37
150pts.

RUN

Enciclopedia Práctica del Spectrum

Nueva Lente/Ingelek

LAS VARIABLES DEL SISTEMA

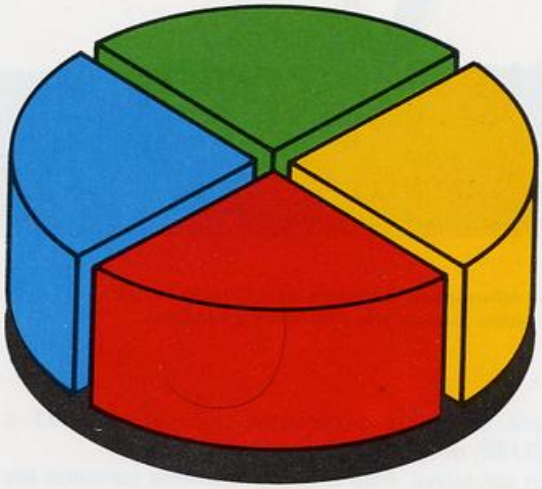
La ROM del Spectrum incorpora un amplio conjunto de rutinas escritas en código máquina del microprocesador Z80, las cuales conforman el Sistema Operativo y el Intérprete del lenguaje BASIC. A toda esta programación de base, incluida por el fabricante como parte integrante del Equipo cuando lo adquirimos, se la denomina *firmware*.

Si bien es cierto que la totalidad del *firmware* se encuentra almacenado en la ROM, no lo es menos que éste necesita de una pequeña parte de la memoria RAM para guardar cierta información de trabajo; cosa lógica si pensamos que la memoria ROM es de sólo lectura, como su nombre indica, no permitiendo que se escriba sobre ella.

En algunos ordenadores, la zona de trabajo del Sistema se encuentra justo al comienzo de la memoria RAM, más concretamente en el primer bloque de 256 *bytes* denominado también «página cero».

Para los no familiarizados con este término, diremos que la memoria podemos considerarla, a efectos de programación en Código Máquina, como dividida en bloques de datos de 1/4 de Kbyte denominados «páginas».

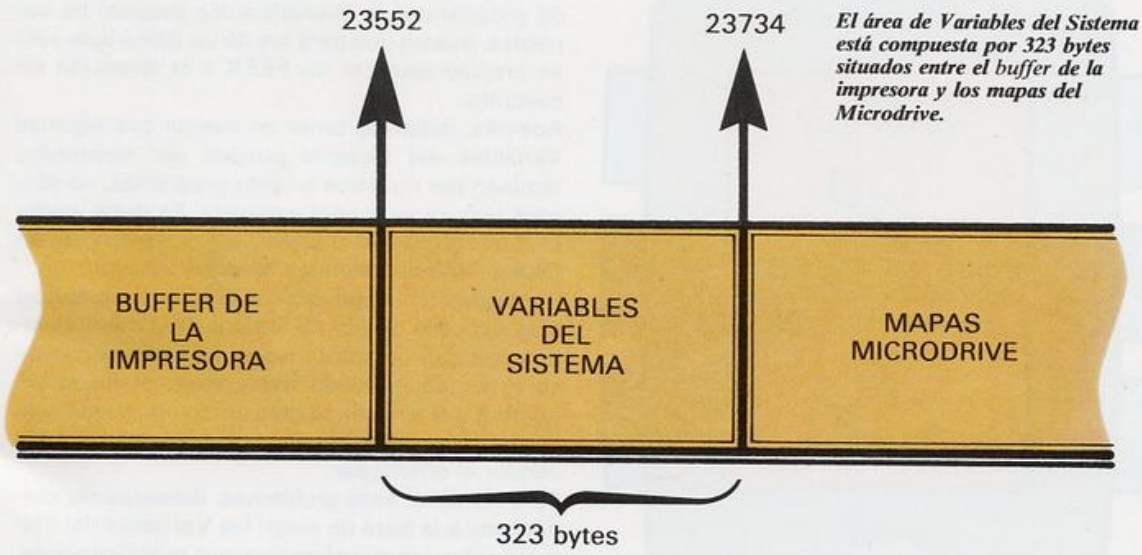
Esto es debido a que ello simplifica enormemente la tarea de codificación en Lenguaje Máquina, por existir un amplio repertorio de instrucciones dedicado a los direccionamientos a esta página,



El Kbyte se puede subdividir en cuatro partes, de 256 bytes cada una, denominadas «páginas».

con un importante ahorro de memoria e incluso, en algunos casos, instrucciones específicas que no pueden ejecutarse más allá de las fronteras de la «página cero».

Sin embargo, no es éste el caso del Spectrum, donde la información de uso general del Sistema se encuentra en un bloque distinto: el compren-



El área de Variables del Sistema está compuesta por 323 bytes situados entre el buffer de la impresora y los mapas del Microdrive.

¡!

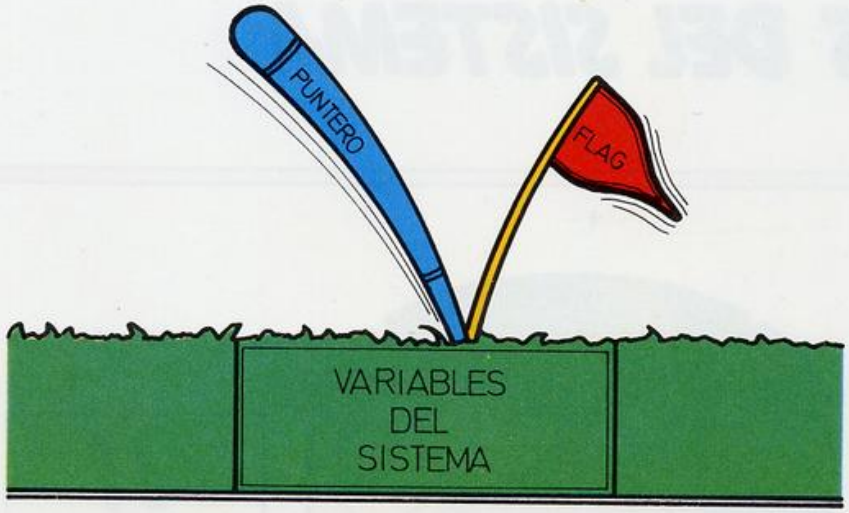
La alteración con un valor inadecuado de algunas de las Variables del Sistema puede hacer «caer el Sistema», nombre por el cual se conoce comúnmente a la pérdida del control que nos obliga a apagar y encender el ordenador.

*

Algunas Variables del Sistema pueden ser empleadas por nuestros programas no sólo para lectura sino para escritura.

*

El formato general de representación para las Variables del Sistema de dos bytes es el estándar de peso bajo y peso alto, o lo que es lo mismo, byte menos significativo y más significativo.



La información contenida en el área de Variables del Sistema corresponde a punteros y banderas.

de cada una de las particiones de la memoria RAM, como por ejemplo al inicio de la zona de texto BASIC, a la del área de variables, etc. En otras ocasiones, precisamente indican lo contrario, como por ejemplo la dirección del final físico de la memoria RAM (**P_RAMT**).
Otro de los contenidos favoritos de este tipo de variables es, sin duda, el de los indicadores o *flags*. Estos, se emplean para indicar al Sistema en todo momento el estado en que se encuentran determinados procesos, informando por ejemplo del «modo» en el cual se encuentra el cursor, etc.

EL FORMATO DE LAS VARIABLES

dido entre las direcciones decimales 23521 y 23733, ocupando un total de 323 bytes.
En definitiva, esta zona de memoria contiene las denominadas Variables del Sistema: conjuntos de uno o dos bytes con sentido propio, conocidos también por el nombre de «punteros» *(pointers)* y «banderas» *(flags)*.
Este nombre se debe a que su contenido suele hacer referencia a las direcciones de comienzo

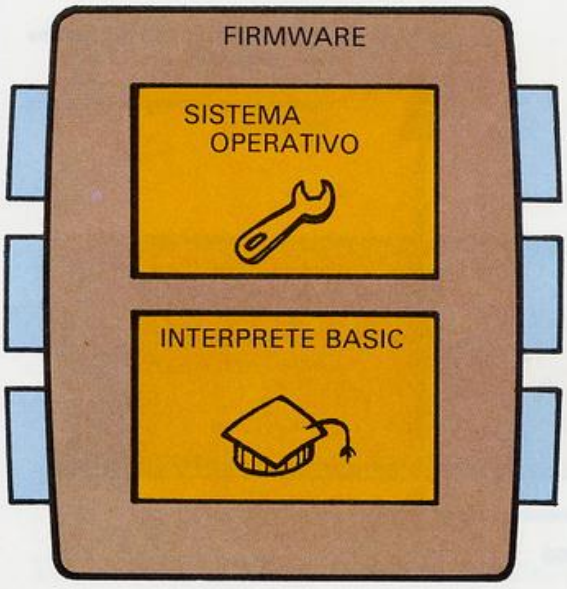
¡!

REPDEL controla el espacio de tiempo entre la pulsación de una tecla y la generación de la autorepetición.

*

VARS contiene la dirección de comienzo del área de variables BASIC. Puede utilizarse para colocar valores calculados por subrutinas en Lenguaje Máquina en el interior de las variables BASIC.

El firmware se compone de Sistema Operativo e Intérprete BASIC.



En cualquier caso, el formato general de representación para las Variables del Sistema de dos bytes es el estándar de peso bajo y peso alto, o lo que es lo mismo, byte menos significativo y más significativo.
Por este motivo, es necesario recurrir a una pequeña artimaña que nos permita interpretar en decimal el valor de las variables de dos bytes, como si de un doble **PEEK** se tratara.
Para ello, podemos definir una función:

```
DEF FN P(X)=PEEK X+256*PEEK (X+1)
```

De forma que suministrando la dirección de comienzo de la Variable de dos bytes (peso bajo) en **X**, podamos interpretar el valor de este byte y el siguiente, codificados en binario. Con ésto, queda solucionada la decodificación decimal de variables, puesto que para las de un único byte sólo es preciso ejecutar un **PEEK** a la dirección en concreto.
Además, debemos tener en cuenta que algunas Variables del Sistema pueden ser empleadas también por nuestros propios programas, no sólo para lectura sino para escritura. Es decir, podemos alterar desde el BASIC o rutinas propias en Código Máquina algunos de estos valores.
Por supuesto, la operación de escritura es mucho más delicada que la de lectura. De hecho, la alteración con un valor inadecuado de algunas de estas variables puede hacer «caer el Sistema», nombre por el cual se conoce comúnmente a la pérdida del control que nos obliga a apagar y encender el ordenador.
Para no tener esos problemas, debemos ser cuidadosos a la hora de elegir las Variables del Sistema sobre las que efectuaremos modificaciones,

asegurándonos previamente de que no sean de las descritas en el cuadro que aparece en el apéndice (o en el manual del equipo antiguo, no Plus), como reservadas al Sistema (las calificadas con una «X»).
Para alterar el valor de las variables del Sistema de un solo byte basta con ejecutar un **POKE**. Para hacerlo con las de dos, podemos recurrir a una rutina del tipo:

```
POKE  D,X-256*INT(X/256):POKE
      D+1,INT(X/256)
```

Donde **D** indica la dirección de comienzo de la Variable y **X** el valor decimal a asignar a ésta, codificado en binario.
Por último, diremos que además de las Variables del tipo puntero y bandera, existen también series de bytes contíguos cuyo cometido es el de servir de almacén temporal para determinados procesos: teclado, canales de comunicación, área de trabajo del calculador, etc.

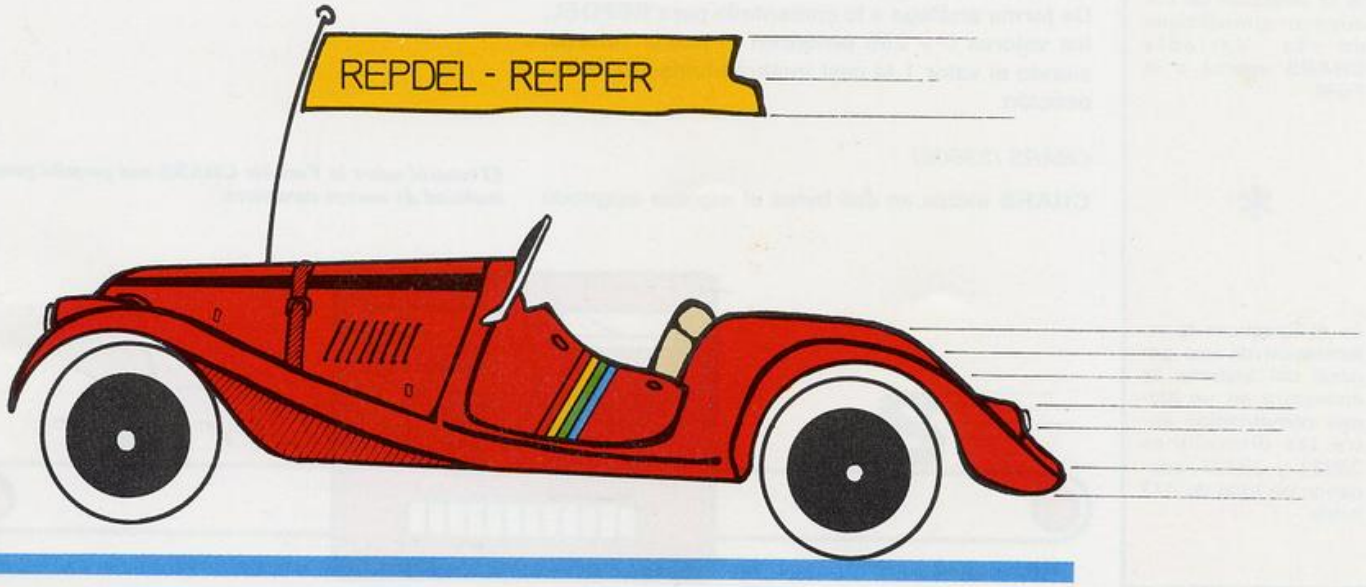
«Molestar» a algunas Variables del Sistema puede hacer que éste se «caiga».

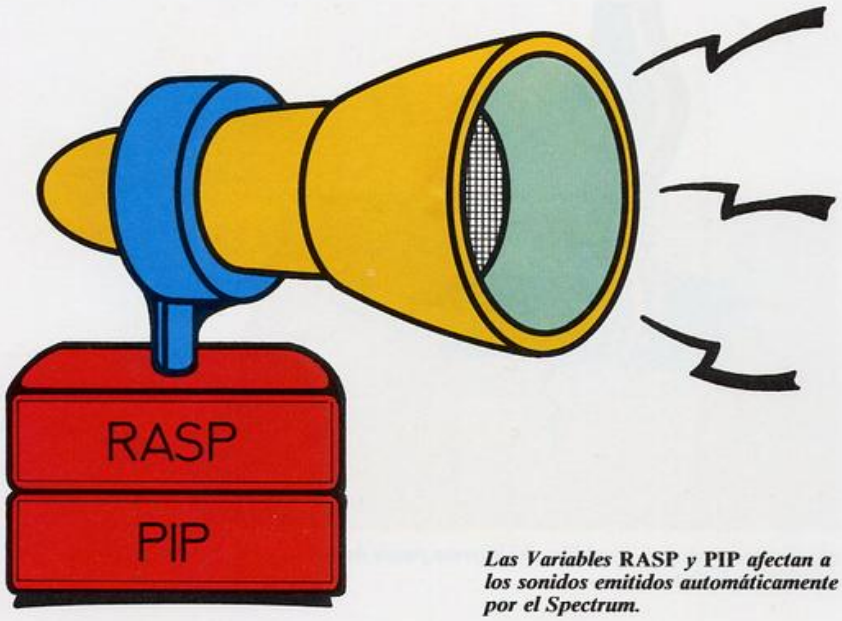
LAS DE MAYOR INTERES

REPDEL (23651)

Por todos es conocido el hecho de que la totalidad de las teclas del Spectrum están dotadas de autorepetición. Pues bien, habremos observado también que entre la pulsación de una tecla y la generación de la autorepetición pasa algún tiempo. Originalmente, este retardo viene ajustado a 35 cincuentavos de segundo, para permitir que no se produzca una autorepetición no deseada por el retardo excesivamente pequeño.
Sin embargo, en determinadas circunstancias puede interesarnos modificar este «tiempo de reacción» para hacerlo más rápido (por ejemplo

La modificación de los valores **REPDEL** *y* **REPPER** *pueden hacer «correr» a nuestro teclado bastante más de lo normal.*





Las Variables **RASP** *y* **PIP** *afectan a los sonidos emitidos automáticamente por el Spectrum.*

para un juego), o quizá todo lo contrario, impedir prácticamente que la autorepetición se produzca. Podemos modificar el contenido de esta Variable del Sistema por medio de la sentencia **POKE** a la dirección 23651 de cualquier número entre 0 y 255. Comprobaremos que los valores 0 y 255 se corresponden con el mayor intervalo de tiempo y el 1 al menor.

¡!

El juego de caracteres está grabado en ROM, por lo que inicialmente la dirección de comienzo almacenada en la Variable **CHARS** apunta a la ROM.

*

En el Spectrum la información de uso general del Sistema se encuentra en un bloque comprendido entre las direcciones 23521 y 23733, ocupando un total de 323 bytes.

REPPER (23562)

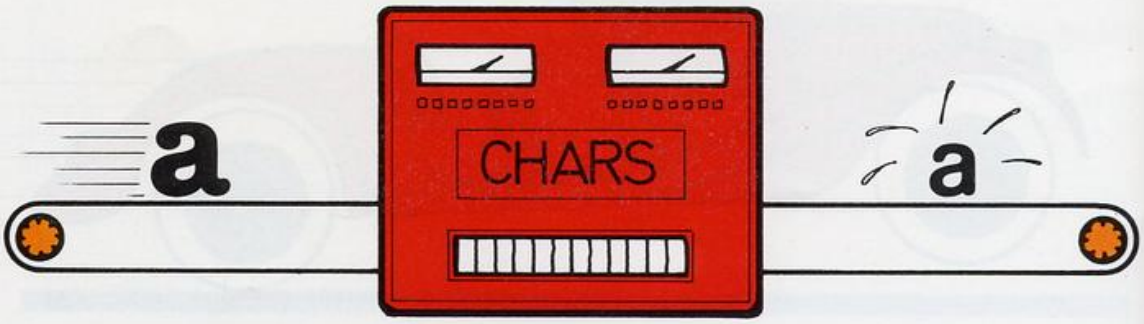
Del mismo modo que **REPDEL** influye sobre el tiempo de retardo antes de efectuar la repetición, **REPPER** lo hace con el tiempo que transcurre entre cada una de éstas, siendo el valor establecido por defecto al conectar el Spectrum de 5 cincuentavos de segundo.
De forma análoga a lo comentado para **REPDEL**, los valores 0 y 255 producen el mayor retardo, siendo el valor 1 la casi instantaneidad de la repetición.

CHARS (23606)

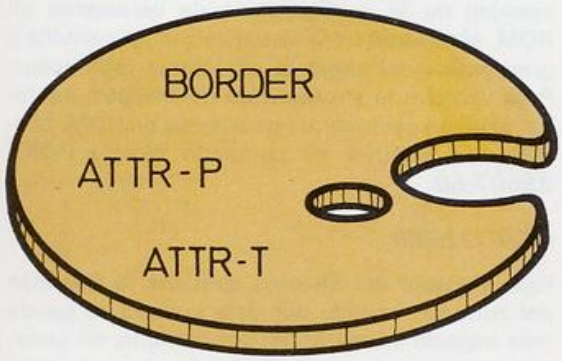
CHARS indica en dos bytes el espacio asignado al juego de caracteres menos 256 (ocupación de los caracteres 0 a 31).
Como sabemos, la definición de un carácter en el Spectrum requiere 8 bytes consecutivos, en los cuales se almacena la configuración bit a bit de cada una de las líneas que lo componen. Por otra parte, existen 128 configuraciones posibles de códigos de caracteres, las cuales, excluyendo los primeros 32 (0 a 31) que no tienen representación, ocupan un total de 768 byte (3/4 Kb.).
El juego de caracteres está grabado en ROM, por lo que inicialmente la dirección de comienzo almacenada en la Variable **CHARS** apunta a la ROM. Sin embargo, es posible alterar su valor para permitirnos definir los caracteres empleando para ello parte de la memoria RAM.
El siguiente programa nos traduce las configuraciones binarias bit a bit incluidas en la zona de definición de caracteres en ROM:

```
  10 REM - JUEGO CARACTERES (C) 1985 LOP
EZ MARTINEZ
  20 DEF FN P(X)=PEEK X+256*PEEK (X+1)
  30 BORDER 0: PAPER 0: CLS : PAPER 7
  40 LET C=31: LET X=23606
  50 FOR I=FN P(X)+256 TO FN P(X)+256+95
*8 STEP 8
  60 FOR J=I TO I+7
  70 LET X=PEEK J: GO SUB 150
  80 FOR K=1 TO 8: PRINT (" " AND X$(K)=
"1");(" " AND X$(K)="0");: NEXT K
  90 PRINT
 100 NEXT J
 110 LET C=C+1: PRINT : PRINT "CHR$ ";("
"+STR$ C)(LEN STR$ C-1 TO )
 120 BEEP .25,0: PAUSE 1: PRINT AT 0,0;
 130 NEXT I
 140 STOP
 150 LET X$=""
 160 LET X$=CHR$ (48+X-2*INT (X/2))+X$:
LET X=INT (X/2): IF X>1 THEN  GO TO 160
 170 LET X$=CHR$ (48+X)+X$: LET X$="0000
00"+X$: LET X$=X$(LEN X$-7 TO )
 180 RETURN
```

El control sobre la Variable **CHARS** *nos permite generar multitud de nuevos caracteres.*



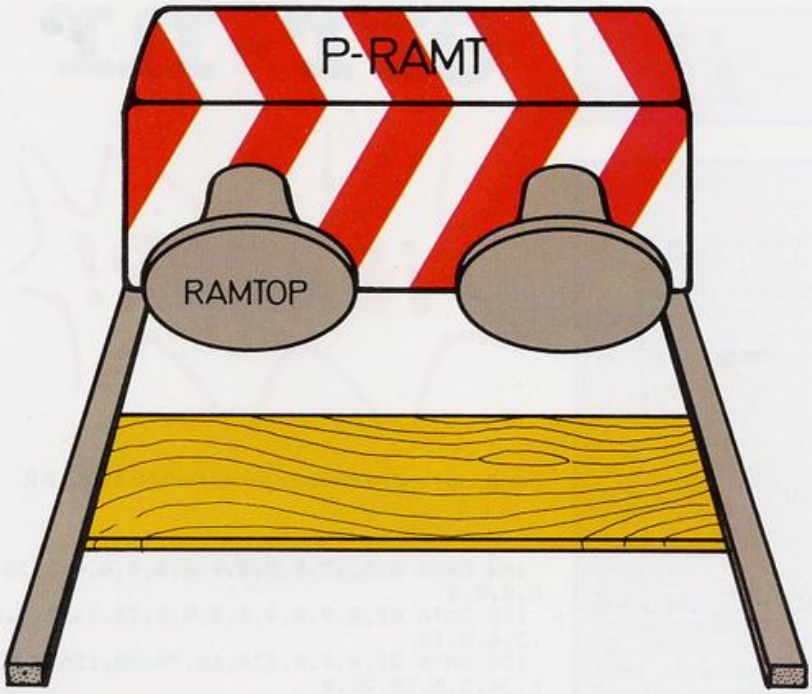
La línea 20 define la función **P(X)** (doble **PEEK**), la cual permite obtener el valor de cualquier Variable del Sistema especificando su dirección de comienzo en **X**.
Las líneas 30 y 40 se ocupan de las inicializaciones. 30 lo hace con el color del **BORDER** y el **PAPER**, y 40 asigna los valores de inicio a **C** y **X**, primer carácter a representar y ubicación del primer byte de la Variable **CHARS**, respectivamente.
En las líneas 50 a 130 se encuentra el ciclo principal de programa, donde se define un bucle **FOR NEXT** para valores de **I** entre la dirección contenida en la Variable **CHARS** más 256 y el final de la configuración del carácter número 95 (último representable).
Las líneas 60 a 100 definen un bucle interior **FOR NEXT** para los 8 siguientes valores de **J**, los cuales contienen la configuración binaria de cada carácter.



Las variables **BORDCR, ATTR_P, ATTR_T** *controlan los atributos empleados para la pantalla.*

Las líneas 110 a 130 se ocupan de la representación del código de carácter correspondiente y de producir un **BEEP** de control.
Por último, las líneas 150 a 180 podemos considerarlas como una subrutina de utilidad, la cual decodifica el valor decimal suministrado en la variable **X** a su formato binario de 8 bits en la variable X$.
La redefinición del juego completo de caracteres en RAM ocupa bastante memoria (3/4 Kbyte), sin embargo, existe la posibilidad más económica de utilizar únicamente el juego de caracteres en mayúsculas (472 bytes) correspondientes a 59 caracteres. Este programa efectúa la redefinición parcial:

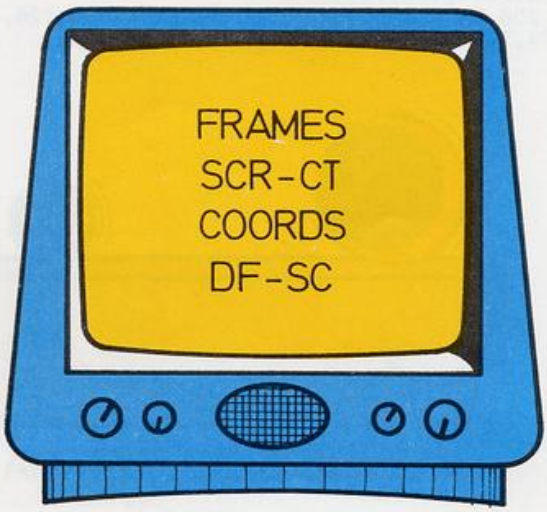
```
  10 REM - REDEFINICION CARACTERES (C) 1
985 LOPEZ MARTINEZ
  20 CLEAR 31487: POKE 23607,122
  30 FOR I=31488 TO 31958: READ X: POKE
I,X: NEXT I
```



La información sobre los topes de memoria corre a cargo de **RAM TOP** *y* **P_RAMT.**

```
  40 STOP
  50 DATA 0,0,0,0,0,0,0,0,0,16,16,16,16,
0,16,0,0,102
  60 DATA 102,0,0,0,0,0,0,36,126,36,36,1
26,36,0,0,8,62,40
  70 DATA 62,10,62,8,0,98,100,8,16,38,70
,0,0,16,40,16,42,68
  80 DATA 58,0,0,8,16,0,0,0,0,0,0,4,8,8,
8,8,4,0
  90 DATA 0,32,16,16,16,16,32,0,0,0,20,8
,62,8,20,0,0,0
```

Diferentes aspectos del manejo de pantalla corren a cargo de **FRAMES, SCR_CT, COORDS** *y* **DF_SZ.**



¡!

PIP permite seleccionar el tono generado por el Spectrum como confirmación de la pulsación de una tecla. Inicialmente se le asigna el valor cero (sin sonido).

*

BORDCR contiene el byte de atributos (**PAPER, INK, BRIGHT** y **FLASH**) empleado en la representación de la parte inferior de la pantalla (zona reservada al Sistema).

¡!

CHARS indica el comienzo del espacio asignado al juego de caracteres menos 256 (ocupación de los caracteres 0 a 31).

*

MODE indica el «modo» en que se encuentra el cursor. Su valor inicial es cero, tomando el valor 2 cuando el cursor está en el modo **G**.

*

UDG contiene la dirección de comienzo del primer gráfico definible por el usuario, coincidiendo con el resultado de la función **USR "A"**.

*

RAMTOP indica la dirección más alta de memoria utilizable por el BASIC. Si forzamos su valor directamente mediante **POKE**, en vez de utilizar **CLEAR <argumento numérico>**, conseguiremos una alteración del **RAM TOP** sin producir los efectos anejos a **CLEAR** (borrado de pantalla y variables, *stack* de subrutinas).

err-nr

Los errores son controlados por la Variable **ERR_NR.**

```
 100 DATA 8,8,62,8,8,0,0,0,0,0,0,8,8,16,
0,0,0,0
 110 DATA 62,0,0,0,0,0,0,0,0,24,24,0,0,0
,2,4,8,16
 120 DATA 32,0,0,0,126,66,70,70,126,0,0,
0,24,8,8,28,28,0
 130 DATA 0,0,126,6,126,64,126,0,0,0,124
,4,126,6,126,0,0,0
 140 DATA 96,102,126,6,6,0,0,0,126,64,12
6,6,126,0,0,0,124,64
 150 DATA 126,70,126,0,0,0,126,6,12,24,2
4,0,0,0,60,36,126,102
 160 DATA 126,0,0,0,126,66,126,6,6,0,0,0
,24,24,0,24,24,0
 170 DATA 0,0,16,0,0,16,16,32,128,0,4,8,
16,8,4,0,0,0
 180 DATA 0,62,0,62,0,0,0,0,48,24,12,24,
48,0,0,60,66,4
 190 DATA 8,0,8,0,0,60,74,86,94,64,60,0,
0,0,126,70,126,70
 200 DATA 70,0,0,0,124,98,124,98,124,0,0
,0,126,70,64,70,126,0
 210 DATA 0,0,126,70,70,70,126,0,0,0,126
,96,126,96,126,0,0,0
 220 DATA 126,96,126,96,96,0,0,0,126,64,
78,70,126,0,0,0,98,98
 230 DATA 126,98,98,0,0,0,24,24,24,24,24
,0,0,0,12,12,12,12
 240 DATA 60,0,0,0,100,100,126,70,70,0,0
,0,96,96,96,96,126,0
 250 DATA 0,0,126,86,86,86,86,0,0,0,126,
70,70,70,70,0,0,0
 260 DATA 126,98,98,98,126,0,0,0,126,98,
126,96,96,0,0,0,124,100
 270 DATA 100,100,126,0,0,0,126,98,124,7
0,70,0,0,0,126,96,126,6
 280 DATA 126,0,0,0,126,24,24,24,24,0,0,
0,98,98,98,98,126,0
 290 DATA 0,0,98,98,98,52,24,0,0,0,106,1
06,106,106,126,0,0,0
 300 DATA 98,98,60,70,70,0,0,0,98,98,126
,24,24,0,0,0,126,6
 310 DATA 24,96,126
```

En la línea 20 se protege la zona de memoria que albergará a los caracteres y se hace el correspondiente **POKE** a la variable **CHARS** de la nueva ubicación («página» 122). A continuación, se coloca la nueva codificación binaria de los caracteres en las direcciones 31488 a 31958.
Si después de ejecutar este programa cargamos y ejecutamos el anteriormente propuesto como ejemplo de la configuración de caracteres en ROM, obtendremos la visualización en pantalla a gran escala del juego de caracteres redefinidos. Para volver a la situación de normalidad, es decir, al juego estándar de caracteres en ROM, bastará con ejecutar en comando directo **POKE 23607,60.**

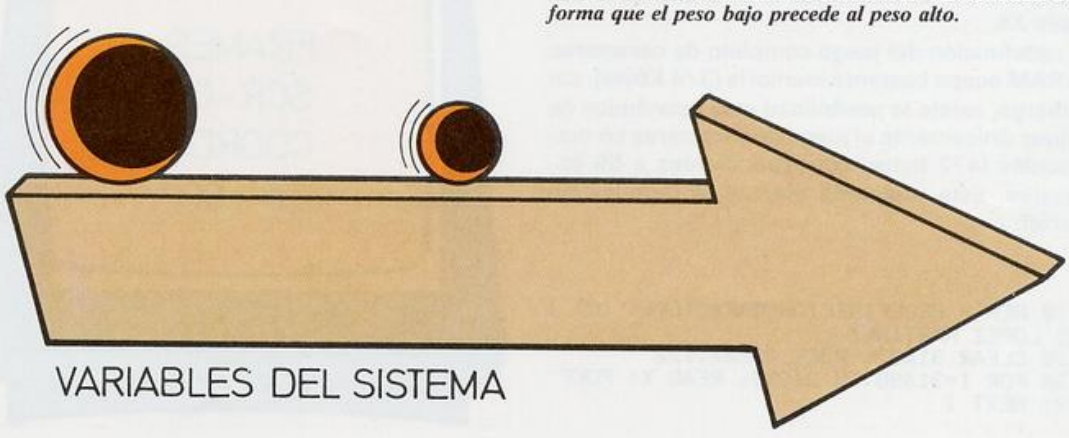
RASP (23608)

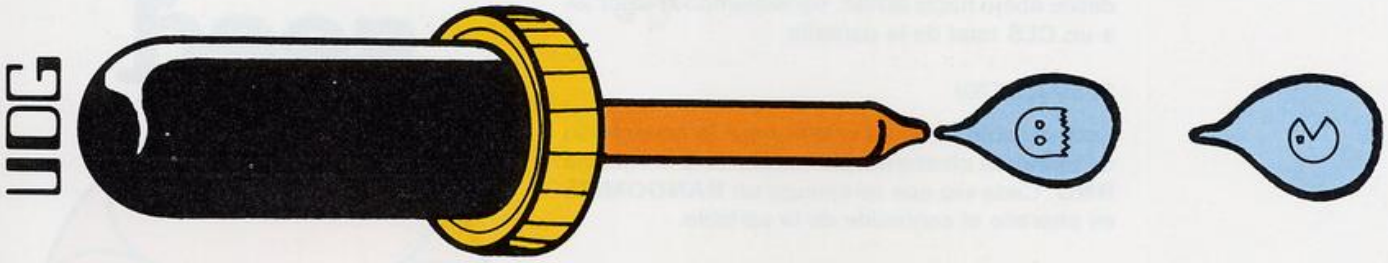
Esta Variable del Sistema contiene la duración del zumbido emitido por éste cuando no queda más espacio en memoria, por ejemplo; en general el zumbido de alarma, inicializado al conectar el Spectrum al valor 64.

PIP (23609)

PIP por su parte, permite seleccionar el tono generado por el Spectrum como confirmación de la pulsación de una tecla. Inicialmente se le asigna el valor cero (sin sonido) al encender el ordenador, valor que podemos modificar a nuestro gusto.

Las Variables del Sistema se encuentran almacenadas de forma que el peso bajo precede al peso alto.





Del control de la ubicación de los gráficos definidos se encarga **UDG.**

Cuanto más alto es el valor más lo es la duración del tono, sin embargo, los valores por encima de 16 empiezan a hacer notorio el retardo en la entrada de datos.

ERR_NR (23610)

Contiene el último código de mensaje informativo menos uno. Se inicializa con el valor 255, correspondiente por tanto al código cero (OK).

MODE (23617)

Esta Variable indica el «modo» en que se encuentra el cursor. Su valor inicial es cero, tomando el valor 2 cuando el cursor está en el modo **G**. Haciendo **POKE** del valor cero a esta dirección después de una entrada de datos por **INPUT**, nos aseguramos de que el cursor no quede en el modo gráfico para la siguiente entrada.

BORDCR (23624)

Esta variable contiene el byte de atributos **(PAPER, INK, BRIGHT** y **FLASH)** empleado en la representación de la parte inferior de la pantalla (zona reservada al Sistema). Haciendo un **POKE** con un valor adecuado, puede darse color a esta zona. También contiene el color del **BORDER** multiplicado por 8.

E_PPC (23625)

Contiene en dos bytes (bajo-alto) el número de línea en que se encuentra posicionado el cursor de programa. Puede modificarse por **POKE** para forzar un listado automático de determinado número de línea.

VARS (23627)

Contiene en dos bytes la dirección de comienzo del área de variables BASIC. Puede utilizarse para colocar valores calculados por subrutinas en Lenguaje Máquina en el interior de las variables BASIC.

PROG (23635)

Señala el comienzo del área de texto BASIC, siendo únicamente útil para investigar en el contenido de esta zona.

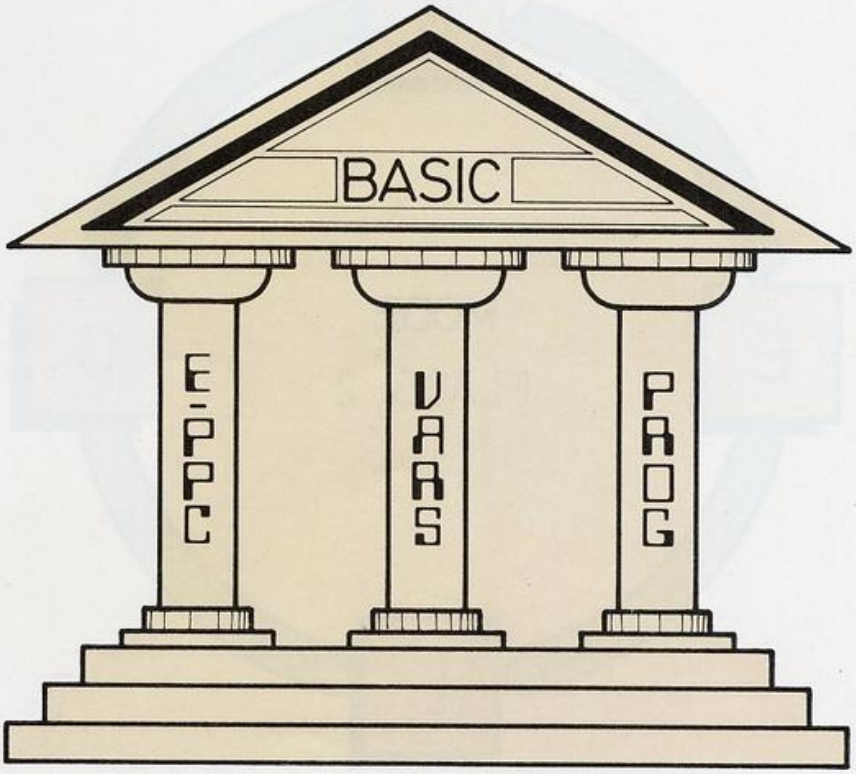
FLAGS2 (23658)

Puede usarse para pasar el cursor del modo **L** a **C** y viceversa. El valor 0 fuerza minúsculas y el valor 8 mayúsculas **(CAPS SHIFT).**

DF_SZ (23659)

Contiene el número de líneas reservado al Sistema en la parte interior de la pantalla (normalmente 2). Un **POKE** con valor cero a esta posición bloquea el Sistema, sin embargo, podemos efectuar un **POKE** entre 3 y 24 indicando el número de líneas que deseamos borrar, contando

Variables como **E_PPC, VARS** *o* **PROG** *influyen decisivamente en la estructura de los programas BASIC.*



desde abajo hacia arriba, equivaliendo el valor 24 a un **CLS** total de la pantalla.

SEED (23670)

Esta variable sirve de origen para la generación de números aleatorios por medio de la sentencia **RND.** Cada vez que se ejecuta un **RANDOMIZE** es alterado el contenido de la variable.

FRAMES (23672)

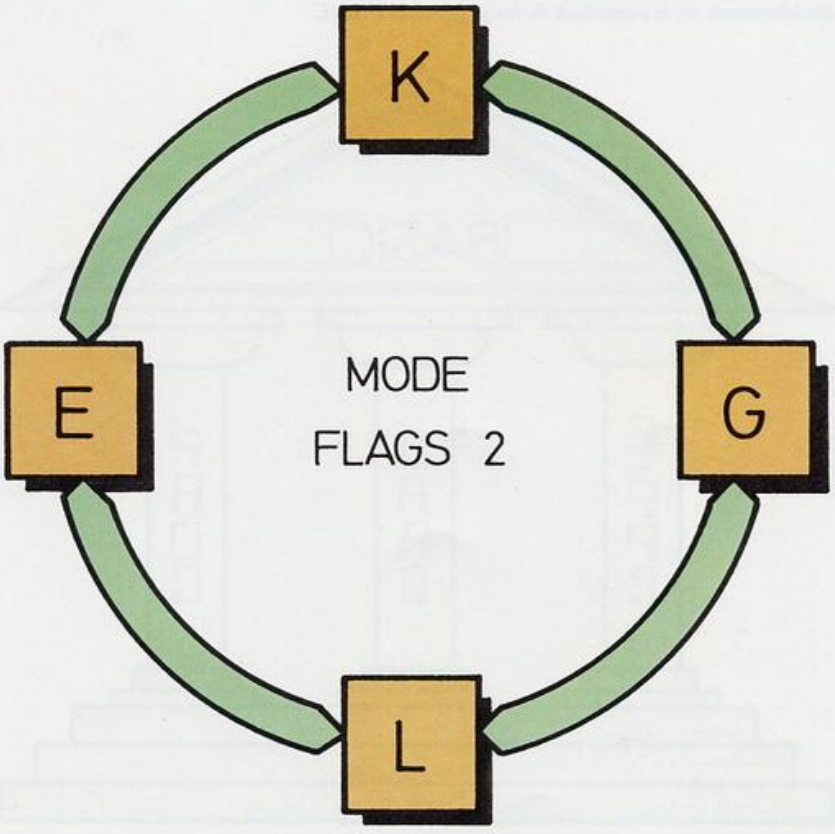
Contiene en tres bytes (de bajo a alto) un contador de «cuadros» de TV. La variable se incrementa automáticamente cada 20 milisegundos, tiempo que transcurre entre cada imagen de TV y la siguiente.

UDG (23675)

Contiene en dos bytes la dirección de comienzo del primer gráfico definible por el usuario, coincidiendo con el resultado de la función **USR "A".**

COORDS (23677)

El primer byte contiene la coordenada X (horizontal) del último punto trazado en alta resolución, y el segundo la coordenada Y (vertical) correspondiente.



MODE *y* **FLAGS 2** *son dos importantes Variables en el manejo del cursor.*

SEED *es una Variable vital para la extracción de números aleatorios.*

SCR_CT (23692)

Contiene el número de líneas que pueden imprimirse en la pantalla antes de que ésta se llene y sea preciso presentar el mensaje del Sistema *scroll?*
Inicialmente tiene valor 21. Puede ejecutarse un **POKE** de un número superior si se desea que pasen mayor número de líneas entre parada y parada.

ATTR_P (23693)

Contiene el byte de atributos permanentes de color en pantalla **(PAPER, INK, BRIGHT** y **FLASH),** determinado por las sentencias correspondientes en el formato genérico.

ATTR_T (23695)

Contiene el byte de atributos temporales determinado por las sentencias de color aplicadas junto con **PRINT,** en el modo temporal.

RAMTOP (23730)

Indica en dos bytes la dirección más alta de memoria utilizable por el BASIC. Coincidirá con el valor de **P_RAMT** siempre que no se efectúe una sentencia **CLEAR** con un valor más bajo, para reservar una zona de memoria.

P_RAMT (23732)

Contiene en dos bytes el total de memoria disponible del equipo. Califica, pues, al modelo de 16 Kbytes y al de 48 Kbytes.

CON BUENA LETRA

La selección de una impresora como complemento a nuestro sistema informático, resulta indispensable cuando destinamos nuestro Spectrum al desarrollo de aplicaciones que proporcionen un volumen considerable de datos, como salida del proceso ejecutado.
En principio, si no somos muy exigentes con el formato de salida, o estamos dispuestos a manejar listados kilométricos, podemos acoplar algunas de las pequeñas impresoras, analizadas en su momento, específicamente diseñadas para funcionar con el Spectrum.
Pero si necesitamos documentos de calidad donde poder combinar distintos tipos de letra, o representar diagramas en alta resolución gráfica, manejando hojas sueltas o papel continuo, la solución a nuestro problema estará servida por alguno de los periféricos o equipo similar a ellos, que comentamos en este capítulo y el próximo, dedicado a mostrar la oferta existente en el mercado, destinada a ser cubierta por las impresoras de tipo medio.

CONFIGURACION BASICA

Como norma general, las impresoras constan de los siguientes bloques principales:

— Fuente de alimentación.
— Circuito de control.
— Mecanismo impresor.

La fuente de alimentación es la encargada de transformar la corriente alterna entrante, en distintas tensiones continuas para facultar el funcionamiento de los componentes internos.
El mecanismo impresor consta de tres elemen-

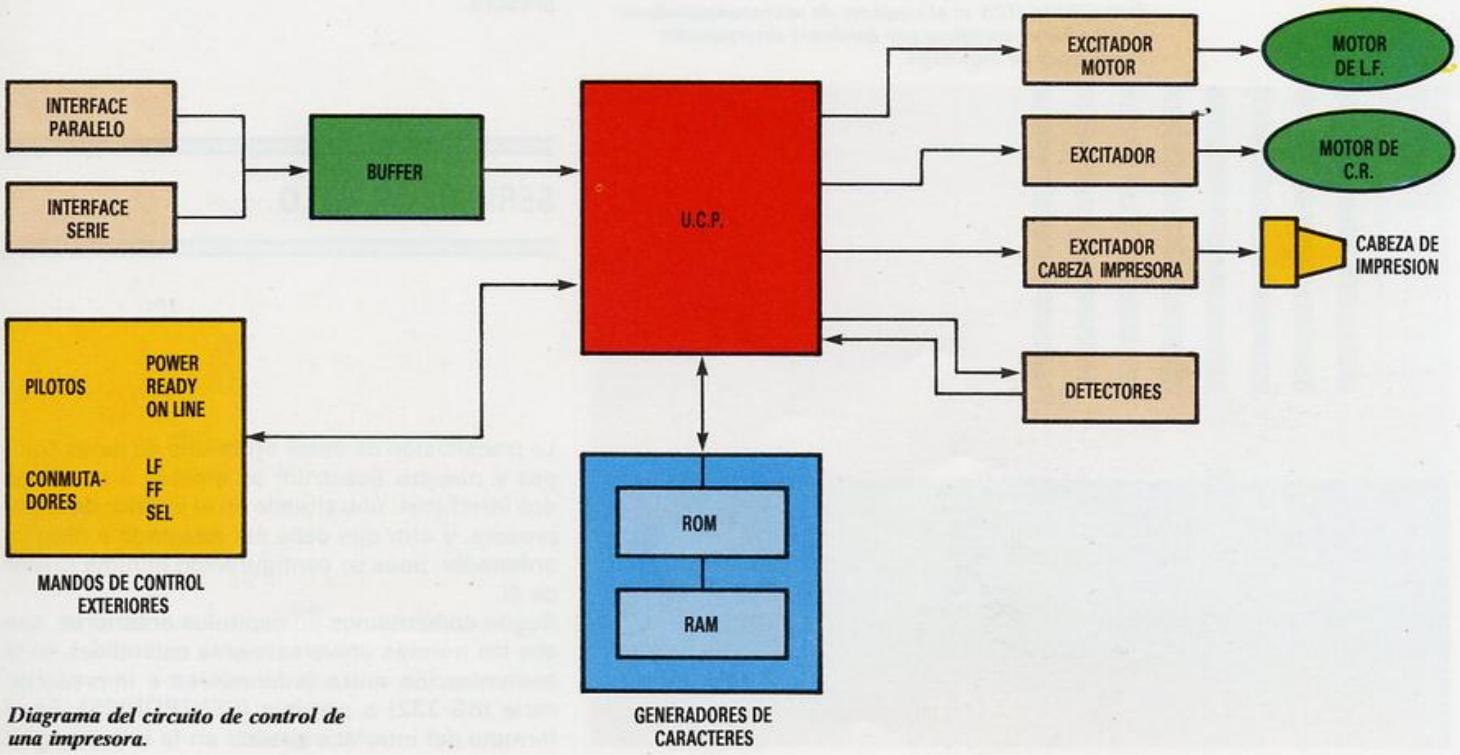
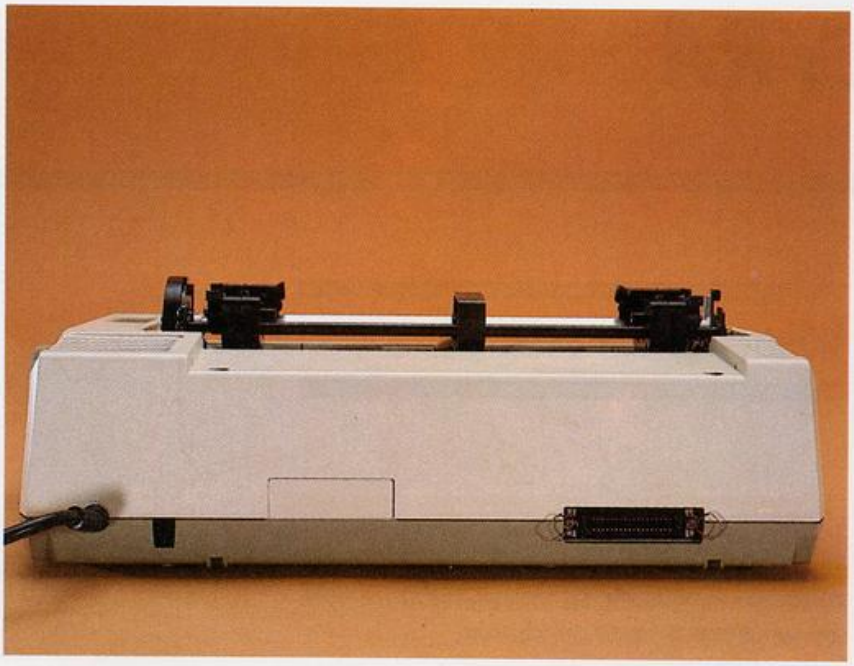


Diagrama del circuito de control de una impresora.

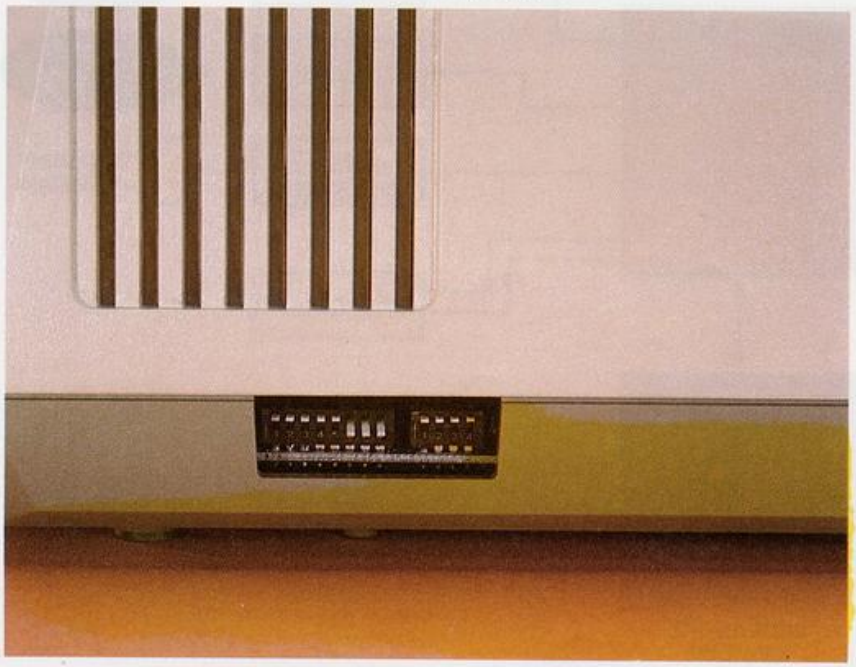


Por lo general, es en la parte trasera de las impresoras en donde se sitúan los conectores de interface.

tos independientes, íntimamente relacionados entre sí:

— Cabezal de impresión.
— Motor de avance del carro.
— Motor de avance del papel.

DIP SWITCHES es el conjunto de microconmutadores destinados a especificar por hardware determinadas condiciones de impresión.



Por tratarse de equipos matriciales, los caracteres se forman tras el impacto sobre la cinta entintada y el papel, de un número variable de agujas, dispuestas verticalmente en la cabeza. Es precisamente la cantidad de éstas (normalmente, entre 7 y 9), la que condiciona, aparte de otros factores, la calidad final del carácter impreso.

El motor de avance del carro es el encargado de mover la cabeza impresora. Está sujeto a las órdenes enviadas por la CPU del circuito de control. Esta, analiza la posición del cabezal y en función de ella y de los siguientes caracteres a imprimir, decide hacia dónde ha de desplazarse ésta para optimizar su movimiento, es decir, permite la escritura bidireccional, de izquierda a derecha y viceversa.

La CPU también controla el motor de avance del papel. Este debe funcionar con extraordinaria precisión, pues en algunos casos debe provocar desplazamientos verticales del orden de 1/216 pulgadas (1 pulgada=2,54 cm).

En la figura pueden apreciarse los bloques funcionales del circuito de control. La mayoría de ellos están presentes en casi todas las impresoras, pero otros como el interface en serie, el buffer o la memoria RAM programable, son opcionales.

Además de otros factores como, por ejemplo, el tipo de papel utilizado (hojas sueltas o continuo), sistema de desplazamiento de eśte (fricción o tracción), la disponibilidad de varios juegos de caracteres, o la posibilidad de modificar el formato de impresión, debemos analizar cuidadosamente los puntos anteriores, y en función de nuestras necesidades reales decidirnos por una u otra impresora.

SERIE O PARALELO

La transmisión de datos entre uno de estos equipos y nuestro Spectrum se efectúa a través de dos interfaces: uno situado en el interior de la impresora, y otro que debe ser adaptado a nuestro ordenador, pues su configuración mínima carece de él.

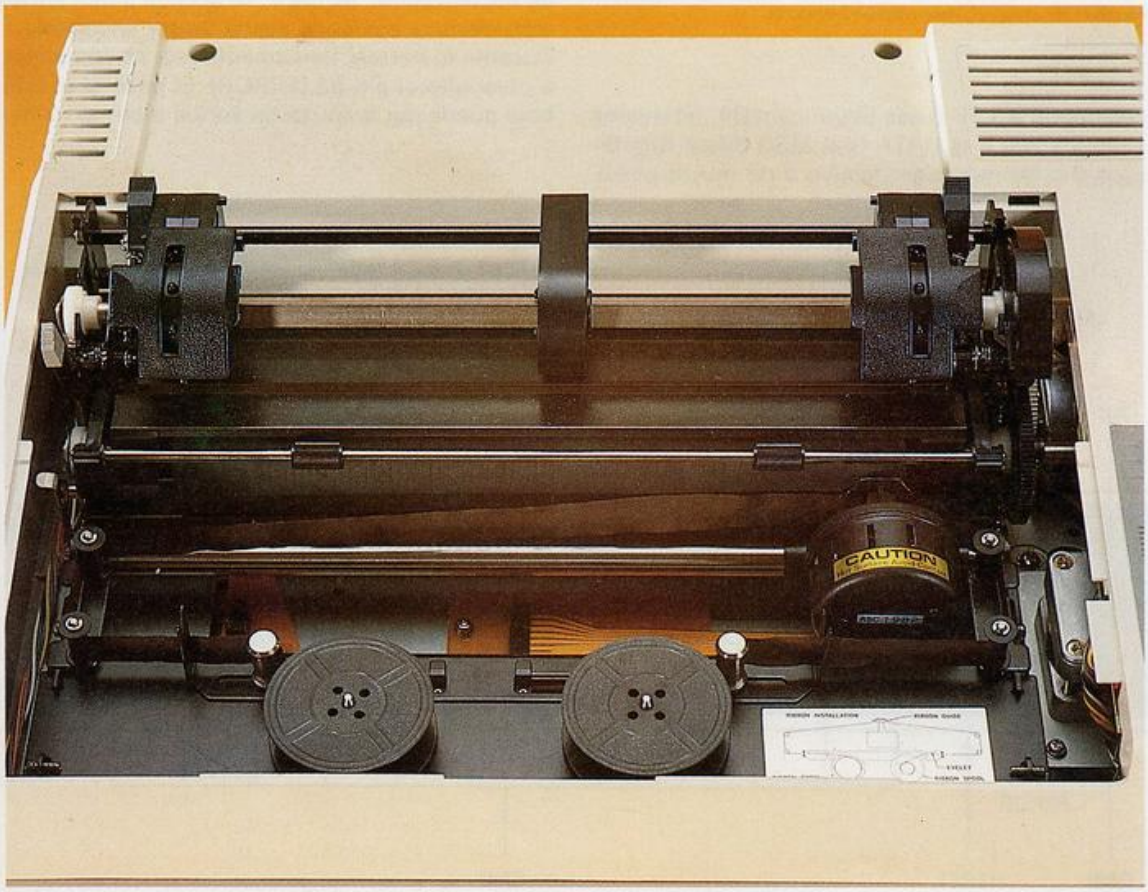
Según comentamos en capítulos anteriores, son dos las normas universalmente extendidas en la comunicación entre ordenadores e impresoras: serie (RS-232) o paralelo (CENTRONICS). Es el formato del interface situado en la impresora, el

que condiciona la elección de uno u otro adaptador para la conexión en el Spectrum.
Es decir, si la impresora incorpora interface en paralelo, necesitaremos un adaptador de igual norma para el Spectrum. Si se utiliza la comunicación en serie, el interface para el ordenador debe ser del tipo serie.
Algunos interfaces e impresoras están preparados para funcionar con las dos normas; por ello, la elección de un sistema u otro, queda solamente condicionado a la facilidad de manejo que el usuario pueda encontrar en él.

DATOS Y SEÑALES

Para el correcto funcionamiento de la impresora son necesarios una serie de protocolos y señales de control, que permitan la sincronización de todo el proceso de impresión. Tengamos en cuenta que la velocidad de transmisión de los datos es mucho mayor que la de impresión, y de no ser por estos impulsos, los resultados podrían ser catastróficos.
Comentemos las señales que circulan a través del conector de la impresora (la figura muestra su forma y la denominación de cada una de las pastillas o pines), en el caso de una vía de comunicación en paralelo. Estos pueden encontrarse en dos estados eléctricos diferentes, alto o bajo, cuya intensidad y duración varía según las impresoras.
A través del pin 1, STROBE, llega a la impresora la señal que indica cuando puede leer un byte de datos. Normalmente, el ordenador la mantiene a nivel alto, y cuando tiene un byte de datos preparado para ser enviado, lo transfiere y cambia la señal a nivel bajo, siendo recogido en los pines 2 a 9 (DATA 1 a DATA 8). A cada uno de ellos, llega un bit de información.
La impresora los reconoce según el nivel alto/bajo de cada uno. Así, para el «1» lógico el nivel eléctrico es alto, y bajo para el «0». El DATA 1 co-

Detalle de los elementos funcionales de una impresora.



¡!

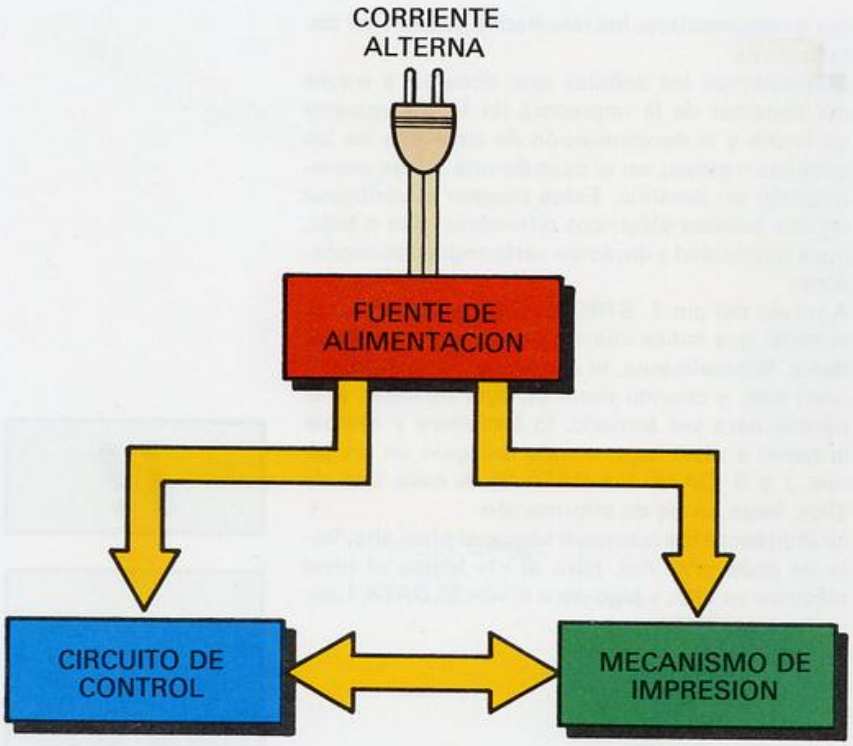
Debemos tener cuidado de no doblar la cinta entintada durante su instalación, ello puede dar lugar a bloqueos de la cabeza de impresión durante su recorrido.

*

Para evitar desagradables quemaduras, debemos evitar el contacto con el cabezal de impresión después de su uso prolongado, pues suele alcanzar temperaturas elevadas.

*

Para el completo borrado de la RAM interna de la impresora, es conveniente que transcurra un tiempo mínimo de un par de segundos entre un apagado y encendido consecutivo.



Como norma general las impresoras constan de los siguientes bloques principales: fuente de alimentación, circuito de control, mecanismo impresor.

rresponde al LSB (Less Significant Bit, bit menos significativo) y el DATA 8 al MSB (Most Significant Bit, bit más significativo o de mayor peso).

El ordenador debe mantener estas señales activas durante algún tiempo (del orden de 1 microsegundo), antes y después del impulso STROBE. Una vez recibido correctamente el dato, la impresora coloca el pin 10 (ACK, *ACKnowl edge,* reconocimiento) a nivel bajo, indicando al ordenador que se encuentra preparada para recibir el siguiente. También se genera esta señal cuando la impresora pasa del estado OFF-LINE (fuera de línea) al ON-LINE (en línea), en el cual está dispuesta a procesar los datos.

El pin 11, denominado BUSY (ocupado) señala que los datos no pueden ser recibidos. Estará a nivel alto durante la transferencia de éstos, cuando la impresora está OFF-LINE, o cuando exista una condición de error.

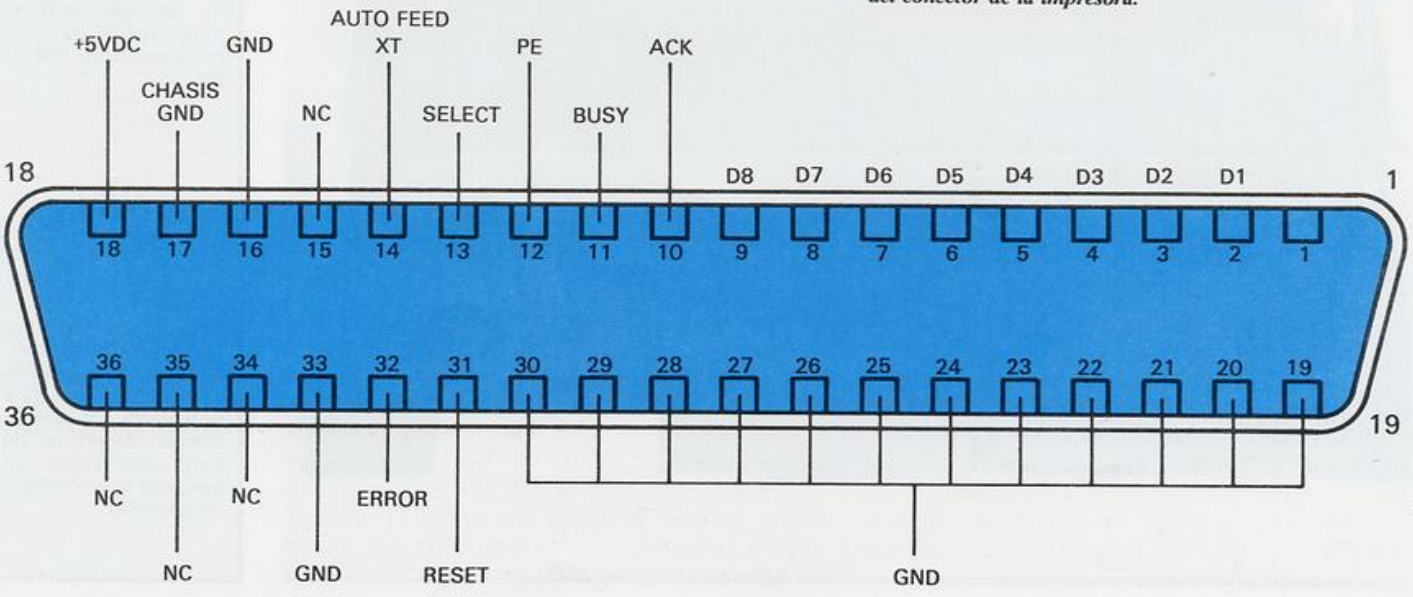
El nivel alto en el pin 12 (PE, *PapEr out,* sin papel), identifica que efectivamente éste se ha terminado. Mientras la máquina está ON-LINE se comprueba esta señal cada vez que se realiza un avance del papel.

Cuando la impresora está ON-LINE el pin 13 (SELECT) se mantiene alto, y en caso contrario, bajo. De esta manera, el ordenador reconoce si el periférico está dispuesto a recibir datos. Si el pin 14 (AUTO FEED XT) se encuentra bajo, se produce tras el retorno del carro, un salto de línea automático.

Si la señal presente en el pin 31 (RESET) se mantiene baja, la impresora comienza una secuencia de inicialización, es decir, se accede a una situación idéntica a cuando acaba de ser encendida.

Durante el normal funcionamiento, se encuentra a nivel alto el pin 32 (ERROR). El paso al estado bajo puede ser a causa de varios motivos, como

Para comprender mejor las señales que circulan a través del conector de la impresora.



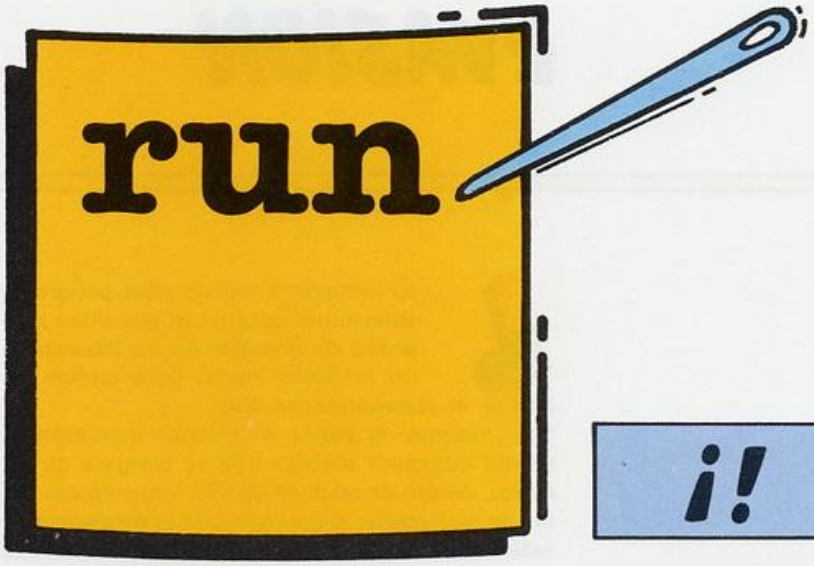
por ejemplo, la detección de un error en la RAM interna durante la inicialización, o la falta de papel. Para salir del primer estado de error, bastará con apagar y volver a encender la impresora.
Los pines 16, 19 a 30 y 33, todos denominados GND (*GrouND,* tierra o masa), son la masa de las distintas señales, mientras que el 17 (CHASIS GND), es la masa del chasis de la impresora, aislada de la masa de señales.
Finalmente, por lo general, el pin 18 (+5VDC) proporciona una tensión de +5V, con objeto de alimentar a un circuito externo, mientras que los pines 15 y 34 a 36 denominados NC (No Conectados), no son utilizados.

DIP SWITCHES

Bajo esta denominación se conoce al conjunto de microconmutadores, destinado a especificar por hardware determinadas condiciones de impresión. Algunas de ellas pueden fijarse también mediante el envío de códigos de control desde el ordenador, es decir, por software.

La transmisión de datos entre una impresora y nuestro Spectrum se efectúa a través de dos interfaces.

Su número, distribución, función y posición (ON/OFF) varía de unos fabricantes a otros, pero su utilidad es común a todas las impresoras, pues son leidos cuando encendemos el equipo, o cuando mandamos un código de control de inicialización **ESC §.** Es decir, mantienen las condiciones de impresión por defecto.



El número de agujas de una impresora (normalmente de siete a nueve) condiciona la calidad final del carácter impreso.

Dos o tres de ellos se destinan a la obtención del juego de caracteres particular de cada idioma. Por ejemplo, si escogemos el Castellano tendremos disponibles la ñ, i o ¿. Con dos podemos acceder a cuatro juegos, mientras que combinando tres switches son posibles hasta ocho diferentes, almacenados todos ellos en la ROM del Sistema. Otras funciones soportadas normalmente por los microinterruptores son las siguientes:

— Activar/desactivar el detector de final de papel.
— Longitud de página (11'' ó 12'').
— Salto automático de una línea tras el retorno de carro, CR+LF *(Carriagbe Return+Line Feed).*
— Longitud de línea (caracteres estándar o comprimidos).
— Impresión normal o en alta calidad.

Otras son específicas de cada equipo en cuestión como, por ejemplo:

— Selección del buffer como almacenamiento intermedio o como RAM programable.
— Activar/desactivar el zumbador de alarma.
— Salto de una pulgada tras el final de página.

Para terminar este capítulo, cabe aconsejar el escrupuloso cumplimiento de las notas escritas en los márgenes de página, además de las específicamente señaladas por cada fabricante. No olvidemos el principio según el cual «si algo puede ir mal, irá mal». Más que una máxima pesimista, es un aviso sobre posibles fuentes de averías que no debemos dejar al azar.

Una precaución a tener en cuenta durante el manejo de las impresoras, es no imprimir en ausencia de papel y cinta.

*

Al utilizar papel continuo, es conveniente asegurarse de que el eje de sujección apoya sobre el papel.

*

Es recomendable no instalar el sistema de tracción cuando se esté empleando el modo de fricción.

*

No es recomendable exponer las impresoras a altas temperaturas ni al polvo.

EVASION

Las mariposas corren grave peligro de exterminio porque un científico loco acaba de inventar en su laboratorio un artilugio mortal para dichos insectos: el cazamariposas láser.
Este aparato es capaz de disecar instantáneamente cualquier insecto que se ponga a su alcance, dentro de su línea de tiro. Los coleccionistas de mariposas, al enterarse de la existencia de esta maléfica arma, han agotado las existencias de dicho artefacto, y se han lanzado al campo con el fin de aumentar sus colecciones.
¿No nos da pena? ¿Vamos a permitir que unos desalmados nos impidan disfrutar del colorido de sus alas, de su vuelo errático y de su poético significado? ¡Hay que salvar a estos inofensivos lepidópteros!

¡!

Las teclas que debemos pulsar para el desplazamiento de la mariposa son las siguientes:
Q-Arriba.
P-Salto hacia la derecha.

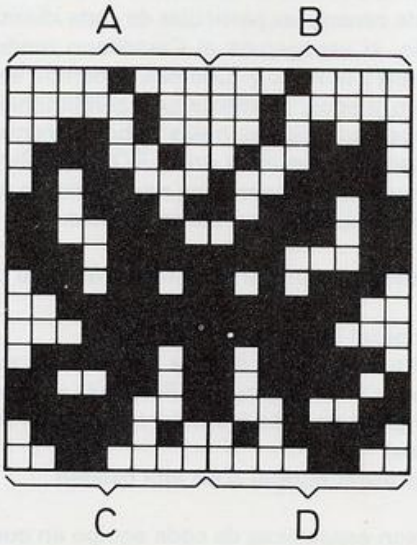
Si no tenemos pulsada ninguna tecla, nuestra mariposa presentará tendencia a ir a su posición inicial.

Los caracteres subrayados corresponden a los gráficos definidos de usuario.

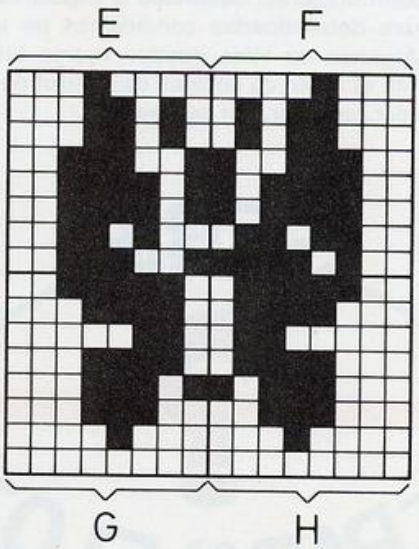
Para la grabación del programa teclearemos el siguiente comando directo: **SAVE "EVASION"**. Si optásemos por la grabación con autoejecución, utilizaríamos la siguiente instrucción: **SAVE "EVASION" LINE 10.**

EL PROGRAMA

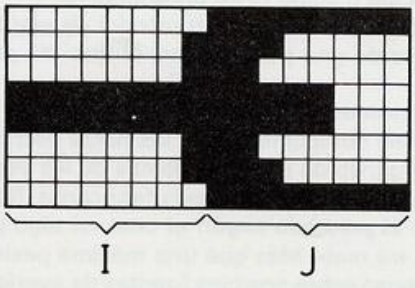
Una vez que el listado del programa ha sido correctamente introducido y ejecutado, nuestro Spectrum nos pedirá que introduzcamos el nivel de dificultad con el que queremos jugar. Para ello, hemos de teclear un número que, como se muestra en la pantalla, ha de estar comprendido entre 1 y 4, ambos inclusive.



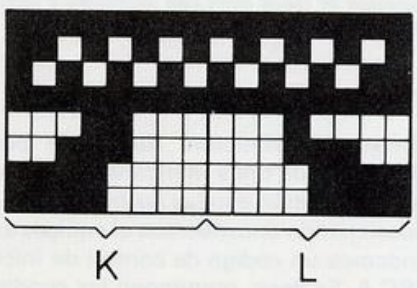
Gráficos definidos para la primera posición de la mariposa.



Gráficos definidos para la segunda posición de la mariposa.



Gráficos definidos para el cazamariposas láser.



Gráficos definidos para el taburete.

Una vez hecho esto, aparecerá en el ángulo inferior izquierdo una mariposa que está batiendo sus alas, y en el otro extremo un taburete que, para el lepidóptero, es sinónimo de salvación.
Nuestra misión consiste en conducir la mariposa a la parte superior del taburete y posarla en él. El juego no es tan fácil como parece, porque desde el primer instante en que comencemos a movernos, el cazamariposas láser hará acto de presencia.
Las teclas que hemos de utilizar para el correcto desplazamiento de nuestra mariposa son las siguientes:

Q - Arriba.
P - Salto hacia la derecha.

La dificultad del juego, independientemente del nivel de complejidad, irá aumentando a medida que nos vayamos acercando a la columna 23, y una vez rebasada ésta, más nos valdrá darnos prisa en colocar la mariposa en el lugar de destino. Si conseguimos hacer llegar a nuestro insecto

sano y salvo a su destino, el Spectrum nos felicitará y nos indicará el tiempo que hemos tardado en efectuar la hazaña.

Para la introducción del programa, recordemos introducir en el modo gráfico las letras que aparecen subrayadas en el listado. Una vez cumplido este requisito, podremos grabar el programa mediante **SAVE "EVASION"** o **SAVE "EVASION" LINE 10**, si vamos a emplear el sistema de autoejecución.

```
 10 REM **********************
 20 REM * J.M.MAYORAL SERRANO *
 30 REM **********************
 40 REM * EVASION    (C) 1985 *
 50 REM **********************
 60 DATA 8,4,52,114,88,221,206,239
 70 DATA 16,32,42,78,158,187,123,227
 80 DATA 109,63,31,125,205,249,122,48
 90 DATA 174,252,248,190,189,147,94,12
100 DATA 16,26,26,57,61,61,52,57
110 DATA 8,88,88,156,188,188,108,244
120 DATA 62,30,6,30,29,28,8,0
130 DATA 124,120,96,120,184,56,16,0
140 DATA 0,1,1,255,255,1,1,0
150 DATA 255,226,192,248,248,192,226,255
160 DATA 255,213,170,255,31,56,240,240
170 DATA 255,85,171,255,240,28,15,15
180 RESTORE
190 LET SW1=0
200 POKE 23658,8
210 LET S=30
220 LET H=20
230 LET D=3
240 LET CE=0
250 LET SH=0
260 LET OG=INT (RND*17)+2
270 LET FR=0
280 LET U=1.5
290 LET U1=.3
300 LET OH=20
310 LET OD=3
320 LET SW=0
330 REM
340 REM
350 REM
360 IF SW1 THEN  GO TO 440
370 REM
380 FOR I=USR "A" TO (USR "L"+7)
390 READ A
400 POKE I,A
410 NEXT I
420 REM
430 REM
440 BORDER 1
450 PAPER 1: INK 9
460 CLS
470 PRINT AT 18,0;" INTRODUCE NIVEL DE DIFICULTAD "
480 PRINT AT 20,7;" 1 - 2 - 3 - 4 "
490 LET A$=INKEY$
500 IF A$="" THEN  GO TO 490
510 IF CODE A$<49 OR CODE A$>52 THEN  BEEP 1,-10: GO
TO 490
520 BEEP .1,40
530 LET P=VAL A$
540 LET P1=P
550 CLS
560 INK 5
570 PRINT AT 19,30;"KL"
580 PLOT 255,0
590 DRAW 0,175
600 DRAW -255,0
610 DRAW 0,-16
620 PLOT 0,23
630 DRAW 0,-23
640 INK 9
650 PRINT #0;"3333333333333333333333333333333333"
660 REM  INICIALIZACION RELOJ
670 FOR N=23672 TO 23674
680 POKE N,0
690 NEXT N
700 GO TO 710
710 REM   JUEGO
720 LET OH=H
730 LET OD=D
740 LET K$=INKEY$
750 LET H=H+(H<19)-(K$="Q" AND H>1)*U
760 LET IH=INT H
770 LET D=D+(K$="P" AND H<19)*U-(D>3)*U1
780 LET ID=INT D
790 IF K$<>"P" THEN  BEEP .001,40
800 IF D>30 THEN  LET D=30
810 IF D>23 THEN  LET P=P-2: GO TO 830
820 LET P=P1
830 IF NOT SH THEN  GO TO 900
840 LET SH=CE
850 PRINT AT OH,OD;"  "
860 PRINT AT OH+1,OD;"  "
870 PRINT AT H,D;"EF"
880 PRINT AT H+1,D;"GH"
890 GO TO 960
900 LET SH=1
910 PRINT AT OH,OD;"  "
920 PRINT AT OH+1,OD;"  "
930 PRINT AT H,D;"AB"
940 PRINT AT H+1,D;"CD"
950 PRINT  INK 5;AT 19,30;"KL"
960 IF D=30 AND INT H>=17 THEN  GO SUB 1300: PRINT A
T 20,0;"TE ESCAPASTE EN ";: GO TO 1210
970 IF NOT FR AND RND<U1*.3 OR H>19 THEN  GO TO 720
980 IF FR THEN  GO TO 1040
990 LET NG=INT (INT (RND*2)+H)-1
1000 LET NG=NG-(NG>18)
1010 IF NG<2 THEN  LET NG=2
1020 LET GM=(NG>OG)-(NG<OG)
1030 LET FR=1
1040 IF NG=OG THEN  GO TO 1100
1050 PRINT AT OG,0;"  "
1060 LET OG=OG+GM
1070 IF OG>=21 THEN  LET OG=20
1080 PRINT  INK 6;AT OG,0;"IJ"
1090 GO TO 720
1100 IF SW<P THEN  LET SW=SW+1: GO TO 720
1110 LET SW=0
1120 PLOT 24,(21-NG)*8+4
1130 DRAW  INK 6;214,0
1140 BEEP .05,40
1150 PLOT  OVER 1;24,(21-NG)*8+4
1160 DRAW  INK 6; OVER 1;214,0
1170 BEEP .01,30
1180 LET RH=(IH+(H>IH))
1190 IF NG<>RH AND NG<>RH+1 OR RH>19 THEN  LET FR=0:
GO TO 720
1200 PRINT AT 10,5;"  PERDISTE GUAPERAS  ": GO TO 124
0
1210 LET T=INT ((((65536*PEEK 23674+256*PEEK 23673+PEE
K 23672)/(50*60))*100)
1220 LET T=60*T/100
1230 PRINT  PAPER 2; INK 7;AT 20,16;T; PAPER 1; INK 9
;" SEGUNDOS"
1240 PRINT AT 15,1; PAPER 6;" PULSA UNA TECLA PARA EM
PEZAR "
1250 IF INKEY$<>"" THEN  GO TO 1250
1260 IF INKEY$="" THEN  GO TO 1260
1270 CLS
1280 LET SW1=1: GO TO 220
1290 REM   FIN DE JUEGO
1300 PRINT  INK 5;AT 17,30;"AB"
1310 PRINT  INK 5;AT 18,30;"CD"
1320 PRINT  INK 7;AT 19,30;"KL"
1330 PRINT  INK 7;AT 20,30;"  "
1340 RETURN
```