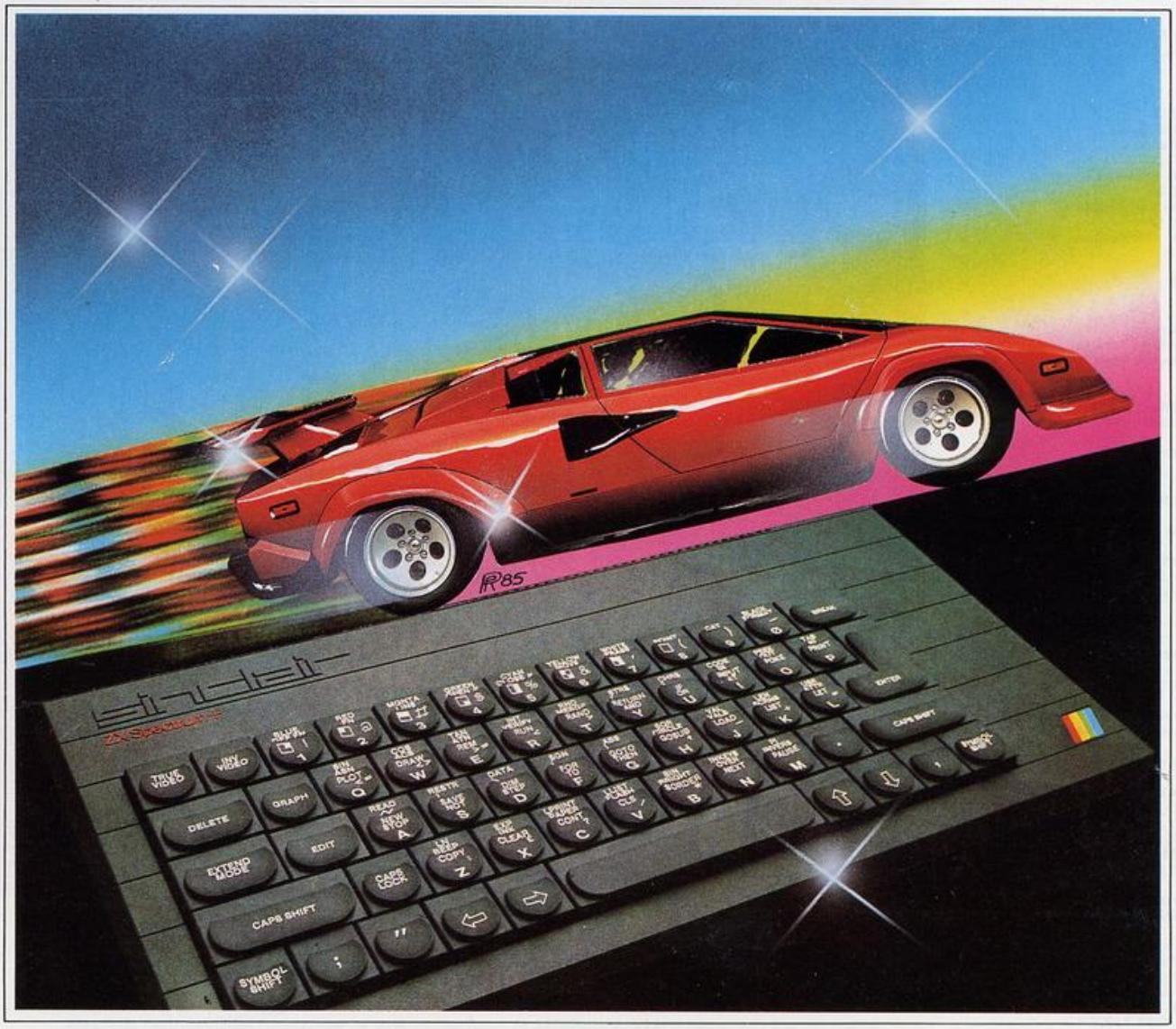
27
150pts.

RUN

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek

PRINCIPIOS GENERALES DEL SONIDO

EL sonido se produce cuando un elemento físico perturba la atmósfera, creando variaciones en la presión del aire que le rodea. Dichas variaciones de presión se propagan en forma de onda esférica y, si son interceptadas por la membrana del tímpano, se transforman en impulsos nerviosos que van al cerebro, donde se interpreta la información recibida.

Si la presión generada pasa instantáneamente a su valor máximo, y permanece en él durante un breve intervalo de tiempo, volviendo bruscamente al valor mínimo, entonces se califica a la onda como «cuadrada», a diferencia de la ya conocida y más suave onda «sinusoidal».

Las ondas sinusoidales son tanto aquellos producidas por medios naturales, como las emitidas por los instrumentos musicales. A este tipo de ondas podemos denominarlas también, atendiendo a la forma en que son producidas, «analógicas».

Decimos que una onda es de tipo analógico o sinusoidal, cuando el paso del valor máximo al mínimo se produce de una forma gradual, siendo su gráfica una curva continua, similar a la de la función seno.

Por el contrario, las ondas de tipo cuadrado o «digital» son las producidas por osciladores electrónicos. Se trata pues de un sonido artificial, en el cual el paso del valor mínimo al máximo se efectúa de una forma brusca, produciendo una onda cuya imagen es similar a las almenas de un castillo.

El sonido se produce cuando un elemento físico perturba la atmósfera.

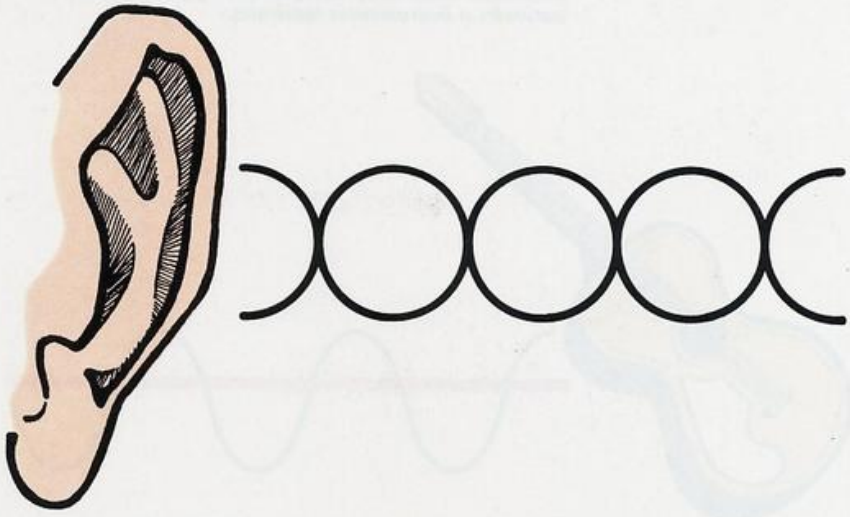
CARACTERISTICAS DE LAS ONDAS

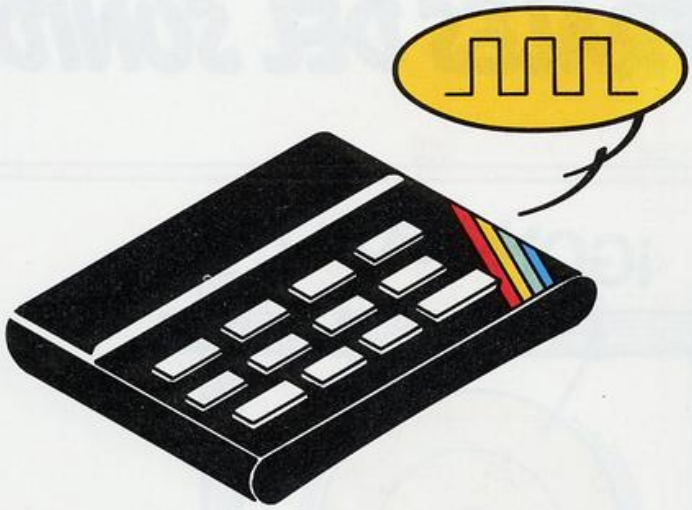
Hagamos un viaje a través de las ondas para conocer sus características; en todo momento, nos apoyaremos en los gráficos adjuntos para llegar a comprender este nuevo vocabulario.

Si consideramos las gráficas de los dos tipos de ondas, las ordenadas (eje vertical) expresarán el «pulso» y las abscisas (eje horizontal) el «tiempo», denominándose «crestas» los picos de mayor altura y «valles» los de menor.

De esta forma, se denomina «amplitud» de onda a la máxima distancia alcanzada entre una cresta o un valle y el eje de abscisas, y «longitud» de onda a la distancia existente entre dos crestas o dos valles consecutivos.

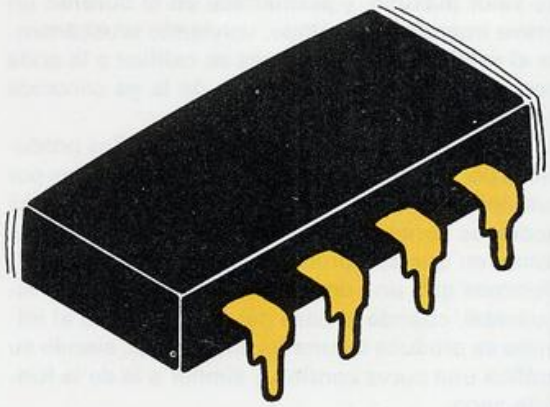
El sonido se propaga en forma de ondas esféricas, que suelen ser interceptadas por la membrana del tímpano.





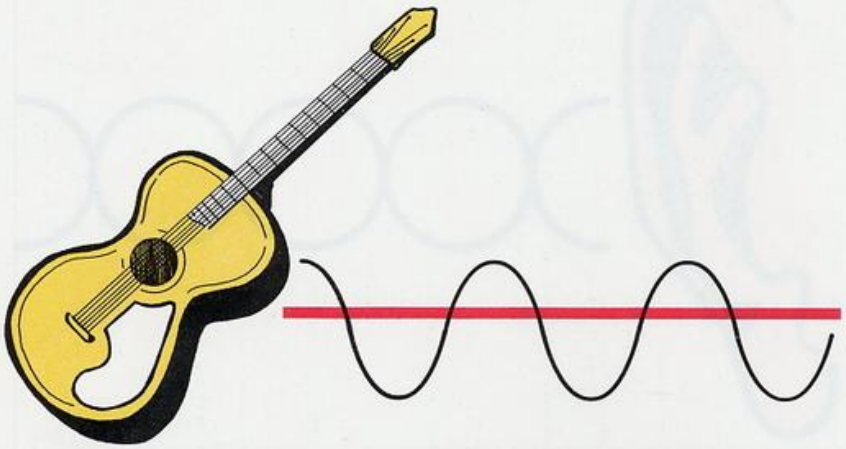
La onda cuadrada es un sonido artificial. Se produce cuando el paso del valor mínimo al máximo es de forma brusca.

El tono de un sonido puede ser más grave o más agudo. Con mayor propiedad, podemos decir que el tono está en relación directa con la frecuencia del sonido producido, pudiéndose definir ésta como la cantidad de variaciones completas que tienen lugar en un segundo, es decir, el número de longitudes de onda descritas en la unidad de tiempo, medida en Hertzios (Hz) o ciclos por segundo (C/S).
La gama de tonos audibles por el ser humano va desde los 40 a los 15.000 Hz., aproximadamente, mientras que el Spectrum puede producir una serie de tonos de onda cuadrada, situados entre 10 y 15.000 Hz., aproximadamente. La forma de generar dichos sonidos consiste en conmutar y desconmutar electrónicamente, a intervalos regulares, una señal de tensión constante, y llevar las variaciones amplificadas a un pequeño altavoz alojado en su interior.
El control del sonido, tanto en la duración como en la frecuencia, se efectúa en el BASIC del Spectrum por medio de la sentencia **BEEP D, N,** donde el parámetro **D** representa la duración en segundos, y **N** el código del tono o frecuencia. Este último puede tomar cualquier valor entero comprendido entre -59 (unos 10 Hz.) y +69 (15.000 Hz. aproximadamente).
Concretamente, cuando **N** es igual a cero, la frecuencia generada corresponde al tono de Do intermedio de la escala del piano, sobre 261 ciclos. La escala completa se consigue, internamente, utilizando las siguientes frecuencias:



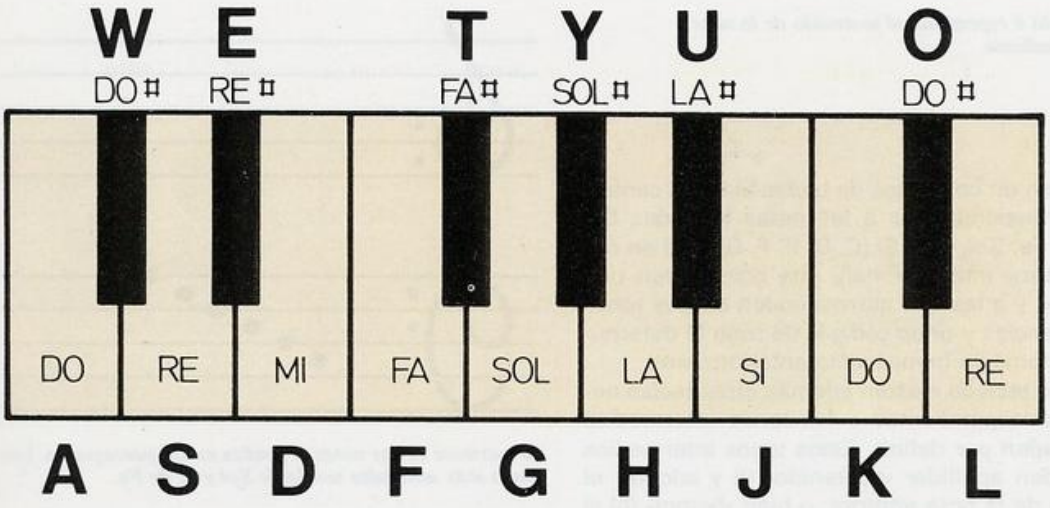
El control del sonido lo efectúa la C. P. U. del Spectrum por medio de **BEEP D, N.**

Las ondas sinusoidales son producidas por medios naturales o instrumentos musicales.



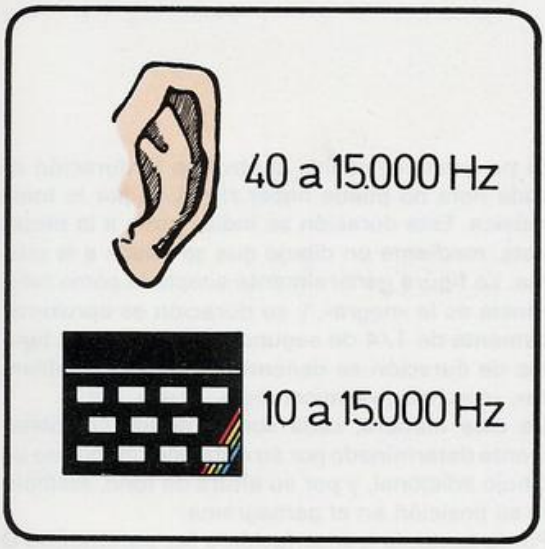
INT.	NOTA	FRECUENCIA
C	DO	261.63
C#	DO#	277.18
D	RE	293.66
D#	RE#	311.13
E	MI	329.63
F	FA	349.23
F#	FA#	369.99
G	SOL	392.00
G#	SOL#	415.30
A	LA	440.00
A#	LA#	466.16
B	SI	493.88

Donde la columna **"INT."** corresponde a la notación internacional y el símbolo «#» al «sostenido» de la nota correspondiente, como explicaremos más adelante.



Correspondencia entre las notas musicales y las letras utilizadas por el ordenador, y su posición en el teclado.

La gama de tonos audibles por el ser humano va desde 40 a 15.000 Hz. El Spectrum produce una gama de tonos de onda cuadrada entre 10 y 15.000 Hz.



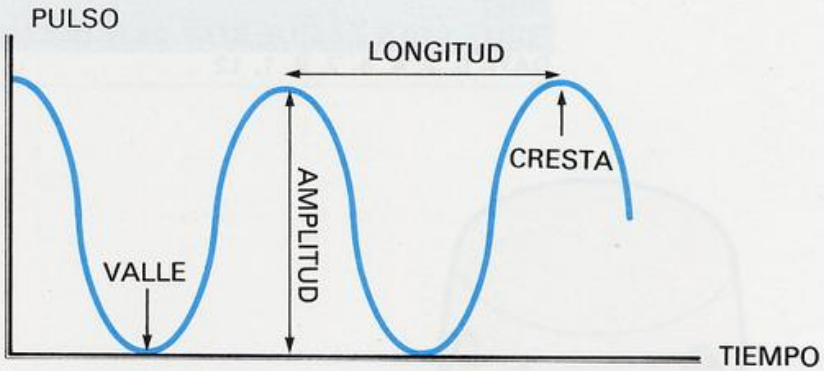
COMPONIENDO MUSICA

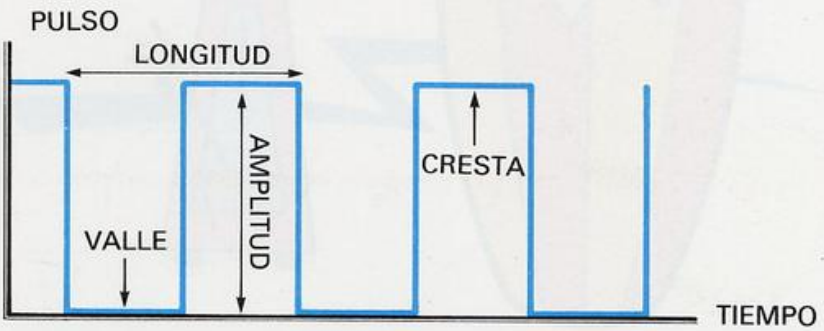
La música se compone fundamentalmente de sonido y ritmo, es decir, de la producción de ciertos tonos durante tiempos determinados.

La diferencia entre el lenguaje musical y el del Spectrum es únicamente cuestión de formas, como ya veremos. Pero, antes de nada, conozcamos los componentes elementales de la música, las notas, cuya representación escrita nos indica la altura del tono en que nos encontramos y la duración del mismo.

Para una más fácil comprensión de los no iniciados, consideraremos el teclado de un piano. En

Representación gráfica de la onda sinusoidal y cuadrada con sus principales características.

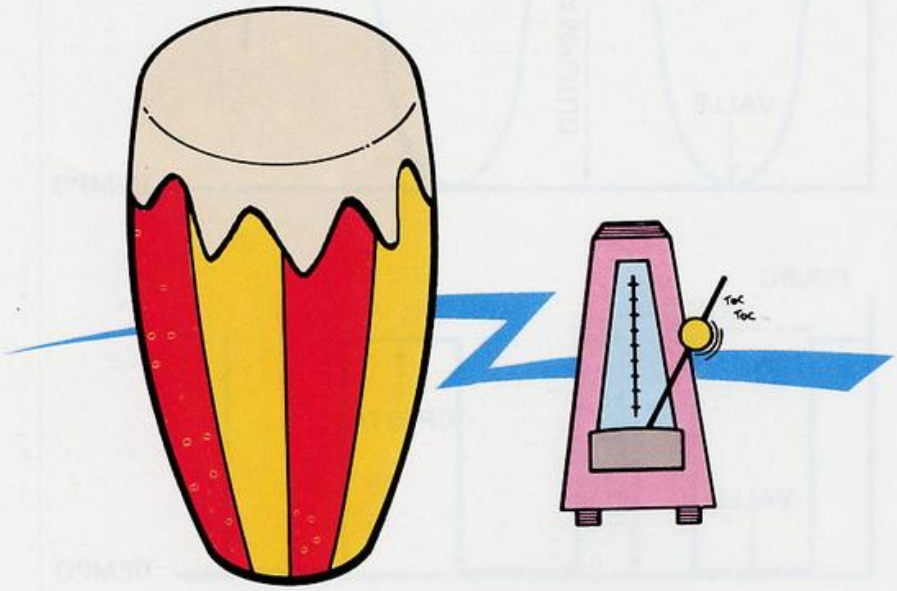




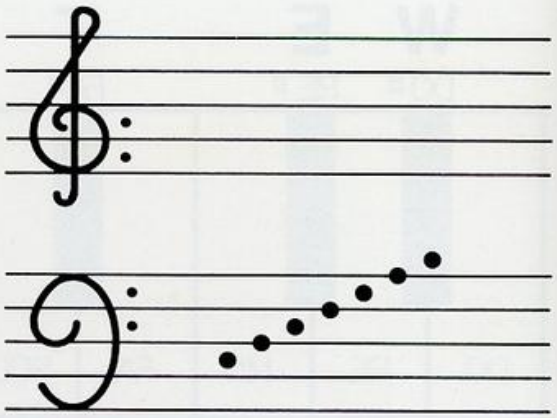
El símbolo # representa al sostenido de la nota correspondiente.

él existen un conjuntos de teclas blancas centrales, correspondientes a las notas llamadas Do, Re, Mi, Fa, Sol, La y Si (C, D, E, F, G, A, B en nomenclatura internacional), que constituyen una «octava», y a las que corresponden ciertos tonos o frecuencias y unos códigos de tono **N** determinados, como ya hemos visto anteriormente.
En dicho teclado existen además otras teclas negras, correspondientes a los tonos intermedios que quedan por definir. Estos tonos intermedios se pueden apellidar «sostenido» ()) y adoptar el nombre de la nota anterior, o bien «bemol» (b) si el nombre adoptado es el de la nota posterior. Por ejemplo, el tono de Do) (Do sostenido) coincide con el de Re b (Re bemol). Cabe añadir que la distancia entre dos tonos consecutivos (dos teclas contiguas blancas o blanca y negra), se denomina «semitono».
La escritura de las notas se realiza sobre un conjunto de cinco líneas, llamado pentagrama. En nuestro caso utilizamos dos de sus muchas versiones: el de clave de Sol en segunda línea y el de clave de Fa en cuarta línea.
Para hacernos una idea más clara de todo lo dicho, veremos a continuación un programa generador de la escala musical:

```
REM - ESCALA MUSICAL J. M. LOPEZ MARTINEZ
FOR I = 1 TO 8: READ N: BEEP. 25,N: NEST N
DATA 0, 2, 4, 5, 7, 9, 1, 12
```



La música se compone fundamentalmente de sonido y ritmo.



La escritura de las notas se realiza en un pentagrama. Las claves más utilizadas son la de Sol y la de Fa.

DURACION DE LAS NOTAS

Si no existe el debido control de la duración de cada nota no puede haber ritmo, ni por lo tanto música. Esta duración se indica junto a la propia nota, mediante un dibujo que se añade a la misma. La figura generalmente aceptada como referencia es la «negra», y su duración es aproximadamente de 1/4 de segundo. Las restantes figuras de duración se denominan «redonda», «blanca», «corchea», «semicorchea», «fusa», etc.
De esta manera, cada sonido queda completamente determinado por su duración, mediante un dibujo adicional, y por su altura de tono, mediante su posición en el pentagrama.
Por lo tanto, la transcripción a los parámetros **D** y **N** resulta trivial y puramente mecánica. No obstante, para facilitar aún más la labor, podemos tomar como referencia el valor de la duración de la fusa, que es 1/32 de segundo. De esta forma, los valores de las duraciones del resto de las notas serán las sucesivas potencias de dos, lo cual facilitará en gran medida la labor de programación.
El puntillo colocado a la derecha de cualquier nota aumenta su duración en la mitad, y los si-

REDONDA NEGRA BLANCA CORCHEA SEMICORCHEA FUSA SEMIFUSA

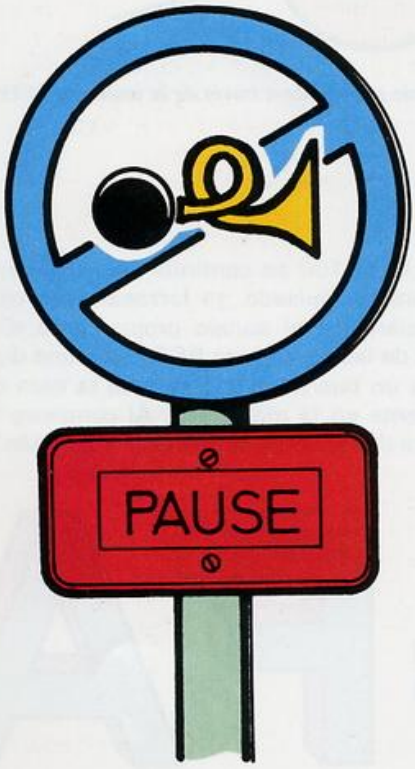
Mediante estas figuras queda determinada la duración del sonido, y la altura del tono por su posición en el pentagrama.

lencios son los tiempos muertos en los cuales el compositor ha decidido que no debe haber sonido alguno. Existen dos alternativas para interpretar los silencios y, como es obvio, ambas utilizan la sentencia **PAUSE.**

La primera consiste en poner un signo - en la duración, que será detectado con una comparación. La segunda es añadir un tercer conjunto de datos de entrada, de manera que si el dato leído es cero, la comparación lo ignore y genere el tono; de no ser así (si no es cero), el ordenador estará en presencia de un silencio y desviará el control hacia la sentencia **PAUSE** correspondiente.

El único y mínimo inconveniente es que **PAUSE** funciona en unidades de 1/50 de segundo, y la coincidencia de duraciones con los tonos puede no ser absoluta.

Para interpretar los silencios utilizaremos la sentencia **PAUSE.**



EMPEZANDO A COMPONER

Con los conocimientos básicos que hasta ahora hemos adquirido, estamos en condiciones de construir un programa capaz de emitir sonidos como un piano, que nos permita componer melodías.

Nuestro piano, consta de 9 teclas blancas (Do, Re, Mi, Fa, Sol, La, Si, Do, Re), localizadas en el teclado del Spectrum sobre la línea media de las letras (A, S, D, F, G, H, J, K, L), y de 6 teclas negras (Do); (Re); (Fa); (Sol); (La); (Do), situadas en W, E, T, Y, U y O de la fila superior de letras. De cualquier forma, nuestro piano es generosamente didáctico, y nos va a mostrar en pantalla la distribución de las teclas. Además, cada vez que pulsemos una, nos lo indicará a través de señales visuales para orientarnos sobre la posición de los dedos:

```
  10 REM - PIANO - J.M.LOPEZ MARTINEZ
  20 DIM N(25)
  30 FOR I=1 TO 25: READ N(I): NEXT I
  40 FOR I=1 TO 4
  50 PRINT AT 7+I,3;"   ";
  60 FOR J=1 TO 7
  70 IF J=2 OR J=6 THEN  PRINT "   ";: G
O TO 90
  80 PRINT "   ";
  90 NEXT J
 100 NEXT I
 110 PRINT AT 6,4;"D  R     F  S  L
D";AT 9,4; OVER 1;"W  E     T  Y  U
O"; OVER 0;AT 14,3;"A  S  D  F  G  H  J
 K  L";AT 17,3;"D  R  M  F  S  L  S  D
R"
 120 FOR I=0 TO 8: PLOT 16+I*24,48: DRAW
 24,0: DRAW 0,64: DRAW -24,0: DRAW 0,-64
: NEXT I
 130 LET X$=INKEY$: IF X$="" THEN  GO TO
 130
 140 IF X$<>"w" AND X$<>"e" AND X$<>"t"
AND X$<>"y" AND X$<>"u" AND X$<>"o" AND
X$<>"a" AND X$<>"s" AND X$<>"d" AND X$<>
"f" AND X$<>"g" AND X$<>"h" AND X$<>"j"
```

¡!

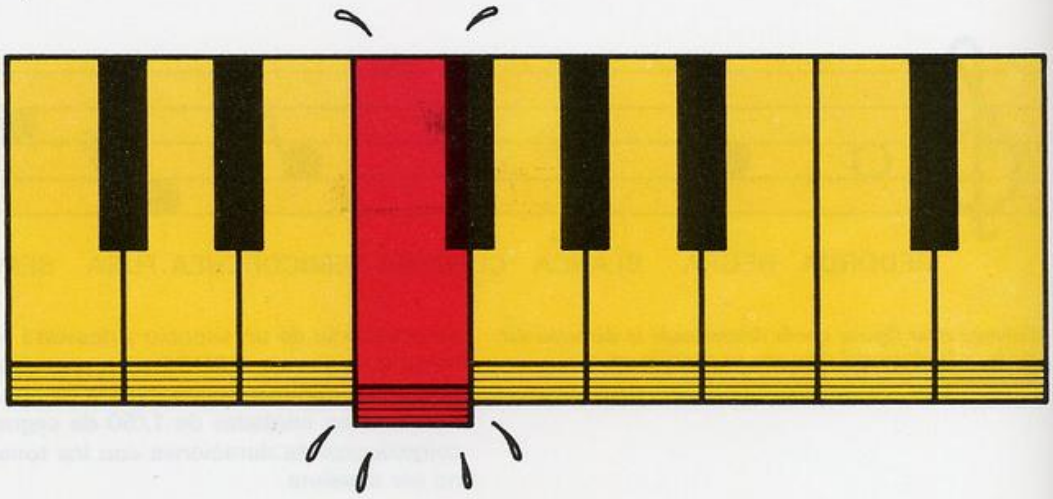
Se dice que una onda es cuadrada cuando la presión pasa instantáneamente a su valor máximo, y permanece con él durante un breve intervalo de tiempo, volviendo bruscamente al valor mínimo.

*

Las ondas de tipo cuadrado o «digital» son las producidas por osciladores electrónicos. En ellas, el paso del valor mínimo al máximo se efectúa de una forma brusca, produciendo una onda cuya imagen es similar a la de las almenas de un castillo.

*

El sonido se produce cuando un elemento físico perturba la atmósfera, creando variaciones en la presión del aire que le rodea.



En nuestro piano, cada vez que pulsemos una tecla, nos lo indicará, para orientarnos sobre la posición de los dedos.

¡!

Las ondas sinusoidales son las producidas por medios naturales, y se denominan también «analógicas». En ellas, el paso del valor máximo al mínimo se produce de una forma gradual, siendo su gráfica una curva continua, similar a la gráfica de la función seno.

*

La distancia entre dos tonos consecutivos (dos teclas contiguas blancas o blanca y negra), se denomina «semitono».

*

Existen unas teclas negras, que corresponden a los tonos intermedios. Estos tonos de la nota anterior, o «bemol» (b) si el nombre adoptado es el de la nota posterior.

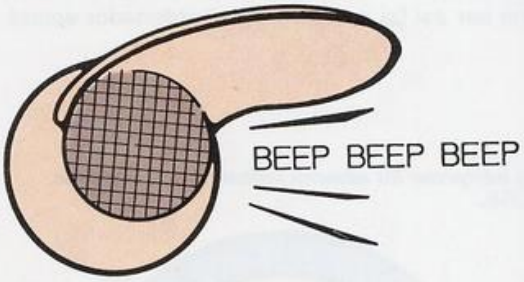
```
AND X$<>"k" AND X$<>"l" THEN  GO TO 130
 150 GO SUB 160: BEEP .25,N(CODE X$-96):
 GO SUB 160: GO TO 130
 160 LET X=(3 AND X$="a")+(4 AND X$="w")
+(6 AND X$="s")+(7 AND X$="e")+(9 AND X$
="d")+(12 AND X$="f")+(13  AND X$="t")+(
15 AND X$="g")+(16 AND X$="y")+(18 AND X
$="h")+(19 AND X$="u")+(21 AND X$="j")+(
24 AND X$="k")+(25 AND X$="o")+(27 AND X
$="l")
 170 LET Y=14: IF X$="w" OR X$="e" OR X$
="t" OR X$="y" OR X$="u" OR X$="o" THEN
 LET Y=9
 180 PRINT AT Y,X; OVER 1; INVERSE 1;" "
: RETURN
 190 DATA 0,0,0,4,3,5,7,9,0,11,12,14,0
 200 DATA 0,13,0,0,0,2,6,10,0,1,0,8
```

En las líneas 20 y 30 se dimensiona la matriz **N(25)** que contiene las notas asignadas a las teclas del piano, efectuándose su carga a partir de las líneas de **DATA** de las intrucciones 190 y 200.

En las líneas 40 a 100, se dibujan las teclas «negras» del piano en las posiciones convenientes. En 110 se imprimen las teclas que podemos pulsar y, sobre ellas, los correspondientes valores de las iniciales de las notas. Debemos destacar el empleo de la función **OVER**, que permite la distinción de las letras impresas, en posiciones donde se solapan las teclas «blancas» y las «negras». Para terminar con este bloque, la línea 120 imprime por medio de la función **DRAW**, los contornos de las teclas «blancas».

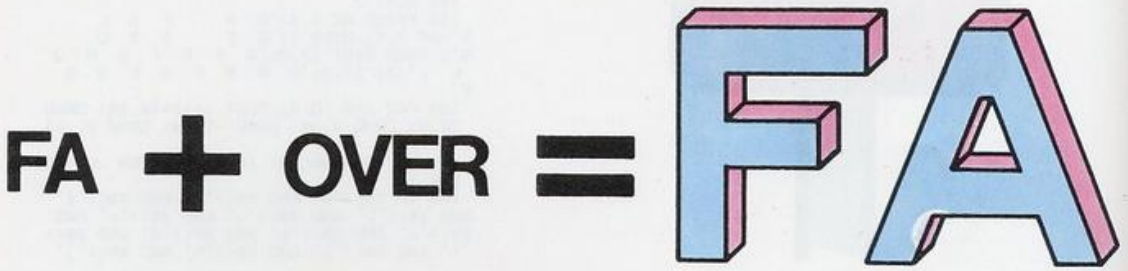
El empleo de **OVER** *permite el resalte de las letras impresas, correspondientes a cada nota musical.*

En el bloque formado por las instrucciones 130 y 140, se aceptan caracteres del teclado por medio de la sentencia **INKEY$**, eliminándose en un primer paso las respuestas nulas y en un segundo las pulsaciones de todas aquellas teclas diferentes de las reseñadas como utilizables en el teclado del piano.



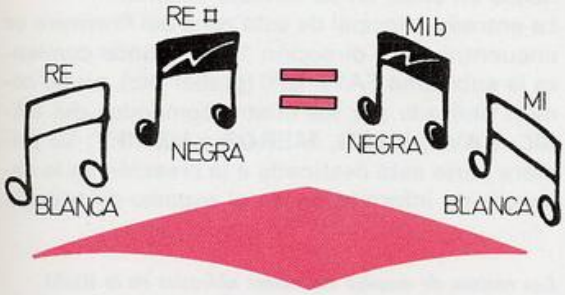
El sonido se produce a través de la sentencia **BEEP**.

En la línea 150 se continúa con el tratamiento del carácter pulsado, ya forzosamente correcto, produciéndose el sonido propiamente dicho, a través de las sentencias **BEEP**, con una duración fija de un cuarto de segundo de la nota correspondiente en la matriz **N()**. Al comienzo y final de esta línea, antes de volver a la petición de da-



El puntillo colocado a la derecha de cualquier nota aumenta su duración en la mitad.

tos, se bifurca a la subrutina 160 encargada de encender y apagar la tecla pulsada en el lugar correspondiente del teclado.
Esta es la parte más complicada del programa, precisamente la subrutina de 160 a 180. En la línea 160 se analiza el valor que se ha de dar a la variable **X**, la cual representa el primer parámetro (horizontal) de la sentencia **AT**. Para ello, se recurre a la suma de las evaluaciones lógicas con relación de tipo **AND**, entre cada una de las posibilidades de teclas pulsadas, con el número de columna que les correspondería en la impresión.



Los tonos de las notas negras se suelen apellidar: sostenido (#) y bemol (b).

Como sólo una de las evaluaciones puede ser correcta, la que lo sea dejará a **X** con el valor de la columna en la cual se debe realizar la impresión, produciendo el resto de las evaluaciones la suma de ceros, sin alterar el valor correcto de **X**. En la instrucción 170, se hace lo propio con el valor de la fila en la cual se ha de realizar la impresión. Sin embargo, esta vez se opta por un nuevo sistema, consistente en establecer de entrada el valor más frecuente, para ser modificado, caso de que no sea correcto, por la sentencia de comparación que le sigue.

Por último, en la línea 180, se efectúa la impresión por medio de **AT**, en la fila y columna especificados por las variables **X** e **Y**. Esta impresión se realiza en **OVER 1**, de un cuadrado de color de la **INK**, por ello, la primera vez que se entra en la subrutina la letra se coloca en video inverso, y la segunda se pasa a video normal, apoyándonos en las propiedades de sobreimpresión con **OVER**.

LOS GRANDES COMPOSITORES

A continuación, como muestra de las posibilidades de la sentencia **BEEP** en la composición de música, reproducimos el listado de un programa, que ejecuta algunos compases de la Marcha Turca de W. A. Mozart:

```
 10 REM MARCHA TURCA - W.A.MOZART
 20 FOR I=0 TO 259
 30 READ A
 40 BEEP .04,A-2
 50 FOR J=0 TO 3: NEXT J
 60 NEXT I
 70 DATA 11,9,8,9,12,12,12,12
 80 DATA 14,12,11,12,16,16,16,16
 90 DATA 17,16,15,16,23,21,20,21
100 DATA 23,21,20,21,24,24,24,24
110 DATA 21,21,24,24,23,23,21,21
120 DATA 19,19,21,21,23,23,21,21,19,19,
21,21,23,23,21,21,19,19,18,18,16,16
130 DATA 69,69,69
140 DATA 11,9,8,9,12,12,12,12
150 DATA 14,12,11,12,16,16,16,16
160 DATA 17,16,15,16,23,21,20,21
170 DATA 23,21,20,21,24,24,24,24
180 DATA 21,21,24,24,23,23,21,21
190 DATA 19,19,21,21,23,23,21,21,19,19,
21,21,23,23,21,21,19,19,18,18,16,16
200 DATA 69,69,69
210 DATA 16,16,17,17,19,19,19,19,21,19,
17,16,14,14,2,2
220 DATA 16,16,17,17,19,19,19,19,21,19,
17,16,14,14,2,2
230 DATA 12,12,14,14,16,16,16,16,17,16,
14,12,11,11,-1,-1
240 DATA 12,12,14,14,16,16,16,16,17,16,
14,12,11,11,-1,-1
250 DATA 11,9,8,9,12,12,12,12
260 DATA 14,12,11,12,16,16,16,16
270 DATA 17,16,15,16,23,21,20,21
280 DATA 23,21,20,21,24,24,24,24
290 DATA 21,21,23,23,24,24,23,23,21,21,
20,20,21,21
300 DATA 16,16,17,17,14,14,12,12,12,12
310 DATA 11,12,11,9,8,9,9,9,9,9
```

¡!

En las gráficas de las ondas sinusoidal y cuadrada, la ordenada expresa el «pulso» y la abscisa el «tiempo», llamándose «crestas» los picos de mayor altura y «valles» los de menor.

*

En el teclado de un piano existen un conjunto de teclas blancas correspondientes a las notas Do, Re, Mi, Fa, Sol, La y Si (C, D, E, F, G, A, B en nomenclatura internacional), que constituyen una «octava», y a las que corresponden ciertos tonos o frecuencia.

*

Se denomina «amplitud» de onda a la máxima distancia entre una cresta o un valle y el eje de abscisas, y «longitud» de onda a la distancia entre dos crestas o dos valles consecutivos.

TURBO TRANSMISION

FRECUENTEMENTE, al interrogar a cualquier usuario del Spectrum sobre su experiencia manejando el casete, contestará, casi con toda seguridad: «Es lento». La queja es, en cierto modo, justificada, aunque otros equipos similares llegan a agotar la paciencia durante las labores de carga y grabación, al utilizar velocidades de transmisión todavía menores.

Debe estar claro para todos, en este momento, que la velocidad de transmisión de los datos entre el Spectrum y el casete no depende en absoluto de la rapidez de giro de la cinta. Es más, si por alguna razón el arrastre de la cinta por el motor sufre fluctuaciones, puede acarrear fallos de interpretación por parte de nuestro microordenador al variar las frecuencias correspondientes al cero y uno lógicos.

Dicha velocidad de transmisión está controlada por las rutinas implementadas en la R. O. M. del Sistema. Al estar almacenadas en la memoria de sólo lectura, son fijas e inalterables por parte del usuario, pero como más adelante veremos, podemos actuar con habilidad sobre ellas, si previamente las hemos reubicado en la memoria R. A. M., donde procederemos a modificarlas.

Frecuentemente al preguntar a cualquier usuario del Spectrum sobre el manejo de un casete, contestará: «Es lento».

¡!

La subrutina ME-CONTRL ejecuta la orden de MERGE en tres etapas: carga del bloque de datos, mezcla de las líneas de los dos programas, y mezcla de las variables.

*

Entre las múltiples funciones que realiza la subrutina LD-BYTES, se encuentra la separación de las operaciones de carga y verificación, comparando los **bytes** uno a uno.

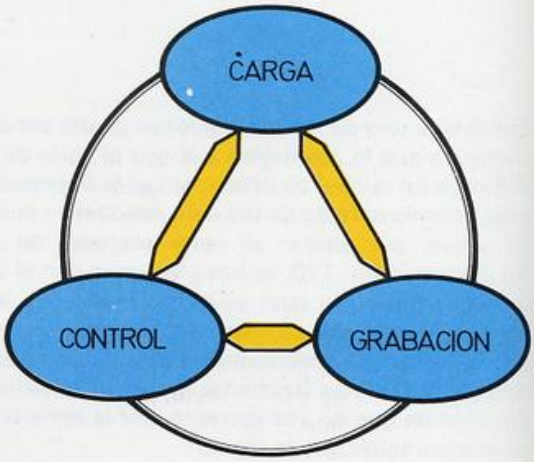
ESCRITO EN LA R. O. M.

Las rutinas de manejo del casete, situadas en la R. O. M. entre las direcciones decimales 1.218 y 2.547, están divididas en once subrutinas, que podemos agrupar en tres grandes grupos: grabación, carga y control de estas operaciones. Como ya estudiaremos más adelante, no son independientes entre sí, es decir, se efectúan constantes llamadas de una a las otras, y algunas pueden ser comunes a varios comandos.

Las rutinas que estamos tratando se encargan de gestionar cualquier operación que involucre el uso de instrucciones relacionadas con el almacenamiento de datos en casete. A partir de ahora, y siempre que no se indique lo contrario, nos referiremos a las direcciones y valores almacenados en ellas, en su formato decimal.

La entrada principal de esta zona del *firmware* se encuentra en la dirección 1.541, donde comienza la subrutina SAVE-ETC (grabar-etc), punto común utilizado por los cuatro comandos del BASIC, **SAVE, LOAD, MERGE** y **VERIFY**. La primera parte está destinada a la creación de la cabecera de información en el espacio de trabajo.

Las rutinas de manejo del casete ubicadas en la **ROM** *están divididas en tres grupos: grabación, carga y control, que efectúan constantes llamadas entre sí.*

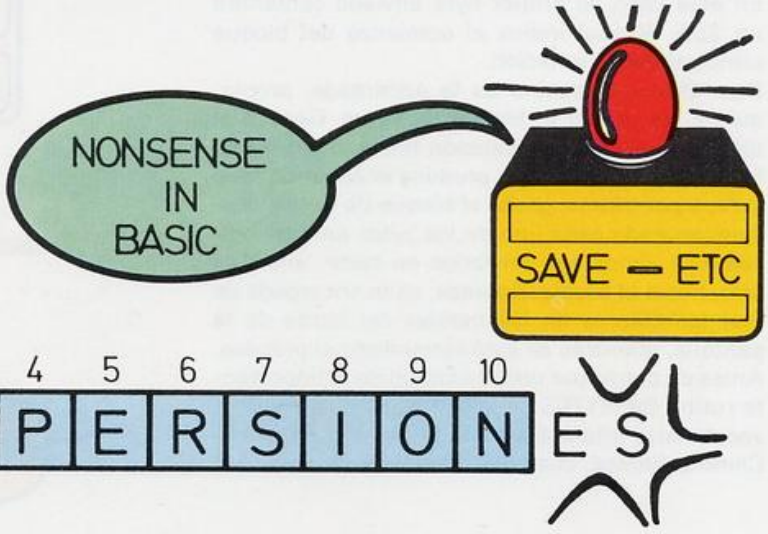


Primeramente, evalúa el tipo de comando; a continuación, reserva diecisiete localizaciones de memoria en el caso de **SAVE,** y treinta y cuatro para los otros tres.
Allí se almacenarán los datos concernientes al nombre de la información, longitud, etc. (ver comentario del programa INDEX). La razón por la cual al efectuar **LOAD, MERGE** o **VERIFY,** se reservan otros diecisiete bytes, queda justificada por la necesidad de comprobar, en estas operaciones, si la cabecera que nosotros generamos tras uno de estos comandos coincide con la recibida desde la cinta. Caso de no ser así, se emite el correspondiente mensaje de error o permanece a la espera de recibir una nueva cabecera.
Entre las funciones de la mencionada rutina de entrada, está comprobar si la longitud del nombre es mayor que diez caracteres, en cuyo caso emite el mensaje **F Invalid file name** (nombre no apto), si el comando era **SAVE,** o desprecia los caracteres en exceso, si se trata de cualquiera de los otros tres. Se cerciora así mismo si la cadena-nombre es nula, no aceptándola en el caso de grabación, pero sí en los de lectura.
Por otra parte, la rutina SAVE-ETC no admite la posibilidad de **MERGE,** cuando se trata de bloques de bytes, pantallas o datos; comprueba si un programa ha sido almacenado con autoejecución mediante **LINE,** evitando igualmente la opción de mezcla en este caso. Otra de sus misiones es imprimir en la pantalla el mensaje apropiado sobre el tipo de lectura: **Program, Bytes, Number array o Character array,** seguido del nombre, y finalmente, si todo fue bien, da paso a las subrutinas que gestionan cada comando.

En **ROM** *se encuentran las rutinas de manejo del casete.*

SUBRUTINAS DE GRABACION

Una vez verificadas todas las funciones anteriores, si estamos efectuando una orden **SAVE,** la rutina anterior salta a la dirección 2.416, donde comienza la subrutina SA-CONTRL (control de



SAVE-ETC *comprueba si la longitud del nombre introducido es mayor de 10 caracteres. En ese caso, si el comando es* SAVE *emite el mensaje* **Nonsense in basic (con Interface 1)** *o* **Invalid file name.**

¡!

Las rutinas implantadas en ROM controlan la velocidad, pero podemos actuar sobre ésta reubicándolas en la memoria RAM.

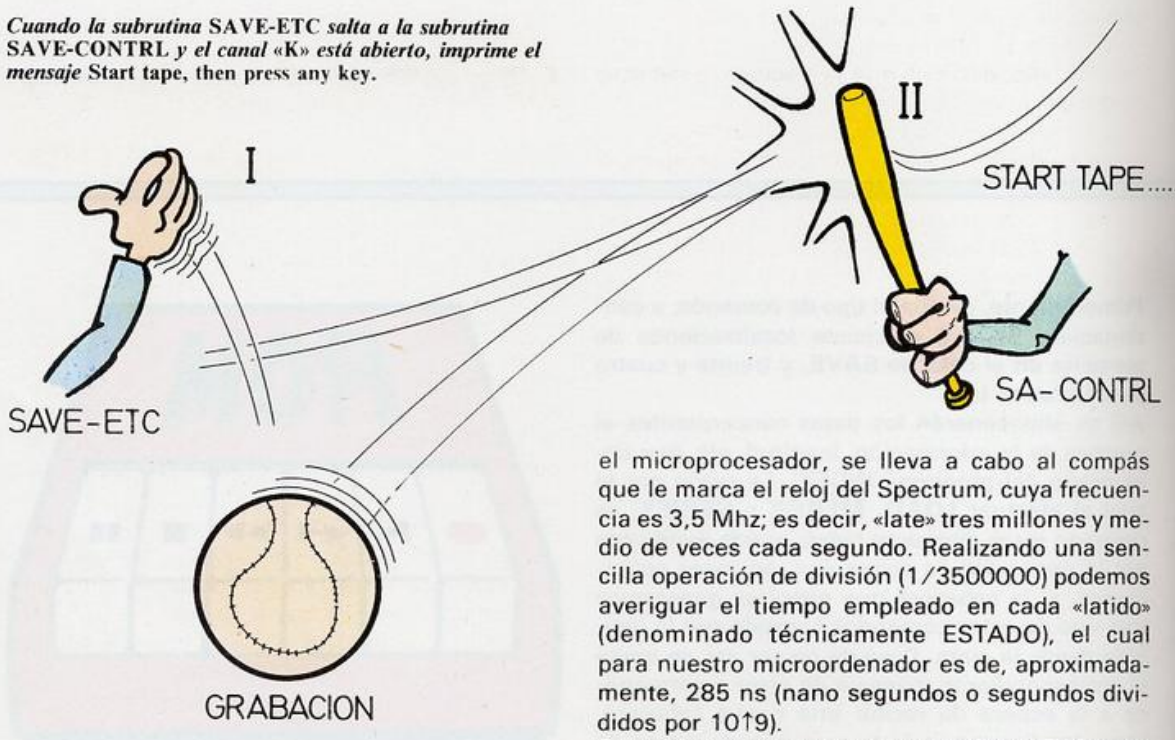
*

Las rutinas de control del casete residen en la ROM del ordenador.

*

Las rutinas de manejo de casete están divididas en tres grandes grupos: grabación, carga y control.

Cuando la subrutina **SAVE-ETC** *salta a la subrutina* **SAVE-CONTRL** *y el canal «K» está abierto, imprime el mensaje* **Start tape, then press any key.**



grabación). Esta se asegura que el canal «K», correspondiente al teclado, está abierto e imprime el mensaje: **Start tape, then press any key** (ponga en marcha el casete y pulse cualquier tecla), que señala el comienzo de toda grabación en cinta, para posteriormente esperar la pulsación de una tecla.

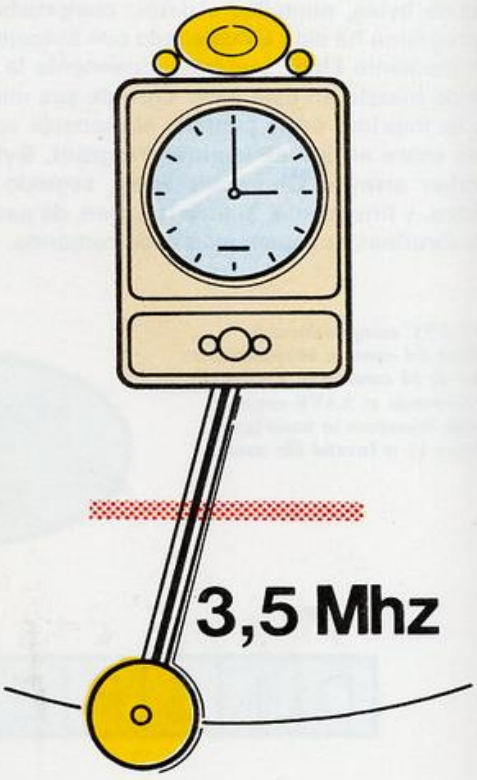
Al hacerlo, realiza dos llamadas sucesivas a la subrutina SA-BYTES (grabar bytes), ubicada a partir de la dirección 1218. En la primera de ellas envía los 17 bytes que componen la cabecera, a los cuales añade a su comienzo, a modo de bandera, un código 0 (cero) que indica el tipo de información que sigue (cabecera), y por detrás un byte de paridad. A continuación, produce el espacio muerto entre los tonos guías, busca la longitud y comienzo del bloque de datos a ser grabado, y realiza la segunda llamada a SA-BYTES. En este caso, el primer byte enviado contendrá un 255, el cual indica el comienzo del bloque principal de información.

Esta última subrutina es la encargada, propiamente, de grabar el bloque de datos. Genera el tono guía previo a la emisión hacia el casete de la cabecera, graba esta, produce el segundo tono guía, y por último, graba el bloque de datos, descomponiendo cada uno de los bytes en bits individuales. Estos son enviados en serie, uno tras otro, hacia el casete. Además, es la encargada de fijar los colores en las bandas del borde de la pantalla, mientras se está ejecutando el proceso.

Antes de continuar con el estudio de la importante rutina SA-BYTES, precisamos ampliar nuestro vocabulario informático con la palabra ESTADO. Como sabemos, cualquier operación que ejecute el microprocesador, se lleva a cabo al compás que le marca el reloj del Spectrum, cuya frecuencia es 3,5 Mhz; es decir, «late» tres millones y medio de veces cada segundo. Realizando una sencilla operación de división (1/3500000) podemos averiguar el tiempo empleado en cada «latido» (denominado técnicamente ESTADO), el cual para nuestro microordenador es de, aproximadamente, 285 ns (nano segundos o segundos divididos por 10↑9).

Pues bien, cuando la subrutina SA-BYTES tiene que enviar ceros y unos hacia el casete, lo hace siempre de la misma manera: mantiene la señal que sale por MIC en alto durante 855 estados, y en bajo durante otros 855 latidos, si está enviando un cero; mientras que si el bit a grabar es un

Cualquier operación que ejecute el microprocesador, es llevada a cabo al compás que le marca el reloj del Spectrum regulado a 3,5 Mhz.



¡!

SAVE-ETC comprueba si la longitud del nombre introducido es mayor de 10 caracteres. En este caso, si el comando es **SAVE** emite un mensaje de error, si se trata de cualquiera de los otros tres, se desprecian los caracteres en exceso.

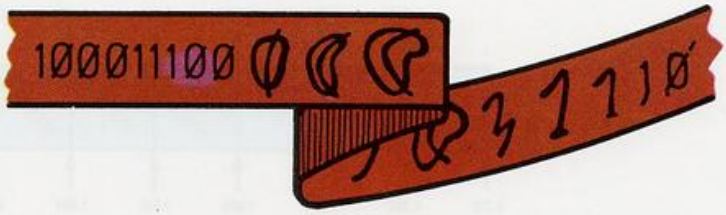
*

La velocidad de grabación se obtiene como promedio, ya que el número de ceros y unos es prácticamente el mismo, compensándose los tiempos de grabación.

*

La rutina LD-BLOCK tiene como misión averiguar si la información que recibe pertenece a una cabecera o a un bloque de datos, y fijar los colores rojo y cyan del borde de la pantalla.

uno, sigue el mismo proceso, pero a los 855 anteriores en alto-bajo, les añade otros 855 estados. Por ello, decimos que la frecuencia del cero es doble a la del uno lógico.
Este sistema de grabación da pie a un hecho curioso: cualquier zona de memoria, por ejemplo una pantalla, llena de ceros tarda la mitad de tiempo en ser grabada o cargada que una llena de unos (255). Así pues, la velocidad de grabación es un promedio, dado que estadísticamente, el número de ceros y unos en cualquier grabación es prácticamente igual y, por tanto, sus tiempos de grabación se compensan.

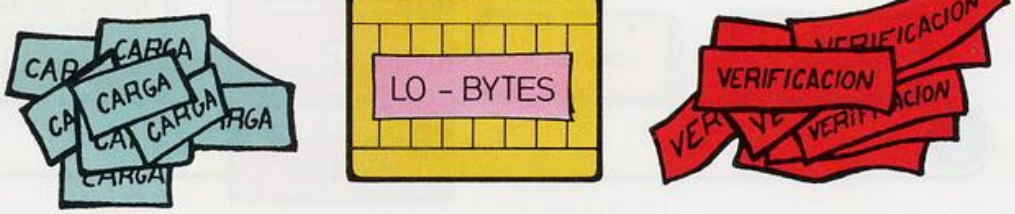


Si el arrastre de la cinta sufre fluctuaciones, puede acarrear fallos de interpretación en nuestro ordenador.

SUBRUTINAS DE CARGA

Una vez comparadas las cabeceras, la indicada por nosotros tras un comando **LOAD, VERIFY** o **MERGE** y la que proviene de la cinta, en caso de coincidencia, la rutina SAVE-ETC pasa a considerar, independientemente, cada uno de los comandos.
Si la orden efectuada fue **LOAD,** salta a la subrutina LD-CONTRL (control de carga), ubicada a partir de la dirección 2.056. Es la encargada de supervisar la carga de un programa BASIC y sus variables, así como variables suscritas (matrices). Comienza considerando si se trata de un programa BASIC o de una matriz. En el primer caso, emite el mensaje **4 Out of memory,** si no dispone de suficiente espacio en la memoria para almacenarlo. Si lo hay, lo reserva, de manera que tengan cabida tanto el programa como sus variables. Finalmente, salta a la subrutina LD-BLOCK (cargar bloque) ubicada entre las direcciones 2.050 y 2.055.
Esta subrutina es común en todos los procesos que impliquen la carga de cualquier bloque de información. Su misión consiste en realizar una llamada a la subrutina LD-BYTES (carga de bytes) que ocupa las direcciones de la R. O. M. comprendidas entre la 1.366 y la 1.506. Esta rutina es llamada dos veces: una para cargar la cabecera de información y otra para los bloques de datos.
Su misión consiste, como primer paso, en averiguar si la información que recibe pertenece a una cabecera o a un bloque de datos. Además, se hace distinción entre los comandos **LOAD** y **VERIFY.** A continuación, toma un byte del *port* 254 (EAR), y lo interpreta bit a bit, lo cual constituye la lectura de datos propiamente dicha. Es también la encargada de fijar los colores rojo y cyan que observamos en el borde de la pantalla, cuando la subrutina, tras ejecutar cualquier comando de carga, está a la espera de recibir la información.
Está preparada para recibir la información contenida en la cabecera primero, y después, tras el segundo tono guía, la del bloque principal de datos. Para ello se realizan varias llamadas a otras subrutinas, denominadas LD-EDGE-1 y LD-EDGE-2, comprendidas entre las direcciones 1.507 y 1.540. La traducción podría ser «cargar bordes», quizás haciendo referencia a que conforme se va recibiendo la información, las bandas laterales de la pantalla van variando según se trate de unos, de ceros o del tono guía para el sincronismo de cabecera.
Estas últimas rutinas son las más importantes en todo proceso en el cual intervenga la carga o verificación de un bloque de datos. Son las encargadas de medir la duración de un pulso completo, así como el tiempo empleado antes de la siguiente transición entre los estados en que el pulso se encuentra alto o bajo. En las páginas an-

La subrutina **LD-BYTES** *separa las funciones de carga y verificación.*



¡!

SA-BYTES es la subrutina encargada de grabar el bloque de datos, generar los tonos guías, descomponer los **bytes** en **bits,** y fijar los colores en las bandas del borde de la pantalla.

*

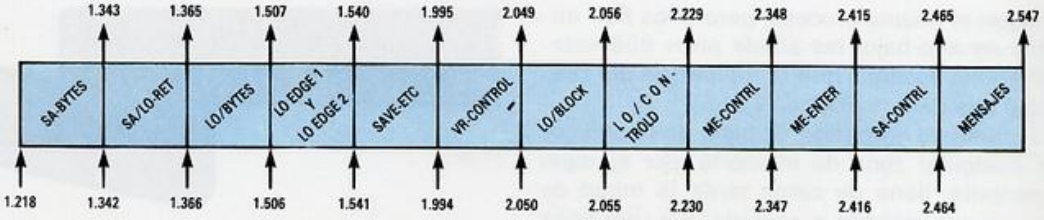
El punto de entrada común utilizado por **SAVE, LOAD** y **VERIFY** es la subrutina SAVE-ETC.

*

La rutina SA-CONTRL se asegura de que el canal del teclado está abierto emitiendo el mensaje: **Start tape, then press any key.**

*

SAVE-ETC no admite el **MERGE** de bloques de **bytes,** pantallas o datos.



Mapa de rutinas de **ROM** *para el manejo de casetes.*

teriores, bajo el epígrafe «transmisión de datos», se dan las explicaciones suficientes sobre el aspecto de los pulsos correspondientes al cero, uno, y cabecera de transmisión.

La subrutina LD-BYTES, completa su trabajo separando las funciones de carga y verificación. Durante esta última operación, compara los bytes que recibe de la cinta, uno a uno, con los almacenados en la memoria, emitiendo el mensaje **R Tape loading error** (error de lectura) cuando no coinciden ambos. Conviene aclarar, que los bytes recibidos no se almacenan en la memoria R. A. M., sino que se comparan directamente.

La supervisión de este proceso recae en la subrutina VR-CONTRL (control de verificación), ubicada entre las direcciones 1.995 y 2.049. Es utilizada, además, como rutina de control en la carga de programas que han sido grabados con las especificaciones **SCREEN$** o **CODE**. Es decir, por ejemplo, para recuperar una pantalla, el proceso sería el mismo descrito anteriormente para un programa BASIC, sólo que las rutinas de control son diferentes.

SUBRUTINAS DE MEZCLA

Cuando efectuamos una orden **MERGE** (mezcla) la rutina SAVE-ETC, tras realizar todos los pasos señalados anteriormente, entre ellos, comprobar si se trata de un programa BASIC salta a ME-CONTRL (control de mezcla), situada a partir de la dirección de memoria 2.230.

Esta subrutina se ejecuta en tres etapas independientes:

1) Carga el bloque de datos en el espacio de trabajo.
2) Mezcla las líneas del nuevo programa, con el antiguo presente en la memoria.
3) Mezcla las variables.

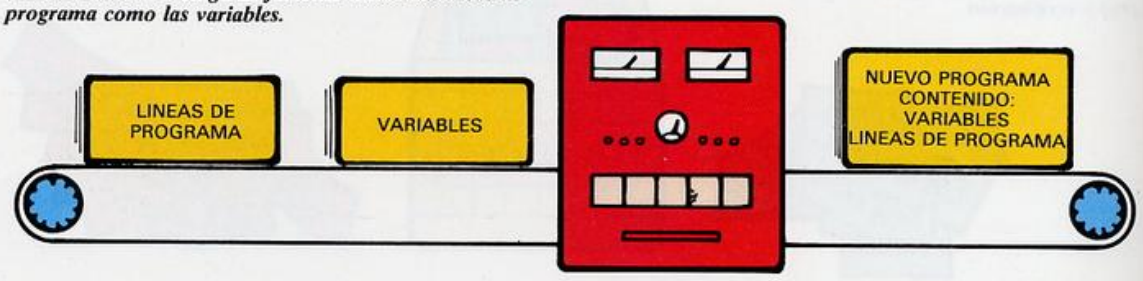
La primera parte, reserva el espacio de memoria suficiente para albergar el programa, y realiza una llamada a la subrutina LD-BLOCK, encargada de cargar el nuevo bloque de datos en este espacio.

Durante el proceso de unión, compara uno a uno, los números de línea entre los programas nuevo y antiguo. Si coinciden, la dirección de comienzo, tanto de la antigua como de la nueva será la misma. A continuación, averigua la dirección de memoria donde introducir la nueva. En ese momento realiza una llamada a la subrutina ME-ENTER, ubicada entre las direcciones 2.348 y 2.415, la cual se encarga de fusionar tanto las líneas de programa, como las variables. Por ello, cuando ejecutamos una orden **MERGE,** si existen números de línea idénticos en los dos programas, serán los del nuevo los que prevalezcan.

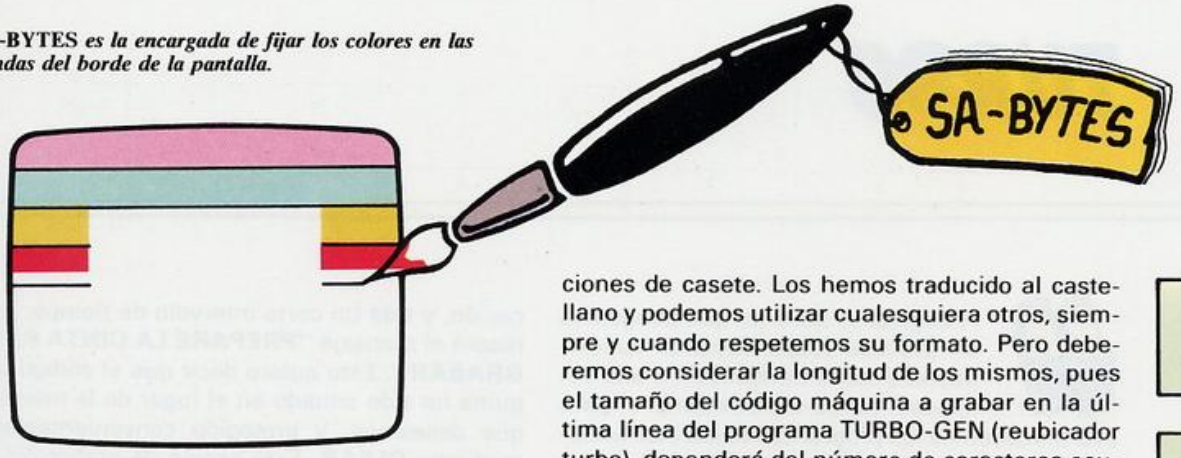
Cuando ha terminado de mezclar todas las líneas, comienza un proceso similar con las variables de ambos programas. Compara los nombres byte a byte, y de ser iguales, el contenido de la nueva quedará almacenado, perdiéndose el de la antigua. Una llamada a ME-ENTER, cada vez que sustituya o añada una variable, completa el proceso.

Finalmente, hay que indicar que la subrutina SA/LD-RET (1.343 a 1.365), retorno de cargar/salvar, es común a ambos procesos, devolviendo al borde su color original y leyendo por última vez si la tecla **BREAK** fue pulsada. Tras ella,

ME-ENTER *se encarga de fusionar tanto las líneas de programa como las variables.*



SA-BYTES *es la encargada de fijar los colores en las bandas del borde de la pantalla.*



se devuelve el control al BASIC al terminar el uso de las rutinas de manejo del casete.
Entre las direcciones 2.465 y 2.547, están almacenados los mensajes que aparecen durante los procesos relativos al almacenamiento de información en la grabadora, tales como **Start tape, then press any key, program:**, etc.

REUBICANDO

El programa «Reubicador Turbo» que aparece en las próximas páginas, transfiere la zona de la R. O. M. que contiene todas las rutinas comentadas anteriormente, menos la tabla de mensajes, a la parte de la R. A. M. escogida por nosotros. Una vez aquí, se efectúan los cambios destinados a modificar la velocidad de transmisión de los datos entre Spectrum y casete.
En primer lugar, hemos realizado una nueva lista de los mensajes relacionados con las operaciones de casete. Los hemos traducido al castellano y podemos utilizar cualesquiera otros, siempre y cuando respetemos su formato. Pero deberemos considerar la longitud de los mismos, pues el tamaño del código máquina a grabar en la última línea del programa TURBO-GEN (reubicador turbo), dependerá del número de caracteres ocupado por dichos mensajes.
Las líneas **DATA** que contienen valores numéricos son las que encierran el verdadero secreto del método de transmisión de datos a velocidad variable. Cada una está formada por parejas de valores: el primero es una dirección y el segundo el nuevo valor almacenado en ella. La primera línea corresponde a las constantes de tiempo que influyen durante el proceso de grabación, y las siguientes a las relacionadas con las operaciones de carga.
A continuación, se procede a reubicar todos las instrucciones en código máquina de las rutinas de manejo del casete que comporten saltos absolutos o llamadas a otras subrutinas. No queda otros remedio que hacerlo así, pues al estar nuestra rutina ubicada en la R. A. M., los saltos y llamadas deben ser entre direcciones de ésta para asegurar su perfecto funcionamiento.
Finalmente, los **POKE** K+1.935 y K+1.936, son los encargados de fijar el nuevo origen para los mensajes que aparecen en cualquier operación de manejo del casete. Si no actuamos sobre estas direcciones, será vano nuestro esfuerzo al tratar de conseguir, por ejemplo, el mensaje «Pon en marcha la grabadora» y seguirá apariendo el ya conocido, **Start tape, then press any key.**
Pongamos en práctica nuestros conocimientos en la sección de PROGRAMA.

¡!

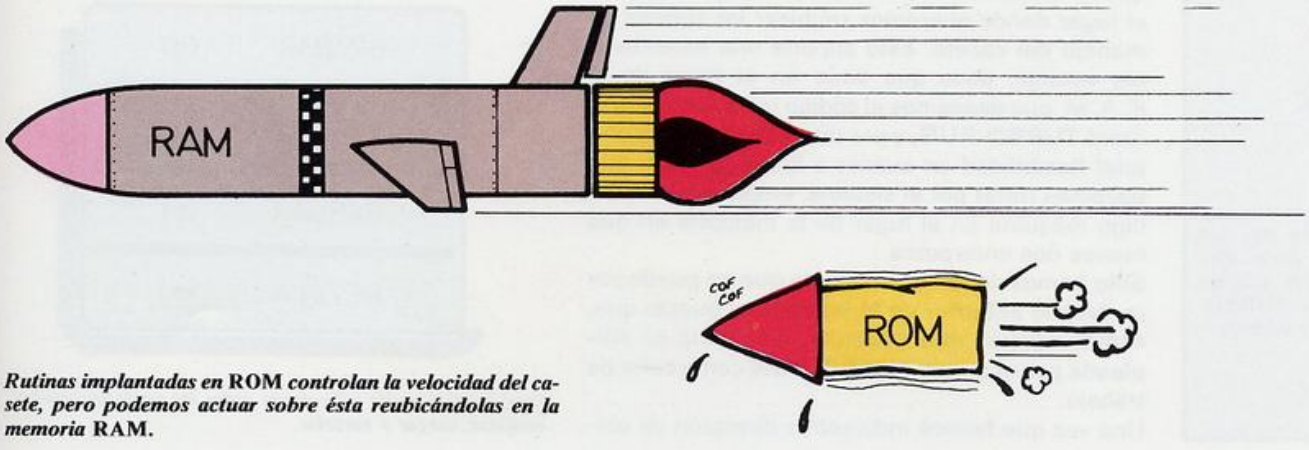
La subrutina LD-CONTRL es la encargada de supervisar la carga de un programa BASIC, sus variables y matrices.

*

La frecuencia de grabación del 0 lógico es doble que la del 1.

*

La subrutina encargada de fusionar las líneas de programa y las variables de ME-ENTER.



***Rutinas implantadas en* ROM *controlan la velocidad del casete, pero podemos actuar sobre ésta reubicándolas en la memoria* RAM.**

TURBO

HA llegado la hora de que pongamos en práctica todos nuestros conocimientos sobre transmisión a alta velocidad (turbo-transmisión), y para ello utilizaremos los programas cuyas instrucciones de manejo detallamos a continuación.

Nuestro sistema de turbo-transmisión se fundamenta, como hemos visto en las páginas anteriores, en la reubicación de la zona de R. O. M. que contiene las rutinas de manejo del casete, para posteriormente realizar sobre ellas una modificación que aumente la velocidad de transmisión, en nuestro caso, nada más y nada menos que al doble de la normal: 3.000 baudios.

Disponemos de dos programas bien diferenciados: el primero de ellos, denominado **TURBO-GEN,** realiza la función de reubicar las rutinas afectadas a una zona de la R. A. M., generando un código máquina que será posteriormente utilizado por el segundo programa **(TURBO-RUN),** cuya misión es proporcionar un apoyo BASIC para la cómoda utilización del sistema de turbo-transmisión.

¡!

La turbo-transmisión se fundamenta en la reubicación de la ROM, para realizar una modificación que aumente la velocidad de transmisión en el doble.

*

Disponemos de dos programas para ello, **TURBO-GEN** (código máquina) y **TURBO-RUN** (apoyo BASIC).

TURBO-GEN

El manejo de **TURBO-GEN** es bien sencillo; casi podríamos decir que automático. El único dato que nos reclama, al comienzo de su ejecución, es el lugar donde queremos reubicar las rutinas de manejo del casete. Esto supone una considerable ventaja, dado que sitúa en el lugar de la R. A. M. que deseemos el código máquina que utilizará **TURBO-RUN,** y por tanto, proporciona una total flexibilidad en cuanto a los programas que podemos tratar por el sistema, emplazando el código máquina en el lugar de la memoria en que menos nos entorpezca.

Sólo hemos de tener en cuenta que no puede ser reubicado al buffer de la impresora, puesto que, como veremos más adelante, esta zona es empleada por el propio **TURBO-RUN** como zona de trabajo.

Una vez que hemos indicado la dirección de ubicación, y tras un corto intervalo de tiempo, aparecerá el mensaje **"PREPARE LA CINTA PARA GRABAR".** Esto quiero decir que el código máquina ha sido situado en el lugar de la memoria que deseamos, y protegido convenientemente mediante **CLEAR.** Esta opción de grabación del código máquina generado, nos evitará sucesivas veces recurrir al programa **TURBO-GEN.**

Pese a todo, si por cualquier motivo no nos interesara grabarlo, podemos interrumpir el programa mediante **BREAK,** y borrar el soporte BASIC con **NEW,** en la certeza de que el código máquina se encuentra correctamente reubicado y protegido por el RAMTOP.

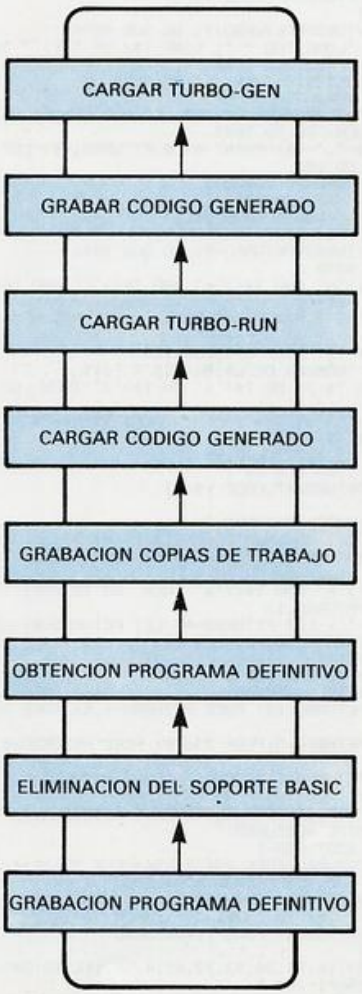
TURBO-RUN

El sistema empleado por **TURBO-RUN** es de una gran simplicidad: se simula en una zona de trabajo (el buffer de la impresora) la instrucción BASIC que se desea ejecutar, y a continuación se efectúa una sencilla llamada al punto de reubicación de **SAVE-ETC,** con lo cual el resto del trabajo corre a cargo del firmware reubicado en la R. A. M.

Sabiendo esto, y con unos pequeños conocimientos de la estructura de las instrucciones BASIC,

Este menú permite seleccionar entre las opciones: grabar, verificar, cargar y mezclar.

no nos será nada complicado utilizar el código máquina de **TURBO-GEN** sin necesidad de tener que recurrir a **TURBO-RUN**.
Para aquellos a los que nos resultaría complicado generar la instrucción pertinente en el área de trabajo, el programa **TURBO-RUN** realiza el trabajo en nuestro lugar, asegurándose además de la correcta sintaxis del comando.
El pograma **TURBO-RUN** debe ser ejecutado a partir de la línea 9881, en la cual le indicaremos el lugar donde se encuentra ubicado, o se va a ubicar, el código máquina. Acto seguido se aguarda la entrada del código máquina desde el casete. Si este se encontrara en memoria, debido a la reciente ejecución de **TURBO-GEN**, el programa puede ser ya interrumpido y se encuentra listo para el funcionamiento.
Llegados a este punto, podemos comenzar a programar nuestro propio BASIC, o bien efectuar un **MERGE** del programa que deseamos grabar a alta velocidad, teniendo en cuenta que el soporte de **TURBO-RUN** ocupa todas las líneas desde la 9800 a la 9893, ambas inclusive, por lo cual dichos números de instrucción no pueden ser empleados para nuestro propio uso.
La finalidad de nuestro sistema de turbo-transmisión es aumentar la velocidad con que utilizamos el casete para las copias de trabajo, de forma que una vez que hemos llegado a una versión definitiva de nuestro programa, podamos eliminar el soporte de **TURBO-RUN**, y grabar a velocidad normal, y más segura, la copia definitiva del programa.
Así pues, cuando grabamos programas en turbo, el soporte BASIC de **TURBO-RUN** se graba con ellos. ¿Cómo hemos de hacer para eliminarlo en la última copia? Dado que esta será efectuada a velocidad normal, no precisamos ninguna de las líneas de **TURBO-RUN**, si bien es cierto que eliminar, una por una, las 94 líneas que lo componen es muy pesado.
La solución se encuentra en la línea 9892; ésta contiene un breve código máquina que destruye todo vestígio de la presencia de **TURBO-RUN**. Sólo hemos de editar esta línea y suprimirle el número de instrucción mediante **DELETE**, para que sea ejecutada en forma de comando directo.



Organigrama de utilización de los programas **TURBO-GEN** *y* **TURBO-RUN**.

Este menú permite determinar qué clase de datos deseamos almacenar o recuperar del casete.

¡!

Para obtener una copia de trabajo accederemos a la línea 9800, seleccionando la opción adecuada.

*

Si no nos interesa grabar el código máquina, después de situado en la memoria, podemos interrumpir el programa para borrar el soporte BASIC con **NEW**.

Una vez que **TURBO-RUN** ha sido inicializado, es decir, se ha efectuado la ejecución a partir de 9881 y el código máquina se encuentra en la memoria, ya no necesitamos recurrir más veces a este punto hasta que no borremos la memoria del ordenador.

Cada vez que deseemos realizar una copia de trabajo deberemos acceder a la línea 9800, donde se sucederán una serie de menús en los cuales podremos escoger de manera muy sencilla el comando que deseamos ejecutar.

¡!

TURBO-GEN solicita el lugar donde reubicar las rutinas de manejo del casete.

*

Al efectuar un **MERGE** del programa que deseamos grabar a alta velocidad, no debemos emplear las líneas del 9800 al 9893.

*

No podemos reubicar el buffer de impresora, por necesitarlo **TURBO-RUN** como zona de trabajo.

*

Si tenemos una versión definitiva de un programa, y deseamos eliminar el soporte BASIC de **TURBO-RUN**, basta editar la línea 9892, suprimiendo el número de instrucción para ejecutarla en comando directo.

```
9800 REM REUBICADOR TURBO - C.DE LA OSSA & F.LOPEZ MA
RTINEZ
9801 INPUT "DIRECCION DE UBICACION ? ";D
9802 RANDOMIZE D
9803 CLEAR D-1: LET D=PEEK 23670+256*PEEK 23671
9804 FOR I=23296 TO 23307
9805 READ A: POKE I,A
9806 NEXT I
9807 DATA 33,194,4,17,PEEK 23670,PEEK 23671,1,224,4,2
37,176,201
9808 RANDOMIZE USR 23296
9809 LET K=D-1218: LET A=2466+K
9810 FOR I=0 TO 4
9811 READ A$
9812 FOR J=1 TO LEN A$
9813 POKE A,CODE A$(J)+(128 AND J=LEN A$): BEEP .01,0
9814 LET A=A+1
9815 NEXT J
9816 NEXT I
9817 DATA "Pon en marcha la grabadora"
9818 DATA CHR$ 13+"Programa: "
9819 DATA CHR$ 13+"Matriz numerica: "
9820 DATA CHR$ 13+"Matriz de cadena: "
9821 DATA CHR$ 13+"Memoria: "
9822 FOR I=0 TO 19
9823 READ A,B
9824 POKE A+K,B: BEEP .01,10
9825 NEXT I
9826 DATA 1238,2,1245,15,1263,13,1271,14,1289,1
9827 DATA 1272,29,1305,33,1311,31,1326,24,1339,29
9828 DATA 1371,15,1384,2,1441,3
9829 DATA 1446,216,1479,218,1487,230,1492,216,1512,12
9830 DATA 1536,8
9831 DATA 1541,0
9832 FOR I=0 TO 28
9833 READ A
9834 LET B=PEEK A+256*PEEK (A+1)
9835 RANDOMIZE K+B: LET RB=PEEK 23670: LET RA=PEEK 23
671
9836 RANDOMIZE K+A: LET BB=PEEK 23670: LET BA=PEEK 23
671
9837 POKE BB+256*BA,RB: POKE BB+256*BA+1,RA: BEEP .01
,20
9838 NEXT I
9839 DATA 1219,1254,1276,1292,1320,1336,1375,1389,140
4,1411
9840 DATA 1426,1436,1483,1494,1508,1694,1887,1903,198
8
9841 DATA 1993,2051,2161,2228,2251,2284,2344,2424,244
3,2463
9842 POKE K+1935,89: POKE K+1936,239
9843 PRINT "PREPARE LA CINTA PARA GRABAR"
9844 BEEP 3,30
9845 SAVE "C/M TURBO"CODE D,1330
```

```
9800 REM TURBO-RUN - C.DE LA OSSA & F.LOPEZ MARTINEZ
9801 CLS
9802 PRINT  INK 9'' INVERSE 1;"G"; INVERSE 0;"RABAR''
'' INVERSE 1;"V"; INVERSE 0;"ERIFICAR''' INVERSE 1;"C"
; INVERSE 0;"ARGAR''' INVERSE 1;"M"; INVERSE 0;"EZCLA
R"
9803 LET T$=INKEY$
9804 IF T$="G" OR T$="g" THEN  LET V$="SAVE ": POKE 2
3728,0: GO TO 9809
9805 IF T$="V" OR T$="v" THEN  LET V$="VERIFY ": POKE
 23728,2: GO TO 9809
9806 IF T$="C" OR T$="c" THEN  LET V$="LOAD ": POKE 2
3728,1: GO TO 9809
9807 IF T$="M" OR T$="m" THEN  LET V$="MERGE ": POKE
23728,3: GO TO 9809
9808 GO TO 9803
9809 PRINT AT 0,0;V$: LET PTTURBO=LEN V$
9810 INPUT "NOMBRE ? ";T$: IF LEN T$>10 OR T$="" AND
NOT PEEK 23728 THEN  GO TO 9810
9811 LET T$=CHR$ 34+T$+CHR$ 34
9812 FOR T=1 TO LEN T$: POKE T+23311,CODE T$(T): NEXT
 T: LET PKTURBO=T+23311
9813 PRINT AT 0,PTTURBO;T$: LET PTTURBO=PTTURBO+LEN T
$
9814 PRINT AT 0,0; INK 9'' INVERSE 1;"P"; INVERSE 0;"
ROGRAMA''' INVERSE 1;"I"; INVERSE 0;"MAGEN DE PANTALL
A''' INVERSE 1;"M"; INVERSE 0;"EMORIA''' INVERSE 1;"N
"; INVERSE 0;"UMERICA (MATRIZ)''' INVERSE 1;"C"; INVE
RSE 0;"ADENA (MATRIZ)"
9815 POKE 23303,224+PEEK 23728
9816 LET T$=INKEY$
9817 IF T$<>"P" AND T$<>"p" THEN  GO TO 9830
9818 IF PEEK 23728 THEN  GO TO 9870
9819 INPUT "AUTOEJECUCION ? "; LINE T$: IF T$="" THEN
 LET T$="0"
9820 LET VARTURBO=INT VAL T$: IF VARTURBO<0 OR VARTUR
BO>9999 THEN  GO TO 9819
9821 IF NOT VARTURBO THEN  GO TO 9870
9822 LET V$=" LINE "+T$: PRINT AT 0,PTTURBO;V$
9823 POKE PKTURBO,202
9824 FOR T=1 TO LEN T$
9825 POKE PKTURBO+T,CODE T$(T)
9826 NEXT T
9827 LET PKTURBO=PKTURBO+T
9828 GO SUB 9876
9829 GO TO 9870
9830 IF PEEK 23728=3 THEN  GO TO 9816
9831 IF T$<>"I" AND T$<>"i" THEN  GO TO 9835
9832 LET V$="SCREEN$ ": PRINT AT 0,PTTURBO;V$
9833 POKE PKTURBO,170: LET PKTURBO=PKTURBO+1
9834 GO TO 9870
9835 IF T$<>"M" AND T$<>"m" THEN  GO TO 9858
9836 LET V$="CODE ": PRINT AT 0,PTTURBO;V$: LET PTTUR
BO=PTTURBO+5
9837 POKE PKTURBO,175
9838 INPUT "DIRECCION DE INICIO ? "; LINE T$: IF T$<>
"" THEN  GO TO 9841
9839 IF PEEK 23728 THEN  LET PKTURBO=PKTURBO+1: GO TO
 9870
9840 GO TO 9838
9841 LET VARTURBO=INT VAL T$: IF VARTURBO<0 OR VARTUR
BO>65535 THEN  GO TO 9838
9842 PRINT AT 0,PTTURBO;T$: LET PTTURBO=PTTURBO+LEN T
$
9843 FOR T=1 TO LEN T$
9844 POKE PKTURBO+T,CODE T$(T)
9845 NEXT T
9846 LET PKTURBO=PKTURBO+T: GO SUB 9876
9847 INPUT "LONGITUD ? "; LINE T$: IF T$<>"" THEN  GO
 TO 9850
9848 IF PEEK 23728 THEN  GO TO 9870
9849 GO TO 9847
9850 LET VARTURBO=INT VAL T$: IF VARTURBO<1 OR VARTUR
BO>65535 THEN  GO TO 9847
9851 LET V$=","+T$: PRINT AT 0,PTTURBO;V$: LET PTTURB
O=PTTURBO+LEN V$
9852 LET PKTURBO=PKTURBO-1
9853 FOR T=1 TO LEN V$
9854 POKE PKTURBO+T,CODE V$(T)
9855 NEXT T
9856 LET PKTURBO=PKTURBO+T: GO SUB 9876
9857 GO TO 9870
9858 IF T$<>"N" AND T$<>"n" AND T$<>"C" AND T$<>"c" T
HEN  GO TO 9816
9859 PRINT AT 0,PTTURBO;" DATA ": LET PTTURBO=PTTURBO
+6
9860 LET V$=T$: POKE PKTURBO,228
9861 INPUT "NOMBRE DE LA MATRIZ ? ";T$
9862 IF LEN T$<>1 OR T$<"A" OR T$>"Z" THEN  GO TO 986
1
9863 IF V$="C" OR V$="c" THEN  LET T$=T$+"$"
9864 LET V$=T$+"()"
9865 PRINT AT 0,PTTURBO;V$
9866 FOR T=1 TO LEN V$
9867 POKE PKTURBO+T,CODE V$(T)
9868 NEXT T
9869 LET PKTURBO=PKTURBO+T
9870 PRINT #0;"COMANDO CORRECTO (S/N) ?"
9871 LET T$=INKEY$: IF T$="N" OR T$="n" THEN  GO TO 9
800
9872 IF T$<>"S" AND T$<>"s" THEN  GO TO 9871
9873 POKE PKTURBO,13
9874 INPUT "": LET PTTURBO=0: LET PKTURBO=0: LET VART
URBO=0: LET ADDTURBO=0: LET T$="": LET V$="": FOR T=0
 TO 1: NEXT T
9875 RANDOMIZE USR 23296
9876 RANDOMIZE VARTURBO
9877 POKE PKTURBO,14: POKE PKTURBO+1,0: POKE PKTURBO+
2,0
9878 POKE PKTURBO+3,PEEK 23670: POKE PKTURBO+4,PEEK 2
3671
9879 POKE PKTURBO+5,0: LET PKTURBO=PKTURBO+6
9880 RETURN
9881 INPUT "UBICACION DEL TURBO ? ";ADDTURBO
9882 RANDOMIZE ADDTURBO
9883 CLEAR ADDTURBO-1
9884 LET ADDTURBO=PEEK 23670+256*PEEK 23671
9885 LOAD "C/M TURBO"CODE ADDTURBO
9886 RANDOMIZE ADDTURBO+324
9887 FOR T=23296 TO 23311
9888 READ VARTURBO: POKE T,VARTURBO
9889 NEXT T
9890 DATA 33,16,91,34,93,92,62,0,50,116,92,205,PEEK 2
3670,PEEK 23671,207,8
9891 GO TO 9801
9892 RESTORE 9893: FOR T=23296 TO 23313: READ PKTURBO
: POKE T,PKTURBO: NEXT T: RANDOMIZE USR 23296: CLEAR
9893 DATA 33,72,38,205,110,25,229,33,166,38,205,110,2
5,209,205,229,25,201
```