no por el canal. Por supuesto, a cada uno de éstos le corresponde una determinada gama de frecuencias.

En principio, diremos que el espectro de frecuencias se divide en dos zonas denominadas VHF *(Very High Frecuency,* muy alta frecuencia) y UHF *(Ultra High Frecuency,* ultra alta frecuencia). Dentro del VHF están las bandas I (canales 2, 3 y 4) y III (canales del 5 al 11). En la zona de UHF están las bandas IV (canales 21 al 35) y V (canales 36 al 69).

Las señales R-Y y B-Y pasan al chip LM-1889, donde se combinan para convertirse en una señal única.



¡!

Los monitores monocromos, ya sean en fósforo verde, ámbar o gris, disminuyen considerablemente el cansancio visual con respecto a los de color.

*

El Spectrum sigue un sistema de emisión de imagen preparado para la norma P.A.L. (línea con alternancia de fase), que es la empleada en España y en la mayoría de los países europeos.

¡!

El principal inconveniente del monitor CUB de MICROVITEC reside en que el efecto aparente de la instrucción **BRIGHT** queda completamente anulado; por otra parte, su facilidad de conexión (a través del bus de expansión, como un interface cualquiera) y su gran calidad de imagen son condiciones dignas de tener en cuenta.

¡¡OJOS CANSADOS!!

* TENEMOS LA SOLUCION A SU PROBLEMA
* USE APARATOS CON PANTALLA DE FOSFORO VERDE O AMBAR

Una pantalla de fósforo verde o ámbar produce menos cansancio visual que el provocado por una de color.

El proceso de modulación consiste básicamente en sumar las señales de vídeo y radiofrecuencia, a la vez que se recorta la altura de la señal, de forma que la resultante sea apta para ser recibida por el televisor. De esta manera, el modulador de vídeo del Spectrum genera una frecuencia correspondiente al canal 36 de UHF (aproximadamente a los 590 MHz.).

La salida del modulador, aquélla en la cual insertamos el conector coaxial del cual parte el cable hacia el televisor, es de 75 ohmios. Esto obliga a utilizar cables apantallados de esta impedancia, y en el caso de disponer de un aparato de televisión con entrada de antena de 300 ohmios deberemos proveernos de un adaptador, que fácilmente encontraremos en el mercado.

CONEXION A UN MONITOR DE VIDEO

En páginas anteriores de esta misma sección tuvimos oportunidad de ver unas fotografías de la creación de una salida de vídeo, tomando la señal a la entrada del modulador. Tal y como hemos comentado anteriormente, esta señal es la denominada compuesta: suma de luminancias y sincronismos.

Ya sabemos que esta señal se encuentra duplicada en el pin 15B del conector trasero, y, en un intento de ser puristas, deberíamos proveernos del conocido adaptador de 56 patillas, empleado por la mayoría de los interfaces de uso con nuestro Spectrum y tomar la señal de allí. Pero seguramente resultaría excesivo, pues tan sólo debemos emplear dos de sus patillas (la 15B y la 6B, GND o *GrouND,* correspondiente a masa).

Un sencillo conector del tipo RCA y una perforación en la caja del Spectrum para insertarlo son la solución más económica, a la vez que práctica, para dejar nuestra salida de vídeo compuesta preparada para la conexión a un monitor.

Existen en el mercado otro tipo de monitores, los cuales toman directamente las señales primarias emitidas por la U.L.A. (R-Y, B-Y e Y) del *slot* trasero del Spectrum (pines 16B, 17B y 18B, correspondientes a Y, V y U, respectivamente), donde éstas se hallan duplicadas. Son los que proporcionan una mayor calidad y definición de imagen; se les denomina monitores de entrada RGB *(Red, Green, Blue,* rojo, verde, azul).

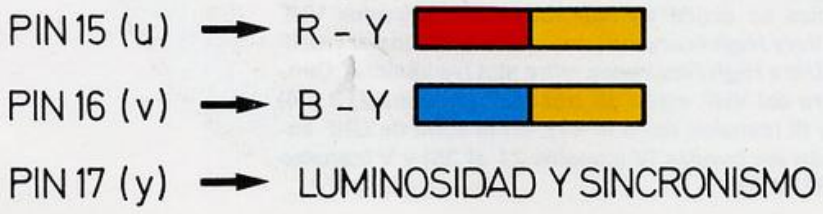
Tal es el caso del CUB de MICROVITEC. Se trata de un monitor compacto, en color, de 14 pulgadas, el cual ofrece resultados realmente excelentes, sobre todo si lo comparamos con un televisor doméstico. Mediante un conmutador TTL/SP situado en la parte trasera se selecciona la opción Spectrum, quedando ya dispuesto para su uso sin ningún otro tipo de conexiones especiales. Actuando sobre el mando *SPECTRUM INPUT LEVEL* (nivel de entrada del Spectrum) se consigue el ajuste fino de la imagen.

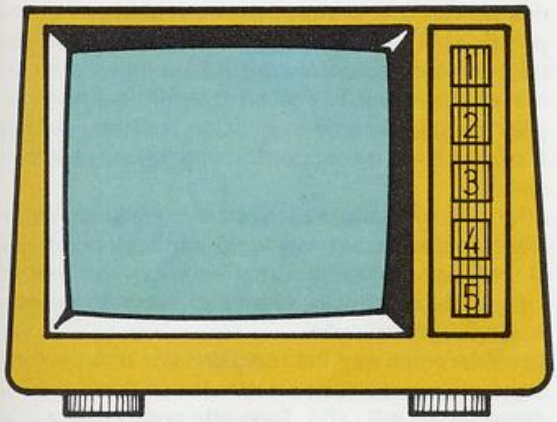
Sólo hemos encontrado un fallo en el equipo: no admite la posibilidad de brillo, es decir, una vez fijado cualquier color, al ejecutar la orden BRIGHT 1, la pantalla permanece tal y como estaba antes de la citada instrucción. Esto, lamentablemente, puede convertirse en un muy grave inconveniente, dado que reduce a la mitad el de por sí parco repertorio de colores de nuestro Spectrum.

Algunos televisores incorporan una entrada de radiofrecuencia auxiliar (RF AUX). En el caso del SONY KV-1430E (TRINITRON), está colocada en la parte frontal, lo cual facilita la conexión con el ordenador. Además, ofrece la ventaja de no tener que estar conectando y desconectando la antena de televisión cada vez que vayamos a poner en funcionamiento nuestro ordenador. Esta tiene entrada independiente en la parte trasera, seleccionándose una entrada u otra mediante un botón marcado GT.

Según señala el fabricante, la calidad de imagen obtenida empleando la conexión trasera o la RF

A los pines 15, 16 y 17 corresponde la emisión de las señales R-Y, B-Y y luminosidad y sincronismo.





En el caso de aparatos de televisión cada emisora se identifica por un canal.

es prácticamente la misma, aunque la RF limita la entrada a equipos que suministren una señal entre los canales 2 a 12 de VHF o 21 a 68 de UHF. Resulta evidente, cómo a medida que profundizamos en las posibilidades gráficas de nuestro micro, que la calidad proporcionada por un televisor convencional deja bastante que desear, sobre todo si pasamos un respetable número de horas delante de él.

Pero el uso de éstos está justificado en función de la facilidad de acceso a ellos, es decir, prácticamente todos disponemos en nuestros hogares de un aparato, ya sea en color o blanco y negro, capaz de sernos útil, desde el primer momento, sin necesidad de realizar un desembolso adicional como el que supone la adquisición de un videomonitor.

Pero si nos revelamos como plusmarquistas en la prueba de resistencia manejando nuestro micro, o somos auténticos «estilistas» confeccionando la presentación gráfica de nuestros programas, pronto echaremos de menos la calidad, nitidez, ausencia de parpadeos, etc., obtenida manejando monitores.

No obstante, no todos estos equipos son iguales. Con ello queremos dejar claro cómo la elección debe ir encaminada hacia aquel que se ajuste mejor a nuestras necesidades. Por ello, quien piense que un monitor de color es mejor que otro monocromo, es aconsejable que vaya desechando la idea. Imaginemos aplicaciones del tipo del tratamiento de textos o controles de producción, nóminas, etc.; en todos ellos la presencia de colorines puede que resulte muy espectacular, pero la auténtica finalidad del ordenador, y de uno de estos programas, es proporcionarnos datos y resultados de gran importancia práctica, aunque todavía existe quien se pregunta cuál es el color de los números y las letras.

Pensemos, además, cómo una pantalla de fósforo verde o ámbar produce, desde luego, mucho menos cansancio visual del provocado por una en color. Incluso en blanco y negro, lo anterior queda plenamente confirmado. Pues bien, como vemos, lo mejor es efectuar un análisis de nuestras necesidades reales y luego, entre la oferta existente, evaluar los siguientes factores:

— COLOR: Entre los monocromos existen varias opciones (verde, ámbar, gris). En los de color, considerar si son capaces de representar los atributos: vídeo inverso, parpadeo, brillo.

— NUMERO DE CARACTERES SOBRE LA PANTALLA: Queda determinado por el número de filas por columnas. Lo habitual son 1.920, correspondientes a 24 filas de 80 caracteres. No quiere ello decir que al conectar nuestro Spectrum a uno de estos equipos el formato de pantalla vaya a cambiar (recordemos que éste es de 24×32, fijado en la R.O.M.), si no que el monitor sería capaz de representar con total nitidez dicho formato.

— RESOLUCION EN MODO GRAFICO: Al igual que en el apartado anterior, es erróneo pensar que podamos modificar los límites para los comandos PLOT y DRAW. La resolución es el número de puntos que el monitor puede representar. Por ejemplo, en el caso del CUB 1431, es de 452×585 puntos.

— TAMAÑO: Suele oscilar entre las 9 y 20 pulgadas, aunque lo habitual, no obstante, está entre las 12 y 14.

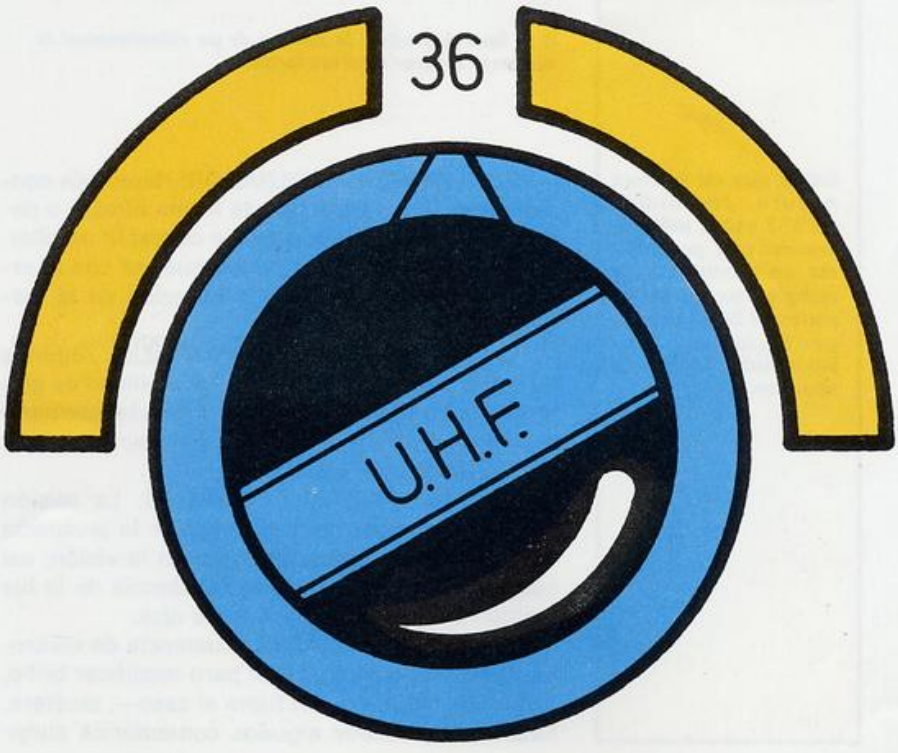
¡!

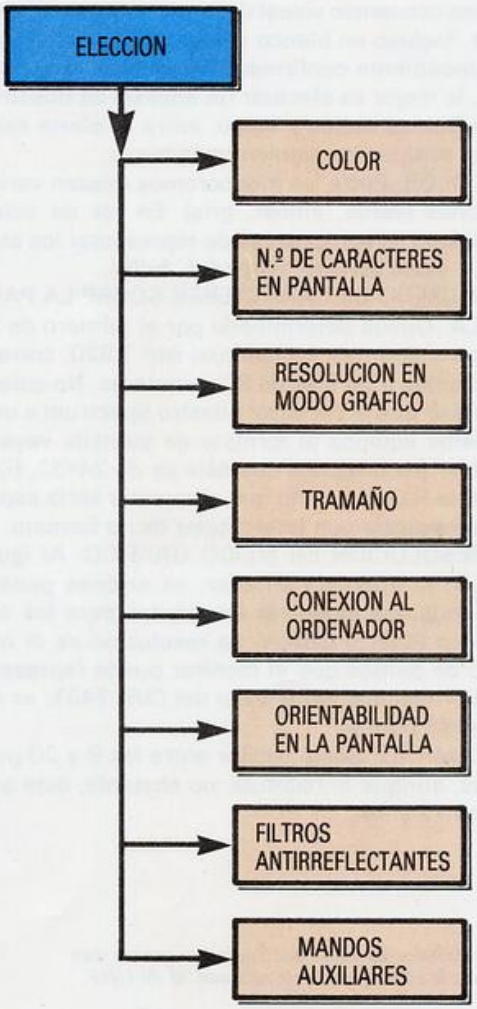
El Spectrum emite la señal de televisión modulada en el canal 36 de UHF.

*

En honor al físico alemán Roetgen, descubridor de los rayos X, la unidad de medida de las emisiones radioactivas es el roetgen.

El modulador de vídeo del Spectrum genera una frecuencia correspondiente al canal 36 de UHF.





A la hora de realizar la compra de un videoterminal es necesario evaluar diversos factores.

¡!

Saber que de manera natural recibimos 13.333 veces más radiación que por utilizar un monitor, aleja completamente de nosotros el fantasma legendario de la radioactividad de estos aparatos.

— CONEXION AL ORDENADOR: Hemos de considerar si ésta se efectúa de forma directa o debemos practicar algún tipo de conexión auxiliar. Los siguientes factores tienen que ver con la ergonomía; es decir, con la adaptación de la máquina al hombre:

— ORIENTABILIDAD DE LA PANTALLA: Algunos de estos terminales ofrecen la posibilidad de giro tanto horizontal como vertical. Además, permiten regular la altura y el grado de inclinación, según las exigencias del usuario.

— FILTROS ANTIRREFLECTANTES: La misión principal de éstos es la de impedir la profusión de reflejos, los cuales dificultarían la visión, así como disminuir el grado de incidencia de la luz emitida por la pantalla sobre los ojos.

— MANDOS AUXILIARES: Existencia de controles sobre los cuales actuar para modificar brillo, contraste, volumen —si fuera el caso—, etcétera.

Finalmente, aclarar algunos comentarios surgidos en torno a la supuesta nocividad provocada por la emisión de radiaciones procedentes del tubo de rayos catódicos del videomonitor. La norma española (UNE) fija en 0,5 mR/h (miliRoetgen/hora) la tasa de exposición máxima, medida a cinco centímetros, para los aparatos electrónicos.

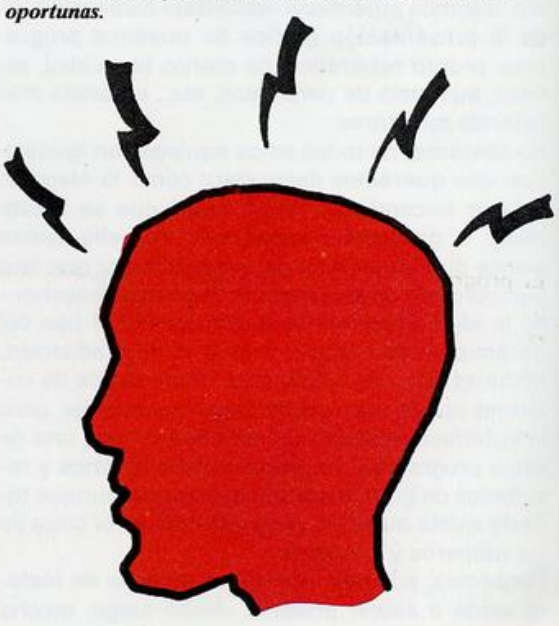
Informes confeccionados por el consejo de seguridad nuclear muestran los resultados obtenidos al analizar las radiaciones emitidas por varias marcas de monitores, siendo el máximo detectado de 0,006 mR/año.

Consideremos que las recibidas por una persona expuesta a radiaciones naturales es de aproximadamente 80 mR/año. Todo ello contribuye a alejar cualquier intento de sensacionalismo o descalificación de los videoterminales, dado que no sólo nos encontramos ante una magnitud 730.000 veces inferior a la admitida por la UNE sino 13.333 veces por debajo de las radiaciones recibidas por cualquiera de nosotros por el mero hecho de habitar en el Planeta Azul.

Los problemas de cansancio visual y la aparición de desagradables dolores de cabeza son de fácil solución siempre y cuando se tomen las medidas oportunas: elección de equipos de alta definición y posibilidad de ajuste de la intensidad de la imagen, de forma tal que ésta no impacte abusivamente sobre los ojos.

En resumen, a los microadictos, y a todo aquél que dedica un considerable número de horas a «plantarse» delante de su televisor, les aconsejamos que tomen las precauciones necesarias antes de verse expuestos a los problemas señalados anteriormente.

Los problemas de cansancio visual y aparición de dolores de cabeza tienen fácil solución si se toman las medidas oportunas.



MOTORA

MOTORA es un juego lleno de una acción trepidante que pone a prueba los reflejos del jugador, tal como si éste fuese el arriesgado piloto de un fuera-borda.
Pese a estar escrito en lenguaje BASIC, el programa presenta una velocidad más que aceptable debido a dos características fundamentales:

a) La minimización de las decisiones en el bucle principal.
b) La reducción de dicho bucle a una toma de datos y unas pocas sentencias **PRINT**, que son ejecutadas con gran rapidez por el BASIC de nuestro Spectrum.

El juego consiste en dirigir una lancha que aparece en las dos primeras líneas de pantalla a través de un río, evitando los escollos que aparecen en su camino. Las dificultades que surgen en el avance de la motora, dependen del nivel de dificultad que hayamos introducido al principio del programa.
A efectos de cronometraje, el programa indica el tiempo en segundos invertido en el recorrido, siempre y cuando hayamos conseguido llegar a la meta, indicada ésta por una serie de juncos al final del sinuoso recorrido fluvial.

EL PROGRAMA

El programa se basa en el desplazamiento por pantalla **(USR 3582)** de una matriz S$, la cual contiene el perfil del río más un cierto número de escollos, proporcionales al nivel de dificultad.
A la hora de introducir el programa en nuestro Spectrum es importante que sigamos detalladamente los siguientes pasos:

a) Introducir el programa principal y almacenarlo con auto-run en la línea 10 **(SAVE «MOTORA» LINE 10)**.
b) Ejecutar el programa cargador RIO. Este ini-

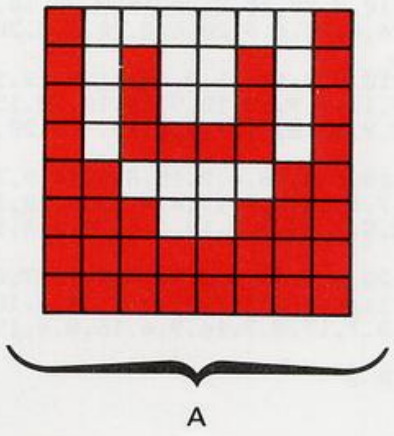


Gráfico definido para la parte trasera de la motora.

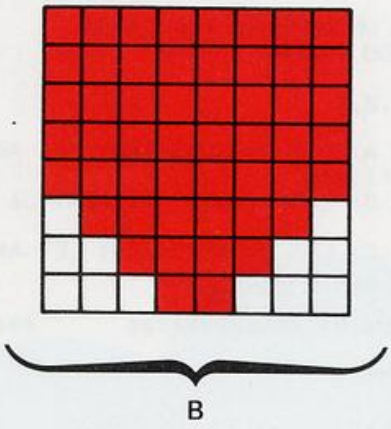


Gráfico para la parte delantera de la motora.

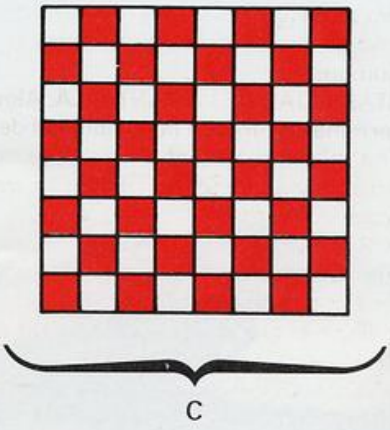


Gráfico para la selva alrededor del río.

¡!

Los caracteres subrayados corresponden a los gráficos definidos de usuario.

*

El programa principal (MOTORA) debemos almacenarlo con auto-run en la línea 10: **SAVE "MOTORA" LINE 10**. Posteriormente efectuar un **NEW** e introducir el programa cargador (RIO) como si fuera totalmente independiente.

*

El programa efectúa un **USR 3582** (ROM) para el Scrolling de la pantalla, lo cual simula el avance de la motora, aunque realmente se trata del desplazamiento del río.

PROGRAMA CARGADOR

```
  10 REM RIO $ J.M.MAYORAL SERRANO & F.LOPEZ MARTINEZ
  20 DATA 85,170,85,170,85,170,85,170
  30 DATA 85,42,213,41,90,133,42,85
  40 DATA 173,136,85,162,77,162,84,169
  50 DATA 161,90,165,90,213,74,173,82
  60 DATA 1,7,225,242,248,252,126,62
  70 DATA 64,128,5,141,159,61,125,124
  80 DATA 6,20,6,6,20,6,6,20,6,6,20,6,6,20,6,4
,20,6,3,20,9,2,20,10,2,20,10,2,20,10,2,20,10,
3,20,9,4,20,6,5,20,7,6,20,6,6,20,6,6,20,6,6,20,6,6,20
,6
  90 DATA 6,20,6,6,19,7,6,18,8,7,14,11,8,13,11,8,13,1
1,5,17,10,4,18,10,4,19,9,8,15,9,9,13,10,9,10,10,10,15
,7,11,14,7,11,14,7,11,13,8,9,15,8,7,17,8,6,19,7,6,20,
6,6,20,6,6,20,6
 100 DATA 6,20,6,6,20,6,7,19,6,8,18,6,9,15,8,9,14,9,7
,15,10,7,16,9,7,17,8,8,17,7,8,16,6,9,19,4,9,19,4,10,1
6,6,11,13,8,11,10,11,10,13,9,7,18,7,7,19,6,6,20,6,6,1
9,7,6,20,6
 110 DATA 6,20,6,6,20,6,6,20,6,7,18,7,8,17,7,9,16,7,8
,15,9,7,16,9,7,14,11,6,15,11,6,16,10,7,21,4,8,20,4,10
,17,5,10,17,5,8,17,7,8,16,8,7,17,8,7,16,9,6,16,8,6,19
,7,6,20,6
 120 INK 9: PAPER 2: BORDER 2
 130 CLS
 140 FOR I=146 TO 151
 150 FOR J=0 TO 7: READ A
 160 POKE USR CHR$ I+J,A
 170 NEXT J
 180 NEXT I
 190 PRINT AT 11,2;"ESPERA UN MOMENTO POR FAVOR."
 200 PRINT AT 13,8;"CARGANDO TABLA"
 210 DIM M$(110,32)
 220 FOR I=1 TO 86: READ A,B,C
 230 PRINT AT 16,14;I
 240 LET A$="": FOR J=1 TO A: LET D=RND*5+146: LET A$
=A$+CHR$ D: NEXT J: LET M$(I)=A$
 250 LET A$="": FOR J=1 TO B: LET A$=A$+" ": NEXT J:
LET M$(I,A+1 TO )=A$
 260 LET A$="": FOR J=1 TO C: LET D=RND*5+146: LET A$
=A$+CHR$ D: NEXT J: LET M$(I,A+B+1 TO )=A$
 270 NEXT I
 280 LET M$(89)=CHR$ 16+CHR$ 0+"############      ###
########"
 290 CLS : SAVE "M$" DATA M$()
```

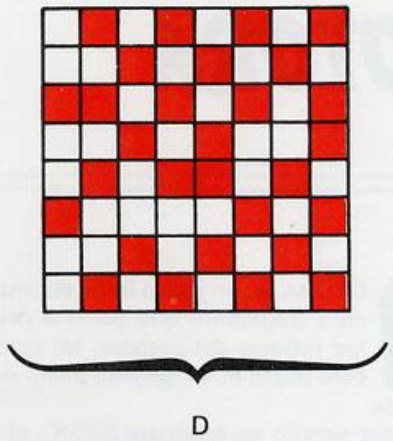
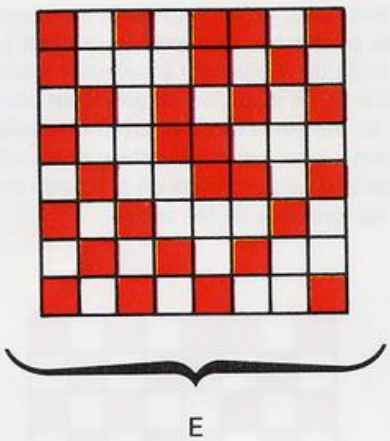


Gráfico para la selva alrededor del río.

cializa la matriz madre M$ con el perfil del río, de la cual se saca una copia para la matriz S$. A su término aparecerá el mensaje *Start tape, then press any key;* situando la cinta a continuación del programa principal ya almacenado, efectuaremos la nueva grabación.

Como podemos observar, la importancia del programa cargador RIO no reside en sí mismo, sino en la tabla M$, de cuya generación se encarga, y que es la información que realmente almacenamos a continuación del programa principal.
Dado que la generación del perfil del río por el programa cargador es aleatoria, no estaría de más que conserváramos una copia de dicho programa, con el fin de que, una vez hayamos llegado a dominar el río obtenido, podamos crear un nuevo cauce sobre el cual poner a prueba nuestros reflejos.



E

Gráfico para la selva alrededor del río.

NAVEGANDO

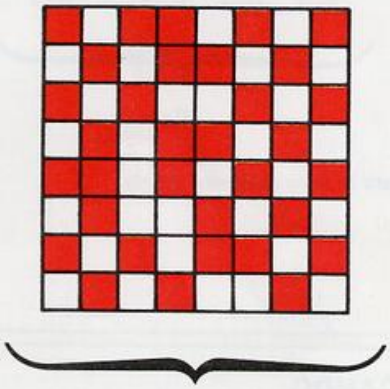
Si por alguna razón el programa principal se interrumpiera por efecto de una acción sobre la tecla **BREAK,** éste puede relanzarse con **GO TO 60** o, simplemente, con **CONTINUE,** pero nunca con **RUN,** puesto que se borraría la tabla M$, desapareciendo por tanto las turbulentas aguas escenario de nuestra aventura.
Los controles de la lancha motora son los siguientes:

IZQUIERDA..... 1
DERECHA..... 2
FRENO..... 0
ACELERADOR..... CUALQUIER OTRA TECLA

Hemos de tener en cuenta que la lancha es tanto más difícilmente controlable cuanto mayor velocidad adquiere, y que llegado el momento en que ésta no puede ser aumentada, a cada pulsación de la tecla de acelerado (cualquier tecla distinta de 1, 2 ó 0), se escucha un pitido de advertencia. Sobre la introducción del programa sólo hemos de advertir que, como es habitual, los caracteres que aparecen subrayados en el listado corresponden a los gráficos de las teclas.

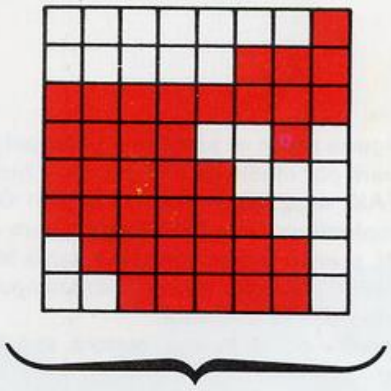
PROGRAMA PRINCIPAL

```
  10 REM ************************
  20 REM * J.M.MAYORAL SERRANO *
  30 REM * & F. LOPEZ MARTINEZ *
  40 REM ************************
  50 LOAD "M$" DATA M$(): POKE 23658,8
  60 DATA 129,165,165,189,195,231,255,255
  70 DATA 255,255,255,255,255,126,60,24
  80 DATA 85,170,85,170,85,170,85,170
  90 DATA 85,42,213,41,90,133,42,85
 100 DATA 173,138,85,152,77,162,84,169
 110 DATA 181,90,165,90,213,74,173,82
 120 DATA 1,7,255,242,248,252,126,62
 130 DATA 64,128,5,141,159,61,125,124
 140 BORDER 5: PAPER 5: INK 0: CLS
 150 FOR I=144 TO 151
 160 FOR J=0 TO 7: READ A
 170 POKE USR CHR$ I+J,A: NEXT J: NEXT I
 180 DIM H(10): FOR I=1 TO 10: LET H(I)=1E38: NEXT J
 190 DIM S$(110,32)
 200 FOR I=1 TO 110: LET S$(I)=M$(I): NEXT I
 210 INPUT "DIFICULTAD?";S: IF S<1 OR S>10 THEN  GO T
O 210
 220 BORDER 4: FOR J=1 TO S
 230 LET A=INT (RND*86)+2: LET B=INT (RND*20)+6: IF S
$(A,B)<>" " THEN  GO TO 230
 240 LET S$(A-1,B)="E": LET S$(A,B-1)="D": LET S$(A,B
)="E": LET S$(A,B+1)="F": LET S$(A+1,B)="D"
 250 NEXT J
 260 LET F=16: LET B=27: LET D=3: POKE 23674,0: POKE
23673,0: POKE 23672,0
 270 LET Q=1
 280 IF INKEY$="" THEN  GO TO 330
 290 IF INKEY$="1" THEN  LET F=F-1: GO TO 330
 300 IF INKEY$="2" THEN  LET F=F+1: GO TO 330
 310 IF INKEY$="0" THEN  LET B=B+1: GO TO 330
 320 LET B=B-3: IF B<0 THEN  LET B=0: BEEP .25,0
 330 FOR I=1 TO B: NEXT I
 340 IF RND*60-B>0 THEN  LET F=F+(1-2*(RND>.5))
 350 IF SCREEN$ (2,F)<>" " THEN  GO TO 530
 360 LET I=USR 3582
 370 PRINT AT 0,F-2;"  "; PAPER 7; INK 2;"A"; PAPER 5
;" "; INK 2;TAB F;"B"
 380 PRINT AT 21,0; INK 4;S$(Q)
 390 LET Q=Q+1
 400 IF Q<86 THEN  GO TO 280
 410 IF Q>89 AND Q<111 THEN  GO TO 280
 420 LET D=D-1: IF D>0 THEN  GO TO 270
 430 IF NOT D THEN  LET Q=89: GO TO 280
 440 CLS
 450 LET C=PEEK 23674: LET B=PEEK 23673: LET A=PEEK 2
3672: LET D=(A+B*256+C*65536)/50
 460 IF D<H(S) THEN  LET H(S)=D: PRINT AT 9,0;"BATIO
EL RECORD EN SU DIFICULTAD": GO TO 480
 470 PRINT AT 9,3;"FELICIDADES! LO CONSIGUIO."
 480 PRINT AT 11,16-(16+LEN STR$ D)/2;"EMPLEO ";D;" S
EGUNDOS"
 490 PRINT AT 13,3;"DESEA INTENTARLO DE NUEVO"
 500 IF INKEY$="S" THEN  CLS : BORDER 5: GO TO 190
 510 IF INKEY$="N" THEN  STOP
 520 GO TO 500
 530 PRINT AT 0,F-2;"      ";TAB F; INK 2;"GH"
 540 FOR I=0 TO -10 STEP -1: BEEP .025,I: NEXT I
 550 PAUSE 100: PRINT AT 1,F;"  ": BORDER 5: GO TO 49
0
```



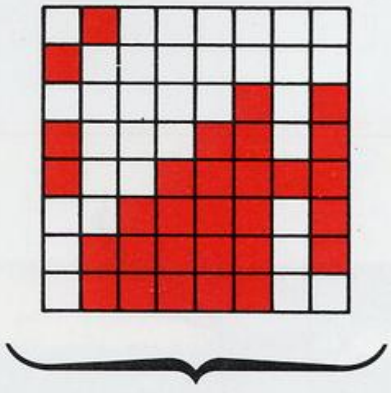
F

Gráfico para la selva alrededor del rio.



G

Gráfico de la parte izquierda de la motora al colisionar.



H

Gráfico de la parte derecha de la motora al colisionar.