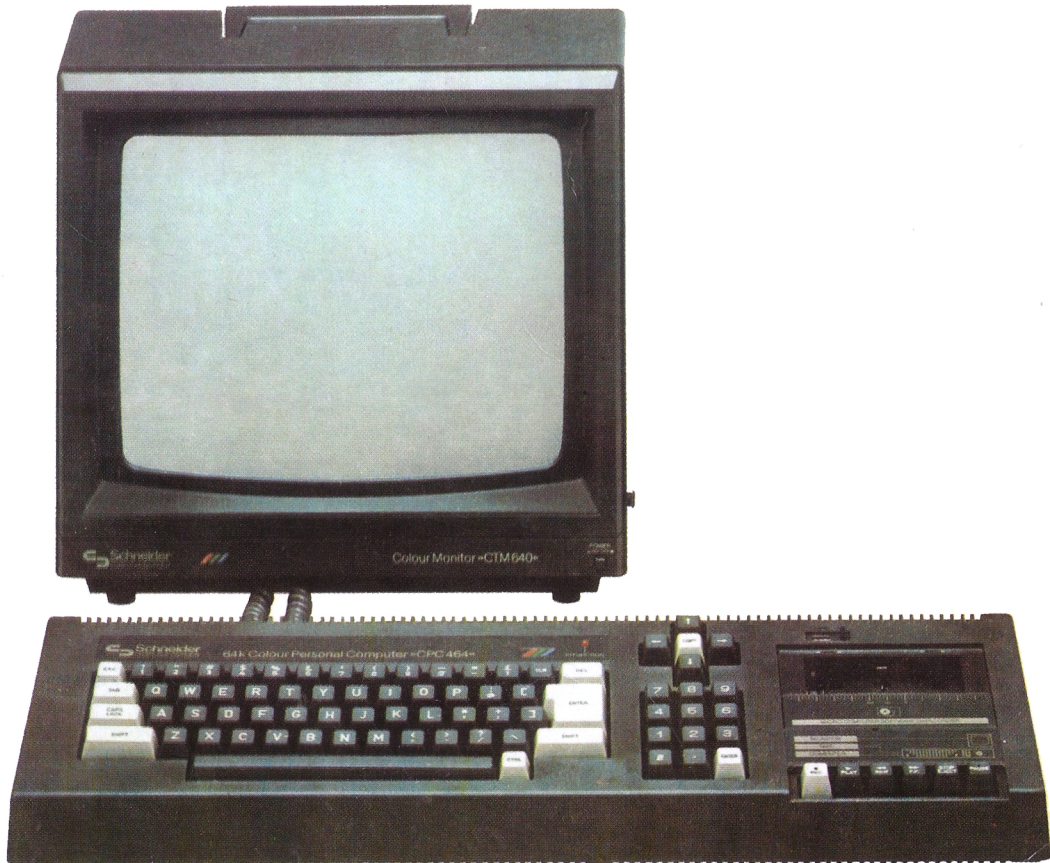




**MANUAL
DE
INSTRUCCIONES**
Colour Personal Computer
»CPC 464«



Erste Ausgabe 1984

Originalausgabe in Englisch

Original Copyright © 1984 AMSOFT, AMSTRAD Consumer Electronics plc and Locomotive Software Limited

Deutsche Übersetzung: Peter Eschenbacher, Wulf-Dietrich Wenzel

Umschlag: Baur-Design

Fotosatz: Lerchl-Druck, Freising

INDICE

	<u>pág.</u>
IMPORTANTE	3
INSTALACION	5
MONITOR de Pantalla Verde SCHNEIDER GT 64	5
El MONITOR a Color SCHNEIDER CTM 640	6
Modulador/Alimentador SCHNEIDER MP1 y un receptor de Televisión a color (UHF) doméstico	6
JOYSTICK	7
Cinta de Bienvenida	7
CARGANDO OTRAS CASSETTES de Programas	8
Guardando Programas en Cinta	9
FAMILIARIZACION CON EL TECLADO	11
Una Introducción a las Palabras Clave del BASIC SCHNEIDER	12
Edición de Programas	13
Aritmética Simple	13
Cálculos Combinados	16
GRAFICOS MODOS Y SONIDOS	19
Colores	19
Colores Parpadeando	21
Programa de Demostración	21
Gráficos	21
UBICANDO EL CURSOR	22
CIRCULOS	23
ORIGEN	23
GOSUB RETURN	24
Sonidos	24
LOS CARACTERES ASCII Y LOS CARACTERES GRAFICOS DEL CPC 464	27
INTRODUCCION AL CPC 464 PARA EXPERTOS	43
Modos de Pantalla	44
1. Hardware	45
NOTAS Y PERIODOS DE TONO	52
CODIGOS DE ERROR Y Palabras Reservadas	57

PALABRAS CLAVE EN EL BASIC SCHNEIDER	61
LA PANTALLA	63
Modos, Resolución, Colores y Tintas	63
La pantalla de Texto y los Caracteres Definibles por el Usuario	63
Cauces hacia Pantalla y Ventana de Texto	64
Pantalla de Gráficos, Sistema de Coordenadas, Ventanas y Origen	65
Resumen de Comandos y Funciones de Pantalla	66
TECLADO Y JOYSTICKS	67
INTRODUCCION DE LINEAS, EDICION Y COPIADO	68
CAUCES DE ENTRADA Y SALIDA	69
GENERACION DE SONIDO	70
INTERRUPCIONES EN BASIC	73
PANORAMA DE COMANDOS	75
LOS COMANDOS Y LAS FUNCIONES INTRINSECAS	78
SCHNEIDER BASIC Manual de Referencia	83
Introducción	83
LOS ELEMENTOS DE BASIC	84
MODOS DIRECTO Y PROGRAMA	93
RESUMEN DE COMANDOS	95
FUNCIONES DEL BASIC	131

IMPORTANTE

Cuando leas esta Guía de Usuario, deberás tener presente que usamos diferentes estilos de letra que indican diferentes formas de hacer referencias a los programas; que hay teclas que se encuentran en el ordenador pero que no exponen ningún carácter en la pantalla al ser pulsadas; y “descripciones generales” asociadas con palabras de programación, pero que no se teclean como parte de la instrucción donde aparecen.

1. Conecta el cable de alimentación siempre a una clavija de 3 patillas.
2. Nunca conectes el Teclado, el Monitor o el Alimentador/Modulador a ninguna parte de equipo o red eléctrica diferente a la descrita en esta Guía.
3. Mantén los ceniceros, bebidas, etc. lo suficientemente alejados del teclado de tu ordenador, del Alimentador/Modulador y del Monitor.
4. No bloques ni cubras las ranuras de ventilación situadas en la parte superior y trasera del Teclado, del Monitor y del Alimentador/Modulador.
5. Toda interrupción en el abastecimiento de energía eléctrica producirá la pérdida de toda la información almacenada en la memoria de lectura y escritura del CPC464.
6. Se recomienda el uso de cintas magnéticas específicamente diseñadas para usar con ordenadores. Es perfectamente aceptable el empleo de cintas que no sean CRO₂ o cintas de “metal”, y que no tengan duración superior a 90 minutos (C-90). Nos permitimos sugerirte que uses cintas C-12 (6 minutos por cara).
7. Considera que aquellas cintas que contienen programas de otros tipos de ordenador no pueden ser empleadas en el CPC464, salvo que hayan sido oficialmente declaradas “compatibles” por nosotros.
8. La tecla de Grabación no actuará si la lengüeta de seguridad ha sido cortada.
9. Recuerda que has de asegurar que la parte inicial de la cinta se encuentra ya enrollada en el carrete receptor antes de comenzar la grabación de tu programa.
10. Procura no operar ni conservar ninguna de las unidades expuestas a la luz solar directa; en zonas con temperaturas excesivamente altas o excesivamente bajas; en áreas de mucha humedad o mucho polvo, o sujetas a fuertes vibraciones.

11 .El cuidado general de tus cintas y la limpieza regularmente de tu equipo de cassettes te mantendrá fuera de problemas.

12 .No intentes manipular en el interior de tu equipo.

INSTALACION

El Ordenador Personal a Color CPC464 puede ser instalado eligiendo entre:

1. 1 Monitor de Pantalla Verde SCHNEIDER GT64
1. 2. Monitor Polícromo SCHNEIDER-ETM640
1. 3. Alimentador/Modulador SCHNEIDER MPI y un receptor doméstico de Televisión (UHF)

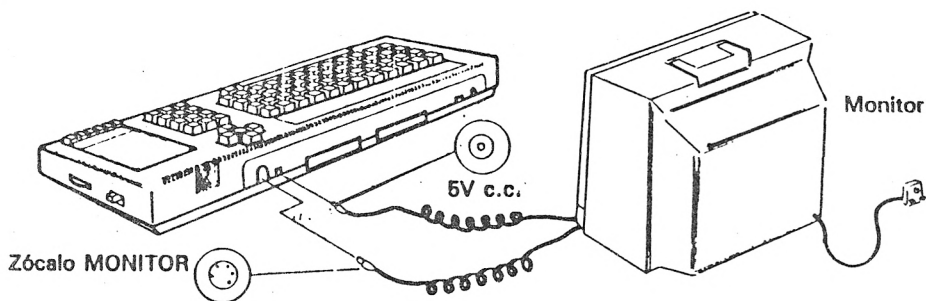
1. 1 Monitor de Pantalla Verde SCHNEIDER GT64

ADVERTENCIA

Desconecta el enchufe de red, la clavija del cable de alimentación siempre que no estés usando el monitor.

Conecta el cable con clavija más grande (6 pines DIN) del Monitor al enchufe hembra marcado MONITOR en la parte posterior del ordenador. Conecta el cable del Monitor con enchufe más pequeño (corriente continua 5 Volt) al enchufe marcado 5 V. D.C al enchufe en la parte posterior del ordenador.

Figura 1



Asegúrate que el botón POWER del Monitor se encuentre en la posición OFF (apagado). Conecta la clavija de red del Monitor al enchufe de (240 volts) corriente alterna.

Enciende el monitor, y posteriormente el ordenador usando el interruptor deslizante marcado POWER (energía)

El testigo luminoso rojo ON situado en la parte superior central del teclado del ordenador deberá encenderse, y el Monitor mostrar en la pantalla el siguiente mensaje:

Schneider 64K Microcomputer
1984 Schneider Consumer Electronics plc
and Locomotive Software Ltd.

BASIC 1. Ø

Ready

cursor

Ajusta el mando del brillo; también deberás ajustar el mando de contraste a la mínima posición que le permita una visión cómoda.

1. 2 El Monitor a Color SCHNEIDER CTM640

Se conecta de igual modo que en apartado anterior.

1. 3 Modulador/Alimentador SHNEIDER MP1 y un receptor de televisión a color (UHF) doméstico

El MP1 es un accesorio conveniente y que desearás comprar, si habitualmente operas tu ordenador CPC464 con el Monitor de Pantalla Verde GT64. El MP1 te permitirá usar el ordenador con un Televisor a Color Doméstico, y por tanto gozar con todas las facilidades de color que te ofrece tu ordenador CPC464.

Conecta el cable del MP1 la clavija más grande (6 pines DIN) al enchufe marcado MONITOR en la parte posterior del ordenador. Conecta el cable del MP1 con la clavija más pequeña (corriente continua 5 voltios) al enchufe marcado 5V. D.C., de la parte posterior del ordenador. Inserta el conector del cable de antena del MP1 al conector de entrada de antena del Televisor.

Comprueba que el interruptor **POWER** del ordenador situado en el costado derecho, se encuentra en la posición **OFF** (apagado); ahora conecta la clavija de alimentación **MP1** al enchufe de energía eléctrica (240 volts. corriente alterna).

Reduce el control de volúmen del Televisor al mínimo.

1. 4 JOYSTICK

El joystick **AMSOFT** modelo **JY1** puede ser conectado a la parte posterior del ordenador usando el conector hembra de 9 vías marcado **USER PORTS (I/O)**. El ordenador **SCHNEIDER CPC464** puede operar simultáneamente con dos joystick de ese estilo. El segundo de ellos deberá ser conectado al conector hembra provisto en la parte del primer joystick.

1. 5 Cinta de Bienvenida

Levanta la cubierta de la ductora de cintas en cassette presionando la tecla marcada **STOP/EJECT** para parar e insertar la cassette con la cinta de bienvenida en los carriles de soporte.

Presiona la tecla de rebobinado, marcada **REW** hasta que te cerciores de que la cinta está debidamente colocada en su principio. Tan pronto como la cinta se detenga, oprimes la tecla **STOP/EJECT**.

Pulsa la tecla blanca de control del teclado, marca **CTRL** y **AL MISMO TIEMPO** la pequeña tecla, marcada **ENTER** en la parte inferior derecha del teclado numérico separado y situado inmediatamente al lado de la unidad ductora de cinta.

La pantalla responderá con el mensaje:

RUN
Press PLAY then any key

Ahora oprimes la tecla de reproducción, marcada **PLAY** hasta que se queda firmemente trabada; luego pulsas cualquier letra, número, o cualquiera de las dos teclas **ENTER**. La cinta comenzará a correr, y después de un corto periodo, aparecerá en la pantalla el mensaje:

Loading WELCOME 1 block 1

La cinta tardará cerca de 5 minutos en ser cargada. El programa "rula" continuamente, así que cuando hayas terminado de observarlo, presiona la tecla de escape, marcada **ESC** dos veces seguidas. Eso hace que se detenga el programa, y ya puedes ahora sacar la cinta, dar la vuelta y ponerla por la cara 2.

Para leer la cara 2 realiza los mismos pasos que con la cara 1.

Sigue las instrucciones dadas en la pantalla, y el programa te invitará a que participes escribiendo diversas instrucciones, más complejas a medida que va progresando el programa y tú.

1. 6 CARGANDO OTRAS CASSETTES DE PROGRAMAS

Los programas en BASIC —sin protecciones especiales— podrán se cargados con los siguientes métodos alternativos. Rebobina la cinta que has insertado en la unidad ductora de cinta, presionando la tecla **REW** hasta que la cinta se detenga; que será el momento en que deberás oprimir la tecla **STOP/EJECT**.

Restaura las condiciones iniciales en el sistema (los angloparlantes llaman **RESET** a esta operación), pulsando sucesivamente las siguientes teclas: **CTRL, SHIFT Y ESC** pero manteniendo presionadas las dos primeras hasta que pulses la tecla **ESC**.

Escribe:

Load "Nombre Programa" ENTER

El ordenador te pide:

Press PLAY then any key:

Ahora presionas la tecla **PLAY** en la fila de teclas de la parte delantera del "almacenador" de datos que se enganche firmemente y luego cualquiera de las del teclado, incluso la barra espaciadora.

La cinta comenzará a pasar y aparecerá en la pantalla este mensaje:

Loading "nombre del programa" block 1

Los números de bloque continuarán aumentando hasta que la cinta haya terminado de cargarse, momento en que aparecerá:

Ready

En ese momento puedes escribirle:

Run ENTER

... y el programa comenzará a ejecutarse.

Si mientras el ordenador busca el programa indicado a lo largo de la cinta se encuentra con otros programas con título distinto al que le has mandado cargar, te mostrará el siguiente mensaje en la pantalla:

Found "otro título" block 1

que indica que ha encontrado un programa con ese título distinto.

Para que el programa sea ejecutado directamente, inmediatamente después de que haya sido cargado por el ordenador, puedes teclear:

RUN "título" ENTER

para obligarle no sólo a que lo cargue, sino a que lo ejecute inmediatamente después.

1. 7 Guardando programas en cinta

Inserta una cinta en el equipo cassette. Oprime la tecla **REW** para rebobinar la cinta y colocarla al principio. Luego tecleas simplemente:

save "título del programa" ENTER

El ordenador te responderá con el mensaje:

Press REC and PLAY then any key:

Para indicarle que en este caso han de pulsarse las teclas de grabación y de marcha al mismo tiempo; y que luego pulses cualquiera de las letras, números, teclas **ENTER**, o incluso barra espaciadora para que comience la operación.

Después de haberlo hecho, el ordenador te mostrará en pantalla el mensaje:

Saving título del programa block 1

Para indicar que ha comenzado la operación de guardar en la cinta el programa que tenía en memoria y con el título que le has dado.

Cuando haya terminado de grabar el programa, la cinta se detendrá y verás aparecer en pantalla la palabra:

Ready.

FAMILIARIZACION CON EL TECLADO

ENTER

Existen dos teclas marcadas **ENTER** en tu ordenador. Cada una de estas teclas “envían adentro” la información que has tecleado, indicando al ordenador que ya has terminado de darle la instrucción y que puede pasar a tramitarla.

DEL

Esta tecla se usa para borrar o “delectar” el carácter situado inmediatamente a la izquierda del cursor; es decir, para suprimir una letra, número, o símbolo que sea innecesario.

SHIFT

Pulsando cualquiera de ellas y manteniéndola pulsada mientras que al mismo tiempo se oprime una tecla correspondiente a una letra, lo que aparece en la pantalla es esa letra pero en mayúsculas.

CAPS LOCK

Pulsando esta tecla una sola vez se logra que las teclas correspondientes a las letras que hay en el teclado pasen a enviar siempre al ordenador el símbolo correspondiente a las mayúsculas. Por supuesto, que si has vuelto al modo de mayúsculas trabadas, el efecto basculador de la tecla **CAPS LOCK** hará que a la siguiente pulsación de esa tecla, vuelvas al modo minúsculas.

CLR

Esta tecla se usa para borrar (“influido por el inglés, diríamos CLA + REAR”) para borrar el carácter situado en la misma posición que el cursor.

ESC

Esta tecla se usa para ESCAPAR de la función que el ordenador esté desarrollando en ese momento. Pulsando una sola vez la tecla **ESC**, sólo se consigue que el ordenador realice una pausa en el desarrollo de la función vigente, y que la continúe tan pronto como sea pulsada cualquier otra tecla distinta de **ESC**.

Si en esa situación pulsas **ESC**, habrán sido dos pulsaciones sucesivas de la tecla **ESC**, lo que tiene como efecto que el ordenador termine y deje definitivamente la función que estaba desarrollando en ese momento. El ordenador pasará entonces al modo normal en que espera que le dictes algún otro comando o instrucción.

UNA INTRODUCCION A LAS PALABRAS CLAVE DEL BASIC SCHNEIDER

CLS

LIMPIA la pantalla

PRINT

Hace que el ordenador EXPONGA en pantalla lo que tú deseas, sean palabras, datos o figuras.

RUN

Hace actuar inmediatamente el mandato indicado o el programa cargado. Un programa es simplemente una serie de instrucciones almacenada en la memoria del ordenador, y no se ejecuta hasta que le mandamos que lo ejecute, usando el comando **RUN**.

LIST

Después de que el programa haya sido almacenado en la memoria del ordenador, sacando un LISTADO del programa.

GOTO

Dice al ordenador que VAYA desde la línea que se encuentra hasta la línea cuyo número se reseña en esa instrucción.

INPUT

Para IMPONER un dato por teclado.

EDICION DE PROGRAMAS

Si cualquiera de las líneas del programa ha sido tipografiada incorrectamente, provocará un error sintáctico u otra clase de error, y será necesario revisar y corregir esa línea en lugar de teclearla de nuevo.

Hay tres métodos para editar un programa:

1.º) Consiste simplemente en volver a teclear correctamente la línea.

2.º) Método de Edición.

para corregir la equivocación de la línea, teclea el comando:

edit "número de línea" ENTER

El ordenador entra en el modo de edición, y nos muestra dicha línea.

3.º) Método de copia moviendo el cursor

Para corregir las equivocaciones puedes hacerlo manteniendo pulsada la tecla **SHIFT** y pulsando la tecla de subida del cursor, marcada (**↑**), hasta que el "cursor que copia" se sitúa en el principio mismo de la línea. Pulsa ahora la tecla de **COPIAR** marcada con la palabra **COPY**, hasta que el cursor de copiado se sitúe entre el espacio en blanco a corregir.

ARITMETICA SIMPLE

Para comprender esta forma de operar, realiza los siguientes ejemplos. Como ya sabes, podemos usar el signo de interrogación (?) en lugar de teclear la palabra **PRINT**, y así abreviar. Las respuestas se expondrán en pantalla tan pronto como pulses la tecla **ENTER**.

ADICION

Teclea:

? 3 + 3 ENTER

6

Teclea:

? 8 + 4 ENTER
12

SUSTRACCION

Teclea:

? 4 - 3 ENTER
1

Teclea:

? 8 - 4 ENTER
4

MULTIPLICACION

Teclea:

? 3 * 3 ENTER
9

Teclea:

? 8 * 4 ENTER
32

DIVISION

Teclea:

? 3 / 3 ENTER
1

Teclea:

? 8 / 4 ENTER
2

RAIZ CUADRADA

Teclea:

? s q r (1 6) ENTER
4

Teclea:

? s q r (100) ENTER
10

EXPONENCIACION

Usa directamente el signo de ‘elevado a’, marcado con **↑** . (Está debajo de la **z** .)

Teclea:

? 3 ↑ 3 ENTER
27

esto significa 3^3

Teclea:

? 8 ↑ 4 ENTER
4096

eso significa 8^4

RAIZ CUBICA

Fácilmente puedes calcular raíces cúbicas, si recuerdas que son equivalentes a elevar el número a la potencia $1 / 3$.

Teclea:

? 2 7 ↑ (1 / 3) ENTER
3

Teclea:

? 1 2 5 ↑ (1 / 3) ENTER
5

CALCULOS COMBINADOS (+, -, *, /)

Los cálculos combinados son bien interpretados por el ordenador, siempre y cuando se calculen de acuerdo con las prioridades estipuladas para cada operación.

La primera prioridad corresponde a la multiplicación, luego la división, luego la adición, y finalmente la sustracción.

Si el cálculo fué:

$$3 + 7 - 2 * 7 / 4$$

Puedes pensar que podría calcularse en la forma siguiente:

$$\begin{aligned} 3 + 7 - 2 * 7 / 4 \\ = 8 * 7 / 4 \\ = 56 / 4 \\ = 14 \end{aligned}$$

Pero de hecho, el ordenador lo calcula como:

$$\begin{aligned} 3 + 7 - 2 * 7 / 4 \\ = 3 + 7 - 14 / 4 \\ = 3 + 7 - 3.5 \\ = 10 - 3.5 \\ = 6.5 \end{aligned}$$

Puedes cambiar la manera en que el ordenador calcula una expresión si empleas paréntesis. El ordenador tratará primero las operaciones incluidas dentro de los paréntesis de acuerdo con las prioridades establecidas, y luego pasará a las que están fuera de los paréntesis.

Si quieres usar en tus cálculos números muy grandes o muy pequeños, es conveniente que uses la denominada notación científica. En esta notación se usa la letra E para indicar exponentes de los números con relación a la base 10. Puedes usar la e minúscula o la E mayúscula.

1. 30 x 10

Puedes teclear en el ordenador:

```
? 30 * 10 ENTER  
300
```

o puedes teclear:

```
? 3 e 1 * 1 e 1 ENTER  
300
```

2. 3000 x 1000

Teclea:

```
? 3 e 3 * 1 e 3 ENTER  
300000
```

3. 3000 x 0,001

Teclea:

```
? 3 e 3 * 1 e - 3 ENTER  
3
```


GRAFICOS, MODOS Y SONIDOS

El Ordenador Personal SCHNEIDER CPC464 dispone de tres MODOS de operación en lo referente a la presentación de imágenes en pantalla: Modo 0, Modo 1 y Modo 2.

Cada vez que se pone en marcha el ordenador, automáticamente adopta el Modo 1.

- Modo 0 = 20 caracteres por línea
- Modo 1 = 40 caracteres por línea
- Modo 2 = 80 caracteres por línea

COLORES

Hay una gama de 27 colores.

En el modo 0, hasta 16 de los 27 colores disponibles pueden aparecer en la pantalla en cualquier momento.

En el modo 1, hasta 4 de los 27 colores pueden aparecer en pantalla en cualquier momento.

En el Modo 2, hasta 2 de los 27 colores pueden aparecer en pantalla en cualquier momento

También podrás cambiar el borde de la pantalla (**BORDER**), el área en que aparecen los caracteres (**PAPER**) o la PLUMA con que el ordenador pintará los caracteres (**PEN**), y estas tres facetas, todas independientemente una de otra.

- Modo 0, hasta 16 de los 27 colores
- Modo 1, hasta 4 de los 27 colores
- Modo 2, hasta 2 de los 27 colores

Para cambiar el color del BORDE que rodea al papel:

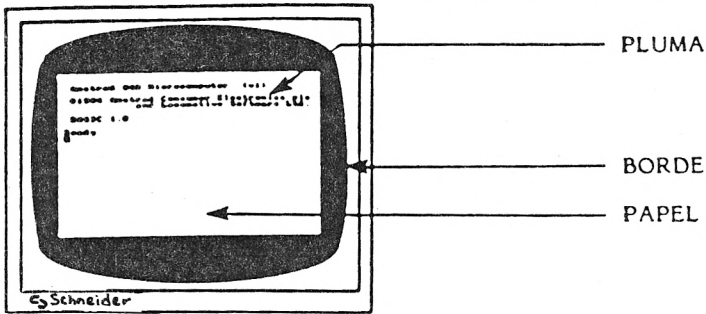
border o ENTER

Para ver como cambia el color del PAPEL; teclea:

paper 2 ENTER

Para ver como cambia el color de la PLUMA, teclea:

PEN 3 ENTER



Los 27 colores disponibles se enumeran en la tabla 1, cada uno con el número que sirve de referencia a la TINTA que se usa tanto para el papel como para la pluma.

TABLA MAESTRA DE COLORES

Nº Tinta	Color Tinta	Nº Tinta	Color Tinta
0	Negro	14	Azul Pastel
1	Azul	15	Naranja
2	Azul Brillante	16	Rosa
3	Rojo	17	Magneta Pastel
4	Magneta	18	Verde Brillante
5	Malva	19	Verde Marino
6	Rojo Brillante	20	Ciano Brillante
7	Púrpura	21	Verde Lima
8	Magneta Brillante	22	Verde Pastel
9	Verde	23	Ciano Pastel
10	Ciano	24	Amarillo Brillante
11	Azul Cielo	25	Amarillo Pastel
12	Amarillo	26	Blanco Brillante
13	Blanco		

Podemos continuar cambiando estos colores y podemos hacerlo usando el comando **INK**. Este comando de TINTA, tiene dos números. El primero es el número de la PLUMA o del PAPEL que queremos cambiar; el segundo es el número que queremos asignar al PAPEL o a la PLUMA respectivamente.

COLORES PARPADANDO

Es posible hacer que los colores de los caracteres alternen entre uno y otro, logrando el parpadeo o destello. Eso puede lograrse añadiendo un número extra de color dentro del co-

mando **INK** relativo a la **PLUMA**. Por ejemplo, para ver los caracteres alternativamente en blanco brillante y rojo brillante, teclea:

ink 1, 26, 6 ENTER

COLOR DE LA TINTA

Nº Papel/Pluma	Modo 0	Modo 1	Modo 2
0	1	1	1
1	24	24	24
2	20	20	1
3	6	6	24
4	26	1	1
5	0	24	24
6	2	20	1
7	8	6	24
8	10	1	1
9	12	24	24
10	14	20	1
11	16	6	24
12	18	1	1
13	22	24	24
14	Parpadeo 1,24	20	1
15	Parpadeo 16,11	6	24

PROGRAMA DE DEMOSTRACION

Para mayor demostración de los colores disponibles, teclea y ejecuta el siguiente programa:

```
10 mode 0: ink 0,2 ink 1,24: paper 0 ENTER
20 pen 1: for b = 0 to 26: border b ENTER
30 locate 3,12: print "COLOR DEL MARGEN"; B ENTER
40 sound 4, (40 - b) ENTER
50 for t = 1 to 600: next b: cls ENTER
60 for p= 0 to 15: paper p: pen 5: print "FONDO" p = print ENTER
70 for n = 0 To 15: pen n: print "TRAZO"; n ENTER
```

```

80 sound 1, (n 20 + p) ENTER
90 for t = 1 to 100: next t:next n ENTER
100 for t = 1 to 1000: next t: cls: next p ENTER
110 cls: paper 0: pen 1: locate 7,12: print "FIN": for t = 1 to 2000: next t ENTER
120 mode 1: border 1: ink 0,1: ink 1,24: paper 0: pen 1 ENTER
run ENTER

```

GRAFICOS

Hay cierto número de caracteres en la memoria del ordenador que corresponden a los símbolos (letras, cifras, signos, etc.) que usamos para las comunicaciones con el ordenador. Para obtener cualquiera de esos símbolos, usamos la palabra clave chr\$(X), siendo la X incluida dentro de los paréntesis el número que corresponda a ese símbolo, que debe estar dentro de la gama 32 a 225.

Para ver todos los símbolos que pueden aparecer en pantalla, junto con su número asociado de carácter, teclea el siguiente programa:

```

10 for n = 32 to 255: print n; chr$(n) 20 next n
run

```

UBICANDO EL CURSOR

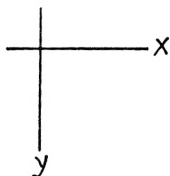
Existe un comando que se usa para ubicar el cursor de texto en una parte específica de la pantalla. Mediante el comando **LOCATE**, el cursor de caracteres comienza en la esquina superior izquierda de la pantalla, que corresponde a las coordenadas x (eje horizontal) e y (eje vertical) de valores 1,1. En el modo de imagen 1, sabemos que hay 40 columnas y 25 líneas dentro del área de exposición de caracteres.

En el modo 0, sólo hay 20 columnas, pero las mismas 25 líneas.
 En el modo 2, hay 80 columnas y 25 líneas.

```

PLOT (x, y)
x = 0 → 640
y = 0 → 400

```



Se usa para determinar la posición del cursor interno usado para gráficos, que usa otros ejes de coordenadas, y con diferentes medidas. De hecho, determina la resolución en pantalla, y lo que se pinta es un diminuto punto que solemos llamar MOTA (o bien pixel como abreviatura de elemento pictórico).

Puedes pintar 640 motas dentro de una línea horizontal, y 400 motas dentro de cada línea vertical. Usando las coordenadas x, y, se toma como origen la esquina inferior izquierda de la pantalla.

DRAW (x,y)

Nos permite DIBUJAR una línea a partir de la corriente posición del cursor de gráficos.

CIRCULOS

Los círculos pueden ser pintados (mota a mota) o dibujados (a trazos o líneas). Un método para formar un círculo, es pintar las coordenadas x, y de cada uno de los puntos de la circunferencia. Consulta la figura inferior y verás cómo el punto p de la circunferencia puede pintarse usando como coordenadas x e y las dadas por las ecuaciones:

$$x = 190 * \cos(a)$$

$$y = 190 * \sin(a)$$

Si queremos situar el círculo en el centro de la pantalla, tendríamos que pintar el centro del círculo en las coordenadas 320, 200; y luego pintar todos los puntos del círculo con relación a ese centro, añadiéndole las coordenadas x, y como función del ángulo.

Por lo tanto, para pintar el círculo, sería algo como esto:

```
new
5 cls
10 for a=1 to 360
15 deg
20 plot 320,200
30 plot 320+190*cos(a), 200+190*sin(a)
40 next
run
```

Un método, es pintar las coordenadas x, y de cada uno de los puntos de la circunferencia.

ORIGEN

Con él señalamos el centro de un círculo, simplemente después sólo tenemos que precisar las coordenadas x, y de todos los puntos de la circunferencia (en pasos de un grado) con respecto a este nuevo origen.

Para verlo, teclea:

```
new
5 cls
10 for a=1 to 360
15 deg
20 origin 320,200
30 plot 190 cos(a), 190 sin(a)
40 next
run
```

GOSUB RETURN

Si hay una serie de instrucciones dentro de un programa que efectúan una tarea determinada que se realiza un cierto número de veces en el programa de forma "rutinaria"; pueden considerarse estas instrucciones como una subrutina, y hacer que el ordenador VAYA a efectuar esa tarea, mediante el comando **GOSUB** seguido del número de línea en que comienza la subrutina.

EL final de dichas subrutinas, se marca tecleando la instrucción que hace que el ordenador VUELVA (**RETURN**) al punto en que se desvió. Es decir, cuando el ordenador encuentra la instrucción **RETURN**, volverá a la instrucción que va inmediatamente detrás de aquella en que se produjo el desvío mediante el comando **GOSUB**.

SONIDOS

Los efectos sonoros son producidos por un altavoz incorporado dentro del propio ordenador.

El comando **SOUND**, tiene 7 parámetros para el sonido. Los primeros 2 de ellos, deben usarse siempre, el resto es opcional. El comando se teclea en la forma:

SONIDO estado del canal, periodo del tono, duración, volumen, envolvente de volumen, envolvente de tono, periodo del ruido.

Duración

Cuando no se especifica duración del sonido generado, sonará durante 0,2 segundos, la unidad de duración es de 0,01 segundos.

Volúmen

Este número especifica el volúmen con que arranca una nota. El número debe estar en la gama 0 a 7. Pero sin embargo, si se especifica un envolvente del volúmen, la gama se amplía de 0 a 15. Si no se utiliza en el comando ningún número, se presupone 4 que es el valor prescrito para las omisiones.

Envolvente de volúmen

El comando para el envolvente de volúmen usa la palabra reservada `env`. `env` número del envolvente, número de pasos, amplitud (tamaño) del paso, tiempo del paso.

Número del envolvente

Es el número dado a un determinado envolvente, de modo que pueda especificarse en el comando **SOUND**. La gama de número de envolventes es de 0 a 15.

Número de pasos

Se usa en conjunción con el tiempo del paso. La gama de valores posibles para los números de pasos es de 0 a 127

Amplitud del paso

Cada paso puede variar en amplitud desde un nivel 0 hasta un nivel 15 con respecto al paso precedente. Los 15 niveles de amplitud son los mismos que se emplean en el comando **SOUND**. Sin embargo, cada paso puede ajustarse desde -128 a +127 de manera que no solamente se puede variar la amplitud subiéndola o bajándola en forma obvia, sino que también puede variarse usando números mayores de 15 para obtener algunos efectos especiales y extraños.

Tiempo del paso

Este número especifica el tiempo transcurrido entre los pasos en unidades de 0,01 segundo (1 100 segundo). La gama de números para el tiempo del paso es de 0 a 255.

Envolvente de Tono

El comando para la envolvente de tono es **ent**.

Tiene normalmente 4 parámetros.

ent número de envolvente, número de pasos, periodo de tono del paso, tiempo del paso.

Número de envolvente

Es el número dado a la envolvente particular de forma que pueda especificarse en el comando **SOUND**. La gama de números de envolvente es de 1 a 15.

Número de pasos

Se usa en conjunción con el tiempo del paso. La gama de números de paso es de 0 a 239.

Periodo de Tono del Paso

El periodo del tono de cada paso puede variar entre -128 a $+127$. Los valores negativos aumentan la frecuencia de las notas (haciendo las notas más agudas). Los números para periodos de tono del paso, deben estar en la gama de -128 a $+127$.

Tiempo del Paso

Este número especifica el tiempo del paso en unidades de 0,01 segundo. La gama de números para el tiempo del paso es e 0 a 225. Por lo tanto, el tiempo del paso más largo es de 2,55 segundos.

Ruido

Se puede añadir ruido al final del comando **SOUND**. Se dispone de una gama de ruidos con números del 1 al 31.

LOS CARACTERES ASCII Y LOS CARACTERES GRAFICOS DEL CPC464

1 ASCII

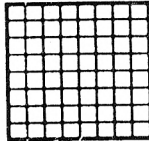
Para referencia, reproducimos aquí el repertorio de caracteres standard ASCII usando las notaciones decimal, octal y hexadecimal, juntamente con el carácter "visivo" o de control que le corresponde. Asimismo, la composición gráfica de los caracteres usados en el CPC464 se muestran a continuación.

DEC	OCTAL	HEX	ASCII characters	DEC	OCTAL	HEX	ASCII	DEC	OCTAL	HEX	ASCII
0	000	00	NUL (CTRL@)	50	082	32	2	100	144	64	d
1	001	01	SOH (CTRLA)	51	083	33	3	101	145	65	e
2	002	02	STX (CTRLB)	52	064	34	4	102	146	66	f
3	003	03	ETX (CTRLC)	53	065	35	5	103	147	67	g
4	004	04	EOT (CTRLD)	54	066	36	6	104	150	68	h
5	005	05	ENO (CTRL E)	55	067	37	7	105	151	69	i
6	006	06	ACK (CTRL F)	56	070	38	8	106	152	6A	j
7	007	07	BEL (CTRL G)	57	071	39	9	107	153	6B	k
8	010	08	BS (CTRL H)	58	072	3A	;	108	154	6C	l
9	011	09	HT (CTRL I)	59	073	3B	:	109	155	6D	m
10	012	0A	LF (CTRL J)	60	074	3C	<	110	156	6E	n
11	013	0B	VT (CTRL K)	61	075	3D	=	111	157	6F	o
12	014	0C	FF (CTRL L)	62	076	3E	>	112	160	70	p
13	015	0D	CR (CTRL M)	63	077	3F	?	113	161	71	q
14	016	0E	SO (CTRL N)	64	100	40	@	114	162	72	r
15	017	0F	SI (CTRL O)	65	101	41	A	115	163	73	s
16	020	10	DLE (CTRL P)	66	102	42	B	116	164	74	t
17	021	11	DC1 (CTRL Q)	67	103	43	C	117	165	75	u
18	022	12	DC2 (CTRL R)	68	104	44	D	118	166	76	v
19	023	13	DC3 (CTRL S)	69	105	45	E	119	167	77	w
20	024	14	DC4 (CTRL T)	70	106	46	F	120	170	78	x
21	025	15	NAK (CTRL U)	71	107	47	G	121	171	79	y
22	026	16	SYN (CTRL V)	72	110	48	H	122	172	7A	z
23	027	17	ETB (CTRL W)	73	111	49	I	123	173	7B	{
24	030	18	CAN (CTRL X)	74	112	4A	J	124	174	7C	
25	031	19	EM (CTRL Y)	75	113	4B	K	125	175	7D	}
26	032	1A	SUB (CTRL Z)	76	114	4C	L	126	176	7E	-
27	033	1B	ESC	77	115	4D	M				
28	034	1C	FS	78	116	4E	N				
29	035	1D	GS	79	117	4F	O				
30	036	1E	RS	80	120	50	P				
31	037	1F	US	81	121	51	Q				
32	040	20	SP	82	122	52	R				
33	041	21	!	83	123	53	S				
34	042	22	"	84	124	54	T				
35	043	23	#	85	125	55	U				
36	044	24	\$	86	126	56	V				
37	045	25	%	87	127	57	W				
38	046	26	&	88	130	58	X				
39	047	27	'	89	131	59	Y				
40	050	28	(90	132	5A	Z				
41	051	29)	91	133	5B	[
42	052	2A	*	92	134	5C	\				
43	053	2B	+	93	135	5D]				
44	054	2C	,	94	136	5E	^				
45	055	2D	-	95	137	5F	_				
46	056	2E	.	96	140	60	`				
47	057	2F	/	97	141	61	a				
48	060	30	@	98	142	62	b				
49	061	31	1	99	143	63	c				

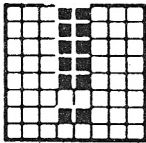
2 Repertorio de caracteres específico de la máquina CPC464

Los caracteres reproducidos aquí, están mostrados usando una retícula de 8x8, que es la que se emplea para exponer los caracteres en la pantalla del CPC464.

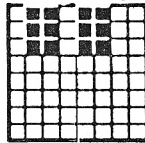
Los caracteres definidos por el usuario pueden ser agrupados y colocados unos contiguos con otros para lograr figuras especiales. (Véase el comando **SYMBOL** en el capítulo 8).



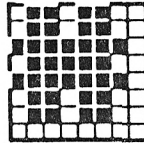
32 &H20
&X00100000



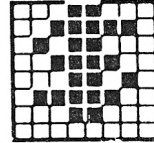
33
&H21
&X00100001



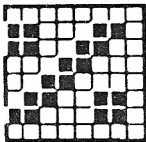
34
&H22
&X00100010



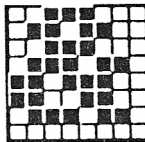
35
&H23
&X00100011



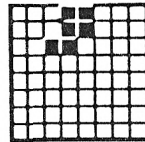
36
&H24
&X00100100



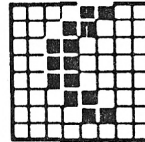
37
&H25
&X00100101



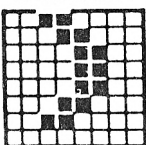
38
&H26
&X00100110



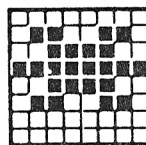
39
&H27
&X00100111



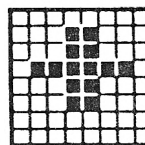
40
&H28
&X00101000



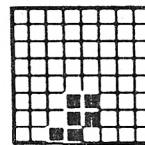
41
&H29
&X00101001



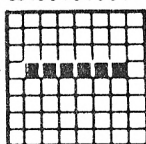
42
&H2A
&X00101010



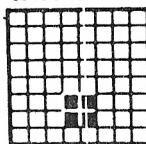
43
&H2B
&X00101011



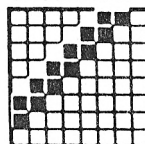
44
&H2C
&X00101100



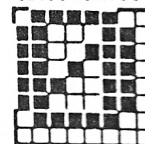
45
&H2D
&X00101101



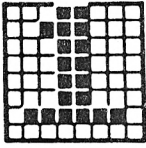
46
&H2E
&X00101110



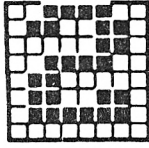
47
&H2F
&X00101111



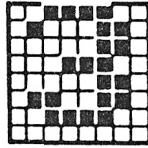
48
&H30
&X00110000



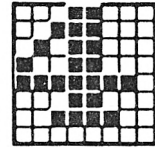
49
&H31
&X00110001



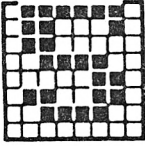
50
&H32
&X00110010



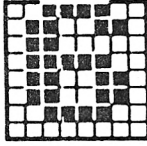
51
&H33
&X00110011



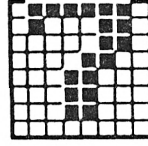
52
&H34
&X00110100



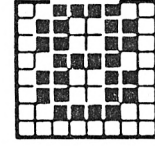
53
&H35
&X00110101



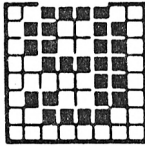
54
&H36
&X00110110



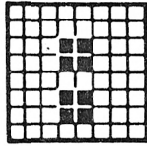
55
&H37
&X00110111



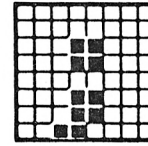
56
&H38
&X00111000



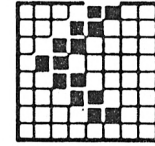
57
&H39
&X00111001



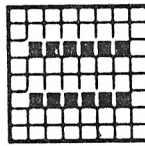
58
&H3A
&X00111010



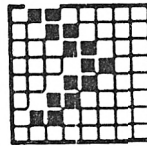
59
&H3B
&X00111011



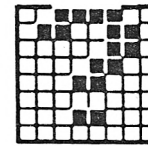
60
&H3C
&X00111100



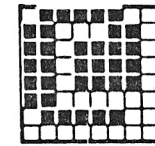
61
&H3D
&X00111101



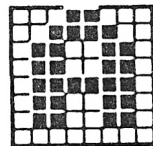
62
&H3E
&X00111110



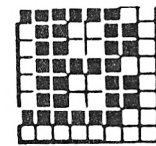
63
&H3F
&X00111111



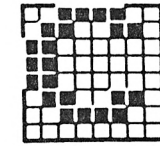
64
&H40
&X01000000



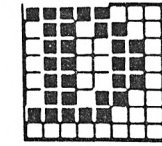
65
&H41
&X01000001



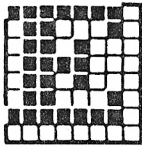
66
&H42
&X01000010



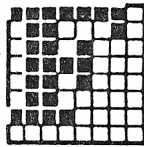
67
&H43
&X01000011



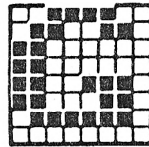
68
&H44
&X01000100



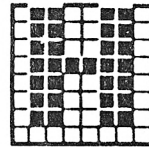
69
&H45
&X01000101



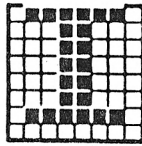
70
&H46
&X01000110



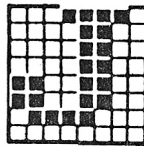
71
&H47
&X01000111



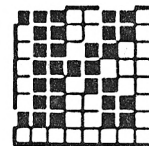
72
&H48
&X01001000



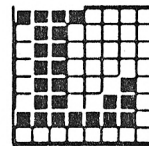
73
&H49
&X01001001



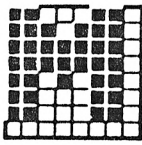
74
&H4A
&X01001010



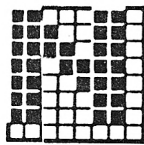
75
&H4B
&X01001011



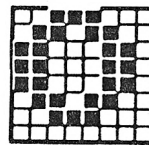
76
&H4C
&X01001100



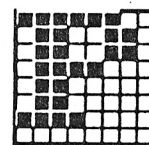
77
&H4D
&X01001101



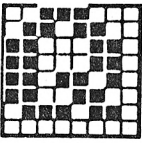
78
&H4E
&X01001110



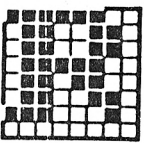
79
&H4F
&X01001111



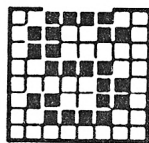
80
&H50
&X01010000



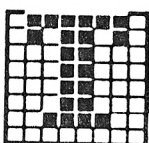
81
&H51
&X01010001



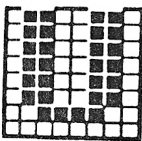
82
&H52
&X01010010



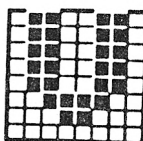
83
&H53
&X01010011



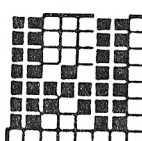
84
&H54
&X01010100



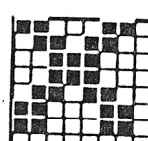
85
&H55
&X01010101



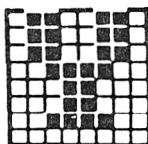
86
&H56
&X01010110



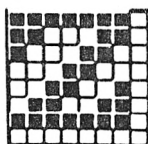
87
&H57
&X01010111



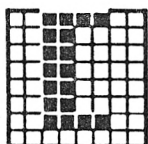
88
&H58
&X01011000



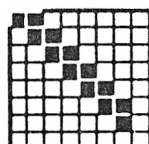
89
&H59
&X01011001



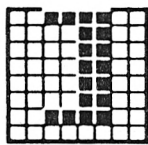
90
&H5A
&X01011010



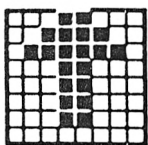
91
&H5B
&X01011011



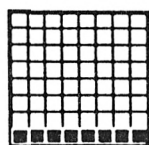
92
&H5C
&X01011100



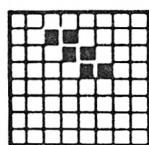
93
&H5D
&X01011101



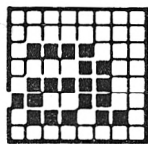
94
&H5E
&X01011110



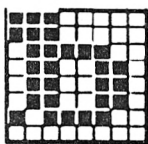
95
&H5F
&X01011111



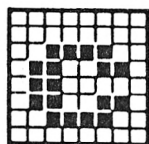
96
&H60
&X01100000



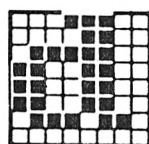
97
&H61
&X01100001



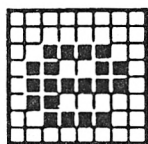
98
&H62
&X01100010



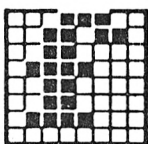
99
&H63
&X01100011



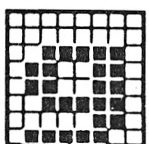
100
&H64
&X01100100



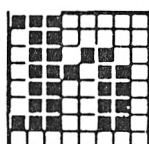
101
&H65
&X01100101



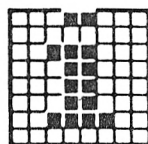
102
&H66
&X01100110



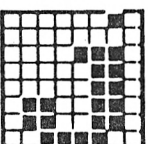
103
&H67
&X01100111



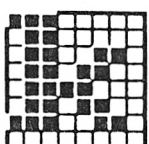
104
&H68
&X01101000



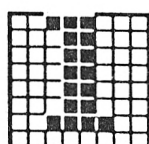
105
&H69
&X01101001



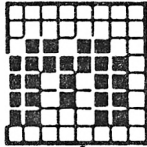
106
&H6A
&X01101010



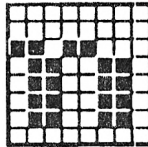
107
&H6B
&X01101011



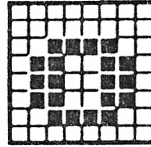
108
&H6C
&X01101100



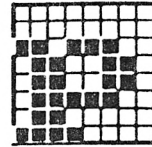
109
&H6D
&X01101101



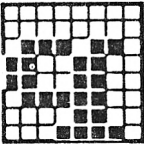
110
&H6E
&X01101110



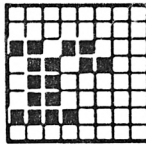
111
&H6F
&X01101111



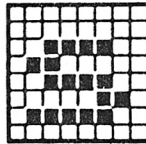
112
&H70
&X01110000



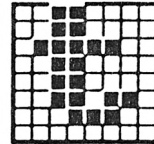
113
&H71
&X01110001



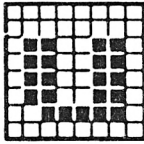
114
&H72
&X01110010



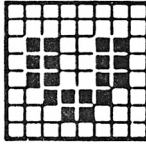
115
&H73
&X01110011



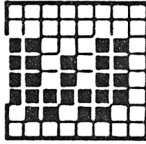
116
&H74
&X01110100



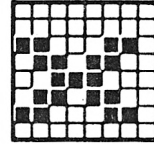
117
&H75
&X01110101



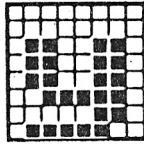
118
&H76
&X01110110



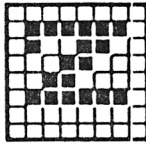
119
&H77
&X01110111



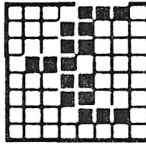
120
&H78
&X01111000



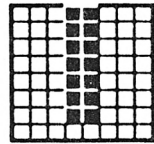
121
&H79
&X01111001



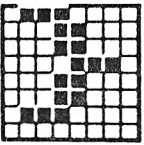
122
&H7A
&X01111010



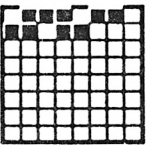
123
&H7B
&X01111011



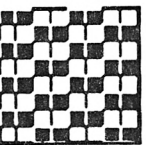
124
&H7C
&X01111100



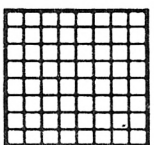
125
&H7D
&X01111101



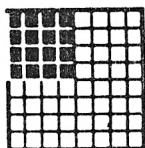
126
&H7E
&X01111110



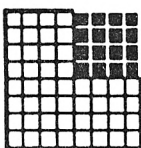
127
&H7F
&X01111111



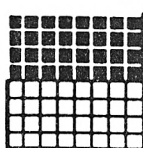
128
&H80
&X10000000



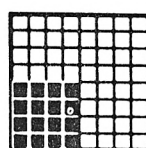
129
&H81
&X10000001



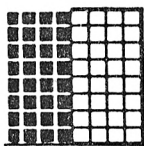
130
&H82
&X10000010



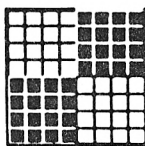
131
&H83
&X10000011



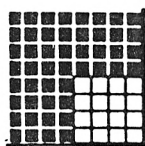
132
&H84
&X10000100



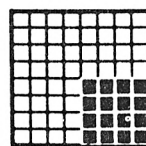
133
&H85
&X10000101



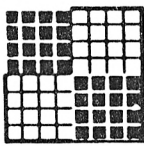
134
&H86
&X10000110



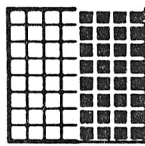
135
&H87
&X10000111



136
&H88
&X10001000



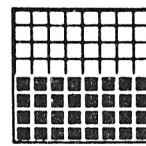
137
&H89
&X10001001



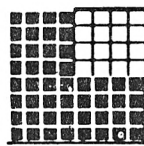
138
&H8A
&X10001010



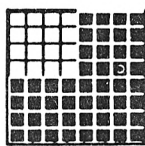
139
&H8B
&X10001011



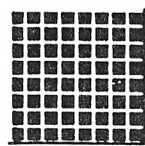
140
&H8C
&X10001100



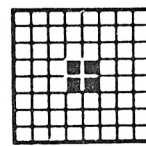
141
&H8D
&X10001101



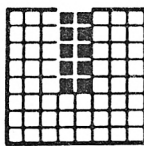
142
&H8E
&X10001110



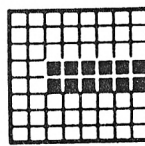
143
&H8F
&X10001111



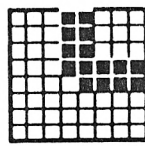
144
&H90
&X10010000



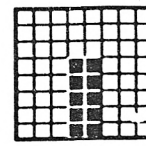
145
&H91
&X10010001



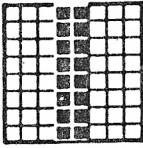
146
&H92
&X10010010



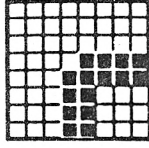
147
&H93
&X10010011



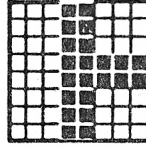
148
&H94
&X10010100



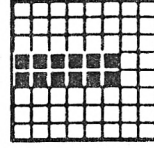
149
&H95
&X10010101



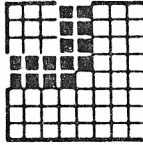
150
&H96
&X10010110



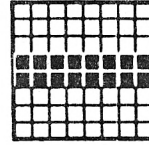
151
&H97
&X10010111



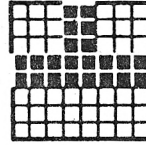
152
&H98
&X10011000



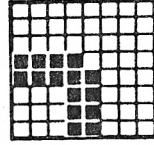
153
&H99
&X10011001



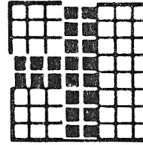
154
&H9A
&X10011010



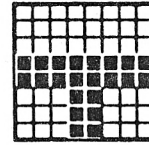
155
&H9B
&X10011011



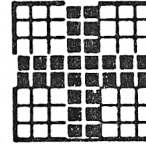
156
&H9C
&X10011100



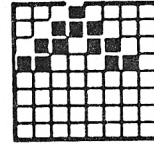
157
&H9D
&X10011101



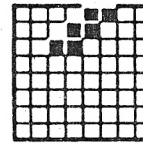
158
&H9E
&X10011110



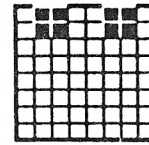
159
&H9F
&X10011111



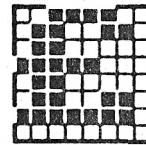
160
&HA0
&X10100000



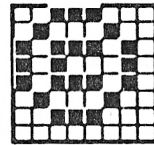
161
&HA1
&X10100001



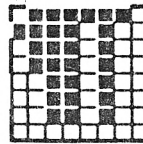
162
&HA2
&X10100010



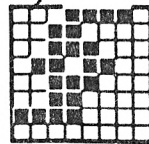
163
&HA3
&X10100011



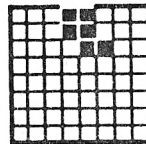
164
&HA4
&X10100100



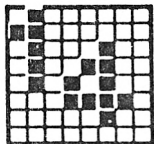
165
&HA5
&X10100101



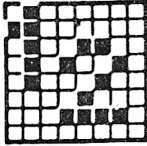
166
&HA6
&X10100110



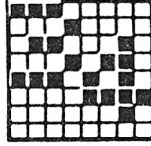
167
&HA7
&X10100111



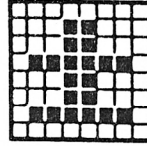
168
&HA8
&X10101000



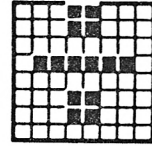
169
&HA9
&X10101001



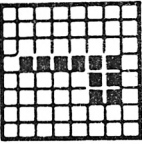
170
&HAA
&X10101010



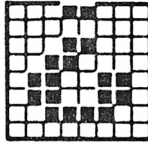
171
&HAB
&X10101011



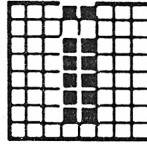
172
&HAC
&X10101100



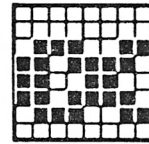
173
&HAD
&X10101101



174
&HAE
&X10101110



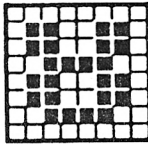
175
&HAF
&X10101111



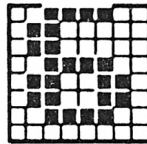
176
&HB0
&X10110000



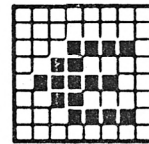
177
&HB1
&X10110001



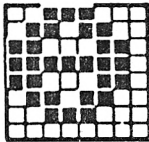
178
&HB2
&X10110010



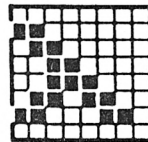
179
&HB3
&X10110011



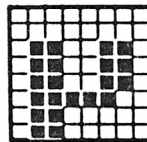
180
&HB4
&X10110100



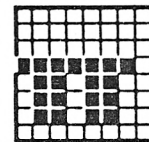
181
&HB5
&X10110101



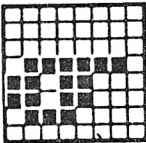
182
&HB6
&X10110110



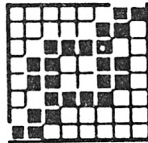
183
&HB7
&X10110111



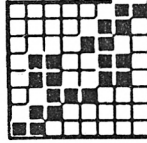
184
&HB8
&X10111000



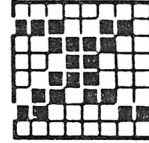
185
&HB9
&X10111001



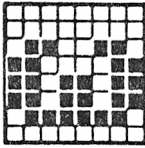
186
&HBA
&X10111010



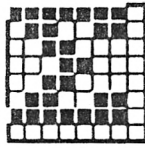
187
&HBB
&X10111011



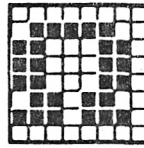
188
&HBC
&X10111100



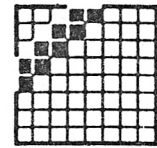
189
&HBD
&X10111101



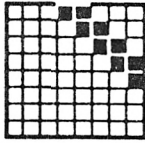
190
&HBE
&X10111110



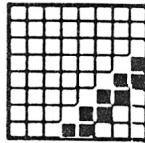
191
&HBF
&X10111111



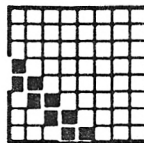
192
&HC0
&X11000000



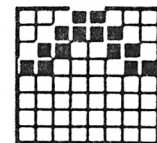
193
&HC1
&X11000001



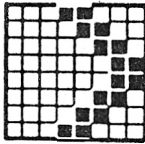
194
&HC2
&X11000010



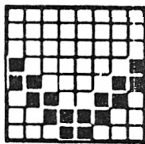
195
&HC3
&X11000011



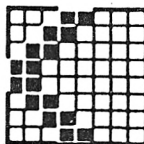
196
&HC4
&X11000100



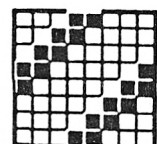
197
&HC5
&X11000101



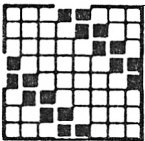
198
&HC6
&X11000110



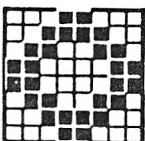
199
&HC7
&X11000111



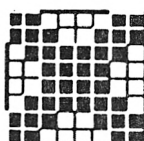
200
&HC8
&X11001000



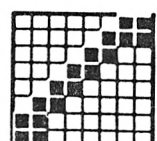
201
&HC9
&X11001001



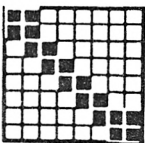
202
&HCA
&X11001010



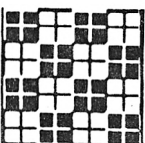
203
&HCB
&X11001011



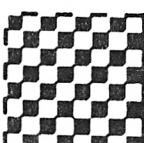
204
&HCC
&X11001100



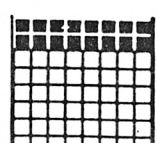
205
&HCD
&X11001101



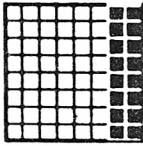
206
&HCE
&X11001110



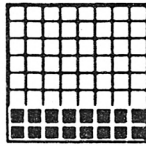
207
&HCF
&X11001111



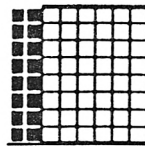
208
&HDO
&X11010000



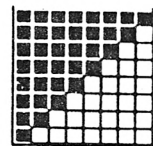
209
&HD1
&X11010001



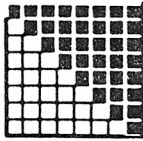
210
&HD2
&X11010010



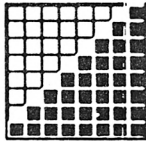
211
&HD3
&X11010011



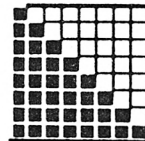
212
&HD4
&X11010100



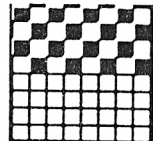
213
&HD5
&X11010101



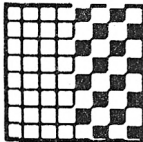
214
&HD6
&X11010110



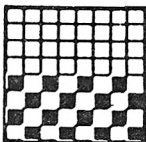
215
&HD7
&X11010111



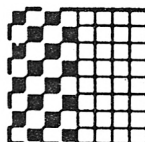
216
&HD8
&X11011000



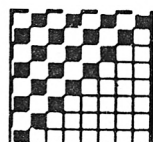
217
&HD9
&X11011001



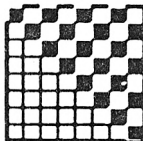
218
&HDA
&X11011010



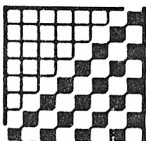
219
&HDB
&X11011011



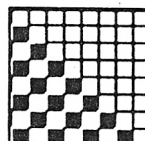
220
&HDC
&X11011100



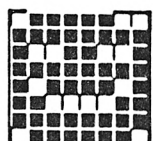
221
&HDD
&X11011101



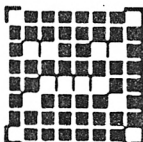
222
&HDE
&X11011110



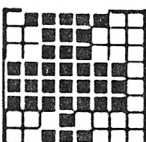
223
&HDF
&X11011111



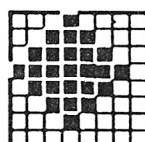
224
&HE0
&X11100000



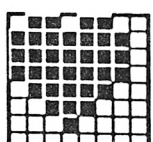
225
&HE1
&X11100001



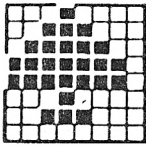
226
&HE2
&X11100010



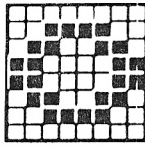
227
&HE3
&X11100011



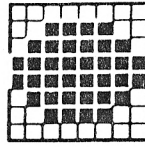
228
&HE4
&X11100100



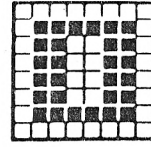
229
&HE5
&X11100101



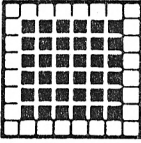
230
&HE6
&X11100110



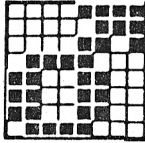
231
&HE7
&X11100111



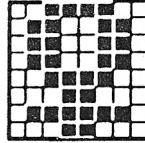
232
&HE8
&X11101000



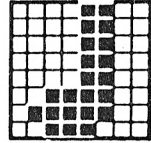
233
&HE9
&X11101001



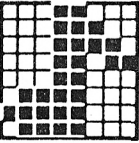
234
&HEA
&X11101010



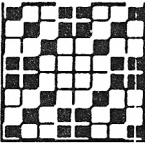
235
&HEB
&X11101011



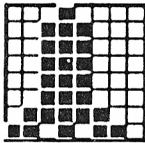
236
&HEC
&X11101100



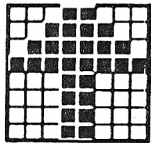
237
&HED
&X11101101



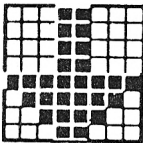
238
&HEE
&X11101110



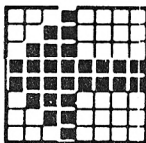
239
&HEF
&X11101111



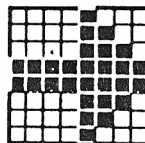
240
&HF0
&X11110000



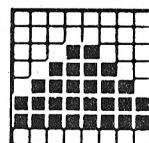
241
&HF1
&X11110001



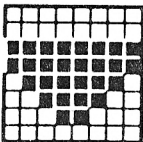
242
&HF2
&X11110010



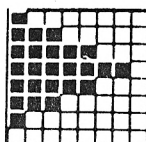
243
&HF3
&X11110011



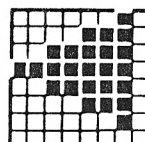
244
&HF4
&X11110100



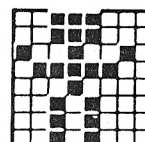
245
&HF5
&X11110101



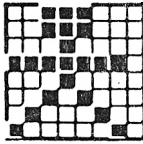
246
&HF6
&X11110110



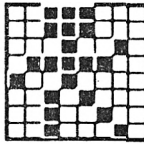
247
&HF7
&X11110111



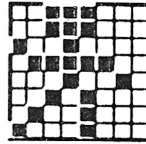
248
&HF8
&X11111000



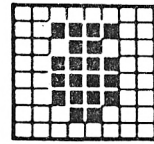
249
&HF9
&X11111001



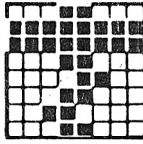
250
&HFA
&X11111010



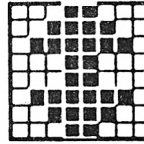
251
&HFB
&X11111011



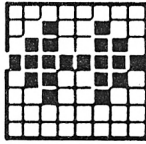
252
&HFC
&X11111100



253
&HFD
&X11111101

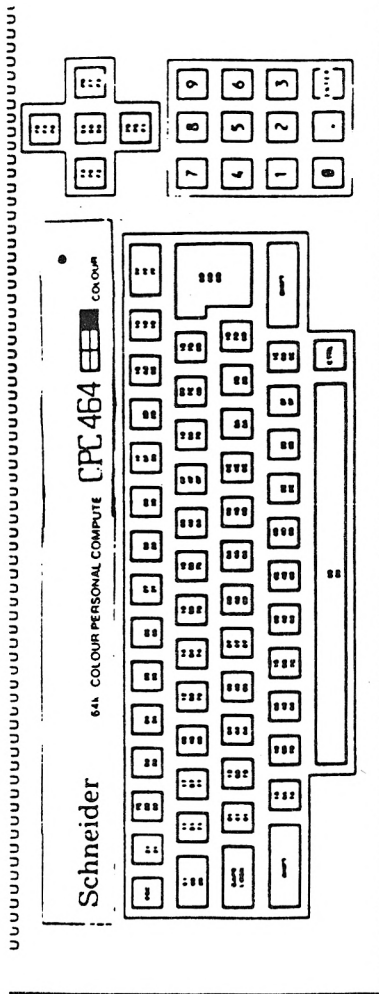


254
&HFE
&X11111110

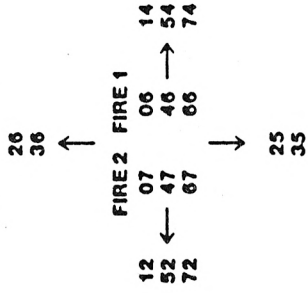


255
&HFF
&X11111111

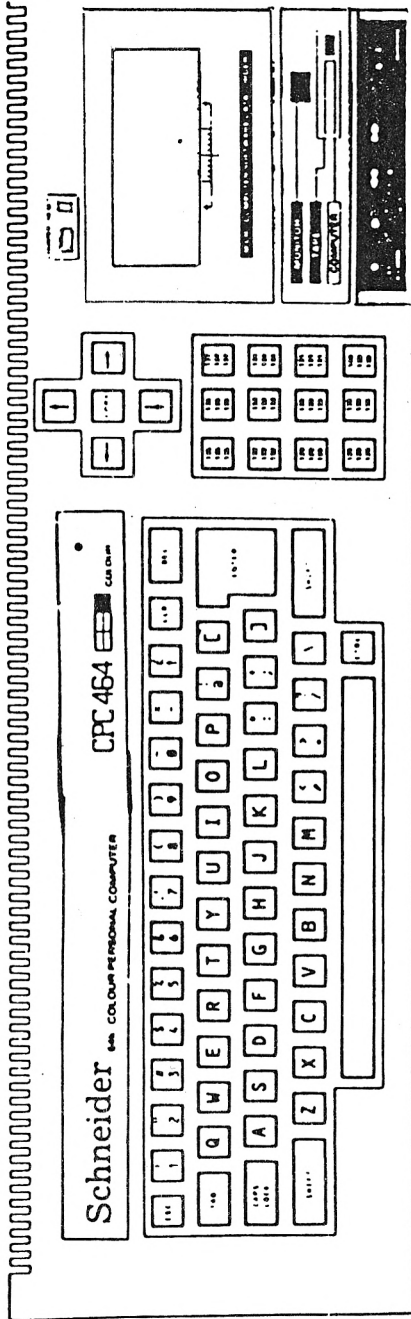
Valores ASCII prescritos como standard



JOYSTICK 1



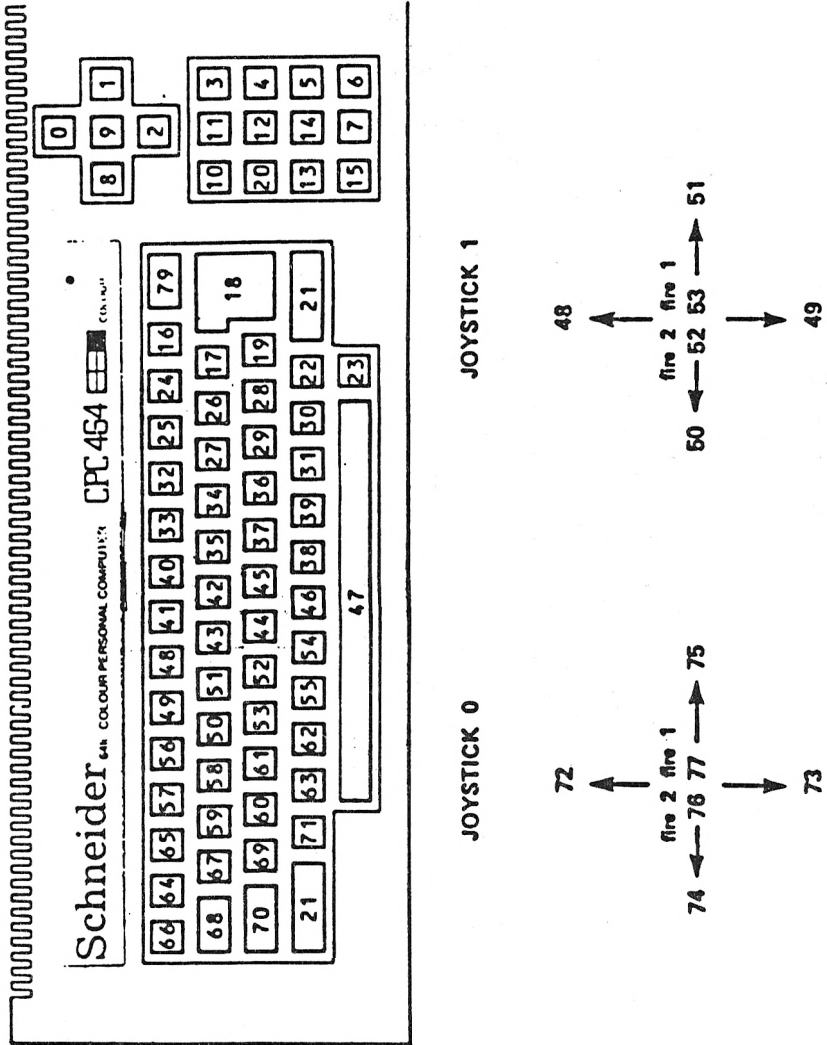
Códigos ASCII y caracteres CPC464



CARACTERES FUNCIONALES
STANDARD, SITUACION
Y VALORES

exp .char	value	ascii
128	0	30
129	1	31
130	2	32
131	3	33
132	4	34
133	5	35
134	6	36
135	7	37
136	8	38
137	9	39
138	(enter)	2E
139	run"	0D
140		52,55,4E,22,0D

Códigos de entrada para teclas y joysticks



INTRODUCCION AL CPC464 PARA EXPERTOS

Las características primordiales del sistema son:

Z80CPU

El microprocesador más ampliamente usado en los ordenadores personales a lo ancho del mundo.

64K RAM

Se suministra como standard una cantidad generosa de RAM, con más de 42K disponibles realmente para el usuario, gracias a la implementación en ROM de técnicas de solapa al implementar el BASIC/.

Pantalla

El CPC464 admite tres modos básicos de manejo en pantalla, incluyendo textos de 80 columnas, un surtido de 27 colores y resolución de hasta 640 x 200 pixels.

Teclado Real

Un teclado de estilo mecanográfico, con un racimo de teclas para cursor y un tablero standard separado para entrada numérica, que dobla el uso para propósitos de teclas funcionales.

Casette incorporado

BASIC

Un BASIC standard en la industria escrito en Inglaterra, más rápido y más versátil de lo que se pudiera esperar de tales BASIC.

Repertorio de caracteres extendido

Un repertorio completo de caracteres de 8 bits, incluyendo símbolos y gráficos es accesible en su mayoría vía el teclado, y usando las funciones **CHR\$(n)**.

Teclas definidas por el usuario

Hasta 32 teclas pueden ser definidas por el usuario con sargas de 32 caracteres.

Subrutinas

Muchas subrutinas ensambladas están disponibles para ser citadas desde BASIC.

MODOS DE PANTALLA

Hay tres modos de operación en pantalla:

a) Normal

Modo 1: 40 columnas x 25 líneas, 4 tintas en texto
320 x 200 pixels, direccionables individualmente en 4 colores.

b) Modo multicolor

20 columnas x 25 líneas, 16 tintas en texto
160 x 200 pixels, individualmente direccionables en 16 colores.

c) Modo de alta resolución

80 columnas x 25 líneas, 2 tintas en texto
640 x 200 pixels, individualmente direccionables en 2 colores.

Selección de color

Ventanas

El usuario puede elegir hasta ocho ventanas de textos en las que se escriben caracteres, y también una ventana de gráficos en la que se pueden efectuar "ploteado".

Cursor

El cursor está inhibido durante los períodos en que la CPU no requiere entrada por teclado, actuando así como un prontador automático. El cursor se representa por un cuadrado de color invertido.

Sonido polifónico

Portal de impresora

Apoyo de expansión

Coordenadas

El origen de texto es la esquina superior izquierda de la pantalla, y las posiciones físicas en pantalla cambian con el modo de imagen.

El origen de gráficos es la esquina inferior izquierda, y supone que la pantalla está en todo momento en el modo de alta resolución aunque los cálculos de tintas serán efectuados correctamente en todos los modos de imagen.

NB

ROMS de expansión

Todas las ROMS ocupan los 16K de la cima de la memoria (donde reside el BASIC) y hay facilidades en el firmware para citar a 240 adicionales, 16K extra ROMS. Debe emplearse externamente una cierta cantidad de circuitos de codificadores de direcciones a la máquina base como parte del hardware de expansión ROM.

Panorámica

Un breve resumen de las principales características del hardware y firmware del CPC464.

1) Hardware

1.1) Dentro de la carcasa principal CPC464:

Computador, teclado, cassette y altavoz. Salidas RGB y luminancias.

1.1.1) LSI Chips

Z80A un procesador ejecutando a 4MHz.

64K bytes de 64K x 1 RAM dinámica refrescada por accesos a la memoria de pantalla.

32K bites de ROM conteniendo BASIC y el sistema operativo (OS).

Una ringla lógica customizada, incorpora casi toda la lógica no incluida previamente en los chips LSI; particularmente temporización, generación de color y circuitería DMA.

La pastilla controladora CRT 6845 genera señales de barrido para la RAM video.

1.1.2) Enchufes externos.

Conectores de borde PCB para expansiones de propósito general (12) y una impresora externa (12) (paralelo Centronix). Enchufes para joystick (14) (9 pines D-Tipo), salida estéreo de sonido (15), salida vídeo (10) (RGB y sync o vídeo compuesto o luminancia y sync).

1.2) Fuera de la carcasa

El sistema CPC464 viene con una elección de dos clases de entrada directa de monitor de vídeo, incluyendo cada una, una fuente de alimentación 5V para el ordenador, diseñado para adaptarse a los standards locales de voltaje de red en el país de venta. Además, hay una PSU opcional y un modulador UHF TV, el MPI.

1.3) Especificación de pantalla

La pantalla opera en 16K de memoria. La máquina tiene un surtido de 27 colores que pueden ser elegidos libremente para formar una paleta. El número de tintas en la paleta depende del modo de pantalla.

TABLA MAESTRA DE COLORES

Nº Tinta	Color Tinta	Nº Tinta	Color Tinta
0	Negro	14	Azul Pastel
1	Azul	15	Naranja
2	Azul Brillante	16	Rosa
3	Rojo	17	Magenta Pastel
4	Magenta	18	Verde Brillante
5	Malva	19	Verde Marino
6	Rojo Brillante	20	Ciano Brillante
7	Púrpura	21	Verde Lima
8	Magenta Brillante	22	Verde Pastel
9	Verde	23	Ciano Pastel
10	Ciano	24	Amarillo Brillante
11	Azul Cielo	25	Amarillo Pastel
12	Amarillo	26	Bianco Brillante
13	Blanco		

1.4) Mapa de memoria

Los 64K de RAM se reparten como sigue:

Observa que parte de la ROM solapa la RAM de pantalla, por lo tanto liberando el área máxima posible para la RAM de usuario durante las operaciones BASIC.

1.4) Mapa de Memoria

Los 64K de RAM se reparten como sigue:

Observa que parte de la ROM solapa la RAM de pantalla, por lo tanto liberando el área máxima posible para la RAM de usuario durante las operaciones BASIC.

ROM SECTION 0 <	0000H	
	3FFFH	
	4000H	
	7FFFH	
	8000H	
	BFFFH	
ROM SECTION 1 <	C000H	> RAM [Screen 3]
	FFFFH	

Cuando hay ambas, RAM y ROM, en una dirección, entonces READING accesa la ROM y WRITING accesa la RAM. Además, la sección ROM puede ser desactivada, permitiendo acceder para leer la RAM en la misma dirección.

1.5) Habilidad de ampliación

1.5.1) ROMS colaterales

Está hecha la provisión para que puedan elegirse ROMS adicionales en lugar de cualquier parte de la ROM incorporada.

1.5.2) RAM Adicional

Se puede conmutar RAM adicional en cualquier parte de la RAM incorporada.

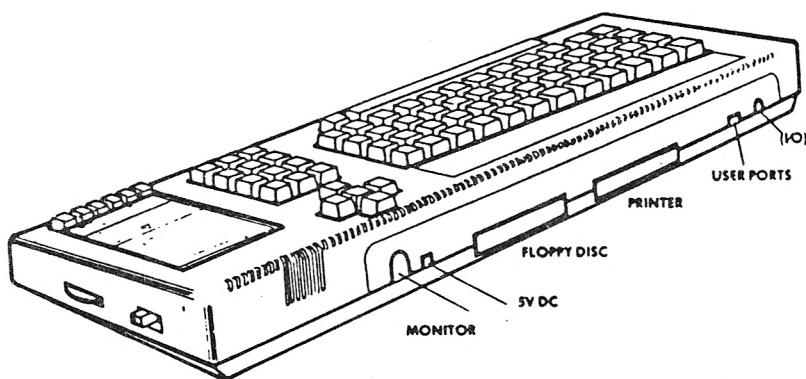
1.5.3) I/O Adicional

La mayoría de las direcciones de portales I/O están reservadas por el ordenador, y en particular las direcciones por debajo 7Fxx no serán usadas en absoluto. Lo siguiente puede ser usado por hardware externo. F8xx, F9xx, FAxx, FBxx.

El bus de expansión para periféricos debe decodificar las direcciones de A0 a A7 mientras que la dirección A10 está baja. Los canales del bus de expansión en la gama de direcciones F800 a FBFF están reservadas como sigue:

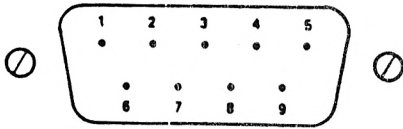
ADDRESSES A0-A7

00 - 78	** Do not use **
7C - 7F	Reserved for disk interface
80 - BB	** Do not use **
BC - BF	Reserved for future use
CD - DB	** Do not use **
DC - DF	Reserved for communications interfaces
EO - FF	Available for user peripherals

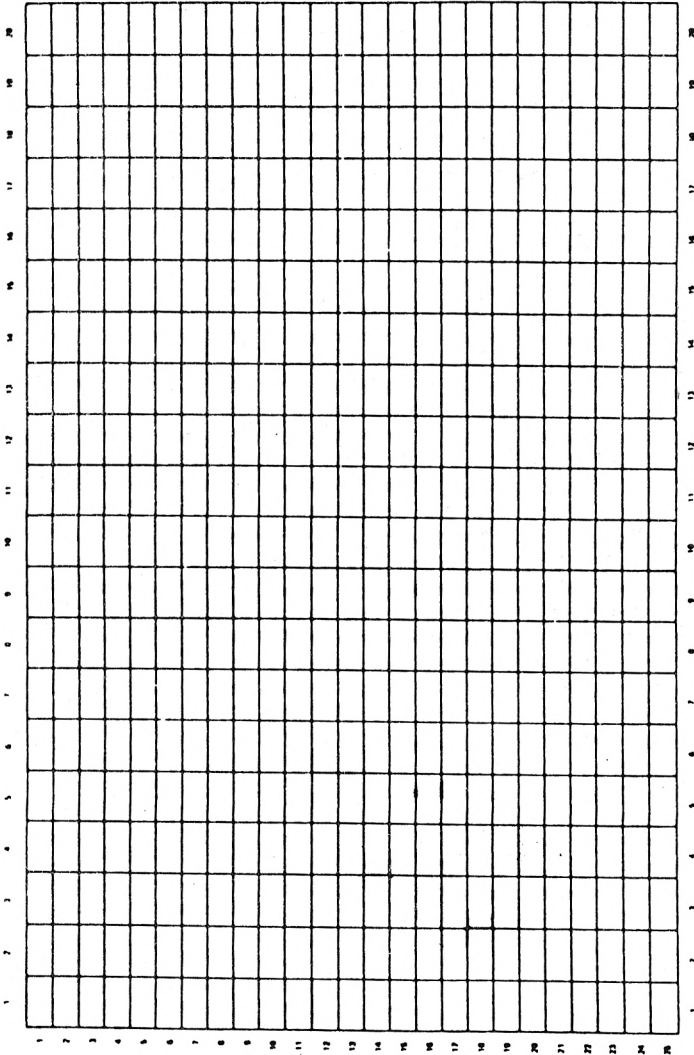


PADDLE PORT CONNECTOR (9 PIN D)

VIEWED FROM REAR

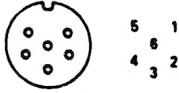


PIN 1	UP	PIN 6	FIRE 2
PIN 2	DOWN	PIN 7	FIRE 1
PIN 3	LEFT	PIN 8	COMMON
PIN 4	RIGHT	PIN 9	COM 2
PIN 5	SPARE		

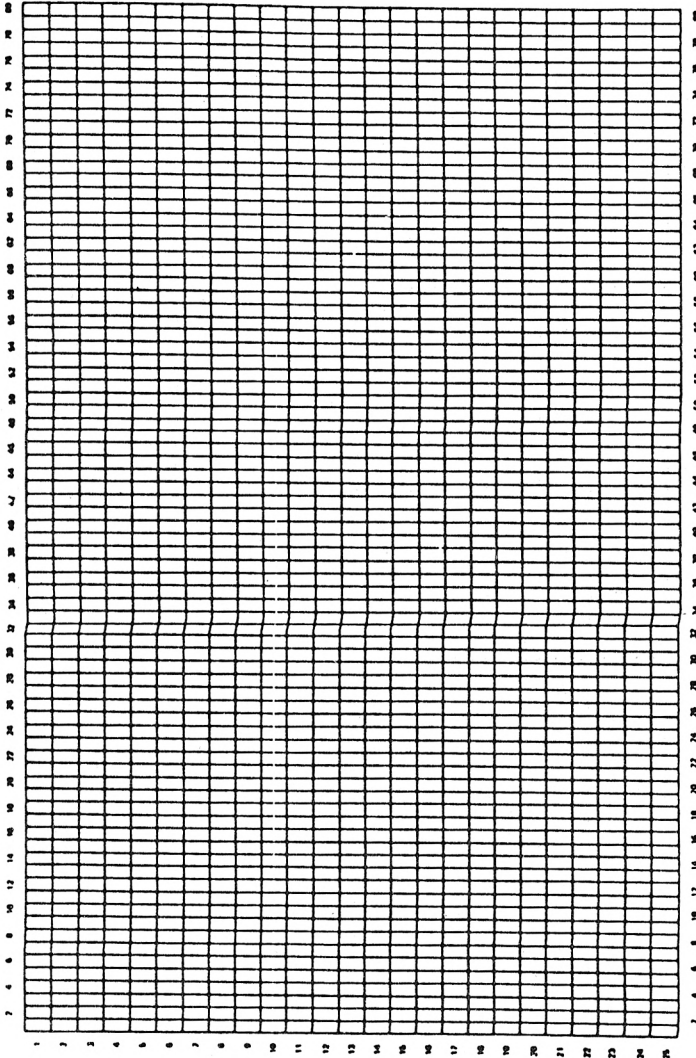


VIDEO OUTPUT CONNECTOR (6 PIN DIN)

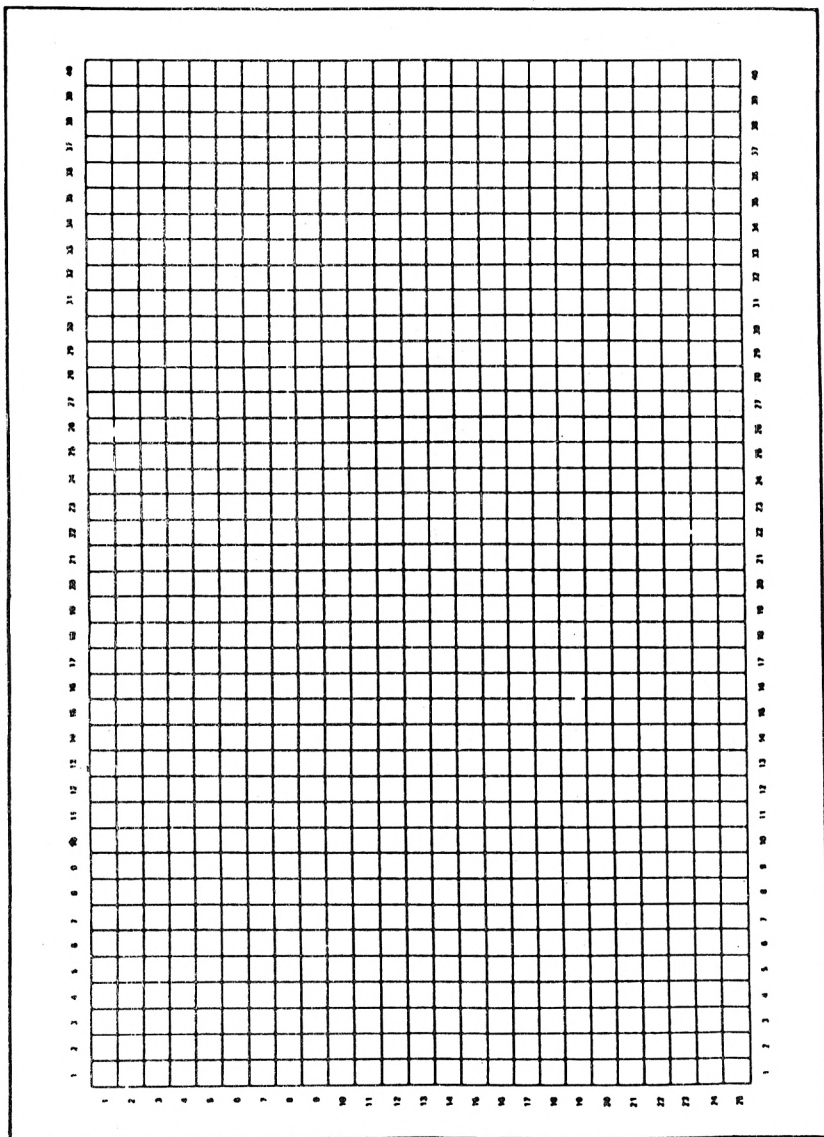
VIEWED FROM REAR



PIN 1	RED	PIN 4	SYNC
PIN 2	GREEN	PIN 5	GND
PIN 3	BLUE	PIN 6	LUM



Reticula de TEXTO y VENTANA
MODO 1 40 columnas



Notas y Períodos de Tono

La tabla que sigue da los ajustes recomendados de períodos de tono para notas en la escala usual de pentagrama para la gama completa de 8 octavas.

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	32.703	3822	-0.007%	
C#	34.648	3608	+0.007%	
D	36.708	3405	-0.007%	
D#	38.891	3214	-0.004%	
E	41.203	3034	+0.009%	
F	43.654	2863	-0.016%	Octave -3
F#	46.249	2703	+0.009%	
G	48.999	2551	-0.002%	
G#	51.913	2408	+0.005%	
A	55.000	2273	+0.012%	
A#	58.270	2145	-0.008%	
B	61.735	2025	+0.011%	
NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	65.406	1911	-0.007%	
C#	69.296	1804	+0.007%	
D	73.416	1703	+0.022%	
D#	77.782	1607	-0.004%	
E	82.407	1517	+0.009%	
F	87.307	1432	+0.019%	Octave -2
F#	92.499	1351	-0.028%	
G	97.999	1276	+0.037%	
G#	103.826	1204	+0.005%	
A	110.000	1136	-0.032%	
A#	116.541	1073	+0.039%	
B	123.471	1012	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	130.813	956	+0.046%	
C#	138.591	902	+0.007%	
D	146.832	851	-0.037%	
D#	155.564	804	+0.058%	
E	164.814	758	-0.057%	
F	174.614	716	+0.019%	Octave -1
F#	184.997	676	+0.046%	
G	195.998	638	+0.037%	
G#	207.652	602	+0.005%	
A	220.000	568	-0.032%	
A#	233.082	536	-0.055%	
B	246.942	506	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	261.626	478	+0.046%	Middle C
C#	277.183	451	+0.007%	
D	293.665	426	+0.081%	
D#	311.127	402	+0.058%	
E	329.628	379	-0.057%	
F	349.228	358	+0.019%	Octave 0
F#	369.994	338	+0.046%	
G	391.995	319	+0.037%	
G#	415.305	301	+0.005%	
A	440.000	284	-0.032%	International A
A#	466.164	268	-0.055%	
B	493.883	253	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	523.251	239	+0.046%	
C#	554.365	225	-0.215%	
D	587.330	213	+0.081%	
D#	622.254	201	+0.058%	
E	659.255	190	+0.206%	
F	698.457	179	+0.019%	Octave 1
F#	739.989	169	+0.046%	
G	783.991	159	-0.277%	
G#	830.609	150	-0.328%	
A	880.000	142	-0.032%	
A#	932.328	134	-0.055%	
B	987.767	127	+0.356%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	1046.502	119	-0.374%	
C#	1108.731	113	+0.229%	
D	1174.659	106	-0.390%	
D#	1244.508	100	-0.441%	
E	1318.510	95	+0.206%	
F	1396.913	89	-0.543%	Octave 2
F#	1479.978	84	-0.548%	
G	1567.982	80	+0.350%	
G#	1661.219	75	-0.328%	
A	1760.000	71	-0.032%	
A#	1864.655	67	-0.055%	
B	1975.533	63	-0.435%	
NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	2093.004	60	+0.462%	
C#	2217.461	56	-0.662%	
D	2349.318	53	-0.390%	
D#	2489.016	50	-0.441%	
E	2637.021	47	-0.855%	
F	2793.826	45	+0.574%	Octave 3
F#	2959.955	42	-0.548%	
G	3135.963	40	+0.350%	
G#	3322.438	38	+0.992%	
A	3520.000	36	+1.357%	
A#	3729.310	34	+1.417%	
B	3951.066	32	+1.134%	
NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	4186.009	30	+0.462%	
C#	4434.922	28	-0.662%	
D	4698.636	27	+1.469%	
D#	4978.032	25	-0.441%	
E	5274.041	24	+1.246%	
F	5587.652	22	-1.685%	Octave 4
F#	5919.911	21	-0.548%	
G	6271.927	20	+0.350%	
G#	6644.875	19	+0.992%	
A	7040.000	18	+1.357%	
A#	7458.621	17	+1.417%	
B	7902.133	16	+1.134%	

Estos valores están todos calculados a partir de la A internacional:

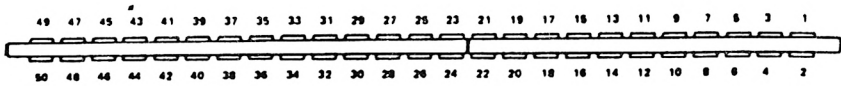
$$\text{FREQUENCY} = 440 * (2^{\uparrow} (\text{OCTAVE} + (10 - N) / 12))$$

$$\text{PERIOD} = \text{ROUND}(125000 / \text{FREQUENCY})$$

donde N es 1 para C, 2 para C#, 3 para D, etc.

EXPANSION PORT 50 WAY 0.1 EDGE CONNECTOR

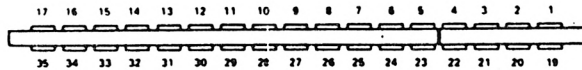
VIEWED FROM REAR



PIN 1	SOUND	PIN 18	A0	PIN 35	INT
PIN 2	GND	PIN 19	D7	PIN 36	NMI
PIN 3	A15	PIN 20	D6	PIN 37	BUSRD
PIN 4	A14	PIN 21	D5	PIN 38	BUSAK
PIN 5	A13	PIN 22	D4	PIN 39	READY
PIN 6	A12	PIN 23	D3	PIN 40	BUS RESET
PIN 7	A11	PIN 24	D2	PIN 41	RESET
PIN 8	A10	PIN 25	D1	PIN 42	ROMEN
PIN 9	A9	PIN 26	D0	PIN 43	ROMDIS
PIN 10	A8	PIN 27	+5v	PIN 44	RAMRD
PIN 11	A7	PIN 28	MREQ	PIN 45	RAMDIS
PIN 12	A6	PIN 29	M1	PIN 46	CURSOR
PIN 13	A5	PIN 30	RFSH	PIN 47	L. PEN
PIN 14	A4	PIN 31	IORQ	PIN 48	EXP
PIN 15	A3	PIN 32	RD	PIN 49	GND
PIN 16	A2	PIN 33	WR	PIN 50	✓
PIN 17	A1	PIN 34	HALT		

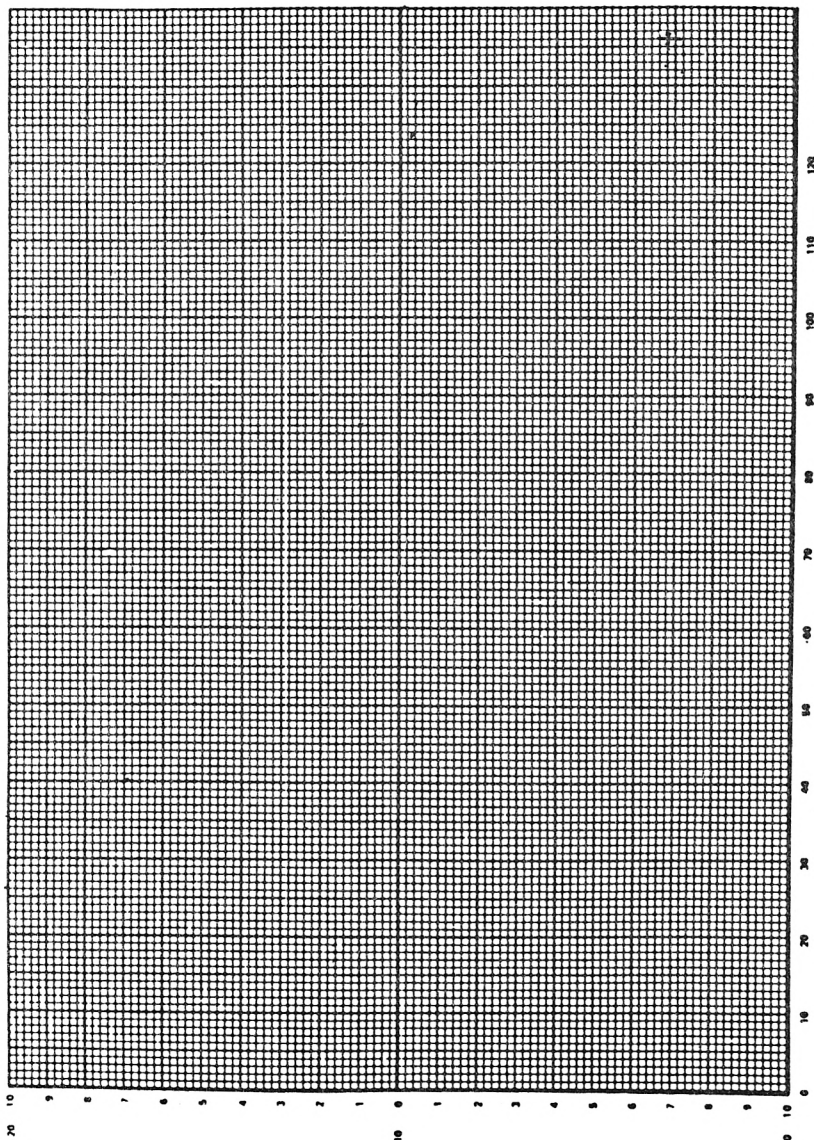
PRINTER PORT 34 WAY 0.1 EDGE CONNECTOR

VIEWED FROM REAR



PIN 1	STROBE	PIN 19	GND
PIN 2	D0	PIN 20	GND
PIN 3	D1	PIN 21	GND
PIN 4	D2	PIN 22	GND
PIN 5	D3	PIN 23	GND
PIN 6	D4	PIN 24	GND
PIN 7	D5	PIN 25	GND
PIN 8	D6	PIN 26	GND
PIN 9	D7	PIN 27	GND
PIN 11	BUSY	PIN 28	GND
PIN 14	GND	PIN 33	GND
PIN 16	GND	All other pins	NC

Retícula para música/envólvntes



Códigos de error y palabras reservadas

Números y mensajes de error

10 ON ERROR GOTO 1000

program

1000 PRINT CHR\$(7): MODE 2 INK 1,0: INK 0,9: CLS: LIST

Syntax Error

CODIGOS DE ERROR Y PALABRAS RESERVADAS

1. Unexpected NEXT

Ha encontrado una instrucción que le manda dar OTRA posible ronda de un bucle, "inesperadamente" porque no estaba repitiendo nada; o la variable de control mencionada en la instrucción **NEXT** no concuerda con la mencionada en la instrucción **FOR**.

2. Syntax Error

Normalmente alguna palabra clave está mal deletreada o no hay igual número de paréntesis de cierre que paréntesis de apertura en una expresión.

3. Unexpected Return

Se ha encontrado con una instrucción en que le pides VUELVA "inesperada", porque no se había desviado en ningún momento anterior.

4. DATA exhausted

Le mandas que "apunte" el siguiente DATO y ya está "exhausta"... la serie de constantes que mencionas en el programa.

5. Improper argument

Tiene un significado general que indica que el valor del ARGUMENTO de una función es

“impropio”, o que el parámetro mencionado en una instrucción “no es válido”.

6. Overflow

El resultado de una operación aritmética hace que se rebase la gama de valores permitida a la variable. Puede corresponder a una variable.

7. Memory full

La memoria está “llena”. El programa es demasiado largo, o simplemente hay demasiadas variables, o innumerables “desvios” y “bucles”.

8. Line does not exist

Mencionas un número de líneas que no existe en el programa.

9. Subscript out of range

Uno de los SUFIJOS (“suscritos” o “subíndices”) usado para seleccionar un elemento de una tabla está fuera de gama, por demasiado grande o demasiado pequeño.

10. Array already dimensioned

La tabla (RINGLA) que mencionas “ya está dimensionada”. Debes previamente hacer que LIBRE el espacio pertinente (ERASE).

11. División by zero

Con esos valores resulta una “división por cero”.

12. Invalid direct command

Le estás dando un comando, usando una palabra clave que sólo admite como instrucción, is, precedida de un número de línea.

13. Typs mismatch

“Clase discordancia”. Das como valor de una variable literal una constante numérica, o viceversa.

14. String space full

Está "lleno el espacio" que internamente reserva para las "sartas" de caracteres (LITROS ENTRECOMILLADOS). Ya no dispone de más espacio, incluso aunque haya hecho "recogida de basura". (Supresión de los viejos valores de las variables literales).

15. String too long

"SARTA demasiado larga", pues sobrepasa de 255 caracteres. Pueden ser resultado del empalme de datos litéricos.

16. String expression too complex

Cuando las "expresiones literales son demasiado complejas" porque pueden dar lugar a demasiados valores intermedios, que exceden de un límite razonable.

17. Cannot CONTinue

"No se puede continuar". Observa que el comando **CONT** es para hacer que SIGA un programa que se ha visto interrumpido por encontrar una instrucción **STOP**, que le hace que se pare.

18. Unknown user function

Esa función definible por el usuario es "desconocida". Necesitas avisarle previamente para que la use.

19. RESUME missing

NO es que "falta" el resumen. Lo que SI falta es la instrucción para que REANUDE en algún punto el programa después de haberse visto "atrapado" en un error, que el propio programa se encarga de tratar.

20. Unexpected RESUME

NO es un "inesperado" resumen. SI es que no puedes decirle "inesperadamente" que reanude cuando no se ha visto "atrapado" en ningún error. Coloca una instrucción para que PARE o TERMINE antes de que llegue al párrafo pertinente.

21. Direct command found

Sólo puede aparecer después de CARGAR un programa desde el cassette, y es porque ha “encontrado” un COMANDO (directo) y no una instrucción (con número de línea delante).

22. Operand missing

En esa expresión “falta” algún operando.

23. Line too long

“Línea demasiado larga”.

24. EOF met

Intentas tomar información del cassette, y te has “topado” con el final del fichero.

25. File type error

“Error en la CLASE (tipo) de fichero”. El fichero que hay en la cinta no es de la clase adecuada.

26. NEXT missing

No puede encontrar la instrucción que le manda hacer OTRA ronda, y estando dentro de un bucle.

27. File already open

Ese “fichero ya está abierto”. Quizás convenga decirse que lo CIERRE, mediante la instrucción **CLOSED**.

28. Unknown command

Comando “desconocido”.

29. WEND missing

No puede encontrar la instrucción **WEND** que siempre tiene que haber en correspondencia con cada instrucción **WHILE**.

30. Unexpected WEND

Encontró una instrucción **WEND** sin estar dentro de un bucle **WHILE**.

PALABRAS CLAVE EN EL BASIC SCHNEIDER

Las siguientes, son las palabras claves en BASIC, y están reservadas para significar exactamente lo que hemos descrito en ésta Guía.

ABS, AFTER, AND, ASC, ATN, AUTO

BIN\$, BORDER

CALL, CAT, CHAIN, CHR\$, CINT, CLEAR, CLG, CLOSEIN, CLOSEOUT, CLS, CONT, COS, CREAL

DATA, DEF, DEFINIT, DEFREAL, DEFSTR, DEG, DELETE, DI, DIM, DRAW, DRAWR

EDIT, EI, ELSE, END, ENT, ENV, EOF, ERASE, ERL, ERR, ERROR, EVERY, EXP

FIX, FN, FOR, FRE

GOSUB, GOTO

HEX\$, HIMEN

IF, INK, INKEY, INKEY\$, INP, INPUT, INSTR, INT

JOY

KEY

LEFT\$, LEN, LET, LINE, LIST, LOAD, LOCATE, LOG, LOGIO, LOWERS\$

MAX, MEMORY, MERGE, MID\$, MIN, MOD, MODE, MOVE, MOVER,

NEXT, NEW, NOT

ON, ON BREAK, ON ERROR GOTO, ON SQ, OPENIN, OPENOUT, OR,
ORIGIN, OUT

PAPER, PEEK, PEN, PI, PLOTR, PLOTA, POJE, POS, PRINT

RAD, RANDOMIZE, READ, RELEASE, REM, REMAIN, RENUM, RESTO-
RE, RESUME, RETURN, RIGH\$, REND, ROUND, RUN

SAVE, SGN, SIN, SOUND, SPACE\$, SPC, SPEED, SQ, SQR, STEP, STOP,
STR\$, STRING\$, SWAP, SYMBOL

TAB, TAG, TAGOFF, TAN, TEST, TESTR, THEN, TIME, TO, TROFF,
TRON

UNT, UPPERS\$, USING

VAL, VPOS

WAITT, WEND, WHILE, WIDTH, WINDOW, WRITE,

XOR, XPOS

YPOS

ZONE

LA PANTALLA

En la pantalla cada punto de imagen (pixel) puede especificarse individualmente. Cada punto es la proyección de un cierto número de bits en un área de la memoria principal.

La pantalla se presenta en dos formas: una pantalla de Textos y una pantalla de Gráficos. El programa para la gestión de la pantalla de texto divide ésta en un cierto número de “células de un carácter” y se preocupa de mostrar caracteres y símbolos.

Modos, Resolución, Colores y Tintas.

La imagen en pantalla es proyección del contenido de una zona de memoria de 16K bytes, con 4, 2 ó 1 bit para determinar el color de cada punto, dependiendo del modo.

La pantalla es capaz de dibujar en 27 colores diferentes (incluyendo blanco y negro).

Los modos de pantalla son los siguientes:

- Modo 0: 25 líneas de 20 caracteres cada una, o bien 160 puntos de anchura por 200 puntos de altura. 16 tintas posibles (4 bits por punto).
- Modo 1: 25 líneas de 40 caracteres cada una, o bien 320 puntos de anchura por 200 puntos de altura. 4 tintas posibles (2 bits por punto).
- Modo 2: 25 líneas de 80 caracteres cada una, o bien 640 puntos de anchura por 200 puntos de altura, 2 tintas posibles (1 bit por punto).

La pantalla de Texto y los Caracteres Definibles por el Usuario

Cuando se usa la pantalla para exponer texto se divide en “células de caracteres”, o cuadrantes, formado por 8 puntos a lo ancho y 8 puntos a lo alto. Eso permite 25 líneas de caracteres en todos los modos de pantalla. Las líneas (la posición “y”) se numeran a partir de 1 que es la parte superior de la pantalla, hasta la línea más inferior que es la línea 25. Los caracteres se numeran a lo largo de cada línea (la posición “x”) a partir de 1 en el extremo izquierdo de la pantalla, de manera que el situado en el extremo derecho de la pantalla es el carácter 20, 40 u 80, dependiendo del modo.

El repertorio de caracteres comprende 256 caracteres diferentes. Los caracteres con valores asociados —códigos— entre 32 y 126 son los caracteres habituales ASCII. Los otros caracteres corresponden a diversos símbolos y “cuadratines gráficos”. El aspecto visual de cada carácter queda definido por una matriz de 8 octetos. Los últimos 16 caracteres son implícitamente definibles por el usuario, es decir, su “matriz de forma” puede cambiarse a discreción aunque los que ya estuvieran expuestos en pantalla no se ven afectados. El número de caracteres definibles por el usuario puede cambiarse usando el comando **SYMBOL AFTER**.

Si el bit en la matriz de forma está puesto a 1, el punto de imagen correspondiente en la célula de carácter aparece en el color designado por la Tinta de la Pluma. Cuando el bit en la matriz de forma del carácter está puesto a 0, el punto de imagen correspondiente queda afectado de la forma siguiente:

en modo Normal o Escritura: el punto aparece en el color correspondiente al Tinte del Papel, de manera que todo el fondo del carácter queda relleno en ese color.

en el modo Transparente: el punto permanece en el color que tuviera, de manera que el color con que aparece corresponde al fondo previo y se muestra a través —de ahí el nombre “transparente”—.

Cauces hacia Pantalla y Ventanas de Texto

Los primeros 8 cauces de salida de información, identificados como cauces 0..7, envían caracteres hacia la pantalla de texto. Cada cauce posee un conjunto separado de atributos.

El área de la pantalla por donde se proyecta información a través de un cauce dado, se conoce como Ventana de dicho cauce. El comando **WINDOW** puede usarse para recuadrar una cierta fracción de la pantalla.

El cursor de texto puede desplazarse fuera de la ventana corriente o vigente, pero antes se obliga automáticamente al cursor para que ocupe una posición legal en la ventana, lo que puede provocar que la ventana se desplace hacia arriba o hacia abajo.

Los caracteres enviados hacia la pantalla de texto se interpretan como sigue:

0..31 Caracteres de Control. Tienen un significado especial para la pantalla de texto, tal como se describe en el Apéndice. Los caracteres de

Control pueden emplearse para desplazar el cursor, limpiar la "ventana", fijar las tintas, y otras gestiones.

Para exponer en pantalla el símbolo asociado a un carácter, con código en la banda 0..31, el carácter debe estar precedido por **CHR\$** (1).

32..127 Caracteres ASCII. La forma del carácter que aparece en pantalla puede alterarse usando el mecanismo de símbolos definibles por el usuario.

128..255 Símbolos surtidos. Los últimos 16 símbolos son implícitamente definibles por el usuario, aunque esta cantidad de símbolos puede cambiarse (subirla o bajarla) mediante el comando **SYMBOL AFTER**.

Pantalla de Gráficos, Sistema de Coordenadas, Ventana y Origen

El sistema de coordenadas para la pantalla de gráficos divide ésta en una gradilla de malla cuadrada con 640 puntos de anchura y 400 puntos de altura.

Las coordenadas usadas en la pantalla de gráficos crecen hacia arriba y hacia la derecha, y parten del valor 0. Por lo tanto, la esquina inferior izquierda de la pantalla tiene como posición en coordenadas absolutas la (0,0); y la esquina superior izquierda de la pantalla tiene la posición (639, 399) como coordenadas absolutas.

El origen de todo el sistema de coordenadas puede desplazarse de su posición base, en la esquina inferior izquierda de la pantalla, mediante el comando **ORIGIN**.

Se puede combinar gráficos y textos en una cierta medida, mediante la posibilidad de texto en la posición del Cursor de gráficos.

El color con que aparecen los puntos de imagen trazadas por los comandos **PLOT**, **DRAW** o **TAG**, depende del Modo de Tinta para Gráficos, que puede ser uno de los siguientes:

- Normal** El punto tiene el color de la tinta nueva
- XOR** El punto toma el color 0-lógico exclusivo entre las tintas vieja y nueva.
- AND** El punto toma el color Y-lógico entre las tintas vieja y nueva.
- OR** El punto toma el color 0-lógico entre las tintas vieja y nueva.

Resumen de Comandos y Funciones de Pantalla

BORDER	fija la tinta del "borde".
CLG	limpia la pantalla de gráficos.
CLS	limpia la ventana de texto del cauce dado.
DRAW	traza recta en pantalla de gráficos —hasta una posición absoluta—.
DRAWR	traza recta en pantalla de gráficos. —hasta una posición relativa—.
INK	fija una tinta dada.
LOCATE	sitúa el cursor de texto de un cauce dado.
MODE	cambia a un nuevo modo de pantalla.
MOVE	mueve el cursor de gráficos, hasta una posición absoluta.
MOVER	mueve el cursor de gráficos, hasta una posición relativa.
ORIGIN	fija el origen de coordenadas para gráficos y la ventana.
PAPER	fija el Tinte del Papel para el cauce dado.
PEN	fija la tinta de la Pluma para el cauce dado.
PLOT	pinta un punto en pantalla de gráficos, en posición absoluta.
PLOTR	pinta un punto en pantalla de gráficos, en posición relativa.
POS	posición horizontal del cursor de textos para el cauce dado.
SPEED INK	fija la rapidez del parpadeo de Tintas alternantes.
SYMBOL	fija la matriz de Forma de los caracteres definidos por el usuario.
SYMBOL AFTER	fija como caracteres definibles por el usuario los de "después" del dado.

TAG	fija el modo Texto en Cursor de Gráficos para el cauce dado.
TAGOFF	quita el modo Texto en Cursor de Gráficos para el cauce dado.
TEST	número de tinta usada para un punto en gráficos, —coordenadas absolutas—.
TESTR	número de tinta usada para un punto en gráficos, —coordenadas relativas—.
VPOS	posición vertical del cursor de textos para el cauce dado.
WINDOW	fija la ventana para proyección de texto a través del cauce dado.
WINDOW SWAP	intercambia, “canjea” las ventanas correspondientes a los dos cauces dados.
XPOS	posición horizontal (coordenada X) del cursor de gráficos.
YPOS	posición vertical (coordenada Y) del cursor de gráficos.

TECLADO Y JOYSTICKS

INKEY	Estado actual de la tecla dada por su número.
INKEYS	Tecla pendiente de tratar, si hay, en el “buffer” del teclado.
JOY	Estado del mando de juegos dado por su número.
KEY	Asocia un nuevo “literal” a la tecla dada por su número.
KEY DEF	Redefine el valor generado por la pulsación de la tecla dada.
SPEED KEY	Fija la demora al arranque y el periodo de repetición de la tecla dada.

INTRODUCCION DE LINEAS, EDICION Y COPIADO

Las teclas del cursor tienen tres significados, dependiendo del "turno" en que se encuentren: como sigue:

"Normal" mueve el cursor de entrada dentro de la línea en curso. Observa que no está permitido sacarlo de dicha línea por ninguno de los extremos.

Sólo mientras no existe línea en curso, mientras está vacía, se puede con estas teclas mover el cursor por toda la pantalla.

con SHIFT "Control" mueve el cursor de copiado por toda la pantalla

[CTRL] y arriba lo mueve hasta el comienzo de la línea lógica actual.

[CTRL] y abajo lo mueve hasta el término de la línea lógica actual.

[CTRL] e izquierda lo mueve hasta el comienzo de la línea física (el "renglón" en pantalla).

[CTRL] y derecha lo mueve hasta el término de la línea física (el "renglón" en pantalla).

Las otras teclas de importancia especial en la edición de líneas, son:

[DEL] Suprime el carácter situado a la izquierda del cursor de entrada.

[CLR] Suprime el carácter situado en la posición del cursor de entrada.

[ESC] Abandona la línea en curso.

[CTRL] - [TAB]

Bascula el modo Inserción/Sustitución.

[ENTER]

Concluye la edición de una línea y la pone adentro para ser tratada por el BASIC.

CAUCES DE ENTRADA Y SALIDA

La entrada y salida de datos hacia el BASIC usa cauces. Hay diez cauces de entrada. Los cauces de salida son:

0..7 Caudes hacia pantalla de texto.

8 Cauce hacia impresora.

9 Cauce hacia un fichero de salida en cassette.

Los cauces de entrada son:

0..7 Del teclado, con avisos enviados hacia el cauce correspondiente de la pantalla.

8 Del teclado, con avisos hacia el cauce de impresora.

9 De un fichero de entrada en cassette.

Caudes de salida

Los comandos para salida de textos, incluyen un parámetro opcional para designar el cauce de envío (si se omite se adopta el cauce 0). Así se consigue que los comandos sean coherentes para todos los diversos equipos y dispositivos.

Caudes de entrada

Sólo hay dos fuentes concretas para introducción de datos hacia el BASIC: el teclado y la cassette. Cuando el BASIC ejecuta un comando **INPUT**, envía un aviso —en forma de signo de interrogación— para indicar que está esperando recibir datos que “impondrá” como valores de las variables. Los cauces de teclado, todos recogen los datos de entrada procedentes del teclado, pero el aviso se dirige hacia el correspondiente cauce de salida, pantalla o impresora.

Ficheros en cassette

Los cauces de entrada y salida hacia el cassette se mencionan como "cauces de fichero".

El BASIC hace la lectura y escritura de ficheros, subdividiéndolos en bloques de 2K bytes. Cuando se abre un fichero como entrada (usando OPENIN) se busca en la cinta el fichero dado, y se hace la lectura del primer bloque trasasándolo a la memoria, y preparándose para su tratamiento. Cuando se abre un fichero de salida (usando OPENOUT) no se efectúa ninguna escritura en la cinta hasta que se haya preparado en el "buffer", o zona reservada de la memoria, el byte número $2K + 1$, y en ese momento el BASIC se ve forzado a vaciar dicho "buffer" y transferir los datos a la cinta.

Los programas que efectúen lectura y escritura de ficheros en cassette deben cerrarlos siempre (usando CLOSEIN Y CLOSEOUT) cuando han acabado de tratarlos. La instrucción END, o la finalización del programa, implícitamente cierran cualquier fichero que hubiera en cassette.

GENERACION DE SONIDO

Los circuitos de generación de sonidos disponen de tres canales independientes (A, B y C). Cada canal es capaz de producir ondas cuadradas en la banda aproximada de 30..120 Kilociclos y simultánea o separadamente ruido pseudoaleatorio. El volumen de cada canal puede fijarse a uno de dieciseis niveles (incluyendo el "silencio"). Los comandos disponibles permiten producir simples sonidos de volumen y tono constante, y ruido en las mismas condiciones; y además permiten generar sonidos más complicados en que el volumen y/o el tono varía mientras se produce el sonido, bajo el control de las correspondientes envolventes.

El circuito de generación de sonido trabaja dándole valores no de frecuencias, sino de períodos. Por lo tanto, para fijar un tono se expresa como "período del tono", siendo un periodo P el que produce un tono de frecuencia F, tal que:

$$F = 125000/P$$

La envolvente de tono permite que el periodo varíe durante la generación de una nota.

El generador de ruido produce un tono cuya frecuencia varía aleatoriamente alrededor de un valor dado. La especificación del ruido también se expresa como un periodo, con la frecuencia y periodo manteniendo la misma relación que para la frecuencia y periodo de la nota musical.

La duración de una nota puede fijarse explícitamente, o puede estar gobernada por la longitud de una "envolvente de volumen".

También puede usarse una envolvente de volumen para variar el volumen de una nota. Cada envolvente puede tener hasta cinco secciones, siendo cada sección un cambio absoluto o relativo en el volumen, o excepcionalmente, una fijación de envolvente estándar en el circuito:

Una sección incremental se especifica como un número de pasos, con un cierto tamaño de paso y una pausa después de cada paso.

Una sección absoluta se especifica como un nuevo nivel de volumen y una pausa antes de la siguiente sección.

La fijación de envolvente por el propio circuito puede usarse para aprovechar la ventaja de las características inherentes para envolventes que posee la "pastilla", cuando sea aconsejable.

Puede usarse una "envolvente de tono" para variar el período de tono de una nota musical. Cada envolvente puede tener hasta cinco secciones, siendo cada sección un cambio incremental o absoluto al período de tono:

Una sección incremental se especifica como un cierto número de pasos, con un tamaño para cada paso y una pausa después de cada paso.

Una sección absoluta se especifica como un nuevo período de tono y una pausa antes de la siguiente sección.

Cada canal puede tener una nota emitiéndose -activa- y hasta cuatro notas en secuencia, esperando ser procesadas. Cuando termina la nota activa, el sistema intenta ejecutar el siguiente sonido que haya en la secuencia de notas del canal en cuestión. La nota comenzará a emitirse a no ser que esté Retenida, o esperando un acorde:

Cuando una nota está Retenida, permanece esperando a ser ejecutada, y el canal no produce ningún sonido, hasta que dicha nota sea Liberada.

El mecanismo de Retención/Liberación permite que las notas de la secuencia sean "marcadas", estipulando que la primera nota de cada canal está Retenida y añadiendo más notas a las secuencias, y luego Liberando los canales.

El mecanismo de Acordes o "Rendezvous", puede usarse para forzar que las notas emi-

tidas por dos o por todos los canales, comiencen conjuntamente.

Cuando se coloca en la secuencia a emitir por un canal una nota dada, puede establecerse que debe ser Acorde con otra nota en uno o en ambos de los otros canales. Cuando comienza a ejecutarse la nota, será retardada hasta que los otros canales tengan notas preparadas para ser ejecutadas, y que tengan estipulado el recíproco Acorde.

Puede emitirse una nota con la marca de Oclusión puesta. Eso hace que se termine inmediatamente cualquier sonido activo y que se desechen los que hubiera en la secuencia de espera, dejándola vacía y el canal inactivo, u "ocluído" y dispuesto para un nuevo sonido.

El estado de la secuencia de notas, contenidas en un canal dado, puede ser comprobado. El sistema informará de cuantas notas es posible enviar hacia ese canal, si una nota es activa o está sonando, si el canal está Retenido o si está esperando un Acorde.

Para facilitar la generación de sonidos "de fondo" puede hacerse que invoque una subrutina determinada del programa en BASIC cuando no tiene completa la secuencia de notas a emitir.

La generación de sonido continúa después de la instrucción **END** o del fin del programa, hasta que deja vacía toda la secuencia de sonidos pendientes, pero las posibles interrupciones para añadir sonidos quedan impedidas.

La suspensión o detención temporal de un programa mientras hay notas activas, provoca que se suspendan también la emisión de sonidos. Cuando se reanuda la ejecución del programa, los sonidos también son reanudados a partir de donde se quedaron. Si se emite un sonido mientras se está en el Modo Directo, también se reanudará cualquier sonido que estuviera en espera de ser emitido.

Resumen de comandos de sonido y funciones:

- | | |
|----------------|--|
| END | Fija la envolvente de tono. |
| ENV | Fija la envolvente de volumen. |
| ON SQ | Establece la subrutina para interrupciones en la secuencia de notas. |
| RELEASE | Libera la emisión de notas retenidas por el canal dado. |
| SOUND | Emite una nota hasta la secuencia del canal dado. |
| SQ | El estado de la secuencia de notas en el canal dado en la función. |

INTERRUPCIONES EN BASIC

Los sucesos que pueden causar interrupciones o desvíos en el programa son los siguientes:

Un grupo de cuatro temporizadores o cronómetros.

Los tres canales de sonido, con sus secuencias incompletas.

Una ruptura en la ejecución (**ESC/ESC**).

Si hay más de un suceso pendiente de ser atendido, el que tenga mayor prioridad determina la subrutina a ejecutar. Las prioridades en orden descendente, son las siguientes:

Ruptura o "corte" en la ejecución (**ESC/ESC**)

Temporizador 3

Temporizador 2 y los tres canales de sonido (todos de igual prioridad)

Temporizador 1

Temporizador 0

Dado que una interrupción puede suceder en cualquier momento entre las ejecuciones de las instrucciones, el programador debe garantizar cuidadosamente que la subrutina de interrupción no ve afectada la "tarea" que realiza por quedar las variables en estados intermedios.

El mecanismo de prioridades permite un método para "exclusiones mutuas". También es posible impedir explícitamente todas las posibles interrupciones (excepto la de corte o ruptura), y permitir las de nuevo, mediante las instrucciones **DI Y EI**.

Los temporizadores

Los cuatro "medidores de lapsos de tiempo" son independientes y tienen prioridades diferentes. Los "cronómetros" cuentan hacia atrás una cierta cantidad de 50-avos (1/50) de segundo, y luego provocan la interrupción. Es posible especificar que el temporizador interrumpa sólo una vez, o que vuelva a reiniciar la cuenta después de cada interrupción, dando así una repetición regular de interrupciones, y consiguiendo por tanto la ejecución de una cierta tarea cada cierto tiempo.

Las interrupciones debidas a los canales de sonido

Los tres canales de sonido provocan interrupciones independientes, pero todos tienen la misma prioridad. Una vez que se ha "citado" —puesto en marca— una subrutina para tratar las interrupciones por sonidos, no se verá por tanto interrumpida por la suscitada en otro canal de sonido —de manera que las subrutinas para interrupciones de sonido pueden compartir las mismas variables.

Cuando está permitida la interrupción por un canal de sonido, provocará dicha interrupción cuando la secuencia de notas para ese canal no está completa; y cuando lo está provocará la interrupción inmediatamente que arranque la siguiente nota y haya por tanto, sitio para una nota más en la secuencia.

La interrupción por "corte" (ESC/ESC)

Esta interrupción, cuando está explícitamente permitida, se provoca al pulsar el par de teclas (ESC/ESC) que por otro lado detiene la ejecución del programa. Esta posibilidad puede usarse para evitar que se detenga por completo la ejecución de un programa mediante ESC/ESC, o para efectuar alguna tarea de limpieza antes de detener la ejecución, o para implementar alguna característica especial.

Resumen de comandos y funciones para interrupciones

AFTER	Fija el intervalo de interrupción para el temporizador dado.
DI	Desactiva (impide) interrupciones (excepto BREAK o "corte").
EI	Permite o "faculta" interrupciones.
EVERY	Fija temporizador para que interrumpa "cada" cierto tiempo.
ON BREAK GOSUB	Señala la subrutina para interrupciones por "corte".
ON SQ GOSUB	Señala la subrutina para interrupciones por secuencia de sonidos.
REMAIN	El tiempo que "resta" o queda en el temporizador dado.

PANORAMA DE COMANDOS

Creación de programas

AUTO, DELETE, EDIT, LIST, MERGE, NEW, RENUM, SAVE

Los comandos de este grupo son utilizados cuando se está preparando un nuevo programa o se está enmendando el existente en memoria.

Carga y Ejecución de Programas

CLEAR, CHAIN, CHAIN MERGE, LOAD, MEMORY, HIMEN, RUN, FRE

Finalización de ejecución

END

Estructuras de Iteración y Selección

FOR, GOSUB, GOTO, IF, ON x GOSUB, ON x GOTO, RETURN, WHILE

El BASIC admite dos formas de bucles: bucles preconfiados o **FOR**, y bucles condicionados o **WHILE**.

Designación de variables

DEFINT, DEFREAL, DEFSTR, DIM, ERASE

Atributos de pantalla

MODE, BORDER, INK, SPEED, INK

Salida de Texto

PRINT, WRITE, ZONE, POS, VPOS

Cauces de envío a pantalla

CLS, LOCATE, PAPER, PEN, TAG, TAGOFF, WINDOW, WINDOW SWAP

El cauce hacia el cassette

CLOSEOUT, OPENOUT

Introducción de texto

INPUT, LINE INPUT

Introducción de datos por cassette

CLOSEIN, EOF, OPENIN

El Cassette en General

CAT, SPEED WRITE

Series de Constantes

DATA, READ, RESTORE

Trazado de Gráficos

CLG, DRAW, DRAWR, MOVE, MOVER, ORIGIN, PLOT, PLOTR, TAG, TAGOFF, TEST, TESTR, XPOS, YPOS

Generación de Sonidos

ENT, ENV, ON SQ, RELEASE, SOUND, SQ

Interrupciones por Temporizador

AFTER, EVERY, REMAIN

Admisión e Inhibición de las Interrupciones

EI, DI

Teclado y Joystick

KEY, KEY DEF, INKEY\$, JOY, SPEED KEY

Caracteres definidos por el Usuario

SYMBOL, SYMBOL AFTER

Funciones Aritméticas

ATS, ATN, COS, DEG, EXP, FIX, INT, LOG, LOGIO, MAX, MIN, PI, RAD, RANDOMIZE, RND, ROUND, SGN, SIN, SQR, TAN

Funciones Literales (String)

BIN\$, DEC\$, HEX\$, INSTR, LEFT\$, LEN, LOWER\$, MID\$, RIGHT\$, SPACES, STR\$, STRING\$, UPPER\$, VAL

Funciones para Conversión de Clases

ASC, CHR\$, CINT, CREAL, UNT

Operaciones a Nivel de Máquina

CALL, INP, OUT, PEEK, POKE, WAIT

Cepos para Atrapar Errores

ERL, ERR, ERROR, ON ERROR GOTO, RESUME, ON BREAK

Desarrollo del Programa

CONT, STOP, TRON, TROFF

LOS COMANDOS Y LAS FUNCIONES INTRINSECAS

<expresión direccional>

Una <expresión numérica> que cuando redondeada a entero produce un valor en la banda $-32768..65535$. Los valores negativos pueden convertirse luego a su equivalente "sin-signo" simplemente añadiéndola 65536.

<nombre variable múltiple>

El nombre de una tabla que puede incluir su "marca de clase".

<color>

Una <expresión entera> que produce un valor en la banda $0..26$, que especifica uno de los 27 colores posibles (véase Apéndice VI).

<expresión>

Cualquier serie de operadores y operandos permitidos que produce un valor de cualquier clase.

<tinta>

Una <expresión entera> que produce un valor en la banda $0..15$, que especifica una de las 16 posibles tintas de la "paleta".

<expresión entera>

Una <expresión numérica> que cuando se redondea a entero, produce un valor en la banda $-32768..32767$.

<número de línea>

Un número decimal en la banda $1..65535$.

<banda de números de línea>

Una <banda de números de línea> hace referencia a todas aquellas líneas cuyos números están dentro de las "cotas" que definen la banda. Puede tomar una de las siguientes formas:

-
- <número de línea> Sólo hay una línea dentro de la banda.
 - <número de línea> Las líneas desde la cota inferior dada hasta el final del programa.
 - <número de línea> Las líneas desde el comienzo del programa hasta la cota superior dada.
 - <número de línea> Las líneas cuyos números son mayor o igual que la cota inferior dada y menor o igual que la cota superior dada.

<expresión lógica>

El BASIC no admite una clase separada de datos booleanos. El valor entero 0 es considerado siempre como falso, y cualquier otro valor entero como cierto. Una <expresión lógica> es por lo tanto, una expresión entera simplemente conocida con nombre.

<tinta reducida>

Una <expresión entera> que produzca un valor en la banda 0..15, que especifica una de las 16 tintas posibles simultáneamente. El valor es a continuación "cribado" para forzarlo en la banda que corresponde al modo corriente de pantalla; por lo tanto:

Modo 0: Criba = 15, banda final = 0..15

Modo 1: Criba = 3, banda final = 0..3

Modo 2: Criba = 1, banda final = 0..1

<expresión numérica>

Una serie de operadores y operandos que al evaluarse produce un valor Real o Entero, incluyendo expresiones que previamente entraban en las categorías denominadas relacional y logical.

<variable numérica>

El nombre que sirve de referencia a una variable, que puede ser un elemento de una tabla, pero cuya clase sea Real o Entera.

<literal entrecomillado>

Un <literal entrecomillado> es una cadena de caracteres que contiene entre 0 y 255 caracteres, y que está delimitada por el signo comillas al principio y por el signo comillas al final de la misma; o bien puede empezar con comillas y estar terminada por la marca de "retorno de carro".

<variable simple>

El nombre que sirve de referencia para una variable pero excluyendo las variables múltiples o tablas. La referencia puede incluir una "marca de clase".

<expresión de cauce>

Una <expresión entera> que produce un valor en la banda 0..9. Toda la entrada y salida de datos se dirigen a su destino a través de dicho cauce.

Los cauces de salida de información son:

- | | |
|------|---|
| 0..7 | Los ocho cauces posibles hacia la pantalla del monitor. |
| 8 | La Impresora. |
| 9 | Los ficheros de salida en Cassette. |

Los cauces de entrada de información son:

- | | |
|------|---|
| 0..7 | El Teclado, con cualquier "aviso" enviado al cauce de pantalla correspondiente. |
| 8 | Teclado, con cualquier aviso enviado por el cauce de impresora. |
| 9 | El fichero de entrada en Cassette, con cualquier "aviso" ignorado. |

<expresión literal>

Una serie de operadores y operandos que produce un valor de índole o clase literal.

<variable literal>

El nombre que sirve de referencia para una variable, que puede ser un elemento de una tabla, cuya clase o índole es literal (string).

<marca de clase>

Uno de los caracteres siguientes:

% indica numeral Entero

! indica numeral Real

\$ indica Literal

<variable>

Un nombre que sirve de referencia a cualquier clase de variable. Incluye variables que son elementos de una tabla, en cuyo caso deberán incluirse al final del nombre los subíndices apropiados.

<nombre de variable>

El nombre de una variable simple, que puede incluir su marca de clase.

SCHNEIDER BASIC Manual de Referencia

1. INTRODUCCION

Los comandos se presentan al BASIC en líneas. Una línea puede constar de varios comandos, separados por dos-puntos, y con la única limitación en la longitud total de la línea. En el modo Directo las líneas se introducen mediante el teclado. En el modo Programa las líneas las lee del programa que en ese momento tenga depositado en memoria.

En el Modo Directo es posible añadir y suprimir líneas del programa en curso y enmendar las líneas existentes.

1. 1 Introducción de Líneas

El BASIC acepta líneas procedentes de la consola con un máximo de 255 caracteres, y terminadas por un retorno de carro.

1. 2 Metalenguaje

Con el fin de describir los comandos y sus parámetros, se utiliza un metalenguaje simple.

Un elemento conceptual viene representado por su nombre encerrado entre corchetes angulados.

(expresión numérica)

Todo lo que no está encerrado entre corchetes angulados se requiere tal y como se escribe.

Parte opcional:

((expresión numérica))

Si una parte opcional de un comando puede repetirse (de forma que aparezca tantas veces como se quiera, o ninguna en absoluto), se incluye un asterisco después del corchete cuadrado de cierre.

(dígito) ((dígito))*

En muchos comandos se usa una lista de parámetros con los elementos separados por comas. Se utiliza para describir una forma abreviada que se comprende mejor mediante un ejemplo. Así:

(Lista de: (expresión) es igual que (expresión) ,(expresión)*

o bien:

(lista de: (//)(número) es igual que (//) (número) (;//)(número))*

Si la lista contiene más de un elemento, cada elemento adicional debe estar precedido por una coma.

2. LOS ELEMENTOS DE BASIC

2. 1 Repertorio de caracteres

El BASIC está utilizando el repertorio de caracteres ASCII, en la banda 0 a 255 (Carácter = Octeto).

Los caracteres con valores inferiores a 32 (and 20, espacio en blanco) se tratan como caracteres no-visivos, y varios de ellos tienen un significado especial para el BASIC. Los caracteres con valores superiores a 126 (and 7E, tilde) generalmente se ignoran.

2. 2 Nombres

Las Variables y las palabras Clave tienen (nombre)s. El primer carácter de un (nombre) debe ser alfabético —una letra— y los demás caracteres pueden ser alfabéticos, numéricos o el punto. Pueden tener como máximo 40 caracteres, y todos ellos son significativos.

Las variables no pueden tener el mismo (nombre) que las palabras clave del lenguaje.

2. 3 Números

Un (número) puede ser cualquiera de lo siguiente:

A) Puede adoptar cualquiera de las siguientes formas:

(dígitos)
(dígitos).((dígitos))
((dígitos)). (dígitos)

siendo (dígitos) uno o una serie de dígitos decimales.

B) Adopta la forma:

(número aritmético)E((signo))((dígitos))

(signo) el signo más o el menos, (dígitos) uno o más dígitos decimales. Se llama exponente a la parte que comprende el (signo) y los (dígitos) que le siguen. El valor de tales números es el valor de (número aritmético) multiplicado por el resultado de elevar 10 al exponente dado.

C) Puede tener cualquiera de las siguientes formas:

and (cifras hexadecimales)

and H (cifras hexadecimales)

and X (cifras binarias)

(cifras binarias) los símbolos empleados en el sistema de base 2 (el 0 ó el 1); (cifras hexadecimales) los símbolos empleados en el sistema de base 16 (los dígitos 0..9 y las letras A..F, o bien a..f). El equivalente en el sistema de base de 10 de todo (número sistemado). (Todo (número sistemado) se trata como ENTERO).

D) Adopta la forma:

(dígitos)

siendo (dígitos) una serie de dígitos decimales con 1, al menos. Los números de línea están limitados a la banda 1..65535.

2. 4 Constantes literales (Strings)

Una (constante literal) es una serie discrecional de caracteres encerrada entre comillas; o que comienza con comillas y se termina por "final de línea". Puede tener 255 caracteres como máximo. (Se suelen llamar cadenas de caracteres).

2. 5 Clase de datos

El BASIC opera sobre dos amplias clases de datos, numéricos y alfanuméricos, (o literales).

Los literales (strings) son series de caracteres que los números pueden ser de una de dos clases: Enteros o Reales.

Enteros ocupan en memoria dos Bytes y están representados en el sistema de complemento

a doses, de forma que pueden tener valores en la banda $-32768..+32767$, con signo, en caso contrario en la banda $0..65535$

Los datos numéricos Reales ocupan en memoria cinco bytes y corresponden al sistema de coma binaria flotante, con cuatro bytes para la mantisa y un byte para el exponente. El valor absoluto máximo es $1.7E+38$; y el más pequeño es $2.9E-39$. La mantisa de cuatro bytes permite una precisión ligeramente superior a nueve cifras.

Un (número sistemado) es siempre Entero.

Un (número aritmético) se trata como entero si no existe punto decimal, y si su valor está incluido dentro de la banda de los Enteros. En caso contrario dicho número se trata como Real.

Todo (número científico) se trata como Real.

2. 6 Variables

Las Variables tienen tres atributos: nombre de la variable, clase de la variable, y organización. Un (nombre variable) tiene la forma (nombre)((designador clase)). Todo (designador clase) es uno de los siguientes:

% para numerales Enteros

! para numerales Reales

\$ para Literales

La clase prescrita para omisiones depende de la inicial, o primer carácter del (nombre variable), y puede estipularse dinámicamente mediante los comandos **DEFINT, DEFREAL Y DEFSIR**.

Las variables pueden ser en cuanto a organización: (variable simple) que corresponde a un solo dato, o (variable múltiple) que es una colección de variables que tienen todas exactamente la misma clase. Se suelen llamar también tablas.

Las variables el espacio que ocupan en memoria queda reservado en cuanto aparece el (nombre) por primera vez, y con un valor de cero o nulo.

2. 7 Tablas

Una tabla es una colección de variables de la misma clase, en la que puede señalarse cada elemento mediante el nombre colectivo de la tabla seguido del adecuado número de sufijos, en la forma:

(nombre variable)((sufijos))

(sufijos), también llamados subíndices, una (lista de: (expresión entera) y cada (expresión entera) da como resultado un valor dentro de la banda correcta.

La clase de variables que componen la tabla está determinada por la clase del (nombre variable).

Las tablas pueden declararse explícitamente mediante el comando **DIM**, o implícitamente utilizando uno de sus elementos. Cuando se declara una tabla, se especifica el número de dimensiones y la cota superior de cada dimensión (en una declaración implícita sólo hay una dimensión y la cota superior es 10). Cualquier referencia a un elemento de la tabla debe usar el mismo número de subíndices. No es posible cambiar las dimensiones de una tabla, si previamente no se ha dejado **LIBRE** el espacio que ocupa mediante el comando **ERASE**.

La cota inferior de todos los subíndices de una tabla es 0.

2. 8 Compatibilidad y Conversión de Clases

Cuando opera sobre numerales de diferentes clases, el BASIC los convertirá automáticamente a la clase que proceda. Los numerales se "amplian" para adaptarse a la "mayor" clase involucrada, de manera que los Enteros se amplian a Reales. Al convertir Reales a Enteros se redondea el número a un Entero, y el valor resultante debe pertenecer a la banda de los Enteros, $-32768..+32767$, o se generará un Error 6 (Overflow = rebase).

2. 8. 1. Redondeo

- A) Truncamiento, o redondeo hacia cero.
Los dígitos sobrantes simplemente se desprecian:

**positivo X X1,
negativo X X2,**

- B) Redondeo hacia-Infinito

**X2 X X1
X X1**

- C) Redondeo hacia + Infinito

X X2

D) Redondeo al más próximo

X X1 ó X2, dependiendo de cuál está más cerca de X.

2. 9 Entero sin-signo

En algunas circunstancias el BASIC requiere un Entero en la banda completa sin-signo de 16 bits, que es de 0..65535; por ejemplo, una dirección de memoria mediante el comando PEEK.

Los números en coma flotante se redondean a enteros, que han de estar en la banda -32768..+65535. El equivalente sin-signo de los números negativos en notación de complemento a doses es la que se usa en esos casos; es decir, los números negativos se tratan como 65536+número.

2. 10 Expresiones

Hay cuatro clases de expresiones: numéricas, literales, relacionales y lógicas. Cuando el orden de evaluación no está forzado por las reglas prescritas de prioridad de operadores, o por paréntesis, la evaluación procede de izquierda a derecha.

2. 10. 1 Expresiones Numéricas

La sintaxis de las expresiones numéricas es:

(expresión numérica) es: (numeral)((operador)(numeral))*
(numeral) es: (menos monádico)numeral
o bien: (constante: numérica)
o bien: (variable numérica)
o bien: (función numérica)
o bien: ((expresión numérica))
o bien: ((expresión relacional))

(variable numérica) una referencia a una variable simple o a un elemento de una tabla de variables numéricas. (Función numérica) es cualquiera de las que en el sistema, o definida por el usuario, da como resultado de la evaluación un valor numérico.

Los (operador) es, en orden de precedencia son:

Grado 1 ^a	Exponenciación	Eleva el valor a la izquierda al valor especificado a su derecha.
Grado 2 ^a	Signo monádico	El signo menos.
Grado 3 ^a	* Multiplicar	Multiplicación Real o Entera.
	/ Dividir	División Real. Si algún valor es Entero se fuerza a ser Real antes de la operación.
Grado 4 ^a	División entera	Fuerza ambos valores a Enteros. El resultado se trunca a Entero.
Grado 5 ^a	MOD Resto División (módulo)	Fuerza ambos valores a Enteros. El resultado es el resto después de División Entera.
Grado 6 ^a	+ Adición	Suma Real o Entera.
	- Sustracción	Resta Real o Entera.

La división por cero genera un Error 11. La división entera por cero provoca el cese de la ejecución del programa; pero la división real por cero no termina dicha ejecución, ya que el cero puede ser el resultado de un "rebase por defecto" en una operación anterior. En lugar de ello, se genera el mensaje "División by zero", y se entrega como valor el máximo número representable (aprox. 1.7E+38) con el signo apropiado, y el programa continúa. En el caso de un "rebase por exceso" cuando los operandos son Enteros, los valores son ampliados a Reales y se repita la operación.

2. 10. 2 Expresiones Literales (Strings)

La sintaxis de las expresiones literal es:

```
(expresión literal) es: (literal) ((+(literal))*
(literal) es: (variable literal)
o bien: (constante literal)
o bien: (función literal)
o bien: ((expresión literal))
```

(Variable literal) una referencia a una variable simple o a un elemento de una tabla de variables literales. (Función literal) es cualquiera de las del sistema, o definida por el usuario.

El efecto del signo de operación + sobre los literales es concatenarlos, o empalmarlos. Se produce un nuevo literal construido por el primer operando literal inmediatamente seguido del segundo operando. Si el literal resultante fuera mayor de 255 caracteres, se genera el Error 15.

2. 10. 3 Expresiones Relacionales

Las expresiones relacionales comparan dos numerales o dos literales. Por lo tanto:

(expresión relacional)

es: (expresión numérica) (operador relación)(expresión numérica)

o bien: (expresión literal)(operador relación)(expresión literal)

El (operador relación) es uno de los siguientes:

Menor que
== Menor que o Igual
= Igual
== Mayor que o Igual
Mayor que
Distinto de (no igual a)

Cuando los argumentos son literales, el significado de las relaciones requiere alguna explicación adicional:

Dos literales son iguales cuando son de la misma longitud, y los caracteres homólogos son los mismos.

Un literal es menor que otro si:

son iguales hasta el final del primero y el segundo es más largo. o bien: el primer carácter que sea distinto en los dos, es más pequeño (y se considera más pequeño cuando el valor asociado al carácter —el código— es de menor valor).

2. 10. 4 Expresiones Lógicas

Toda (expresión Lógica) es de la forma:

(operando)((operador lógico)(operando))*

(operando) es: **NOT** (operando)

o bien: (expresión numérica)

o bien: (expresión relacional)

o bien: ((expresión lógica))

Los operandos sobre los que trabajan los operadores lógicos son forzados a Enteros. Luego se efectúa la operación indicada para cada uno de los 16 bits que representan dichos Enteros usando el sistema de complemento a dos.

El operador monádico **NOT** invierte simplemente cada bit del operando (0 pasa a 1, y viceversa; es la operación lógica "negación").

Los operadores diádicos, en orden de prioridad y su efecto sobre cada bit es el siguiente:

AND El resultado es 0 a no ser que ambos bits operandos sean 1.

OR El resultado es 1 a no ser que ambos bits operandos sean 0.

XOR El resultado es 1 a no ser que ambos bits operandos sean iguales.

El resultado de toda (expresión relacional) es el valor -1 o el valor 0. La representación para -1 , la "certeza", es en el equivalente binario todos los bits a 1; la representación para 0, la "falsedad" es un equivalente binario con todos los bits a 0. El resultado de toda (expresión lógica) es uno de esos dos valores.

2. 11 Funciones

Las funciones en BASIC son subrutinas internas del programa interpretador que "absorben" un determinado número de argumentos que son elaborados según un determinado método, y que entregan un valor que se denomina resultado de la función.

Hay dos clases de funciones en BASIC; las intrínsecas del lenguaje y las "definidas" por el usuario en el programa.

2. 12 Funciones intrínsecas

La clase o índole del resultado no viene necesariamente dictado por el nombre que tenga la función.

2. 13 Funciones de usuario

El usuario puede definir sus propias funciones mediante el comando **DEFFN**. Este comando adscribe o asocia un nombre con una lista de argumentos formales y con una expresión numérica o literal.

El nombre que identifica la función tiene la forma **FN** (nombre-designador-clase) en donde las siglas **FN** sirven solo para distinguir el nombre de las funciones de los nombres de las variables; pero no interviene en la cuenta de los cuarenta posibles caracteres de todo "nombre"

La lista de argumentos formales enuncia un cierto número de variables que son "locales" de esa función. Cuando se "invoca" esa función debe hacerse dándole argumentos actuales que deben corresponder en lugar y en clase con los argumentos formales descritos en la definición.

El cuerpo de la función, la fórmula, es una expresión simple. No es posible tener ninguna forma de estructura de control dentro de una función. Dicha expresión puede incluir referencias a otras funciones y a variables globales, así como a los argumentos formales de la función.

Cuando se evalúa una función, el resultado de la expresión que la define se convierte a un dato de la misma clase que el designado por el nombre de la función, y se entrega como resultado dicho valor convertido.

2. 14 Fracciones Decimales y Coma flotante y Binaria

Algunas fracciones decimales tienen una representación exacta como números binarios en coma flotante. Los números en coma flotante se redondean antes de ser exhibidos en pantalla. A menos que se eviten totalmente las fracciones decimales, multiplicando todos los números por las potencias adecuadas de 10 (tratando todos los valores monetarios como céntimos por ej.) es imposible evitar esos defectos. Sin embargo pueden reducirse estipulando explícitamente la función intrínseca **ROUND**, que redondea números que en pantalla parecen iguales a exactamente el mismo valor.

2. 15 Comandos Externos

Es posible ceder la ejecución de algunos comandos a otros programas interpretadores de la máquina. Estos comandos se distinguirán de los comandos **BASIC**, prefijando el nombre con una barra vertical (|). El **BASIC** encuentra un comando externo, primero buscar el programa externo al que reclamar la ejecución. Si el comando es aceptado el **BASIC** interpreta el resto de dicho comando externo con expresiones separadas por comas. Los nombres de los comandos externos adoptan la forma de un nombre en **BASIC**. Los comandos externos pueden ser de dos clases: de retaguardia (vuelve a entregar el control al **BASIC** una vez que ha terminado su tarea, su propósito es permitir el acceso a tales dispositivos extra como "lápices luminosos", unidades de disco, impresoras inteligentes y demás.

De vanguardia (no devuelve el control al BASIC, sino que se apodera completamente de la máquina (puede ser un juego auto-contenido y otro interpretador de lenguaje tal como Forth).

NOTA: los términos ingleses son BACKGROUND y FOREGROUND para designar lo aquí llamado de retaguardia y vanguardia respectivamente.

3 MODOS DIRECTO Y PROGRAMA

En Modo Directo, el BASIC toma las líneas tecladas en la consola y ejecuta los comandos cuando se pulsa retorno de carro que termina el comando. En Modo Programa, el BASIC toma las líneas del programa actualmente depositado en memoria y ejecuta los comandos que hay en cada una de las líneas.

3. 1 Modo Directo

Puede subdividirse en:

Comandos Directos: se introducen e inmediatamente son ejecutados.

Introducción de Programas: se introducen nuevas o sustitutivas líneas de programas o se quitan algunas existentes.

Enmienda de un Programa: se alteran o modifican líneas de programa existentes.

Es posible usar variables en el Modo Comandos Directos. Ciertos comandos hacen que todas las variables se anulen, y entre ellos se incluyen los comandos **RUN-NEW-LOAD**.

En Introducción de Programas puede hacerse simplemente tecleando una línea que comience con un número dado. Cuando se termina la línea queda incluida en el programa actual, sustituyendo cualquier línea que en él exista con el mismo número.

El comando **AUTO** hace que el BASIC ayude a la introducción de programas generando los números de línea automáticamente.

En Enmienda de un Programa está apoyado por el comando **EDIT**.

3. 2 Modo Programa

En el modo de Programa, el BASIC automáticamente recorre paso a paso las líneas del programa presente en memoria, obedeciendo los comandos que allí encuentra. Las líneas se

toman según el orden de números de línea, excepto cuando de manera explícita, los comandos obligan a hacer otra cosa. El comando **END** señala el punto esperado de finalización de un programa. Se considera que hay un comando **END** implícito, después de la última línea del programa. Las variables persisten con sus valores cuando el BASIC vuelve del Modo Programa al Modo Directo, lo que es muy utilizado al depurar un programa.

El modo Tratamiento de Error es un sub-modo del Modo Programa, y el BASIC entra en dicho sub-modo cuando ocurre un error y además se ha dado un comando —**ON ERROR GOTO nn**—

Pulsando **ESC**: el BASIC volverá al Modo Directo con un mensaje de interrupción (**BREAK**) puede hacerse que siga mediante el comando **CONT**, como si no hubiera sucedido ninguna interrupción en la ejecución.

Pulsando **SPACE**: El BASIC reanuda la ejecución del programa actual. El propio carácter espacio se descarta.

RESUMEN DE COMANDOS

AFTER

Suscita una subrutina después de que ha transcurrido un periodo dado.

AFTER <lapso de tiempo> [<número temporizador>]GOSUB <número línea>

El /lapso de tiempo/ es una /expresión entera/ que especifica cuánto tiempo ha de transcurrir antes de que la subrutina sea llamada. Este tiempo se mide en 1/50-avos de segundo.

El /número temporizador/ es una /expresión entera/ que especifica cuál de los cuatro posibles temporizadores de demora va a usarse.

La expresión debe producir un valor en la banda 0..3. Si se omite la expresión, se supone el 0.

AUTO

Numeración automática en líneas.

AUTO [/número de línea/][,/incremento/]

Si se omite el /número de línea/, se supone 10.

El /incremento/adopta la forma de un /número de línea/. Si se omite el /incremento/, se supone 10.

BORDER

Fija el color del área que bordea la imagen.

BORDER/color/[,/color/]

Si se especifican dos colores, el "borde" alterna entre los dos.

CALL

Cede el control a una subrutina externa.

CALL/expresión direccional[/,/lista de:/parámetro/]

La/expresión direccional/indica la dirección de comienzo de la subrutina.

Los /parámetro/s puede ser bien una /expresión numérica/ o bien: /variable/.

CAT

Muestra el catálogo de los ficheros grabados en una cinta.

CAT El "banderín,, (el testigo) indica la clase de fichero (en este caso hemos puesto una interrogación); y puede ser uno de los siguientes caracteres:

- \$ un fichero con un programa en BASIC.
- % un fichero con un programa "protegido" en BASIC.
- * un fichero con texto en caracteres ASCII.
- un fichero binario (contenido de una zona de memoria).
- un fichero binario "protegido".

CHAIN

CHAIN MERGE

Encadenar un programa en cinta al programa actual.

Permite que desde el programa actual se "traiga" de la cinta y se cargue en memoria otro programa, reteniendo las variables vigentes y parte o todo el programa ya existente en la memoria.

CHAIN/nombre fichero[/,/expresión de número de línea/]
CHAIN MERGE/nombre fichero[/,/expresión de número de línea/]
[,DELETE/gama de número de línea]]

El/nombre de fichero/una/expresión literal/ que indica el nombre del fichero donde está contenido el programa que va a ser “engarzado” al existente en memoria.

La/expresión de número de línea/es una expresión numérica/, que cuando se redondea a Entero produce un valor en la banda —32768..65535.

La opción **DELETE** hace que se supriman del programa corriente las líneas comprendidas en la gama dada, antes de que sea traído de la cinta y cargado en memoria el nuevo programa.

Ambas forman del comando **CHAIN** y **CHAIN MERGE**, tienen los siguientes efectos sobre la situación actual:

- Se olvidan todas las Funciones de Usuario. — Se desactivan los cepos **ON ERROR GOTO**.
- Todos los ficheros abiertos son abandonados (y se pierde cualquier dato en el “buffer” de salida.
- Se efectúa la acción de **RESTORE**.
- Se olvidan todos los comandos **FOR**, **WHILE** y **GOSUB** que hubiera pendientes.

Durante la operación **CHAIN MERGE**, en “ensambla” al programa actual el nuevo procedente del cassette, cualquier línea en el programa nuevo con el mismo número que el de una línea del programa existente, sustituirá a la línea del existente.

No es posible hacer la operación **CHAIN MERGE** con un programa protegido en el cassette.

CLEAR

Anula todas las variables y ficheros.

Para hacer que todas las variables sean cero si son numéricas, o “vacías” si son variables literales.

CLEAR.

CLG

Limpia la pantalla de gráficos.

GLC[/cinta cribada/]

CLOSEIN

Cierra los ficheros de entrada desde cassette.

CLOSEIN

CLOSEOUT

Cierra un fichero de salida hasta el cassette.

CLOSEOUT

CLS

Limpia una ventana de texto.

CLS[#/expresión de cauce/]

La /expresión de cauce/ debe producir un valor entero en la banda 0..7, que corresponde al cauce de envío de texto hacia la pantalla. Si se omite el sistema adopta el cauce #0.

CONT

Seguir la ejecución, continuar después de **(ESC)(ESC)STOP** o bien **END**.

CONT

DATA

Declara datos constantes a usar en un programa.

DATA/lista de:/constante/

/constante numérica/puede estar rodeada de un /espacio blanco/, que será ignorado.

constante literal entrecomillada/ puede estar precedida y seguida de /espacio blanco/que será ignorado. Las comillas posteriores pueden omitirse si el literal es el último elemento de la lista.

/constante literal sin entrecomillar/ es una serie de caracteres, entre 0 y 255, que son tomados como un solo elemento. Una coma, un dos-puntos o un "retorno de carro" termina un literal sin comillas, dentro de este contexto. Una /constante literal sin entrecomillar/ puede estar precedida y seguida de /espacio blanco/ que será ignorado.

DEF FN

Notificar una Función de Usuario.

El BASIC permite que el programa se definan y usen funciones, es la parte que corresponde a la definición.

DEF FN /nombre función/[(/argumentos formales/)] = /expresión/.

El /nombre función/ tiene la misma forma que un /nombre variable/, siendo el nombre completo de la función el **FN** /nombre función/.

/argumentos formales/ adoptan la forma de / lista de:/nombre variable simple/

/expresión/ puede incluir no solo los argumentos formales de la función, sino también cualquier otra constante, variable o función.

DEFINT, DEFREAL, DEFSTR

Prescribir la clase para omisiones de los nombres.

Todos los nombres de variables y de funciones tienen asociada una cierta clase con ellos. Esta clase puede estar explícita o implícitamente designada en el nombre, dependiendo de

lo que se haya prescrito para los nombres mediante los comandos de esta sección.

DEFINT /lista de: /gama de letras

DEFREAL /lista de: /gama de letras/

DEFSTR /lista de: /gama de letras

Siendo /gama de letras/ bien /letra/o bien /letra/letra- /letra y con la anotación /letra/-/letra/ se define una gama de letras inclusiva.

DEFINIT establece para omisiones que sean Enteras.

DEFREAL establece para omisiones que sean Reales.

DEFSTR establece para omisiones que sean Literales.

DEG

Establece el modo grados para las funciones trigonométricas.

DEG.

DELETE

Suprime líneas de programa.

DELETE /banda de números de línea/.

DI

Inhibe las posibles interrupciones.

Impide que se atiendan las señales de interrupción hasta que explícitamente sean “facultades” mediante el comando **EI**.

DIM

Reserva espacio para una tabla de las dimensiones dadas.

DIM /lista de:/variable subindicada

Siendo /variable subindicada/ la forma por /nombre variable/(/lista dimensiones/) y la /lista dimensiones/ viene dada por /lista de:/expresión entera/.

DRAW

Traza una recta en la Pantalla de Gráficos.

Para dibujar una línea en la pantalla a partir de la posición corriente hasta una posición absoluta.

DRAW / coordenada x/,/ coordenada y[/,/tinta cribada/]

DRAWR

Traza una recta en la Pantalla de Gráficos.

Para dibujar una línea recta en la pantalla a partir de la posición corriente hasta una posición relativa a la actual.

DRAWR /incremento x/ ,/incremento y/ [,/tinta cribada/]

EDIT

Revisar una línea de programa.

EDIT/número línea/

EI

Faculta el tratamiento de las interrupciones.

Para volver a habilitar las interrupciones inhibidas mediante un comando **DI**.

EI.

END

Finaliza la ejecución del programa.

END.

ENT

Fija la envolvente de tono para las notas.

Mientras se está produciendo un sonido es posible hacer que varíe su tono.

ENT/número de envolvente/[,/lista de:/secciones de envolvente/]

/número de envolvente/ es una */expresión entera/* cuyo valor absoluto debe estar en la banda 1..15, y especifica cuál de las envolventes de tono va a usarse. Si el */número de envolvente/* es negativo, entonces la envolvente es repetitiva, haciendo que la nota se repita continuamente hasta su conclusión.

Cinco/*secciones de envolvente/*. Cada una puede adoptar una de las dos formas siguientes:

/cuenta de pasos/,/altura del paso/ ,/tiempo del paso/

o bien: */periodo del tono/,/tiempo del paso/*

La primera de las normas especifica un cambio incremental en relación con el valor corriente para el periodo del tono. Con la segunda forma se especifica un valor absoluto para el periodo de tono. Siendo:

<i>/cuenta de pasos/</i>	da en número de pasos en la sección- una/ <i>expresión entera/</i> en la banda 0..239.
<i>/altura del paso/</i>	da la cuantía por la que ha de variarse el periodo de tono en cada paso de la sección-una <i>/expresión entera/</i> que produzca un valor en la banda —128... +127.
<i>/tiempo del paso/</i>	da el tiempo que dura cada paso-una/ <i>expresión entera/</i> especificando el tiempo en centésimas de segundo. La expresión debe producir un valor en la banda 0..255 (donde 0 se trata como 256;)
<i>/periodo del tono/</i>	da un nuevo valor absoluto para el periodo del tono usado en la sección —una <i>/expresión entera/</i> que produce un valor en la banda 0..4095

ENV

Fijar la envolvente de volumen para una nota.

Mientras se está produciendo un sonido es posible hacer que varíe su volumen. Una envolvente de volumen define como ha de ser dicha variación.

ENV /número de envolvente/[,/lista de: /sección de envolvente/]

Cinco /secciones de envolvente/. Cada una puede tomar una de las dos formas siguientes:

/cuenta de pasos/ ,*/altura del paso/* ,*/tiempo del paso/*
o bien: */envolvente prescrita/* ,*/periodo de envolvente/*

La primera de las formas especifica una sección en que el volumen varía incremental o absolutamente bajo control del programa, siendo los parámetros:

<i>/cuentas de pasos/</i>	da el número de pasos en la sección —una /expresión entera/en la banda 0..127
<i>altura del paso/</i>	Si la /cuenta de pasos/no es 0, la /altura del paso/de la cuantía en la que variará el volumen en cada paso sucesivo de la sección—una /expresión entera/ n la banda —128 ..+127.
<i>/tiempo del paso/</i>	especifica el tiempo que dura cada paso de la sección—una /expresión entera/que señala el tiempo en centésimas de segundo. Dicha expresión debe producir un valor en la banda 0.. 255

La segunda de las formas de la instrucción especifica una sección de envolvente que va a ejecutarse bajo control directo de los circuitos procesadores de sonido, siendo:

<i>/envolvente prescrita/</i>	es el valor a depositar dentro del registro de perfil de envolvente (registro 13)
<i>/periodo de envolvente/</i>	es el valor a depositar dentro de los registros de periodo de envolvente (registros 11 y 12).

ERASE

Elimina las tablas definidas en el programa.

ERASE/lista de:/nombre variable/

ERROR

Provoca un Error con la acción subsecuente.

Simular un error con código de error determinado.

ERROR/expresión entera/

EVERY

Suscitar la ejecución de una subrutina cada cierto tiempo.

El sistema mantiene un reloj en el tiempo real. El comando permite que en un programa BASIC se señalen subrutinas cuya ejecución ha de suscitarse a intervalos regulares de tiempo.

EVERY/intervalo tiempo/[,/número temporizador/]GOSUB/número línea/

El /intervalo tiempo/ es una expresión entera/ que especifica cuánto tiempo tiene que transcurrir entre cada ejecución de la subrutina señalada.

El/número de temporizador/ es una /expresión entera/ que identifica cuál de los cuatro posibles temporizadores de demora va a usarse.

FOR

Bucle preconfinado.

Ejecutar una serie de instrucciones un determinado número de veces, incrementando una variable de control entre un valor de comienzo y un valor de finalización.

FOR/variable simple/= /comienzo/ TO /final/ [STEP /incremento/]

La/variable simple/ es la que controla el bucle **FOR**, y debe ser Entera o Real y no puede ser un elemento de una tabla.

El /comienzo/ es una /expresión numérica/ que da el valor límite para la variable de control.

El /final/ es una /expresión numérica/ que da el valor límite para la variable de control.

El /incremento/ es una /expresión numérica/ que da el valor que se añade a la variable de control después de cada iteración o ronda.

Los bucles **FOR** pueden ser anidados.

GOSUB

Desvío de la ejecución hacia una subrutina.

Para poner en marcha una subrutina determinada.

GOSUB /número línea/

Las subrutinas se terminan mediante un comando **RETURN**. Una Subrutina puede contener más de un comando **RETURN**.

GOTO

Saltar o bifurcar a un número de línea.

Para efectuar un salto en la ejecución de un programa hasta un número de líneas dado, y continuar la ejecución a partir de dicha línea.

GOTO/número línea/

IF

Establecer una premisa condicional.

Para hacer que la ejecución de las instrucciones esté supeditada al resultado de una expresión.

IF/expresión logical/ THEN /alternativa- 1/[ELSE/alternativa-2/]
IF/expresión logical/ GOTO /número línea/[ELSE/alternativa/]

Siendo las /alternativa/s bien: /serie de comandos/
o bien: /número línea/
y: /serie comandos

es uno o más comandos consecutivos dentro de la misma línea que la partícula **IF**, y separados por dos-puntos.

INK

Fijar el color que corresponde a la tinta dada.

Dependiendo del modo de pantalla vigente, se dispone de un cierto número de tintas. El color o la pareja de colores, asociada a una tinta puede cambiarse mediante este comando **INK**.

INK/tinta/,/color/[,/color/]

INPUT

Impone la información tomada de un cauce como valores de variables.

Para leer datos procedentes de un cauce de entrada dado. Produce un "aviso" en la pantalla cuando el cauce de entrada es el teclado.

**INPUT[#expresión cauce/],[;][/aviso entrecomillado/;]lista de:/variable/
INPUT[#expresión cauce/],[;][/aviso entrecomillado/;]lista de:/variable/**

Si se omite la /expresión cauce/ se usa el prescrito para omisiones que es cauce **#0**, procedente del teclado.

El /aviso entrecomillado/ es una constante literal entrecomillada.

Datos procedentes del teclado - Cauces **#0..#8**.

Datos procedentes del cassette - Cauce **#9**.

KEY

Adscribir un nuevo mensaje de entrada a una tecla funcional.

Se admiten hasta treinta y dos "códigos de entrada desarrollables" en el teclado. Cuando se pulsa una de esas teclas, la serie de caracteres —el dicho— asignado a ella, es la que

se comunica en lugar del código de entrada correspondiente.

KEY/código de entrada desarrollable/,/expresión literal/

El /código de entrada desarrollable/ es una /expresión entera/ que debe producir un valor en la banda 0..31, y que señala cual de los códigos de entrada es el que se va a emplear.

La /expresión literal/ se evalúa previamente y el resultado obtenido se deposita para reemplazar al código de entrada cuando se pulse la tecla o teclas correspondientes.

KEY DEF

Definir los valores correspondientes a una tecla dada.

Los valores generados cuando se pulsa una tecla, solamente ella o simultáneamente con otra, pueden redefinirse usando este comando.

También permite establecer si la tecla es autorrepetitiva o no.

KEY DEF/número tecla/,/repite/,[/normal/[/,/turnada/[/,/control/]]]

El/número tecla/ es una /expresión entera/ que produce un valor en la banda 0..79,y que especifica la tecla cuyos valores van a redefinirse.

El parámetro/repite/es una /expresión entera/ que produce un valor en la banda 0..1, y que especifica si la tecla es autorrepetitiva (1) o si no lo es (0).

Los parámetros /normal/ ,/turnada/ y /control/ , cuando se especifican , son todos /expresiones enteras/ que producen valores en la banda 0..255.

/normal/ es el valor cuando no se han pulsado además ni **SHIFT** ni **CTRL**.

/turnada/ es el valor cuando también está pulsada **SHIFT**, pero no **CTRL**.

/control/ es el valor cuando también está pulsada **CTRL**.

LET

Asigna un valor a una variable.

[LET]/variable/=/expresión/

LINE INPUT

Introducir una línea completa procedente de un cauce de entrada.

Para recoger una línea completa de texto introducida por un determinado cauce, y asignarla como valor de una variable literal. Produce un aviso en la pantalla cuando el cauce de entrada es teclado.

**LINE INPUT[~~#~~/expresión cauce/,][;][/aviso;/]variable literal/
LINE INPUT[~~#~~/expresión cauce/,][/aviso/,]variable literal/**

Si se omite la /expresión cauce/se presupone que la entrada es por el cauce del teclado (cauce ~~#~~0).

El /aviso/ es opcional y debe ser una constante literal entrecomillada

La /variable literal/ especifica el nombre de la variable a la que se asignará la línea introducida.

LIST

Mostrar las líneas de un programa.

LIST /banda números de líneas/[~~#~~expresión cauce/]

La /banda números de línea/ define las líneas de programa que van a ser listadas, incluyendo las cotas inferior y superior. Si se omite, se mostrarán todas las líneas de programa.

LOAD

Cargar un programa en la memoria.

LOAD /nombre fichero/[~~#~~/expresión direccional/]

El /nombre fichero/ es una /expresión literal/ que da el nombre del fichero a partir del cual

se efectúa la carga.

LOCATE

Sitúa el cursor en una posición dada.

Para desplazar el cursor a una nueva posición.

LOCATE[#/*expresión cauce/* ,]/*coordenada x/* ,/*coordenada y/*

Si esta presenta la /*expresión cauce/* debe producir un valor correspondiente a un cauce de pantalla.

La /*coordenada x/* da la nueva columna que corresponde al cursor; la /*coordenada y/* de el nuevo número de línea que corresponde a la posición final del cursor.

MEMORY

Restaura los parámetros de memoria para el BASIC.

Para cambiar la cantidad de memoria utilizable por el BASIC.

MEMORY/*expresión direccional/*

La /*expresión direccional/* fija la dirección del byte más alto en la memoria que puede ser usado por BASIC.

MERGE

Congrega un programa procedente del cassette con el presente en memoria.

Agregar el contenido de un fichero al programa existente en la memoria

MERGE/*nombre fichero/*

MERGE tiene los siguientes efectos sobre el entorno actual:

-
- Se descartan todas las variables y tablas.
 - Se olvidan todas las funciones de usuario
 - Se desactiva el posible **ON ERROR GOTO**.
 - Se abandonan todos los ficheros abiertos —perdiéndose cualquier salida que hubiera en el “buffer”.
 - Se adopta la acción **RESTORE**.
 - Se restaura lo establecido mediante **DEFINT**, **DEFREAL** y **DEFSTR**.

Durante la operación de **MERGE**, cualquier línea en el nuevo programa cuyo número coincida con una línea del programa existente en memoria, sustituirá a la línea de memoria.

MODE

Fija el modo de gestión para la pantalla.

Para cambiar la manera de operar con la pantalla del monitor.

MODE/expresión entera/

Los modos de pantalla son como sigue:

- 0: 16 Colores 25 líneas por 20 caracteres. 200 por 160 puntos.
- 1: 4 Colores 25 líneas por 40 caracteres. 200 por 320 puntos.
- 2: 2 Colores 25 líneas por 80 caracteres. 200 por 640 puntos.

MOVE

Mover el cursor de gráficos - Coordenadas Absolutas.

Desplazar el cursor de gráficos hasta la posición señalada por coordenadas absolutas.

MOVE/coordenada x/ ,/coordenada y/

MOVER

Mover el cursor de gráficos - Coordenadas Relativas.

Desplazar el cursor de gráficos hasta una posición especificada por coordenadas relativas con respecto a la posición actual.

MOVER/desplazamiento x/ ,/desplazamiento y/

NEW

Prepararse para un nuevo programa.

Antes de comenzar un nuevo programa, se puede usar **NEW** para quitar completamente el contenido presente en la memoria.

NEW

Además de suprimir el programa existente en memoria, se toman las siguientes acciones:

- Se descartan todas las variables y tablas.
- Se olvidan todas las Funciones de Usuario.
- Se desactiva el posible **ON ERROR GOTO**.
- Se abandonan todos los ficheros - perdiéndose cualquier salida en el "buffer".
- Se restaura lo estipulado por **DEFINT**, **DEFREAL** y **DEFSTR**.
- Se desactiva el rastreo puesto por **TRON**.

NEXT

Incrementa la variable de control **FOR** al valor siguiente.

Delimita el extremo de un bucle **FOR**. El comando **NEXT** puede ser anónimo, o pueden mencionar al **FOR** correspondiente.

NEXT[/lista de;/variable/]

siendo: **NEXT**/variable/ ,/lista de;/variable/

equivalente a: **NEXT**/variable/:**NEXT**/lista de;/variable/

ON /expression/ **GOSUB**

ON /expression/ **GOTO**.

GOSUB y **GOTO** dependiente de un valor computado.

Para elegir una entre varias subrutinas a las que desviarse, o entre varias líneas a las que saltar; dependiendo del resultado de una expresión.

**ON/selector/GOSUB/lista de:/número línea/
ON/selector/GOTO/lista de:/número línea/**

ON BREAK GOSUB

Permitir las subrutinas suscitadas por interrupción de corte.

Cuando se pulsa sucesivamente **ESC/ESC**, mientras el **BASIC** está ejecutando un programa, se suspende la ejecución y el **BASIC** regresa al Modo Directo, indicando que se ha producido un corte (“**Break**”) en la ejecución.

El **ON BREAK GOSUB** permite que en lugar que el mensaje correspondiente, el **BASIC** pase a ejecutar la subrutina citada con el comando, cuando se produce una interrupción de esta clase.

ON BREAK GOSUB/número línea/.

ON BREAK STOP

Cancela la detección de interrupciones en la ejecución.

Quita cualquier “cepo” establecido para detectar un corte en la ejecución producido al pulsar **ESC/ESC**, y establecido mediante el comando **ON BREAK GOSUB**.

ON BREAK STOP.

ON ERROR GOTO

Establecer un cepo para detectar errores.

Cuando el **BASIC** detecta un error mientras está siguiendo un programa, puede pasar a ejercer la acción prescrita, que es generar un mensaje de error y regresar al Modo Directo;

o puede hacerse que pase a efectuar una subrutina para resarcimiento de errores, incorporada al propio programa mediante el comando **ON ERROR GOTO**.

**ON ERROR GOTO /número línea/
ON ERROR GOTO O**

ON SQ GOSUB

Establece el cepo para detectar vacíos en la secuencia de notas a emitir.

Atender las señales de interrupción enviadas por los circuitos generadores de sonido cuando detectan un sitio libre en una determinada secuencia de notas que está esperando ser emitida.

**ON SQ <<canal>>
GOSUB <número línea>**

OPENIN

Abre un fichero de entrada en cassette.

Para prepararse a recoger datos de un fichero en cinta.

OPENIN/nombre fichero/

OPENOUT

Abrir un fichero de salida en cassette.

Para preparar la salida de datos hacia un fichero en cintas.

OPENOUT/nombre fichero/

ORIGIN

Estipula el origen y la ventana de la pantalla de gráficos.

Se puede desplazar la posición base 0,0 en el sistema de coordenadas para gráfico. Puede asimismo restringirse la zona de pantalla —la ventana— usada para gráficos.

ORIGIN/x0/,/y0/[**/izquierda/** ,/**derecha/** ,/**arriba/** ,/**abajo/**]

Las coordenadas /x0/ e /y0/ del nuevo origen, debe ser /expresiones enteras/

Los parámetros /izquierda/ ,/ derecha/ ,/arriba/ ,/abajo/ especifican los bordes de la ventana para gráficos. Si se omiten, la ventana actual permanece sin verse afectada.

OUT

Saca un valor hacia el portal de salida de la máquina.

Para enviar un byte a un determinado portal de salida.

OUT/número portal/ ,/**expresión entera/**.

PAPER

Estipula la tinta a usar para el fondo de los textos e imágenes.

Cuando se exponen caracteres en pantalla, la posición que ocupa el carácter se rellena con el Tinte del Papel antes de mostrar el carácter —a no ser que se haya seleccionado el modo Transparente—. El comando **PAPER** especifica el Tinte que va a usarse.

PAPER [**#**]/**expresión cauce/** ,]/**tinta Cribada/**

PEN

Estipular la tinta usada para el frente de los caracteres e imágenes.

Identificar la tinta que va a usarse al exponer caracteres o dibujar figuras en la pantalla.

PEN [**#**]/**[expresión cauce],** <**Tinta Cribada**>

PLOT

Pinta un punto en la pantalla de gráficos.

Para mostrar un punto de color en una determinada posición de la pantalla dada en coordenadas absolutas, y con un tamaño en puntos dependiente del modo de pantalla en que se opere.

PLOT/coordenada x/ ,/coordenada y/[,/tinta cribada/]

PLOTR

Para pintar un punto en la pantalla de Gráficos.

Para mostrar un punto de color en la posición especificada relativa a la posición corriente del cursor de gráficos, y del tamaño en puntos dependiente del modo de gráficos empleados.

PLOTR/desplazamiento x/,/desplazamiento y/[,/tinta cribada/]

POKE

Meter un valor en la memoria de la máquina.

Permite el acceso directo a la memoria de la máquina para escribir un byte.

POKE/expresión direccional/ ,/expresión entera/

La /expresión direccional/da la dirección de la celdilla de memoria en que se va a escribir.

PRINT

Enviar datos a través de un cauce de salida.

Para sacar datos, numéricos o literales, a través de un cauce dado.

PRINT[#/expresión cauce/,][/cláusula usando/][/separador/]

/lista a sacar/ es :/dato a sacar/[/separador//dato a sacar/]*

Siendo /dato a sacar/: /expresión/
o bien: **SPC** (/expresión entera/)
o bien: **TAB** (/expresión entera/)

La cláusula usando/ es: **USING**/literal conformador/; /lista a conformar/ siendo/ lista a conformar/:/expresión/[/separador//expresión/]*
y donde /separador/ es: coma o punto — y — coma.

(Observa que la construcción metalingüística [:•]* significa que dicho objeto es opcional, pero puede repetirse cualquier número de veces).

1. Efecto:

Los datos a sacar/ si los hay dentro de la /lista a sacar/, se evalúan y sacan en “formato libre”.

Si está presente una /cláusula usando/, entonces el /literal conformador/ define una “forma” o “plantilla” que controla el formato en que aparecerán los valores de las expresiones incluidas en la /lista a conformar/.

Cuando se hayan procesado todos los argumentos, se enviará un retorno de carro, a no ser que el comando **PRINT** termine en un /separador/, **TAB** o **SPC**.

2. Exposición de datos en Formato Libre:

Los /datos a sacar/ se evalúan sucesivamente y se van sacando los resultados obtenidos por el cauce especificado.

Una coma que siga a un /dato a sacar/hace que el BASIC avance con espacios en blanco hasta el comienzo de la siguiente Zona de exposición, antes de procesar la siguiente expresión. Un punto — y — como simplemente separa las expresiones, sin incluir ningún espacio en blanco entre ellas.

2. 1. Zonas de Exposición y Anchura del Cauce.

El BASIC divide cada línea de la pantalla en Zonas, cuya anchura está fijada por el comando **ZONE**, y para omisiones 13 caracteres de anchura.

El "borde de un cauce" depende de la clase de cauce de salida:

Cauces de pantalla:	El borde es el borde de la ventana actual.
Cauces de impresora:	El borde está definido por el comando WIDTH . La posición de la impresora se puede saber mediante la función POS .
Cauce de Cassette:	No existe ningún borde que lo limite.

2. 2. Exposición de Número en Formato Libre.

Los números positivos van precedidos de un espacio en blanco; los negativos por un signo menos.

Todos los números se exponen con el mínimo de caracteres. No se saca ningún punto decimal si no hay ningún dígito fraccionario significativo.

Los números reales se exponen según "formato escalado" (con un exponente) si no pueden representarse exactamente mediante un número con nueve o menos dígitos (excluyendo un posible cero delantero).

2. 3. Función SPC para el comando PRINT.

SPC avanza un determinado número de espacios, antes de exponer el siguiente dato.

SPC (/expresión entera/)

2. 4. Función TAB en el comando PRINT

TAB saca los espacios en blanco suficientes para pasar a una determinada posición del cursor, cabezal de impresión o cabezal de lectura, según el cauce especificado.

TAB (/expresión entera/)

3. Usando exposición de datos con Formato.

Las expresiones en la /lista a conformar/ se evalúan una a una sucesivamente y se exponen a través del cauce de acuerdo con la "horma" o "plantilla" de conformación. Si el /literal conformador/ se ha quedado vacío mientras se examina en busca de una especificación de formato, la búsqueda vuelve a comenzar de nuevo a partir del comienzo de /literal conformador/.

3. 1. Plantilla conformadora (también llamada “máscara de formato”)

La “horma” o plantilla conformadora es un literal que el BASIC interpreta carácter a carácter para controlar la manera en que se expone el resultado de cada expresión, por el cauce de pantalla, impresora o grabadora especificado. Se aceptan y reconocen los siguientes caracteres en las especificaciones de la plantilla conformadora:

i \ &#. + - * \$↑ , _

3. 2. Especificaciones de Formato para datos Literales.

“i” Solo el primer carácter del literal es el que se saca.
”/espacios/ Se sacarán los primeros “n” caracteres del dato literal, siendo “n” la longitud en caracteres de /espacios/ (habiendo por tanto “n-2” espacios en blanco entre las barras inclinadas invertidas.
” & ” El dato literal completo se saca “tal cual es”

3. 3. Especificaciones de Formato para datos Numerales.

Los números pueden exponerse por pantalla o impresora, o grabarse en cassette sin la parte exponencial (“sin-escala”) o con notación científica (“escalada” con mantisa y exponente).

La plantilla conformadora para un número no puede sobrepasar de 20 caracteres, sin contar el exponente ni tampoco las opciones de signos en la parte posterior.

Para especificar los dígitos significativos de un dato numérico:

’ # ’ Cada carácter “#” en la plantilla especifica una posición de un dígito del dato a exponer.

“.” Especifica la posición que ha de ocupar el punto decimal. Puede haber como máximo un carácter“.”

“,” Puede aparecer antes de “.” anteriormente mencionado.

Dólar delantero y Asterisco:

Estas opciones deben incluirse en la plantilla, inmediatamente antes de los caracteres que fijan las posiciones de dígito:

“\$\$” Especifica dos posiciones de dígitos. Indica que ha de sacarse un signo “\$” inmediatamente antes del primer dígito significativo o punto decimal.

“***” Indica que cualquier espacio blanco delantero que pudiera haber en el dato a exponer, sea sustituido por “*” s.

“**\$” Actúa como combinación de las dos opciones previamente mencionadas.

Signo:

“+” Especifica que se saque el “+” o el “—” según corresponda.

“—” Hace que saque el “—” para datos numéricos negativos, y un espacio en blanco para los positivos.

Opción de Notación Científica, con Exponente:

“###” Seguido de la parte principal del número y precediendo cualquier indicación posterior de signo, permite la notación con exponente.

Rebasamiento de Campo:

Si no se puede exponer un número que cumpla con la plantilla conformadora dada, el BASIC intenta acercarse lo más posible al formato exigido, pero hace que el resultado aparezca precedido de un “%” para señalar que se ha “rebasado el campo”

4. Continuación de la Exposición de datos.

Terminando el comando **PRINT** con un /separador/, **TAB** o **SPC**, impide que el BASIC lance una nueva línea, y que por tanto, el siguiente comando **PRINT** sea tratado como continuación del anterior en cuanto a posiciones.

RAD

Considera los radianes como medida de los ángulos.

RAD

RANDOMIZE

Hace que sea aleatorio el germen actual de la secuencia de números de azar.

Produce una secuencia pseudo-aleatoria, en que cada número depende completamente del número previamente producido.

RANDOMIZE [/expresión numérica/]

READ

Leer las constantes incluidas en instrucciones **DATA**.

READ recoge las constantes numéricas o literales de las instrucciones **DATA** y las “apunta” como valores de variables.

READ /lista de: /variable/

RELEASE

Deja que sean emitidas las notas almacenadas en los canales de sonido.

RELEASE /canales sonido/

Bit 0 (el bit menos significativo) especifica el canal A.

Bit 1 Especifica el canal B.

y Bit 2 Especifica el canal C.

REM

Incluir comentarios en un programa.

REM /resto de línea/

RENUM

Renumerar las líneas del programa en curso.

Para cambiar la numeración de todo el programa o parte de él.

RENUM [/nuevo número línea/] [,/viejo número línea/] [,/incremento/]

El /nuevo número línea/ es un /número línea/ que da el nuevo número que se quiere para la primera línea de programa que se vaya a renumerar. Si se omite se supone 10.

El /viejo número línea/ es un /número línea/ que da la primera línea del programa que va a ser renumerada. Si se omite, se supone la primera línea que haya en dicho programa.

El /incremento/ es /número línea/ que da el salto que va a usarse en la nueva numeración de líneas. Si se omite, se supone 10.

RESTORE

Hacer que el puntero repunte a una instrucción **DATA** determinada.

Desplazar el "puntero" interno que lleva la cuenta de las constantes ya usadas de las instrucciones **DATA**. se puede usar el comando **RESTORE** haciendo que señale a un determinado número de línea del programa (típicamente, para que vuelva a señalar a la primera instrucción **DATA** del programa).

RESTORE [/número línea/]

El /número línea/ fija la línea hacia la que se debe señalar el puntero interno de **DATA**. Si se omite, dicho puntero señala de nuevo a la línea inicial del programa.

RESUME

Reanudada la ejecución después de tratar un error.

Cuando se ha detectado, se ha "atrapado" un error y subsecuentemente se ha pasado a ejecutar la subrutina señalada mediante el comando **ON ERROR GOTO** para resarcirse del error, el comando **RESUME** permite que continúe la ejecución normal del programa a partir de diversos puntos del mismo.

RESUME o bien RESUME /número línea/ o bien RESUME NEXT

RESUME sin ningún parámetro, devuelve el control de la ejecución al principio de la instrucción en que se detectó primeramente el error.

RESUME /número línea/ devuelve el control a la línea que se especifique por su número.

RESUME NEXT devuelve el control de la ejecución a la instrucción que va inmediatamente detrás de aquella en que se detectó el error en un principio.

RETURN

Volver de una subrutina.

Marca el final de una subrutina.

RETURN

RUN /filename/

Carga y comienza a ejecutar un programa.

Para cargar un programa desde el cassette y comenzar automáticamente a ejecutarlo.

RUN /nombre fichero/.

RUN [/line number/]

Comenzar la ejecución del programa existente en memoria.

Iniciar la ejecución del programa en curso, ya sea al principio del mismo o bien a partir de una línea dada.

RUN [/número línea/]

Si se omite al /número línea/, la ejecución comienza con la primera línea que haya en el programa.

SAVE

Guardar en cassette un programa o el contenido de la memoria.

SAVE /nombre fichero/ [,/clase fichero/ [,/parámetros binarios/]]

El /nombre fichero/ es una /expresión literal/ que da el nombre con que quedará registrado en el cassette el fichero.

La /clase fichero/, si está presente, define la modalidad de fichero que va a ser creada.

- A— el programa actual en BASIC se guarda como un fichero de texto en ASCII.
- P— el programa actual en BASIC se guarda en la forma protegida interna.
- B— se guarda un cierto área de la memoria como un fichero Binario.

Los /parámetros binarios/ se exigen para la /clase fichero/ B, pero en los demás casos es ilegal.

SAVE, B crea un fichero Binario que contiene una copia del recinto de memoria delimitado por los /parámetros binarios/ especificados en el comando y que tienen la siguiente forma:

/dirección comienzo/, /longitud/ [, /punto de entrada/]

donde:

/dirección comienzo/ es una /expresión direccional/ que da la dirección del primer byte que va a escribirse.

/longitud/ es una /expresión direccional/ que da la cantidad de bytes que van a escribirse.

/punto de entrada/ es una /expresión direccional/ que da la dirección a partir de la cual comenzará la ejecución si el fichero corresponde a un programa, mediante **RUN**.

SOUND

Incluir una nota sonora dentro de la secuencia de sonidos a emitir.

SOUND /estado canal/, /período tono/ [,/duración/ [,/volumen/[,/envolvente volumen/ [,/envolvente tono/ [,/período ruido/]]]]]

/estado canal/ es una /expresión entera/ en la banda 1..255.

El valor dado está “calibrado en binario” en la forma siguiente:

Bit 0: envía la nota al canal A

Bit 1: envía la nota al canal B

Bit 2: envía la nota al canal C

Bit 3: hay “acorde” con el canal A

Bit 4: hay “acorde” con el canal B

Bit 5: hay “acorde” con el canal C

Bit 6: se marca como “retenida”

Bit 7: hace que se “evacue” (vacíe) la secuencia de notas sonoras.

El /período tono/ es una /expresión entera/ en la banda 0..4095

Fija el tono de la nota a emitir. Un /período tono/ P produce un tono de frecuencia F, donde $F = 125000/P$

La /duración/ es una /expresión entera/ en la banda $-32768..+32767$

El efecto del parámetro /duración/ depende del signo y del valor:

> 0 especifica la duración del sonido en centésimas de segundo.

=0 especifica que el sonido ha de durar hasta que termine la envolvente de volumen.

<0 especifica que la envolvente de volumen debe repetirse **ABS** (/duración/)

Si no se especifica ninguna /duración/, se supone 20, con lo que el sonido durará un quinto de segundo.

El /volumen/ es una /expresión entera/ en la banda 0..15.

Especifica el volumen inicial de la nota.

Si no se especifica ningún /volumen/ se supone el valor 12.

La /envolvente volumen/ es una /expresión entera/ en la banda 0..15.

Especifica cuál de las envolventes de volumen, si hay alguna, se va a usar para generar la nota sonora.

Si no se especifica se emplea la envolvente 0.

Es una constante, mantiene el volumen dado por el parámetro /volumen/ durante dos segundos.

La /envolvente tono/ es una /expresión entera/ en la banda 0..15.

Especifica cuál de las envolventes de tono, si hay alguna, va a ser usada al generar la nota sonora.

El /período ruido/ es una /expresión entera/ en la banda 0..31.

Especifica cuál de los ruidos, si hay alguno, se va a añadir a la nota generada. Si se omite el parámetro o se le dá el valor cero, significa que no se añade ninguna clase de ruido.

SPEED INK

Gradúa la rapidez con que parpadea la tinta.

Este comando establece los períodos, las duraciones que tienen los cambios entre colores.

SPEED INK /período/, /período/

El primero especifica el lapso de tiempo en que estará presente el primer color asociado a la Tinta; el segundo especifica el lapso de tiempo para el segundo color.

SPEED KEY

Gradúa la rapidez de autorrepetición de las teclas.

Establece el tiempo de demora antes de que comience a repetirse el envío del valor, y además la rapidez con que se efectuarán dicho envío a partir de ese momento.

SPEED KEY /demora comienzo/, /período repetición/

La /demora comienzo/ es una /expresión entera/ en la banda 1..255, y especifica en 1/50-avos de segundo el retardo que ha de haber entre la primera pulsación de la tecla y la repetición automática del primer valor.

El /período repetición/ es una /expresión entera/ en la banda 1..255, y especifica el 1/50-avos

de segundo el tiempo que ha de transcurrir entre cada repetición automática del valor, una vez que haya comenzado a actuar la repetición automática.

SPEED WRITE

Gradúa la rapidez con que se transfieren datos al cassette.

Se puede transferir datos al cassette en una de dos “cadencias”.

SPEED WRITE /expresión entera/

La /expresión entera/ debe producir un valor especificando cuál de las cadencias de transferencia va a usarse, en la forma siguiente:

0: Nominal 1000 bits por segundo.

1: Nominal 2000 bits por segundo.

STOP

Para la ejecución de un programa.

STOP

SYMBOL

Define la matriz de los Símbolos Definibles por el Usuario.

SYMBOL /número carácter/, /lista de: /filas/

El /número carácter/ es una /expresión entera/ en la banda n..255, siendo n el primero de los símbolos Definibles por el Usuario.

La /lista de: /filas/ es una lista con uno y ocho elementos posibles, en que cada uno de ellos son /expresiones enteras/ que han de producir el valor en la banda 0..255. Si se especifican menos de ocho filas para la matriz que determina la forma, el “tipo” del carácter, las filas no especificadas se colocan al valor cero.

SYMBOL AFTER

Establece una nueva cantidad de Símbolos Definibles por el Usuario.

SYMBOL AFTER /expresión entera/

La /expresión entera/ debe producir un valor en la banda 0..256, y especifica el número del caracter a partir del cual todos son definibles por el usuario.

TAG

Texto en la posición del cursor de gráficos.

TAG [# /expresión cauce/]

La /expresión cauce/ debe producir un valor coherente con el cauce de pantalla. Si se omite, se supone el cauce #0.

TAGOFF

Quitar el estado de Texto a Gráficos.

Cancela la posibilidad de mostrar texto en la posición ocupada por el cursor de gráficos, para el cauce dado.

TAGOFF [# /expresión cauce/]

La /expresión cauce/ debe producir un valor coherente con un cauce de pantalla. Si se omite, se supone el cauce #0.

TRON

TROFF

Para poner y quitar el mecanismo de "rastreo" de la ejecución.

El BASIC incluye la facilidad de poder rastrear la ejecución de un programa presentando

en pantalla el número de cada línea del programa,justamente antes de que pase a ser ejecutada.

TRON TROFF

TRON faculta la posibilidad de rastreo, **TROFF** la cancela.

Mientras está habilitado el mecanismo de rastreo, va apareciendo en el monitor el número de cada línea del programa, inmediatamente antes de que sea ejecutada.

WAIT

Vigila el estado de un portal de entrada/salida.

WAIT /número portal/, /máscara/ /inversión/

El /número portal/ es una /expresión direccional/.

Tanto el parámetro de /máscara/ como el /inversión/ han de ser /expresiones enteras/ que produzcan valores dentro de la banda 0..255.

WEND

Marca el límite final de un bucle **WHILE**.

Un bucle **WHILE** hace que se ejecute repetidamente una serie de líneas de programa hasta que una determinada condición sea "cierta". El comando **WEND** define el extremo de dicho bucle.

WEND

WHILE

Iniciar un bucle **WHILE**.

Un bucle **WHILE** hace que se ejecute repetidamente una serie de líneas consecutivas de programa "mientras" sea cierta una condición dada.

El comando **WHILE** define la línea inicial de dicho bucle y establece la condición que ha de ser cierta para que el bucle sea ejecutado.

WHILE /expresión logical/

WIDTH

Fija la anchura de línea en la impresora.

WIDTH /expresión entera/

La /expresión entera/ da la anchura de la impresora y debe producir un valor en la banda 1..255.

WINDOW

Fija una ventana de texto en la pantalla.

WINDOW [# /expresión cauce/,] /izquierda/ /derecha/ /arriba/ /abajo/

La /expresión cauce/ identifica al cauce cuyas dimensiones de ventana se están estipulando.

Todos los otros parámetros especifican las posiciones de los cuatro bordes de la ventana de forma absoluta según las filas y columnas de la ventana.

/izquierda/ La columna extremo-izquierda incluida en la ventana

/derecha/ La columna extremo-derecha incluida en la ventana

/arriba/ La fila más superior (con mínimo número) incluida en la ventana.

/abajo/ La fila más inferior (con máximo número) incluida en la ventana.

WINDOW SWAP

Canjea —intercambia— ventanas de texto

Este comando permite, entre otros efectos, que se intercambie un cauce cualquiera con el cauce #0 para “redirigir” el texto enviado hasta el cauce #0.

WINDOW SWAP /expresión cauce/, /expresión cauce/.

Donde ambas /expresiones cauce/ deben producir un valor coherente con un cauce de pantalla.

WRITE

Escribir datos a través de un cauce de salida.

Para sacar los valores de un cierto número de expresiones hasta un cauce dado, separándolos por comas y encerrándolos entre dobles comillas.

WRITE [# /expresión cauce/,] [/lista a escribir/]

La /expresión cauce/ especifica el cauce a través del que va a escribirse la información. Si se omite, se supone el cauce #0.

La /lista a escribir/ es: /expresión/ [/separador/ /expresión/*
donde /separador/ puede ser una coma o un punto —y— coma, intercambiabilmente.

WRITE es similar a **PRINT**, exceptuando que:

- las zonas de exposición se ignoran.
- los literales se escriben encerrados entre dobles comillas.
- se añaden comas entre los elementos de la lista a escribir.
- **WRITE** no permite la opción de separadores posteriores.

WRITE está pensado para ser usado con el cauce de cassette.

ZONE

Fija el tamaño de la zona de exposición.

Para cambiar la anchura de la Zona de Exposición empleada con el comando **PRINT**.

ZONE /expresión entera/

La /expresión entera/ fija la nueva anchura de las zonas de exposición en pantalla, y debe producir un valor en la banda 1..255.

FUNCIONES DEL BASIC

ABS

Valor absoluto.

Determinar el valor absoluto de una expresión dada.

ABS (/expresión numérica/)

ASC

Dice el valor ASCII del carácter.

Conseguir el valor numérico del primer carácter de un literal, dado que la codificación de caracteres ASCII es la empleada.

La inversa de **CHR\$**

ASC (/expresión literal/)

ATN

Arcotangente.

El resultado obtenido es un ángulo bien en radianes (en la banda $-\pi/2.. +\pi/2$) o en grados (en la banda $-90..+90$) dependiendo del modo vigente.

ATN (/expresión numérica/)

BIN\$

Literal binario.

Para producir una serie de símbolos representando el valor de la expresión dada.

BIN\$ (/expresión entera sin-signo/ ,/anchura campo/)

La /anchura campo/ opcional es una /expresión entera/ dando el mínimo tamaño de la serie literal (string) que se produce. La expresión debe resultar en un valor en la banda 0..16

CHR\$

Convierte a carácter.

Para convertir un valor numérico a su equivalente "literal" o carácter.

Es la inversa de **ASC**

CHR\$ (/expresión entera/)

CINT

Convierte a entero.

Para convertir el valor dado a su representación Entera, si es posible.

CINT (/expresión numérica/)

COS

Coseno de un ángulo.

En el modo Radianes, el ángulo viene expresado en radianes. En el modo Grados el ángulo viene expresado en grados.

DOS (/expresión numérica/)

CREAL

Convierte un valor a Real.

CREAL (/expresión numérica/)

DEC\$

Genera una representación literal conformada de un numeral.

Para crear una representación literal del valor de la expresión dada en la función, de acuerdo con una determinada "plantilla de conformación".

DEC\$ (/expresión numérica/, /plantilla de conformación/)

La /plantilla de conformación/ es una /expresión literal/ que se usa a modo de "horma" para dar forma específica a la serie de dígitos resultantes de aplicar la función.

La /plantilla de conformación/ solo puede contener los siguientes caracteres:

+ - \$ * # , . ↑

EOF

Comprueba el final de fichero.

Para saber si al hacer la lectura final de un fichero de entrada en cassette se ha alcanzado la posición de final de fichero. La función da el valor -1 (cierto) si se está en el final del fichero. Da el valor 0 (falso) en todas las otras situaciones.

EOF

ERR

ERL

Código de Error y Línea de Error.

Estas funciones que pueden verse como variables del sistema y no usan argumentos, pueden usarse en las subrutinas para resarcimiento de errores, como medio de descubrir el código de error (**ERR**) y el número de línea (**ERL**) que se estaba ejecutando cuando se detectó o produjo el error.

EXP

Exponencial.

Calcular el resultado de elevar el número e a una potencia dada. El número e es el número cuyo logaritmo natural es el 1, y aproximadamente es 2.7182818.

Esta función también se conoce como antilogaritmo.

EXP (/expresión numérica/).

FIX

Da la parte “fija” de un número, haciéndolo entero.

FIX (/expresión numérica/)

FRE

Mide el espacio libre.

Examinar cuanta memoria libre permanece sin ser usada por el BASIC.

Existen dos formas de la función, y una de ellas hace que el BASIC efectúe una “recogida de basura” antes de decir el espacio libre.

FRE (/expresión numérica/)

FRE (/expresión literal/)

La función **FRE (0)** dice la cantidad de memoria que queda todavía por usar.

La función **FRE (””)** obliga al BASIC a efectuar una “recogida de basura” de manera que cualquier espacio desaprovechado dentro del área literal sale transferido al área disponible.

HEX\$

Dar la representación hexadecimal de un numeral.

HEX\$ (/expresión entera sin-signo/ [,/anchura campo/]).

La /anchura campo/ opcional es una /expresión entera/ que concreta la longitud mínima del literal hexadecimal que se produce, y debe ser un entero en la banda 0..16

HIMEM

Máxima dirección de memoria usable por el BASIC.

La cantidad de memoria usable por el BASIC puede alternarse mediante el comando **MEMORY**. La función intrínseca **HIMEM** dice cual es la máxima dirección que se usa en un programa concreto.

HIMEM

INKEY

Comprobar el estado de una determinada tecla.

INKEY puede usarse para probar si una tecla dada está pulsada o no, y si lo está, si además están pulsadas las teclas **/SHIFT/** Y **/CTRL/**.

INKEY (/número tecla/)

La función da como resultado un valor entero con el significado siguiente:

	Tecla	[SHIFT]	[CTRL]
-1	Quitada.	Desconocida	Desconocida
0	Pulsada.	Quitada	Quitada.
32	Pulsada.	Pulsada.	Quitada.
128	Pulsada.	Quitada.	Pulsada.
160	Pulsada.	Pulsada.	Pulsada.

INKEY\$

Recoger la tecla últimamente pulsada.

INKEY\$

INP

Introduce el valor depositado en un portal de entrada a la máquina.

INP (/número portal/)

INSTR

Buscar si un literal dado está incluido en otro.

INSTR ([/posición comienzo/], /donde busca/, /lo que busca/)

La /posición comienzo/ es una /expresión entera/ que fija la posición dentro del literal donde busca, a partir de la cual empieza a comprobar si aparece el literal que busca.

El literal /donde busca/ es una /expresión literal/ que señala la serie de caracteres que van a ser examinados.

El literal /lo que busca/ es una /expresión literal/ que señala la serie de caracteres cuya pertenencia al otro literal se trata de comprobar.

INT

Obtener el entero más próximo.

Para redondear un valor dado obteniendo el entero más próximo a él.

INT (/expresión numérica/)

JOY

Inspecciona el estado de un mando para juegos dado.

JOY (/número del mando/)

LEFT\$

Extraer la parte situada a la izquierda de un literal.

LEFT\$ (/expresión literal/, /longitud requerida/).

La /longitud requerida/ es una /expresión entera/ que especifica cuantos caracteres han de extraerse, y debe producir un valor en la banda 0..255.

LEN

Determina la longitud de un literal dado.

LEN (/expresión literal/)

LOG

Logaritmo natural.

En base e de un número.

LOG (/expresión numérica/)

LOG 10

Logaritmo decimal.

En base 10 de un número.

LOG10 (/expresión numérica/)

LOWERS\$

Convierte un literal a letras minúsculas.

LOWERS\$ (/expresión literal/)

MAX

Determina el valor máximo de una serie de ellos.

MAX (/lista de; /expresión numérica/)

MID\$

Sustituye parte de un literal por otro.

Devuelve la parte del medio de un literal.

MID\$ especifica parte de un literal (un sub-literal) que puede usarse tanto como destino de una asignación (**MID\$** como un comando), o como un argumento en una expresión literal (**MID\$** como una función).

MID\$ (/literal/, /posición comienzo/ [,/longitud sub- literal/])

Para **MID\$** como comando, el /literal/ debe ser un nombre de variable literal parte de cuyo contenido es el que va a ser alterado.

Para **MID\$** como Función, el /literal/ es una /expresión literal/, que será evaluada previamente y parte de la cual será devuelta como resultado de la función.

La /posición de comienzo/ es una /expresión entera/ que especifica la posición del carácter dentro del /literal/ que va a ser el primer carácter del sub-literal. La expresión debe producir un valor en la banda 1..255.

La /longitud sub-literal/ es una /expresión entera/ que especifica la longitud que ha de tener el sub-literal segregado. Si se omite, el sub-literal se extiende hasta el final del /literal/ original. La expresión debe producir un valor en la banda 1..255.

MIN

Determinar el valor mínimo de una serie de valores.

MIN /lista de; /expresión numérica/)

PEEK

Examina y dice lo que hay en una celdilla de memoria.

Para observar el contenido de un byte determinado de la memoria de la máquina.

PEEK (/expresión direccional/)

Resultado un valor en la banda 0..255, en base 10.

PI

Obtener el valor del número trascendente π

PI

Pi, es aproximadamente 3.1415926535898. La representación interna en máquina más cercana es aproximadamente 3.1415926534683.

POS

Posición horizontal del cursor en el cauce dado.

Para observar la posición horizontal que ocupa en ese momento el cursor correspondiente a un cauce de salida dado.

POS (//expresión cauce/)

1. Cauces de pantalla - Cauces #0-#7

POS devuelve la coordenada "X" de la posición actual del cursor.

2. Cauce de Impresora - Cauce #8

Devuelve la posición corriente del cabezal de la impresora. El borde de la parte izquierda de la impresora es la posición.

3. Cauce de Salida hasta el Cassette - Cauce#9

El BASIC mantiene la posición de un cursor lógico en los cauces de fichero contando todos los caracteres visivos enviados a través de dicho cauce desde el envío del último retorno de carro.

REMAIN

Obtiene la cuenta que queda en el temporizador de demora.

Para conocer el tiempo que queda en cada uno de los "cronómetros" posibles y al mismo tiempo desactivarlo.

REMAIN (/número temporizador/)

RIGHT\$

Extrae la parte de un literal situada a la derecha.

Para extraer una determinada cantidad de caracteres de un literal, tomándolos a partir del extremo derecho del mismo.

RIGHT\$ (/expresión literal/, /longitud requerida/)

RND

Número aleatorio.

Para conseguir un número aleatorio.

RND [(/expresión numérica/)]

ROUND

Redondeo de valores numéricos.

Para conseguir un dato numérico con un determinado número de posiciones fraccionarias, o elevando a una determinada potencia a diez.

ROUND (/expresión numérica/ [,/decimales/])

El parámetro opcional /decimales/ especifica la cantidad de posiciones decimales que han de considerarse en el redondeo.

El valor de la /expresión numérica/ se redondea hasta el número de posiciones fraccionarias especificadas por el parámetro /decimales/ en la forma siguiente:

- | | |
|----------------------------|---|
| /decimales/ mayor que cero | el valor se redondea hasta el número dado de dígitos después del punto decimal. |
| /decimales/ cero o ausente | el valor se redondea hasta un entero. |
| /decimales/ menor que cero | el valor se redondea hasta dar ABS (/decimales/) ceros antes del punto decimal. |

SGN

Signo de un valor numérico.

SGN (/expresión numérica/)

SGN da como resultado un valor Entero:

- 1 si la /expresión numérica/ es $\lt 0$
- 0 si la /expresión numérica/ es $= 0$
- +1 si la /expresión numérica/ es $\gt 0$

SIN

Seno de un ángulo.

En el modo Radianes, el ángulo se expresa en radianes. En el modo grados el ángulo debe expresarse en grados sexagesimales.

SIN (/expresión numérica/)

SPACE\$

Serie literal de espacios en blanco.

Para crear una constante literal formada por espacios en blanco y de una longitud dada.

SPACE\$ (/longitud/)

SQ

Comprueba el estado de la secuencia de notas sonoras.

Observar cuantos "sitios libres" hay todavía en la secuencia de notas a emitir por un canal dado, para comprobar si el canal está activo, y si no lo está porque la nota situada en la cabeza de secuencia está esperando a ser emitida.

SQ (/canal/)

El /canal/ es una /expresión entera/ que produce uno de los siguientes valores:

- 1: para el canal A
- 2: para el canal B
- 3: para el canal C

La función **SQ** entrega como resultado un valor Entero que está "calibrado en binario", de acuerdo con los siguientes campos:

Bits 0..2 el número de sitios libres en la secuencia, en la banda 0..4

Bits 3..5 el estado de Acorde en la cabeza de la secuencia (si lo hay)

Bit 6 está "alzado" si la cabeza de la secuencia está Retenida.

Bit 7 está activo en ese momento.

mútuamente excluyentes.

SQR

Raíz cuadrada.

SQR (/expresión numérica/)

STR\$

Representación literal de un valor numérico.

Para convertir un valor numérico dado en su representación literal -alfanumérica.

STR\$ (/expresión numérica/)

STRING\$

Literales con una serie de caracteres repetidos.

Para construir una constante literal formada por un determinado carácter que está repetido una cierta cantidad de veces.

STRING\$ (/longitud/, /especificador carácter/)

La */longitud/* es una */expresión entera/* que da la longitud requerida del literal resultante de la función.

El */especificador carácter/* puede ser uno de los dos siguientes:

/expresión entera/ que indica **CHR\$ (/expresión entera/)**

/expresión literal/ que indica el primer carácter, el inicial, del literal dado.

TAN

Tangente de un ángulo.

En el modo Radianes el valor del ángulo se expresa en radianes. En el modo Grados el valor del ángulo se expresa en ángulos sexagesimales.

TAN (/expresión numérica/)

TEST

Comprueba la Tinta usada en una posición de gráficos dada.

TEST puede usarse para examinar la tinta que se ha empleado en una determinada posición de la pantalla.

TEXT (/coordenada x/, /coordenada y/)

TESTR

Comprueba la tinta usada en una determinada posición de gráficos.

TESTR puede emplearse para saber el número de tinta usado en una determinada posición de la pantalla, que viene señalada por sus coordenadas relativas a la posición actual del cursor de gráficos.

TESTR (/desplazamiento X/, /desplazamiento y/)

TIME

Tiempo transcurrido.

El sistema mantiene la cuenta del tiempo transcurrido, desde que se puso en marcha la máquina.

TIME

El tiempo se controla en unidades de 1/300-avos de segundo.

UNT

Convierte un valor a Entero sin-signo.

Para hallar el valor entero equivalente a un valor numérico sin-signo dado.

UNT (/expresión direccional/)

UPPER\$

Convierte a mayúsculas los caracteres de un literal.

Para crear un nuevo literal que es copia de otro literal con todos los caracteres alfabéticos en minúsculas convertidos a sus equivalentes en mayúsculas.

UPPER\$ (/expresión literal/)

VAL

Convierte un literal a su valor numérico.

Para tomar la representación literal de un dato numérico y convertirlo a su valor de "índice" numérica igual.

VAL (/expresión literal/)

VPOS

Halla la posición Vertical actual.

Para establecer la posición vertical que corresponde actualmente al cursor de un cauce dado.

VPOS /expresión cauce/)

La /expresión cauce/ debe producir un valor coherente con un cauce de pantalla.

XPOS

Posición horizontal del cursor de gráficos.

Para conocer la posición X que ocupa actualmente el cursor de gráficos.

XPOS

YPOS

Posición vertical del cursor de gráficos.

Para conocer la posición Y que ocupa actualmente el cursor de gráficos.

YPOS

AMSTRAD

CPC



MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA



<https://acpc.me/>

[FRA] Ce document a été préservé numériquement à des fins éducatives et d'études, et non commerciales.

[ENG] This document has been digitally preserved for educational and study purposes, not for commercial purposes.

[ESP] Este documento se ha conservado digitalmente con fines educativos y de estudio, no con fines comerciales.