



DRAGON 200



DRAGON 200

MANUAL DE REFERENCIA

MANUAL DE REFERENCIA

DRAGON 200

Editado por EUROHARD, S. A.

Prohibida la reproducción, grabación
o transmisión por cualquier medio
de esta publicación,
sin permiso del editor.

Diseñado por Diréctica, S. A.
Teléfono 234 94 36

Impreso por Artes Gráficas GALA, S. L.
Miguel Yuste, 36 - 28037 Madrid

Depósito Legal M. 14.198-1985

Indice

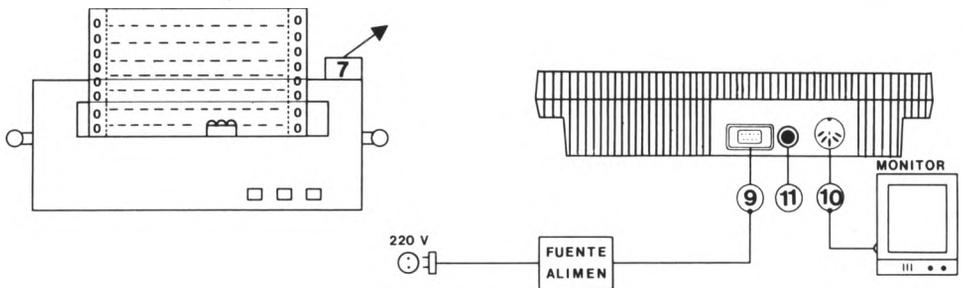
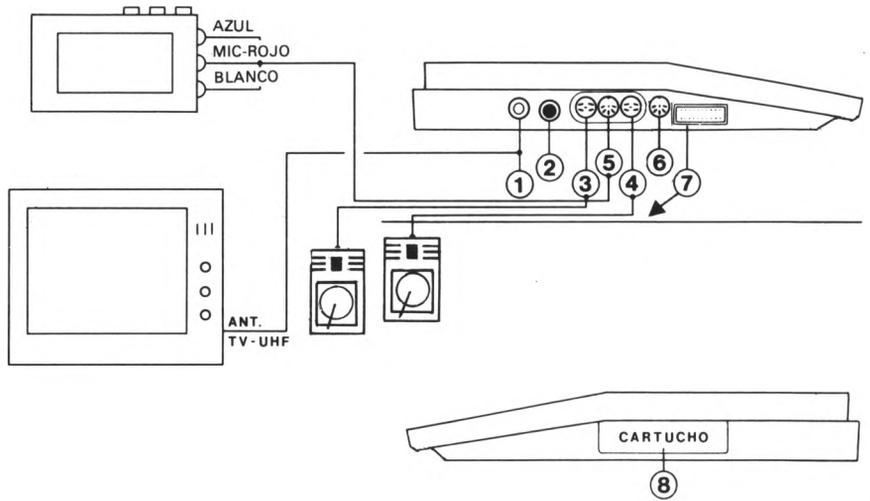
1. Advertencias Generales. 5
2. Conexiones. 7
3. Teclado. 11
4. Características del Dragón 2 0 0. 13
5. Comandos. 15
6. Sentencias Generales. 25
7. Funciones. 43
8. Gráficos y sonidos. 49
9. Códigos de error. 59

Apéndices

- A. Cassette. 63
- B. Esquemas. 65
- C. Código ASCII. 67
- D. Modos Gráficos. 71
- E. Mapas de Memoria. 83
- F. Retículas. 85

Advertencias Generales

1. Asegúrese de que no deja cables colgando en sitios peligrosos.
2. Antes de enchufar el transformador a la red compruebe que el Dragón está desconectado.
3. Asegúrese de que todas las conexiones están bien hechas antes de encender el ordenador.
4. Si utiliza unidad de discos compruebe que está conectada antes de encender el Dragón.
5. Nunca conecte o desconecte un cartucho con el DRAGON encendido.
6. Desconéctelo todo y apague el ordenador cuando no lo esté utilizando.
7. Mantenga todo tipo de líquidos lejos del ordenador.
8. Para limpiar la carcasa y el teclado, primero desconecte totalmente la unidad de la red. Con un trapo humedecido ligeramente frote la carcasa y el teclado. No utilice ningún producto con alcohol.
9. No abra nunca el Dragón ni intente desmontar el transformador.
10. No utilice un transformador distinto del que se suministra con su Dragón.
11. Es peligroso mantener las cassettes (o discos si tiene unidad) introducidos en el periférico. Acostúmbrese a guardarlos.
12. Si tiene algún problema consulte a su distribuidor.



1. **TOMA PARA TELEVISION**
Conexión para toma standard de antena de televisión.
2. **BOTON DE RESET**
Se utiliza para inicializar el sistema. Detiene inmediatamente la ejecución del programa, u operaciones de entrada/salida. Cualquier programa que resida en memoria permanecerá allí después de pulsar reset.
3. **TOMA PARA EL JOYSTICK IZQUIERDO**
4. **TOMA PARA EL JOYSTICK DERECHO**
Para el 3 y 4. Tomas Din de 5 pines utilizadas para conectar joysticks, accesorio opcional.
5. **TOMA PARA ENTRADA/SALIDA DE CASSETTE**
Toma DIN de 5 pines para conectar un cassette.

6. **SALIDA SERIE I/O RS232**
7. **PORT PARA IMPRESORA PARALELO**
Conexión para una impresora de tipo centronics por medio de un cable centronics standard.
8. **TOMA PARA CARTUCHOS DE PROGRAMAS**
Utilizado para conectar cartuchos de programas. Deben conectarse cuando el computador esté desconectado.
9. **TOMA DE ALIMENTACION**
Para conectar el cable de alimentación.
10. **TOMA PARA EL MONITOR**
Para conectar un monitor en color PAL.
11. **INTERRUPTOR DE ENCENDIDO.**

Conexiones

Para conectar el Dragón debe seguir los siguientes pasos:

Asegúrese que el botón de encendido núm. 11 está desconectado.

DESCONECTADO



CONECTADO



Conecte el transformador a la entrada del Dragón (núm. 8) y después el otro extremo a la red.

Si va a utilizar un T.V. en color busque en la caja el cable de antena y conéctelo al Dragón (núm. 1) y al T.V.

Si desea utilizar un Monitor PAL lo debe conectar a la salida núm. 10 mediante un cable adecuado. La descripción de esta salida se muestra en el apéndice B.

Encienda el T.V. y el Dragón pulsando el botón de encendido (núm. 11). Verá que se enciende el Led situado en la parte inferior del teclado.

Sintonice el canal 36 de U.H.F. y seleccione el color hasta que la imagen quede nítida.

En pantalla le habrá aparecido el copyright después O.K., aviso de que el ordenador está listo para trabajar.

Conexión del Cassette

En la caja de su ordenador encontrará un cable que un extremo es una clavija DIN de 5 pin y el otro son 3 jacks de colores rojo, blanco y azul.

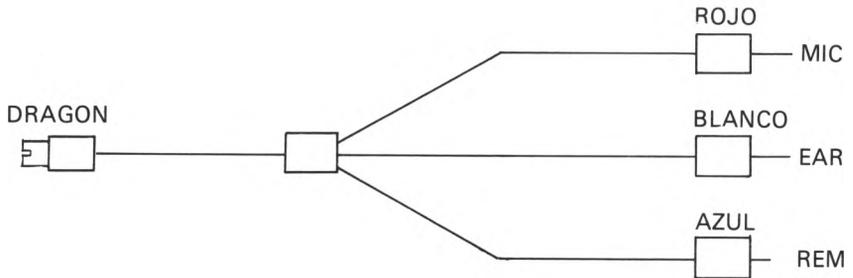
Este es el cable del cassette.

La clavija DIN de 5 pin se conecta al ordenador en la entrada núm. 5 y los tres jacks a cualquier cassette de los existentes en el mercado.

El jack rojo se conecta a la salida MIC (micrófono).

El jack blanco a la salida EAR o AUX (altavoz).

El jack azul a la salida REM (control remoto).



Si desea más información consulte el Apéndice A.

Conexión de los joystick

Los joystick son los mandos de juego que se utilizan en la mayoría de los programas de entretenimiento e incluso en algunas aplicaciones comerciales. Se pueden conectar dos joystick que se llaman joystick derecho e izquierdo a las salidas núms. 4 y 3 respectivamente.

A estas salidas a veces se conectan otro tipo de accesorios (como por ejemplo el lápiz óptico) ya que son en realidad un convertor analógico-digital.

Los esquemas de este interface y otros detalles técnicos se pueden encontrar en el apéndice B.

Conexión de la impresora

La impresora se conecta a la salida núm. 7 que es una puerta paralelo tipo centronics.

Los esquemas de esta puerta y los del cable necesarios se encuentran en el apéndice B.

También se puede utilizar una impresora serie utilizando el interface RS232 (núm. 6) según se explica en el apéndice B.

El interface serie RS232 también se puede utilizar para comunicaciones entre ordenadores por medio de los comandos DLOAD y DLOADM que se explican más adelante.

Conexión del cartucho

La entrada del cartucho (núm. 8) es la única que se encuentra en la parte derecha

del Dragón y en ella se conectan cartuchos de juegos o de aplicaciones, el controlador del disco, la grabadora Eproms, etc.

Los cartuchos siempre deben llevar la etiqueta hacia arriba.

El importante hacer dos observaciones:

- Nunca conecte o desconecte un cartucho con el ordenador encendido.
- Al pasar al modo de 48 K esta puerta se desconecta. (Ver capítulo 4.)

El teclado

El teclado del Dragón 200 es un teclado profesional tipo QWERTY con 53 teclas.

La disposición de los caracteres es el siguiente:



En las teclas que tienen dos caracteres los caracteres superiores se obtienen manteniendo pulsada la tecla SHIFT a la vez que la tecla correspondiente.

La tecla (ENTER) es el retorno de carro, la tecla (CLEAR) borra la pantalla y la tecla (BREAK) aborta la ejecución actual.

Las mayúsculas aparecen como la letra negra sobre fondo verde y las minúsculas en modo invertido, es decir, verdes sobre fondo negro. Para conectar la introducción de minúsculas pulse (SHIFT) y a la vez (Ø) y para desconectarlas la misma secuencia.

La tecla de borrado es la flecha a la izquierda: (←). Al pulsar esta tecla retrocederá una posición el curso borrando el carácter alcanzado. Si pulsa (SHIFT) y a la vez la tecla (←) borrará toda la línea introducida.

La flecha hacia arriba (↑) es el signo de potenciación. Las dos flechas restantes no tienen un código representable en pantalla pero sí captable por software.

La combinación (SHIFT) y @ a la vez detiene la ejecución del proceso que se esté realizando hasta que se pulse otra tecla.

Existen otros caracteres que no aparecen en el teclado y que se obtienen en combinación con la tecla (SHIFT) estos son:

SHIFT y (→) obtiene:]
SHIFT y (↑) obtiene: ←
SHIFT y (↓) obtiene: [

En el lado izquierdo del Dragón 2000 hay un botón que es el RESET o inicialización en caliente. Este botón inicializa el equipo sin dañar los programas existentes en memoria.

Características del Dragón 200

El Dragón 200 está diseñado de forma que admite dos tipos de configuración de memoria.

La primera configuración, llamada mapa tipo 0, es la que se tiene al encender el equipo. Es idéntica a la del Dragón 100 y por tanto compatible con todo el software existente para él.

En esta configuración el usuario dispone de 32K de RAM y 32K de ROM (16K para el intérprete BASIC y 16K para el cartucho. Se reconoce porque el cursor es de color negro.

Cuando se está en el tipo 0, la puerta del cartucho está conectada y cualquier cartucho funcionará.

Sin embargo en la dirección 48000, existe una subrutina que pasa al mapa tipo 1. Para acceder a ella teclee EXEC 48000.

Al pasar al tipo 1, el cursor se vuelve de color azul y el teclado pasa a ser autorepetitivo.

De todas formas al cambiar de modo no se pierden los programas que están en memoria.

En este modo, el intérprete de BASIC se coloca en la parte inferior de la memoria, en la zona que antes estaba reservada para el cartucho, devolviéndonos 16K de memoria RAM pero anulando el cartucho.

Este modo tiene la ventaja del teclado repetitivo es decir, si mantiene pulsada una tecla, ésta se repetirá hasta que usted deje de apretarla.

En la dirección \$11F existe un valor que controla los caracteres que repetirá por segundo y el tiempo de espera antes de empezar la autorepetición.

Por defecto su valor es 5 y esto supone 10 caracteres por segundo y 0,8 segundos de espera.

Esta ventaja se puede incorporar en el modo de 32K (mapa tipo 0) de la siguiente forma:

```
POKE & HFF03, (PEEK (& HFF03) AND & HFE)
POKE & H10D, & HBF
POKE & H10E, & H20
POKE & HFF03, (PEEK (& HFF03) OR 1)
```

Otra ventaja fundamental del modo 48K (mapa tipo 1) es que ahora el intérprete de BASIC se encuentra en RAM y si no lo utiliza puede borrarlo consiguiendo 64K libres.

Esta técnica se utiliza fundamentalmente cuando se usa un sistema operativo residente en disco.

El Dragón 200 posee sistemas operativos en disco como el OS9 o el FLEX, que se inicializan desde el modo de 32K por el comando BOOT del Dragondos.

Cuando se ha introducido un sistema operativo se pasa a mapa tipo 1, se anula el intérprete de BASIC y se mantiene la puerta del cartucho abierta para poder acceder al disco.

En esta situación tenemos 64K libres y se dice que el Dragón 200 está en modo 64K.

Otra característica importante del Dragón 200 es que incorpora un interface RS232 para comunicaciones tipo serie y los comandos DLOAD, DLOADM para su utilización.

El interface RS232 también puede utilizarse como interface normal de impresora en lugar de la interface habitual Centronics (paralelo). Para seleccionar la opción serie, deben utilizarse las siguientes órdenes:

```
POKE & H3FF, 1
```

y para seleccionar la opción paralelo (en ausencia)

```
POKE & H3FF, 0
```

Además del byte de selección de impresora hay otros dos bytes (&H3FD y &H3FE) que especifican un fin de línea cuyo espacio es 16 bit, puesto que lo requieren varias impresoras. La espera se puede incrementar en períodos de 10 milisegundos. Así:

```
POKE &H3FE, 50
```

dará una espera de medio segundo.

Una vez que se ha seleccionado la opción de impresora serie, las órdenes del BASIC, LLIST y PRINT # — 2 sacará caracteres a través de la interface RS232 en lugar de la interface Centronics.

Comandos

Comandos son las órdenes que se dan al ordenador para informarle de lo que se desea hacer con un programa: ejecutarlo, borrarlo, guardarlo en un cassette, etc...

Los comandos se suelen utilizar en modo directo, es decir sin número de línea, fuera de los programas porque rigen el proceso de los mismos.

De todas formas también se pueden utilizar en programas aunque algunos de ellos provocan que se pare la ejecución (por ejemplo EDIT).

Los comandos del Dragón 200 son:

– AUDIO	– DEL	– EXEC	– RENUM
– CLOAD	– CONT	– LIST	– RUN
– CLOADM	– DLOAD	– LLIST	– SKIPF
– CSAVE	– DLOADM	– MOTOR	– TRACE
– CSAVEM	– EDIT	– NEW	

AUDIO

El comando AUDIO controla la salida de sonido del cassette por el altavoz del televisor. AUDIO ON dirige la salida de cassette hacia la televisión. AUDIO OFF desconecta la comunicación.

Veamos un ejemplo:

Coloca una cinta de música en tu cassette e introduce:

AUDIO ON

Escriba ahora:

MOTOR ON

Ahora su Dragón se ha convertido en amplificador de su cassette y puede oír la música por el altavoz del televisor.

CLOAD

El comando CLOAD se utiliza para cargar en memoria los programas en BASIC que se han almacenado en cassette.

Tiene dos formatos: CLOAD carga el primer programa que encuentra en la cinta y CLOAD "nombre de programa" recorre la cinta hasta que encuentre un programa con el nombre especificado. El nombre tiene un máximo de 8 caracteres.

Cuando se introduce CLOAD en cualquiera de sus formas aparece en la esquina superior izquierda una S (abreviatura de Searching —buscando—) que significa que está buscando el programa pedido. Cuando encuentre un programa que no es el deseado aparecerá su nombre a continuación de la S y seguirá buscando.

En el momento en que encuentre el programa que debe cargar cambiará la S por una F y el nombre del programa y significa que está cargando ese programa.

CLOADM

El comando CLOADM funciona igual que el comando CLOAD sólo que carga programas que están en código máquina. Tiene cuatro formatos diferentes:

CLOADM: Carga el primer programa que encuentra en la posición de memoria original desde donde fue salvado (ver comando CSAVEM).

CLOADM "programa": Busca el programa especificado y cuando lo encuentra lo carga.

CLOADM "", n: Carga el primer programa que encuentra sumándole a la posición original de carga el incremento n.

CLOADM "programa", n: Igual que en la forma anterior pero buscando un programa específico.

Veamos con un poco más detalle este último parámetro:

Si se tiene un programa en código máquina que comienza en la dirección 200, acaba en la dirección 600 y su dirección de ejecución es 400, lo salvaremos con CSAVEM "EJEMPLO 1", 200, 600, 400.

Si lo cargamos con CLOADM o CLOADM "Ejemplo 1" se cargará en la dirección 200 hasta la 600 y pondrá el puntero de ejecución a la dirección 400 (ver comando EXEC).

Si se hubiese cargado con CLOADM "EJEMPLO 1", 100, entonces se coloca desde la dirección 300 hasta la 700 y deja el puntero de ejecución en la dirección 400.

Una observación importante es que si la dirección de carga más el incremento supera la última posición de memoria (65535) continúa en la posición 0.

Veamos un ejemplo. Consideremos el programa anterior y carguémoslo con CLOADM "", 65430.

Debería cargarse en la dirección $65430 + 200 = 65630$, pero ésta dirección supera la máxima de memoria entonces se carga en la dirección $65630 - 65536 = 94$.

Hagamos un pequeño ejercicio. Introduzca el siguiente programa:

```
10 PCLEAR 5
20 PMODE 3, 2
30 PCLS
40 SCREEN 1, 0
50 CIRCLE (128, 96), 90
60 PAINT (128, 96)
70 GOTO 70
```

Este programa nos dibuja un gráfico en las páginas 2, 3, 4, y 5 es decir desde la dirección 3072 hasta la dirección 9215.

Si queremos guardar este gráfico en una cinta escribiríamos: CSAVEM "GRAFICO", 3072, 9215, 3072.

Apague y encienda el ordenador. El gráfico se ha perdido. Si quiere recuperarlo escriba este programa:

```
10 PCLEAR 5
20 PMODE 3, 2
30 PCLS
40 SCREEN 1,0
50 CLOADM
60 GOTO 60
```

Si lo desea situar en las páginas 3 a 6 sustituya estas tres líneas:

```
10 PLEAR 6
20 PMODE 3, 3
50 CLOADM "", 1536 (cada página de gráficos ocupa 1536 bytes)
```

Si lo desea en las páginas 1 a 4 sustituya:

```
10 PMODE 3, 1
20 PCLEAR 4
50 CLOADM "", 64000
```

CSAVE

El comando CSAVE graba en el cassette el programa en Basic que en ese momento se encuentre en memoria. Posee dos formatos:

- CSAVE "nombre" donde "nombre" es el nombre que queremos dar al programa y tiene un máximo de 8 caracteres. En este formato el programa se graba en el código interno del ordenador.
- CSAVE "nombre", A. El programa con esta opción se graba en formato ASCII, es decir, de la misma forma que se visualiza por pantalla cuando tecleamos LIST. De esta forma la grabación (y también la lectura) es más larga pero tenemos la ventaja de poder utilizar el programa como si fuera un fichero de datos (ver sentencia Input # — 1).

No se olvide de pulsar Play y RECORD en su cassette para utilizar este comando.

CSAVEM

Graba en una cinta de cassette un programa en código máquina o un trozo de memoria.

Su formato es CSAVEM "nombre", principio, fin, ejecución.

Donde "nombre" tiene un máximo de 8 caracteres, 'principio' es la dirección donde se empieza a salvar, 'fin' la dirección donde se debe acabar y 'ejecución' la dirección de ejecución del programa. Si no es ejecutable se puede poner cualquier dirección (normalmente 0).

CONT

Este comando continúa la ejecución de un programa que ha sido parado por una sentencia STOP o por la tecla BREAK.

DEL

El comando DEL se utiliza para borrar líneas determinadas del programa que está actualmente en memoria.

DEL número de línea 1 — número de línea 2

El comando borrará todas las líneas desde la línea 1 hasta la línea 2 ambas inclusive. Ambos números de líneas son opcionales y el comando se puede utilizar en cualquiera de las siguientes formas:

DEL — Borra el programa completo

DEL — 100 Borra desde el principio hasta la línea 100 inclusive

DEL 300 — Borra desde la línea 300 inclusive hasta el final

DEL 40 Borra la línea 40 exclusivamente

DEL 100 — 200 Borra todas las líneas entre la 100 y 200 ambas inclusive

DLOAD, DLOADM

Una interface en serie RS232 está incluido en el Dragón 200 pudiendo ser utilizado en los modos 32K y 64K. Esta interface se utiliza con las siguientes sentencias BASIC.

DLOAD "nombre del fichero", (relación de baudios seleccionado).

DLOADM "nombre del fichero", (relación de baudios seleccionado), (carga).

DLOAD y DLOAM cargan programas BASIC con formato ASCII y programas en código-máquina, respectivamente, desde un ordenador residente como sus equivalentes en cassette, CLOAD y CLOADM, los parámetros de estas órdenes son opcionales.

El valor de la (selección de relación de baudios), como sugiere su nombre, se utiliza para seleccionar la relación de baudios para la comunicación entre ordenadores. Los valores admisibles para este parámetro son:

Valor de la relación de baudios seleccionada	Baudios resultantes
0	110
1	300
2	600
3	1.200
4	2.400
5	4.800
6	9.600

Si no se indica este parámetro utiliza el valor anteriormente indicado y si no hay un precedente toma 1.200 baudios.

Los baudios resultantes de la interface RS232 pueden alterarse con una orden POKE en BASIC. Por ejemplo:

POKE & HFF07, (PEEK (&HFF07) AND & HF0) OR B

Donde la variable B contienen el valor que especifica una de las tres relaciones de baudios soportadas por el aparato.

B	Baudios resultantes
1	50
2	75
3	110
4	135
5	150
6	300
7	600
8	1.200
9	1.800
10	2.400
11	3.600
12	4.800
13	7.200
14	9.600

Los datos son transmitidos y recibidos en serie con 1 bit de inicio, 8 bits de datos,

2 bits de parada (stop) y sin bits de paridad, por eso los instrumentos conectados a esta interface, como impresoras en serie, pueden conformar esta especificación.

EDIT

El comando EDIT se usa para modificar el contenido de una línea determinada. Su formato es:

EDIT número de línea

Si no existe ese número de línea aparecerá UL? ERROR.

Una vez en el modo de edición se puede utilizar cualquiera de los siguientes comandos.

L	Lista el contenido actual de la línea.
C carácter	Cambia el carácter en la posición del cursores por el carácter introducido.
n C caracteres	Cambia los siguientes n caracteres por los nuevos.
I	Inserta caracteres en la posición del cursor.
D	Borra el carácter debajo del cursor.
n D	Borra los próximos n caracteres.
H	Borra el resto de la línea, desde la posición actual.
X	Se posiciona al final de la línea y se sitúa en modo de inserción.
S carácter	Busca la primera aparición del carácter a partir de la posición del cursor.
K caracter	Borra la línea desde el cursor hasta la primera aparición del carácter.
n K carácter	Borra la línea hasta la n — sima aparición del carácter.
n [ESPACIO]	Avanza el cursor n espacios. Si se omite n avanza 1.
n [←]	Retrocede el cursor n posiciones. Si se omite n se retrocede 1.
[SHIFT] [↑]	Deja el modo de inserción y vuelve al modo de edición.
[ENTER]	Deja el editor y almacena la línea.
A	Anula todas las modificaciones hechas hasta el momento y, continúa en modo de edición.
E	Igual que ENTER.
Q	Anula todas las modificaciones y sale del modo de edición.

EXEC

El comando EXEC ejecuta un programa escrito en lenguaje máquina.

Su formato es: EXEC dirección, donde 'dirección' es la dirección donde debe empezarse a ejecutar el programa y tiene un rango entre 0 y 65535.

Un caso especial es cuando se escribe solamente EXEC sin especificar la dirección. Entonces se asume la siguiente:

- Si no ha cargado ningún programa en lenguaje máquina de cinta ni de cassette ni ha ejecutado ningún programa en código máquina se asume EXEC 48000 que es la subrutina de pasar a modo de 48K.
- En otro caso, se asume la dirección de ejecución del último programa en lenguaje máquina utilizado (tanto cargado de disco o de cinta como salvado o ejecutado).

EXEC 300 ejecuta el programa en la dirección 300.

LIST

El comando LIST muestra en pantalla, el programa que se encuentra en memoria. No puede ser utilizado en programas.

Si el programa es demasiado largo para entrar en la pantalla, se puede detener el listado pulsando la tecla [SHIFT] y la tecla [@] simultáneamente (pero debe ser rápido). Se puede continuar el listado pulsando cualquier otra tecla.

Para listar sólo una parte del programa puede usar

LIST N₁ — N₂

donde N₁ y N₂ son dos números de línea (N₂ deberá ser mayor que N₁)

LIST 40 — 100

presentará las líneas de programa comprendidas entre las líneas 40 y 100 ambas inclusive.

LIST — 80

mostrará todas las líneas de programa desde el comienzo hasta la línea número 80 inclusive

LIST 120 —

mostrará todas las líneas de programa comprendidas entre la línea número 120 y el final de éste incluida la línea 120.

LLIST

El comando LLIST funciona igual que el comando LIST pero dirige la salida a la impresora.

LLIST Lista el programa completo por impresora.

LLIST 40—200 Lista desde la línea 40 a la línea 200 ambas inclusive por impresora.

MOTOR

El comando MOTOR controla el funcionamiento del motor del cassette.

MOTOR ON arranca el motor. MOTOR OFF detiene el motor.

El botón play del cassette deberá estar pulsado para que el comando funcione.

NEW

El comando NEW borra la memoria y coloca todas las variables a cero.

No se suele utilizar en programas.

Es una buena costumbre ejecutar NEW antes de introducir un programa, para asegurarse de que no quedan restos del programa anterior, que puedan interferir en el programa actual.

RENUM

El comando RENUM permite cambiar parte o todos los números de línea del programa. El comando RENUM también cambia los números de línea en sentencias de bifurcación (GOTO, etc.) para asegurar que el flujo del programa discurre de la misma forma.

RENUM nueva línea, línea de comienzo, incremento.

El comando renumerará todas las líneas desde la línea de comienzo, empezando por el número «nueva línea» y el incremento indicado.

Todos los parámetros son opcionales y la sentencia se puede usar en cualquier de las siguientes formas:

RENUM	Renumerar todo el programa. Las líneas se numerarán 10, 20, 30...
RENUM 100	Renumerar todo el programa. Las líneas se numerarán 100, 110, 120...
RENUM 100, 50, 5	Renumerar comenzando en la antigua línea 5. Las líneas se numerarán 100, 105, 110...
RENUM, 20	Renumerar el programa completo. Las líneas se numerarán 10, 30, 50...

Fíjese en que si se omite un parámetro y el siguiente debe ser utilizado, deberá de escribirse la coma. No se puede utilizar RENUM para alterar la secuencia de las líneas.

RUN

El comando RUN se utiliza para comenzar a ejecutar un programa.

No se suele utilizar como sentencia.

Si desea comenzar un programa desde un punto distinto del principio, puede hacerlo escribiendo.

RUN número de línea.

Donde «número de línea» es el punto donde se desea que comience la ejecución.

RUN 250

Si sale UL? ERROR significa que no existe el número de línea especificado.

SKIPF

El comando SKIPF se utiliza para situar la cinta de cassette detrás de un programa concreto.

Tiene dos formatos: SKIPF que avanza la cinta hasta sobrepasar el primer fichero que encuentre (puede ser un programa en BASIC, en lenguaje máquina o un fichero de datos).

SKIPF "nombre de programa" avanza hasta sobrepasar el nombre de programa especificado.

Este comando sólo avanza la cinta y no modifica ninguno de los programas en memoria.

TRACE

Esta función permite seguir el flujo de un programa. A medida que se alcanza una línea se imprime su número en pantalla. La función trace se debe activar antes de ejecutar un programa.

TRON activa el seguimiento (trace)

TROFF desactiva el seguimiento.

Ambos son comandos directos que no necesitan número de línea pero pueden ser utilizados en programas.

Sentencias

Sentencias son las instrucciones del ordenador que normalmente se utilizan desde un programa aunque se pueden utilizar en modo directo. Cuando se utiliza una sentencia en modo directo que no se admite se produce ID? ERROR.

Vamos a ver las sentencias generales que son comunes a la mayoría de los dialectos del BASIC. Estos son:

CLEAR	GOTO	PRINT
CLOSE	IF /THEN	PRINT @
CLS	INPUT	PRINT #
Constantes	INPUT #	PRINT USING
DATA	LINE INPUT	PRINT # USING
DEF FN	ON /GOSUB	READ
DEF USR	ON /GOTO	REM
DIM	OPEN	RESTORE
END	Operadores aritméticos	RETURN
FOR /NEXT	Operaciones de relación	STOP
GOSUB	POKE	Variables

CLEAR

El comando CLEAR borra todas las variables, los dimensionamientos y reserva espacio para el almacenamiento de cadenas y subrutinas en lenguaje máquina.

```
CLEAR 500
```

Reservará 500 bytes para el almacenamiento de variables de cadena.

El comando CLEAR se puede usar también para fijar la dirección mayor de memoria BASIC, con objeto de reservar espacio para rutinas en lenguaje máquina.

```
CLEAR 200, 14000
```

reservará 200 bytes para el almacenamiento de cadenas y fijará la dirección superior

para BASIC en 14000. Ahora se podrán almacenar rutinas en lenguaje máquina a partir de la dirección 14001.

Si no se utiliza CLEAR, se reservan automáticamente 200 bytes de espacio para cadenas.

CLOSE

Cierra un fichero de datos en cassette. Su formato es CLOSE # — 1.

Si utiliza unidad de discos consulte el manual. Esta sentencia varía.

CLS

La sentencia CLS se utiliza para limpiar la pantalla, y ajustar el color del fondo. El color normal del fondo es el verde; si utiliza CLS sin parámetro, éste será el color utilizado.

Para cambiar el color del fondo agregue un número entre 0 y 8 después del comando CLS (CLS 2), o una variable numérica (CLS C).

Los colores disponibles son:

0—Negro	1—Verde	2—Amarillo
3—Azul	4—Rojo	5—Beige
6—Turquesa	7—Magenta	8—Naranja

El matiz de estos colores dependerá de su televisor.

Notará de todos modos, que independientemente del color del fondo, el ordenador presentará el texto en negro y verde.

```
10 CLS
20 PRINT "DEMOSTRACION DE COLORES DE FONDO"
30 PRINT "INTRODUZCA UN NUMERO ENTRE 0 y 8"
40 INPUT C
50 CLS C
60 PRINT "ESTE ES EL COLOR DE FONDO:"; C
70 PRINT "PULSE ENTER"
80 INPUT A$
90 GOTO 10
```

Constantes

Existen dos tipos de constantes: las numéricas y las alfanuméricas.

Las constantes alfanuméricas van siempre entre comillas (ver sentencia DATA) y la longitud máxima es de 256 caracteres.

“HOLA. QUE TAL ESTAS?”

Las constantes numéricas se pueden escribir de diferentes formas:

- Decimal. Es la forma usual de trabajar la coma decimal se representa por el punto. Si la parte entera es cero se puede suprimir.

1.3, 2.4, 5, .78 (=0.78).

- Hexadecimal. Representan números en hexadecimal y se reconocen porque van precedidos de &H.

&HFF = 255 en decimal

&HC00 = 3072 en decimal.

- Octal. representan números en octal y se reconocen porque van precedidos de &O.

&O17 = 15 en decimal

&O230 = 152 en decimal.

- Exponencial. Cuando tenemos un número demasiado grande o demasiado pequeño, como el ordenador sólo puede trabajar con 9 cifras significativas nos representa el número en formato exponencial: n E m, donde n y m son números y E es una constante fija.

n representa el número y se llama mantisa y m la potencia de 10 por la que hay que multiplicar a n se llama exponente.

$1.347 E-2 = 1.347 \times 10^{-2} = 0.01347$

$23.4 E5 = 23.4 \times 10^5 = 2340000$

DATA

La sentencia DATA sirve para almacenar datos dentro de un programa.

Su formato es Data dato1, dato2, dato3...

Los datos pueden ser numéricos o alfanuméricos y es la única sentencia que permite que las constantes alfanuméricas no vayan entre comillas.

Sólo es obligatorio poner las cadenas entre comillas en dos situaciones:

1. Cuando la cadena lleve espacios delante.
2. Cuando la cadena contenga alguna coma (,). (Consultar sentencia READ.)

DEF FN

La sentencia DEF FN se utiliza para definir una función numérica.

DEF FN nombre (variable objeto) = fórmula

El nombre puede ser cualquier letra de la A a la Z.

La variable objeto puede ser cualquier letra, y será reemplazada por el argumento cuando se utilice la función.

La fórmula describe la operación en términos de la variable objeto y/o otras variables.

Las funciones definidas por el usuario no pueden ocupar más de una línea. Una función puede utilizar otras funciones (definidas o del sistema), en la fórmula, pero no deberá llamarse a sí misma.

Una función deberá haberse definido antes de utilizarla, por ello convendrá definirlas al comienzo del programa.

```
10 REM representación gráfica de la función seno
20 PMODE 4, 1
30 PCLS 1
40 SCREEN 1, 1
50 COLOR 0,1
60 DEF FNA(X) = (SIN(X*3.14159/128) + 1)*96
70 LINE (0,0) - (0, 192), PSET
80 LINE (256, 96) - (0, 96), PSET
90 FOR I=0 TO 256
100 LINEA - (I,FNA (I)), PSET
110 NEXT I
120 GOTO 120.
```

DEF USRn — dirección

Se utiliza para definir la dirección de una función USRn n está entre 0 y 9 debe corresponderse con la n en USR. La dirección deberá estar entre 0 y 65535 y contener la dirección de entrada para USRn.

DIM

La sentencia DIM dimensiona matrices. Las matrices pueden ser de una o dos dimensiones y pueden ser numéricas o de cadena. Las reglas para nombrar matrices

son las mismas que para variables.

DIM nombre de matriz (n), nombre de matriz (n,n).

Si la dimensión máxima de la matriz no excede de 10, no es necesario dimensionarla con DIM.

Para referirse a un elemento de la matriz, se debe poner el número del elemento entre paréntesis después del nombre.

A (14,K), N9 (B), L\$ (14,4)

END

La sentencia END termina la ejecución de un programa y devuelve el control al teclado.

FOR... NEXT... STEP

La sentencia FOR... NEXT consta de dos partes:

FOR variable numérica = expresión TO expresión
STEP expresión

Y,

NEXT variable numérica

Las sentencias FOR ... NEXT trabajan conjuntamente para controlar el número de veces que es ejecutada una sección de programa.

Las expresiones se evalúan y el bucle va desde el valor de la primera expresión hasta el valor de la segunda en incrementos definidos por la tercera. El valor actual del contador se mantiene en la variable numérica.

Un bucle se realizará por lo menos una vez, aunque el rango y el incremento sean imposibles. Si se omite la parte STEP de la sentencia, se toma 1 por defecto.

Los bucles se pueden anidar dentro de otros, pero debe hacerse en orden correcto. En este caso, la sentencia NEXT admite la forma NEXT variable numérica, variable numérica, ...

Puede saltar fuera de un bucle utilizando una sentencia GOTO... pero no puede saltar desde fuera al interior de un bucle.

```
10 CLS: CLEAR: DIM L(1000)
20 INPUT "INTRODUZCA UN NUMERO"; N: IF N > 1000 or N < 2 THEN 20
30 CLS: PRINT "NUMEROS PRIMOS ENTRE 2 Y", N
```

```

40 PRINT: FOR I = 2 TO N: IF L(I) > 0 THEN 80
50 PRINT I;
60 IF I > SQR (N) THEN 80
70 FOR J = I TO N STEP 1: L(J) = 1:NEXT J
80 NEXT I

```

GOSUB

La sentencia GOSUB transfiere el control del programa al comienzo de una subrutina. RETURN devuelve de nuevo el control a la línea que viene después de la sentencia GOSUB.

GOSUB viene seguido por un número de línea, que es la primera de la subrutina.

```
GOSUB 1600
```

Una subrutina deberá contener por lo menos una sentencia RETURN.

GOTO

La sentencia GOTO tiene el formato

```
GOTO número de línea
```

El número de línea debe de ser un número (no una variable), y debe existir en algún lugar del programa. Si no se encuentra el número de línea se detendrá el programa con un mensaje ?UL ERROR (línea indefinida).

La sentencia GOTO se ejecuta inmediatamente, por ello, no se deberá poner ninguna otra sentencia en la misma línea después de GOTO, el programa nunca pasará por ellas.

```

20 CLS
30 GOTO 60
40 PRINT "EN LA LINEA 40"
50 GOTO 80
60 PRINT "EN LA LINEA 60"
70 GOTO 40
80 PRINT "FIN DEL PROGRAMA"

```

IF... THEN... ELSE

El formato completo de la sentencia IF es,

```
IF condición THEN acción 1 ELSE acción 2
```

La sentencia evalúa la condición que puede ser CIERTA o FALTA. Si es CIERTA se realizará la acción 1, si es FALSA se realizará la acción 2.

Una condición está formada por una expresión, una relación y otra expresión. Las expresiones pueden ser cualquier expresión en BASIC del mismo tipo (es decir, ambas numéricas o de cadena). El operador de relación puede ser cualquiera de los siguientes:

=	Igual a	<	>	Distinto de
>	Mayor que	<		Menor que
>=	Mayor o igual que	<=		Menor o igual que

Las condiciones también se pueden combinar mediante los operadores lógicos AND, OR, NOT.

“Condición” AND “condición” será sólo cierta, si ambas condiciones lo son.

“Condición” OR “condición” es cierta si lo son cualquiera de las dos condiciones.

NOT “condición” es cierta si la condición es falsa.

Acción 1 y acción 2 pueden ser cualquier sentencia BASIC, incluyendo otra sentencia IF.

La parte ELSE del comando es opcional y se puede omitir, en cuyo caso el programa continuará en la línea siguiente si no se cumple la condición.

```
10 CLS: PRINT "JUEGO DE ADIVINANZAS": N = RND (100):T = 0
20 INPUT "ADIVINA MI NUMERO"; G:T = T + 1
30 IF G = N THEN PRINT "LO ADIVINASTE EN"; T; "INTENTOS":END
40 IF G>N THEN PRINT "NO, ES MENOR" ELSE PRINT "NO, ES MAYOR"
50 GOTO 20
```

La condición puede reducirse a una variable o expresión sin signos de relación. Entonces es cierto si es distinto de cero y sólo es falso cuando toma el valor 0.

INPUT

Cuando un programa llega a una sentencia INPUT, se detiene su ejecución, y espera a que se introduzca algo por el teclado. Después del comando INPUT debe haber uno o más nombres de variable separados por comas.

```
25 INPUT A, B, F$, H7
```

La sentencia anterior requiere que se introduzcan 4 elementos. Esto se puede realizar uno a uno pulsando la tecla (ENTER) después de cada uno, o como una lista separada por comas, es decir

```
146.2, 78.1, CADENA, 3 (ENTER)
```

Es siempre una buena idea presentar un mensaje antes de una sentencia INPUT, para recordarle lo que debe introducir. Esto se puede realizar con una sentencia

PRINT, o incluirlo en una sentencia INPUT como a continuación:

```
35 INPUT "DOS NUMEROS POR FAVOR"; N1, N2
```

Fíjese en el punto y coma que separa la cadena de la lista de elementos.

Debe asegurarse de que introduce el tipo de dato correcto (cadenas para variables de cadena, números para variables numéricas), o de lo contrario el programa se detendrá dando el mensaje ?REDO, entonces tendrá que volver a introducir los datos.

No es necesario incluir las comillas en una cadena cuando se introduce, tanto "CADENA" como CADENA son válidas.

En el siguiente programa fíjese que:

- a) Las líneas 100 v 170. Utilizan el comando INPUT para detener el programa hasta que esté preparado. A\$ no tendrá nada.
- b) El uso de las variables de cadena C\$, P\$ y T\$ para evitar el tener que escribir el mismo mensaje varias veces.

```
10 CLS:T$ = "DEMOSTRACION DE ENTRADAS"
20 P$ = "PULSE (ENTER) PARA CONTINUAR"
30 C$ = "INTRODUZCA 4 NUMEROS"
40 PRINT: PRINT T$:PRINT
50 PRINT C$; "PULSANDO"
60 PRINT "(ENTER) DESPUES DE CADA UNO"
70 INPUT A, B, C, D: PRINT
80 PRINT: PRINT "HA INTRODUCIDO ESTOS VALORES"
90 PRINT A; B; C; D: PRINT
100 PRINT P$: INPUT A$: CLS
110 PRINT: PRINT T$: PRINT
120 PRINT "AHORA, "; C$; " SEPARADOS"
130 PRINT "POR COMAS Y PULSE LA TECLA (ENTER)"
140 INPUT A, B, C, D
150 PRINT: PRINT "ESTA VEZ LOS NUMEROS HAN SIDO"
160 PRINT A; B; C; D
170 PRINT: PRINT P$: INPUT A$: CLS
180 PRINT: PRINT T$: PRINT
190 INPUT "ESCRIBA UNA CADENA Y UN NUMERO, PULSANDO (ENTER) DES-
    PUES DE CADA UNO"; B$, N
200 PRINT: PRINT "LA CADENA QUE HA INTRODUCIDO ES"
210 PRINT B$
220 PRINT: PRINT "Y EL NUMERO ES:"; N
230 PRINT: PRINT "FIN DE LA DEMOSTRACION"
```

INPUT # -1

Esta sentencia es igual a la sentencia INPUT la única diferencia estriba en el lugar de donde provienen los datos.

Los datos en INPUT provenían del teclado y con INPUT # -1 los datos proceden de un fichero almacenado en cassette y que ha sido abierto con la sentencia OPEN.

Los datos se recuperan de la misma forma y en el mismo orden en que fueron grabados.

```
10 OPEN "I", # -1, "FICHERO1"  
20 INPUT # -1, A$, B$
```

Lee A\$ y B\$ del fichero FICHERO1 que se encuentra en el cassette.

Para utilizar esta sentencia debe estar pulsada la tecla PLAY y el control remoto conectado.

El número -1 es el código del cassette y el 0 el código del teclado, así INPUT #0, A\$ es la misma sentencia que INPUT A\$

```
10 INPUT "Donde están los datos (C/T)"; A$  
20 IF A$="C" THEN A=-1 ELSE A=0  
30 OPEN "I", # A, "FICHERO"  
40 INPUT # A, B$  
50 CLOSE #A  
60 PRINT "EL DATO ERA: "; A$  
70 END
```

Su formato es INPUT # expresión, lista de variables donde la expresión es un número, una variable o una expresión pero sólo puede valer 0 ó 1.

LINE INPUT

La sentencia LINE INPUT introduce una línea completa en una variable de cadena, incluyendo comas y espacios en blanco, los cuales no son aceptados por la sentencia INPUT.

LINE INPUT "mensaje"; variable de cadena.

El mensaje puede ser cualquiera que vaya entre comillas. Es opcional, y si se incluye deberá separarse de la variable de cadena por un punto y coma. Sólo puede aparecer una variable de cadena en la sentencia LINE INPUT. La longitud máxima de la línea almacenada por esta sentencia es de 255 caracteres.

```
10 CLEAR 500: CLS  
20 LINE INPUT "INTRODUZCA SU NOMBRE COMPLETO"; N$  
30 LINE INPUT "Y SU DIRECCION"; D$  
40 CLS: PRINT "NOMBRE: "; N$  
50 PRINT "DIRECCION: "; D$  
60 END
```

ON ... GOSUB

La sentencia ON... GOSUB permite una ramificación condicional a subrutinas de forma parecida al comando ON... GOTO.

ON expresión GOSUB lista de números de línea.

Si el valor de la expresión es negativa el programa se detendrá dando un mensaje de error, si es cero o mayor que el número de elementos en la lista de números de línea, se ignorará la sentencia y el programa continuará en la siguiente línea.

```
10 CLS: INPUT "INTRODUZCA UN PAR DE NUMEROS"; A, B
20 INPUT "AHORA INTRODUZCA UN NUMERO ENTRE 1 Y 4"; C
30 ON C GOSUB 100, 200, 300, 400
40 IF C >= 1 AND C <= 4 THEN END
50 PRINT C; "NO ESTA ENTRE 1 Y 4": GOTO 20
100 PRINT "LA SUMA DE"; A "MAS"; B; "ES"; A+B
110 RETURN
200 PRINT "LA DIFERENCIA DE"; A; "MENOS"; B; "ES"; A - B
210 RETURN
300 PRINT "EL PRODUCTO DE"; A; "POR"; B; "ES"; A*B
310 RETURN
400 PRINT "EL COCIENTE ENTRE"; A; "Y"; B; "ES"; A/B
410 RETURN.
```

ON... GOTO

La sentencia ON GOTO provee un método de ramificación múltiple a los números de línea especificados.

ON expresión GOTO lista de números de línea.

Se evalúa la expresión (y se trunca si no da como resultado un entero). El programa después salta a la línea cuya posición en la lista tenga un valor igual al valor de la expresión. Si el valor de la expresión es 4, el comando ON GOTO seleccionará el cuarto elemento de la lista de números de línea. El valor de la expresión no debe ser negativo, ya que se producirá un error. Si el valor de la expresión es cero o mayor que el número de líneas en la lista, se ignorará la sentencia y el programa continuará en la línea siguiente.

```
10 CLS: PRINT "SOLUCION DE LA ECUACION CUADRATICA"
20 INPUT A, B, C:IF A=0 THEN 20
30 R = -B/(2*A):D=B*B-4*A*C:S = SGN (D)
40 P = SQR (D*S)/(2*A)
50 ON S + 2 GOTO 80,60,70
60 PRINT "RAICES PERFECTAS":PRINT R,R:END
70 PRINT "RAICES REALES": PRINT R + P,R - P:END
80 PRINT "RAICES COMPLEJAS":PRINT R, P:PRINT R, -P:END
```

OPEN

Esta sentencia se utiliza para especificar el periférico que vamos a utilizar y si lo vamos a utilizar como entrada o como salida.

Su formato es

OPEN cadena 1, # expresión, cadena2

Cadena 1 puede ser una variable alfanumérica o una constante. Sólo admite dos valores:

“O” se especifica el periférico como salida

“I” se especifica el periférico como entrada

Expresión indica el periférico y sólo puede tomar tres valores:

- Ø indica la consola
- 1 indica el cassette
- 2 indica la impresora

Cualquier otro valor fuera de este rango provocará DN? ERROR.

Cadena 2 indica el nombre que se asigna al fichero y sólo es necesario cuando se utiliza el cassette.

La impresora sólo se puede utilizar como periférico de salida. Si se abre como periférico de entrada ocurrirá FM? ERROR.

Si la consola se abre como entrada se asume el teclado y si se abre como salida se asume el televisor o monitor.

Operadores

El Dragón posee los siguientes operadores:

+	Suma	=	Igual
-	Resta	> =	Mayor o igual
*	Multiplicación	< =	Menor o igual
/	División	< >	Distinto
↑	Potencia	AND	Y lógico
>	Mayor	OR	O lógico
<	Menor	NOT	Negación lógica

Todos estos operadores se pueden utilizar en expresiones combinados con paréntesis.

La jerarquía de operadores es la siguiente:

La expresión se evalúa de izquierda a derecha como sigue:

En primer lugar los paréntesis y las operaciones, según la siguiente escala

- 1.º) ↑
- 2.º) * y /
- 3.º) + y -
- 4.º) <, >, =, <=, >=,
- 5.º) AND, OR y NOT

Las expresiones con <, >, =, <=, >=, <> devuelven 0 si es falsa y -1 si son verdaderas

Ejemplo:

$$X = 3 * 5 > (264 \text{ OR } 25)$$

Primero se efectúa el paréntesis

$$(264 \text{ OR } 25) = 281$$

$$\text{Queda así: } 3 * 5 \uparrow 2 > 281$$

A continuación se evalúa la potencia, $5 \uparrow 2 = 25$

$$3 * 25 > 281$$

luego la multiplicación $3 * 25 = 75$

$$75 > 281$$

y por último el operador > quedando cero de resultado por ser falso.

En consecuencia $X = 0$

POKE

La sentencia POKE se utiliza para colocar un número en un lugar específico de la memoria.

POKE dirección, número

La dirección es un valor 0 y 65535 y número debe estar en el rango 0 y 255.

PRINT

La sentencia PRINT se utiliza para sacar algo por pantalla.

Se puede utilizar para mostrar constantes, el valor de una variable, cadenas y también evalúa expresiones.

Si se incluye más de un elemento en una sentencia PRINT, estos deberán ir separados por una coma (,) o un punto y coma (;).

Una coma provocará que se imprima la salida en dos columnas, cada quince caracteres. (Si la longitud del primer elemento es mayor de quince caracteres se imprimirá completo. El siguiente elemento aparecerá en la línea siguiente).

El punto y coma hace que se comprima la salida. Las cadenas se imprimirán una junto a otra, y los números tendrán un espacio a cada lado. El punto y coma mantiene la posición después de la última impresión, para continuar en la próxima sentencia PRINT.

Una sentencia PRINT sin elementos imprime una línea en blanco.

```
10 CLS
20 PRINT
30 PRINT"    UNA DEMOSTRACION"
40 PRINT"    DEL COMANDO PRINT"
50 PRINT
60 PRINT "COLUMNA UNO", "COLUMNA DOS"
70 PRINT 14.2,13.7
80 PRINT 1,2,7,11
90 PRINT
100 A$ = "COMPRIMIDA":B=3
110 PRINT, "SEGUNDA LINEA"
130 PRINT A$,"LA SALIDA EN LA LINEA", B
140 PRINT
150 PRINT "ESTO APARECERA";
160 PRINT "COMO UNA LINEA",
170 PRINT "DEMOSTRACION",
180 PRINT "TERMINADA"
```

PRINT @

La sentencia PRINT @ se utiliza para dirigir la salida a un punto determinado de la pantalla.

Con este propósito, la pantalla está dividida en una tabla de 16 x 32 casillas, lo que proporciona 512 posiciones. Vea el diagrama del apéndice F para ver cómo se numera la tabla.

El formato del comando PRINT @ es

PRINT @ expresión, lista de elementos

La expresión puede ser un número, una variable o una expresión aritmética, mientras el valor se mantenga entre 0 y 511.

La lista de elementos es la misma que la utilizada en el comando PRINT, pueden ser números, variables, cadenas o expresiones separadas por comas o punto y comas.

Si considera que su pantalla tiene 16 líneas, entonces la sentencia

```
PRINT @ 32* (LINEA-1),A
```

imprimirá el valor de A al comienzo de la LINEA imaginaria de su pantalla. Cada línea dependerá del valor de la variable LINEA (un número entre 1 y 16).

El siguiente ejemplo es una forma deliberadamente confusa de producir un resultado bastante sencillo. Trate de averiguar lo que hace, ejecute el programa, y después escriba una versión más directa de producir el mismo resultado.

Fíjese en la utilización del punto y coma al final de las sentencias para evitar que se borre el resto de la línea. (Si no se lo cree, trate de introducir la línea 80 sin el punto y coma final.)

```
10 CLS:P$ = "PRINT ":N = 1
20 FILA = 12:A$ = "LA PANTALLA"
30 PRINT @ 26* (FILA -1) + 15, "UTILIZANDO"; P$
40 PRINT @ 230 + 3, "EN CUALQUIER PUNTO DE";
50 PRINT @ 160, "ESTE";:PRINT @ 174, "MUESTRA";
60 PRINT @ 16* (FILA-1) + 11, "PUEDE";:PRINT @ 12, "PAGINA";
70 PRINT @ 263 + 25, A$
80 PRINT @ 182, "COMO";
```

PRINT

La sentencia PRINT # tiene la misma estructura que la sentencia PRINT, pero en este formato se puede especificar el dispositivo a donde se debe dirigir al igual que en la sentencia Input #:

PRINT # variable o constante numérica, lista de salida

La variable o constante numérica puede ser:

```
0      = consola (pantalla)
-1     = cassette
-2     = impresora
```

PRINT USING

La sentencia PRINT USING permite un gran control sobre la salida de resultados por pantalla o impresora.

PRINT USING formato; lista de salida

El formato es una constante o variable de cadena conteniendo los especificadores

de campo que indica cómo se debe imprimir la lista de salida. La lista de salida es una lista de cadenas o variables numéricas (o constantes) separadas por comas.

Los especificadores de campo son:

CARACTER	ACCION	EJEMPLO	RESULTADO
#	Formato de número	"####"; 147.2	147
	Presenta una coma a la izq. cada 3 caracteres	"#####.#"; 123456	123,456
**	Rellena los espacios iniciales con *->	"* * ##.####"; 1.47	* * * 1.47
\$	Coloca un signo \$ delante del número	"\$####.#"; 12.689	\$ 12.69
** \$	Signo \$ flotante	"** \$####.#"; 12.689	* * * \$12.69
+	En primera posición hace que se ponga el signo delante. en última posición lo imprime después del número	"#.##+"; -12.689	12.69-
↑↑↑↑~	Imprime en formato exponencial	"#.## ↑↑↑↑ ; 12.689	0.13E + 02
!	Imprime solo el primer carácter	"!"; "CREDITO"	C
%espacios%	Campo de cadena. La longitud del campo es el número de espacios más 2	"% %"; BALANCE	BALANCE

PRINT USING también tiene el formato PRINT # variable, USING formato; lista de salida que coincide con su forma general pero especificando en la variable el dispositivo de salida.

READ

La sentencia READ lee el siguiente o siguientes datos de las sentencias DATA. Si es la primera vez que se utiliza lee el primer dato de la sentencia DATA que tenga menor número de línea.

READ variable1, variable2, ...

El tipo de variable debe coincidir con el tipo de dato que le corresponde.

```
10 CLS:PRINT:PRINT:PRINT
20 READ A,B:IF A = -9999 THEN RESTORE:GOTO 20
30 PRINT A;" + 5 ES ";:INPUT C
40 IF C = B THEN PRINT "CORRECTO" ELSE PRINT "ERRONEO"
50 FOR D = 1 TO 600:NEXT D:GOTO 10
60 DATA 8,13,12,17,5,10,27,32,14,19,3,8
70 DATA 7,12,6,11,1,6, -9999, -9999
```

REM

La sentencia REM se utiliza para introducir comentarios en un programa. El ordenador ignora cualquier cosa que aparezca en la misma línea después de REM (o su abreviatura ' ').

```
10 REM ESTA ES UNA LINEA DE COMENTARIO
35 D = B*B-4*A*C:'CALCULO DEL DISCRIMINANTE
```

RESTORE

Fuerza que el siguiente dato leído con una sentencia READ vuelva a ser el primero de la línea DATA de menor número.

```
RESTORE
```

RETURN

Indica el fin de una subrutina. Cuando se accede a una parte del programa mediante la sentencia GOSUB, se retorna a la sentencia siguiente cuando se encuentra RETURN.

Ver la sentencia GOSUB

STOP

Cuando un programa encuentra esta sentencia rompe la ejecución del programa y

muestra un mensaje "BREAK AT LINE..."; donde se informa de la línea donde se encuentra el STOP.

De todas formas conserva todos los datos necesarios para continuar el programa si se escribe el comando CONT.

Cuando se ha producido el STOP se pueden inspeccionar las variables mediante la sentencia PRINT.

Nombres de variables de cadena

Un nombre de variable de cadena puede constar de cualquier combinación de letras y números, pero debe comenzar con una letra y acabar con el signo \$.

Como con los nombres de las variables numéricas el ordenador sólo reconocerá los dos primeros caracteres de modo que,

RESPUESTA\$, RES1\$, RES \$

Serán consideradas como la misma (RE\$).

Nombres de variables numéricas

Una variable numérica sólo puede almacenar números.

El nombre de una variable numérica puede constar de cualquier combinación de letras y números, pero debe comenzar por una letra.

Ya que el ordenador sólo reconoce los dos primeros caracteres de un nombre, nombres como:

ESTRELLA, ESTADO, ES

serán consideradas como la misma (ES).

Las variables con un nombre de más de dos caracteres, son útiles como medio para recordar su contenido,

NUMERO, CONTADOR, SUMA

serán aceptadas, pero ocuparán más memoria que NU, CO, SU.

Funciones

Las funciones son las órdenes que actúan siempre sobre un dato para transformarlo mediante un algoritmo y devolver un resultado.

Por ejemplo SQR(X) calcula la raíz cuadrada del dato X.

Tienen siempre ir acompañadas de una sentencia y ocupan en ella el lugar que ocuparía una variable.

Pueden ir con cualquier sentencia que utilice la función sin operar sobre ella. Por ejemplo PRINT SQR(3) utiliza el dato SQR (3) para imprimirlo en la pantalla pero INPUT SQR (3) no es válido porque intentará poner en SQR (3) el valor que se introdujese por teclado.

Los operandos que necesitan las funciones para operar correctamente se llaman argumentos.

Existen tres tipos de funciones

a) Numéricas

Devuelven un dato numérico. Su nombre no lleva al final el carácter \$ y necesitan al menos un argumento.

b) Alfanuméricas

Devuelven una cadena. Su nombre acaba con el carácter \$. Necesitan al menos un argumento.

c) Variables del sistema

Contienen valores que maneja el microordenador. No necesitan ningún argumento y devuelven valores numéricos o alfanuméricos según que lleven al final el carácter \$ o no.

Las funciones del Dragón por orden alfabético son:

– ABS	– HEX\$	– MEM	– SQR
– ASC	– INKEY\$	– MID\$	– STR\$
– ATN	– INSTR	– PEEK	– STRING\$
– CHR\$	– INT	– POINT	– TIMER
– COS	– JOYSTK	– POS	– TAN
– EOF	– LEFT\$	– RIGHT\$	– USR
– EXP	– LEN	– RND	– VAL
– FIX	– LOG	– SIN	– VARPTR

NOMBRE DE FUNCION	OPERACION	EJEMPLO
ABS (X)	Valor absoluto de X.	100 A = ABS (D*2-C)
ASC (X\$)	Devuelve el código ASCII del primer carácter de la cadena.	715 P = ASC (F\$)-64
ATN (X)	Arcotangente de X, en radianes. Inversa de TAN (X).	110 PRINT "ANG="; ATN(R3)
CHR\$ (X)	Devuelve el carácter cuyo código es X. X debe estar entre 0 y 255.	20 M\$ = CHR\$(143) + CHR\$(128)
COS (X)	Coseno de X, siendo X un ángulo en radianes.	510 F7 = COS (X+4)
EOF (-1)	Se utiliza para conocer si se ha llegado al final de un fichero en cassette. Devuelve -1 si se ha leído el último registro del fichero y cero en cualquier otro caso.	30 IF EOF(-1)=0 THEN INPUT # -1,A\$ ELSE CLOSE # -1
EXP (X)	Eleva el número e a la potencia X. Inversa del LOG (X).	215 Q = EXP (-A*A)
FIX (X)	Devuelve la parte entera de X (Trunca la parte decimal).	172 N = FIX (Z*.05)
HEX\$ (X)	Calcula el valor hexadecimal del valor X decimal.	42 PRINT HEX\$ (30)
INKEY\$	Chequea el teclado y devuelve la tecla pulsada (si se ha hecho). Devuelve una cadena, luego debe usarse en una expresión de cadena.	146 P3\$ = INKEY\$:IF P3\$ = "" THEN 146

INSTR (P, S\$, T\$)

Explora la cadena S\$ buscando la cadena T\$, a partir de la posición P. Devuelve 0 si no la encuentra, si no su posición.

212 F = INSTR (N,X\$,"AB")

INT (X)

Trunca X si es positivo como FIX. Si X es negativo, redondea por debajo. eje. INT DE - 12.001 es - 13

280 P = INT (100*X)

JOYSTK (X)

Devuelve la posición actual del joystick izq. o derecho, como sigue:
X = 0, joystick izq.horiz-
ontal
X = 1, joystick izq.verti-
cal
X = 2, joystick der.hori-
zontal
X = 3, joystick der.verti-
cal

1040 A = JOYSTK (0)
B = JOYSTK (1)

LEFT\$ (X\$,N)

Devuelve los primeros N caracteres de la cadena X\$

114 A\$ = LEFT\$ (B\$,7)

LEN (X\$)

Devuelve la longitud de la cadena X\$. Se cuentan también los espacios. Si la cadena está vacía da 0

845 N = LEN (N\$)

LOG (X)

Logaritmo natural de X. El valor del argumento debe ser mayor que 0. Inversa de EXP (X).

617 L1 = 5.2*LOG (W4)

MEM

Devuelve el espacio libre de memoria.

PRINT MEM

MID\$ (X\$,M,N)

Devuelve N caracteres de la cadena N\$, a partir de la posición M. Si se omite N se devuelve toda la cadena a la derecha de M. M debe ser mayor que 0.

760 K\$ = MID\$ (W\$,I,4)

PEEK (X)	Devuelve el contenido de la posición de memoria cuya dirección es X.	55 P = PEEK (65280)
POINT(X,Y)	Mira a ver si una celda en baja resolución está en off. X debe estar entre 0—63 (horizontal), y entre 0—31 (vertical). Devuelve el valor 0 si la celda está off, -1 en modo texto, si está activa devuelve de 1 a 8 según el color.	50 IF POINT (5,A) = C THEN 210
POS (X)	Devuelve la posición actual de impresión. Los únicos argumentos son: 0 para pantalla -2 para impresora.	168 IF POS(0) < 30 THEN PRINT A\$
RIGHT\$(X\$,N)	Devuelve los últimos N caracteres de la cadena X\$.	340 T\$ = RIGHT\$(Q\$.B + 7)
RND (X)	Devuelve un número aleatorio entero entre 1 y X. Si X es 0, devuelve un número aleatorio entre 0 y 1	220 PRINT RND (510); ""
SIN (X)	Calcula el seno de X, estando X en radianes.	205 S = SIN (K*PI/180)
SGN (X)	Devuelve el signo del argumento. Si X es negativo da -1 Si es cero devuelve 0 Si X es positivo da +1	412 Y = RND(ABS(N))*SGN (N)
SQR (X)	Raíz cuadrada del argumento, X no debe ser negativo, si lo fuera se devuelve SQR (ABS (X)).	330 C = SQR (A*A + B*B)

STR\$(X)	<p>Convierte un valor numérico en su carácter equivalente.</p>	<p>175 A\$ = STR\$(12.49)</p>
STRING\$(N,C)	<p>Devuelve una cadena de longitud N con el carácter definido por C. C puede ser un número (el código ASCII del carácter) o el propio carácter entre comillas.</p>	<p>400 A\$ = STRING\$(5,67) 410 PRINT STRING\$(32,"**")</p>
TIMER	<p>Devuelve el contenido del timer, un valor entre 0 y 65535. Para ponerlo a 0 use <code>TIMER = 0</code>. Timer aumenta 50 cada segundo.</p>	<p>62 T1 = TIMER - T 63 TIMER = 0</p>
TAN(X)	<p>Tangente de X, donde X es un ángulo en radianes. Es la inversa de <code>ATN(X)</code>.</p>	<p>840 R5 = B/TAN(EQ-5)</p>
USR	<p>USRn le permite llamar hasta 10 (0 a 9) rutinas en lenguaje máquina. El formato es, USRn (argumento) Donde el "argumento" puede ser una cadena o una expresión numérica. Cuando se encuentra una llamada de usuario en un programa, se transfiere el control a la dirección dada en la sentencia <code>DEF USRn</code>. La dirección especifica el punto de entrada de la rutina en lenguaje máquina.</p>	
VAL(X\$)	<p>Convierte una cadena numérica en un número. Si esta empieza por un carácter alfabético, devuelve 0.</p>	<p>92 Z = VAL("123")</p>
VARPTR(X)	<p>Devuelve la dirección donde está almacenada la variable. La variable tiene que haber sido utilizada previamente.</p>	<p>50 X = 0 60 A = VARPTR(X) 70 XX\$ = "" 80 B = VARPTR(XX\$)</p>

Gráficos y Sonido

En este capítulo vamos a describir las sentencias que nos permiten manejar gráficos, color y sonido.

Estas sentencias son:

CIRCLE	LINE	PCOPY	PRESET	SCREEN
COLOR	PAINT	PLAY	PSET	SET
DRAW	PCLEAR	PMODE	PUT	SOUND
GET	PCLS	PPOINT	RESET	

CIRCLE

La sentencia CIRCLE dibujará círculos, elipses y arcos. Sólo se puede utilizar en los modos de alta resolución.

CIRCLE (x,y),r,c,ha, comienzo, fin

x	es la coordenada X del centro del círculo (de 0 a 255)
y	es la coordenada Y del centro del círculo (de 0 a 191)
r	es el radio del círculo medido en puntos de pantalla
c	es un código de color (de 0 a 8), deberá ser uno de los colores disponibles actualmente.
ha	es la relación entre la altura y la anchura (entre 0 y 255). Se utiliza para dibujar elipses. Si se omite se toma 1.
comienzo	es el comienzo del arco del círculo (de 0 a 1). La posición 0 representa las 3 horas. Si se omite se usa 0.
fin	indica el final del arco (de 0 a 1). El dibujo avanza en el sentido de las agujas del reloj desde el comienzo. la posición 0.5 representa las 9 horas. Si se omite se toma 1.

```
10 PMODE 3, 1:SCREEN 1, 0:PCLS
20 CIRCLE (180, 156),28,3: PAINT (180, 156),3,3
30 CIRCLE (110, 156),28,3: PAINT (110, 156),3,3
40 CIRCLE (144, 80), 68, 4, 1, 0,.5
50 LINE (212, 80) — (76, 80), PSET:LINE — (48,32), PSET
60 PAINT (144, 82):CIRCLE (144, 80), 70,4,.8,.79,1
70 LINE (160, 80) — (160, 28), PSET:PAINT (210, 75)
80 GOTO 80
```

COLOR

La sentencia COLOR se usa para cambiar los colores utilizados como fondo y en primer término cuando se trabaja en el modo de gráficos en alta resolución.

COLOR en primer término, fondo

Tanto "en primer término" como "fondo" son números entre 0 y 8, que representan el código del color. Ambos colores deben pertenecer al conjunto de colores posibles dentro del modo actual.

DRAW

La sentencia DRAW dibuja una línea, o serie de ellas, de acuerdo con las instrucciones de una cadena. Sólo opera en el modo de alta resolución de gráficos.

DRAW cadena

La cadena puede ser una constante de cadena o una variable de cadena y puede contener cualquiera de los siguientes subcomandos:

Mx,y	Se mueve a la posición de dibujo x,y
Un	Arriba n puntos
Dn	Abajo n puntos
Ln	Hacia la izquierda n puntos
Rn	Hacia la derecha n puntos
En	n puntos a 45 grados
Fn	n puntos a 135 grados
Gn	n puntos a 225 grados
Hn	n puntos a 315 grados
X	Ejecuta una subcadena
C	Fija el color de la línea
Ak	Gira la siguiente línea un ángulo
	K=0 0 grados K=1 90 grados
	K=2 180 grados K=3 270 grados
Sk	Hace el dibujo a 1/4s de escala, k entre 1 y 62
	K=1 un cuarto de escala, K=8 dobla la escala
	Por defecto K=4
N	No actualizar la posición de dibujo
B	En blanco (no dibuja, sólo mueve)

Se pueden especificar movimientos relativos con el parámetro M en la forma.

M x desplazado, y desplazado

Donde x desplazado e y desplazado son números que especifican la distancia que hay que mover a partir de la posición actual. Ambos números deberán estar precedidos por un signo más (+) o menos (-).

```

10 PMODE 3,1:PCLS:SCREEN 1,1
20 FOR I=0 TO 3: DRAW "A" + STR$(I):PCLS
30 DRAW "BM120,96;R13;D26;L13"
40 DRAW "C7;L6;U6;E6;BR13;F6;D6;16;BU26;H6;G6"
50 FOR D=1 TO 100:NEXT D,I
60 GOTO 60

```

GET

La sentencia GET sólo se puede utilizar en gráficos en alta resolución. GET copiará los gráficos contenidos en un área rectangular determinada de la pantalla, y los almacenará en una matriz. La matriz deberá haber sido dimensionada correctamente antes.

GET (x_1, Y_1) — (x_2, y_2), nombre de matriz, G

X_1, y_1 y x_2, y_2 , son las coordenadas del extremo superior izquierdo y del inferior derecho de un rectángulo de la pantalla.

«nombre de matriz» es el nombre de una matriz previamente dimensionada donde se almacenará el contenido del rectángulo.

La matriz debe ser dimensionada de la forma DIM nombre de matriz ($x_2 - x_1, y_2 - y_1$).

G indica que se almacenen todos los detalles del gráfico, este comando es opcional.

Vea PUT para un ejemplo del uso de GET.

De todas formas se puede dimensionar la matriz con una dimensión, ahorrando bastante más espacio.

La regla para hallar el tamaño de la matriz es la siguiente.

- 1.º) Se calcula el número de elementos gráficos que hay que almacenar, esto es, en GET (x_1, y_1) — (x_2, y_2) son $(x_2 - x_1) \cdot (y_2 - y_1)$.
- 2.º) Si se ha especificado la opción G, divida por 8 en PMODE 3 ó 4, por 16 en PMODE 1 ó 2 y por 32 en PMODE 0. Si no ha especificado la opción G divida por 16, 8 ó 32 respectivamente.
- 3.º) Redondee el resultado al entero inmediato superior.
- 4.º) Divida por 5 y redondee al entero superior. El número obtenido es la dimensión óptima de la matriz.

Por ejemplo, calculemos la dimensión óptima de GET (20,20) — (100,100) en PMODE 3 con la opción G.

$$(100 - 20) \cdot (100 - 20) = 6400$$

$$6400 : 8 = 800$$

$$800 : 5 = 160$$

La dimensión sería pues A(160)

LINE

La sentencia LINE se utiliza para dibujar líneas y rectángulos en el modo de alta resolución de gráficos.

LINE (x1,y1) — (x2,y2),a,b

x1 e y1 son las coordenadas del punto de comienzo de la línea.

x2 e y2 son las coordenadas del extremo final de la línea.

“a” puede ser PSET o PRESET. Si se utiliza PSET se dibujará una línea en el color utilizado en primer término. Si es PRESET la línea se dibujará en el color utilizado para el fondo. “b” es un parámetro opcional; si se utiliza, puede ser B o BF. Si se pone B se dibujará un rectángulo en lugar de la línea, la esquina superior del rectángulo será x1, y1 y la esquina inferior izquierda será x2, y2. Si se pone BF, se dibujará el rectángulo y además se pintará con el color utilizado en primer término.

```
10 PMODE 4,1:SCREEN 1,1:PCLS 5:COLOR 0,5
20 FOR I=1 TO 1000
30 X=X + L*SIN (R):Y = Y+L*COS (R)
40 IF X <- 128 OR X>128 THEN 90
50 IF Y <- 96 OR Y>95 THEN 90
60 LINE — (X+128, y+96), PSET
70 R1=R1+60:R = R1 /57.29578:L = L+0.5
80 NEXT I
90 GOTO 90
```

PAINT

La sentencia PAINT se utiliza en el modo de alta resolución de gráficos, para rellenar una forma con un color determinado.

PAINT (x,y),c,b

x,y son las coordenadas donde se comienza a pintar.

c es el código del color que se va a utilizar para pintar. Deberá estar entre 0 y 8 y ser uno de los colores disponibles en el modo de trabajo actual. Si se omite, se utilizará el color actual utilizado en primer término.

b es el código del color del borde donde se debe detener la pintada. También deberá estar comprendido entre 1 y 8, la pintada continuará sobre los bordes de cualquier otro color. Si se omite se tomará el color actualmente utilizado en primer término.

Vea CIRCLE para un ejemplo de su utilización.

PCLEAR

PCLEAR se utiliza para reservar páginas para gráficos en los modos de alta resolución.

PCLEAR n

n debe ser un número entre 1 y 8. Si se omite la sentencia PCLEAR se toma 4 por defecto.

Puesto que cada página para gráficos utiliza 1536 bytes de memoria, reserve sólo el espacio imprescindible.

PCLS

La sentencia PCLS se utiliza en alta resolución, para borrar la pantalla y dejarla con el color de fondo especificado.

PCLS c

c es el código del color de fondo deseado. Deberá ser uno de los colores disponibles dentro del conjunto del modo de trabajo; si no es así o se omite c se tomará el color de fondo por defecto.

Vea CLS para los códigos de color.

PCOPY

PCOPY es una sentencia para gráficos en alta resolución, que copia el contenido de una página de gráficos en otra.

PCOPY fuente TO destino

Tanto "fuente" como "destino" deben ser números entre 1 y 8 y deben referirse a páginas previamente reservadas con el comando PCLEAR. El espacio necesario para mantener una figura depende del modo utilizado y deberá tenerse en cuenta al utilizar PCOPY.

PCOPY 3 TO 5

PLAY

La sentencia PLAY se utiliza para generar una secuencia musical. El argumento es una expresión de cadena, o una constante, o una variable de cadena. Su formato es,

PLAY música

Donde "música" se forma con los siguientes elementos:

Notas: Una letra de la "A" a la "G" o un número entre 1 y 12. La letra C equivale a DO, la D a RE, E es MI, F es FA, G es SOL, A es LA y B es SI.
Si detrás se pone el signo # o + la nota es sostenida y si se pone - es bemol.

Octava: "O" seguida de un número entre 1 y 5. Por defecto se toma O2. Los valores por defecto los fija el ordenador al encenderlo.

Duración de la nota: "L" seguida de un número entre 1 y 255. Valor por defecto, L4.

Tempo: "T" seguida de un número entre 1 y 255. Por defecto T2.

Intensidad: "V" seguida por un número entre 1 y 31. Por defecto V15.

Duración de un silencio: "P" seguida por un número entre 1 y 255.

Ejecución de subcadenas: "X" seguida de una variable de cadena y un punto y coma.

Una nota sostenida se puede indicar por "+" o "#" y en bemol mediante "-".

El parámetro sobre la duración de una nota se puede modificar mediante un punto (.) adicional después del número; (L2.) representa una nota con puntillo.

La octava, intensidad, tempo y duración de una nota se pueden modificar utilizando uno de los siguientes sufijos:

- + Suma 1 al valor actual
- Resta 1 al valor actual
- > Multiplica por 2 el valor actual
- < Divide por dos el valor actual

10 X\$ = "O3L4EF#L4.EL8AAG#ABL4O+C#O-B"

20 A\$ = "XX\$;O4C#O-AF#O+DC#O-BL2AXX\$;

O+C#DEL8DO-BL<AG#L<AL4.BL80+C#L4

DO-BL4.O+C#L8DL<EC#L4.EL8EEEEEL1

EL4.EL8DC#EDO-BL<AG#L<A"

30 PLAY "T2V20" +A\$

PMODE

Selecciona la resolución y las páginas de gráficos que se van a utilizar.

PMODE x,y

x es el modo de gráficos según se indica en la tabla adjunta e y es la primera página de gráficos que se va a utilizar.

Las páginas que se van a utilizar deben estar reservadas por la sentencia PCLEAR.

NUMERO DE MODO	DIMENSION DEL PANEL	DIMENSION DEL PUNTO	PAGINAS USADAS	JUEGO DE COLORES DISPONIBLES	
				PANTALLA 1,0	PANTALLA 1,1
0	128 X 96		1	Negro (0), Verde (1)	Negro (0), beige (5)
1	128 X 96		2	Verde (1), Amarillo (2) Azul (3), Rojo (4)	Beige (5), Turquesa (6) Magenta (7), Naranja (8)
2	192 X 128		2	Negro (0), Verde (1)	Negro (0), Beige (5)
3	192 X 128		4	Verde (1), Amarillo (2) Azul (3), Rojo (4)	Beite (5), Turquesa (6) Magenta (7), Naranja (8)
4	256 X 192		4	Negro (0), Verde (1)	Negro (0), Beige (5)

PPOINT

Testea una celda en alta resolución de gráficos. Devuelve 0 si la celda está off, si no, devuelve el código del color de la celda. (X comprendido entre 0-255, y entre 0-191). 115C = PPOINT (A1, A2)

PRESET

Es la versión en alta resolución de la sentencia RESET.

PRESET (x,y)

desactiva el punto (x,y), (lo vuelve del color del fondo). X deberá estar comprendido entre 0 y 255 e y entre 0 y 191.

PSET

Es la versión de SET para alta resolución.

PSET (x,y,c)

activa el punto (x,y) con el color c.x deberá estar comprendido entre 0 y 255, e y entre 0 y 191.

c es el código de color entre 0 y 8, y deberá estar dentro del conjunto de colores disponibles.

PUT

La sentencia PUT se utiliza para presentar el contenido de una matriz de gráficos almacenada con el comando GET.

PUT se debe utilizar en el mismo modo que se usó para crear la matriz en el primer lugar, de otra forma los resultados serán impredecibles.

PUT (x₁,y₁) – (x₂,y₂), nombre de matriz, acción

x₁, y₁ son las coordenadas de la esquina superior izquierda de la pantalla donde presentará la imagen y x₂,y₂ es la esquina inferior derecha. El "nombre de matriz" se refiere a la matriz conteniendo el dibujo. El parámetro "acción" es opcional pero se deberá utilizar si se utilizó el parámetro G en el comando GET.

PSET Activa los puntos en el destino como en la matriz fuente.

PRESET Desactiva cada punto fijado en la matriz fuente.

AND Compara las matrices fuente y destino. Si ambos puntos están activados, el punto lo seguirá estando, si no, quedará desactivado.

OR Compara los puntos como antes, si cualquiera de los puntos está activo el punto en pantalla también.

NOT Invierte el estado de cada punto en el área de destino, de acuerdo con la matriz fuente.

El área seleccionada para presentación deberá tener la misma dimensión que la matriz o aparecerá "basura" en la pantalla.

```
10 PCLEAR 4: PMODE 3,1:PCLS:SCREEN 1, 1:DIM 2 (30,30)
20 DRAW "BM10,12;28;R1U3R1D2R2U2R1D3R1D2L
   D2R1D1L2U3L4D3L2U1R1U2L1U2R1U2BR1BD1D2RDU2NL2R2D2L2US)
30 PAINT (11,13),6,5:GET (0,0)– (30,30), W
40 A$ = INKEY$:IF A$ = "" THEN 40
50 PCLS:FOR C=0 TO 100 STEP 20
```

```

60 FOR A=0 TO 200 STEP 20
70 PUT (A,C)-(30+A,30+C), W
80 PUT (A,C+30)-(30+A,60+C),W
90 PUT (A,C+60)-(30+A,90+C),2
100 PLAY "T255;ABFGBA":PCLS:NEXT A,C

```

RESET

La sentencia RESET se utiliza para borrar un punto activado con el comando SET. Se utiliza en baja resolución de gráficos.

RESET (X,Y)

X,Y son las coordenadas del punto a desactivar. X deberá estar entre 0 y 63, e Y entre 0 y 31

El punto se pone del color del fondo, provocando el borrado de lo que hubiera.

Para ver un ejemplo vea SET.

SCREEN

La sentencia SCREEN se utiliza para conmutar la pantalla entre los modos gráfico y de texto.

SCREEN tipo, juego de colores

El tipo puede ser: 0 para texto y gráficos en baja resolución ó 1 para gráficos en alta resolución.

El juego de colores es 0 ó 1

El juego de colores para la pantalla de textos es 0 (negro sobre verde) ó 1 (negro sobre naranja). Para gráficos en alta resolución el juego de colores depende del modo en que estemos trabajando.

PMODE	PANTALLA 1,0	PANTALLA 1,1
0	Negro, verde	Negro, beige
1	Verde, amarillo	Beige, verde
	Azul, rojo	Magenta, naranja
2	Negro, verde	Negro, beige
3	Verde, amarillo	Beige, turquesa
	Azul, rojo	Magente, naranja
4	Negro, verde	Negro, beige

SET

La sentencia SET se utiliza para activar con un color determinado un punto de la pantalla, cuando se trabaja en baja resolución

```
SET (X,Y,C)
```

X e Y son las coordenadas del punto. X deberá estar comprendido entre 0 y 63, e Y entre 0 y 31.

C es el código del color deseado. Deberá ser un número entre 0 y 8.

```
10 CLS0:SET(5,27,8):SET(6,27,8)
20 FOR X=0TO 6:FOR Y= 28 TO 30
30 SET (X,Y,8):NEXT Y,X
40 FOR X=7 TO 63: FOR D=1 TO 200:NEXT D
50 FOR Y=27 TO 30:IF Y=27 THEN RESET (X-2,Y)
60 SET (X,Y,8):RESET (X-7,Y):NEXT Y,X
70 GOTO 70
```

SOUND

La sentencia SOUND genera un sonido con un tono y una duración determinado. Requiere dos argumentos.

```
50 SOUND P,D.
```

P es un número entre 1 y 255. El tono más bajo es 1, el más alto 255. El Do medio del piano es P=89

D puede ser cualquier número entre 1 y 255. D=16 da un tono durante aproximadamente 1 segundo.

```
10 CLS
20 PRINT 6, "DEMOSTRACION DE SONIDO"
30 PRINT 64, "INTRODUCTA UN NUMERO COMPRENDIDO ENTRE 1 y 255";
40 PRINT 111, "PARA ESPECIFICAR EL TONO SONORO": INPUT P
50 PRINT 192, "DE UN NUMERO PARA LA DURACION DE LA NOTA";
60 INPUT D:CLS RND(9) - 1
70 SOUND P,D
80 GOTO 10
```

Códigos de error

Código	Descripción	Indicación
/0	Ha intentado dividir por cero.	Revise las operaciones de su programa. La variable que divide toma un valor equivocado.
A O	Se ha abierto un fichero que ya estaba abierto.	Utilice la sentencia CLOSE antes de abrir un fichero.
BS	El índice de una matriz es erróneo.	Compruebe sus cálculos. El índice de su matriz es mayor que el que ha declarado o es negativo.
CN	Ha intentado continuar un programa que ha sido abortado.	No puede utilizar el comando CONT después de un error, de modificar el programa o de que se ha ejecutado la sentencia END.
DD	Matriz redimensionada.	Si quiere redefinir una matriz utilice la sentencia CLEAR.
DN	Número de dispositivos erróneo.	Los únicos números válidos son 0, -1 y -2.
DS	Error en una sentencia directa.	Normalmente aparece si intenta cargar un fichero de datos con la sentencia CLOAD.
FC	Llamada ilegal de función.	El argumento de una función se sale del rango admisible.
FD	Dato de un fichero erróneo.	Ha intentado leer un dato numérico de un fichero en cassette en una variable de cadena o viceversa.
FM	Tipo de fichero incorrecto.	Ha intentado cargar un programa en BASIC con CLOADM o al revés.
ID	Sentencia directa no admisible.	Ha intentado utilizar una sentencia en modo directo y no es posible.
IE	Lectura pasado EOF.	Ha intentado leer un fichero cuando no hay más datos. Utilice la función EOF.
IO	Error en la lectura del cassette.	El volumen de un cassette no está ajustado o la cinta es defectuosa.

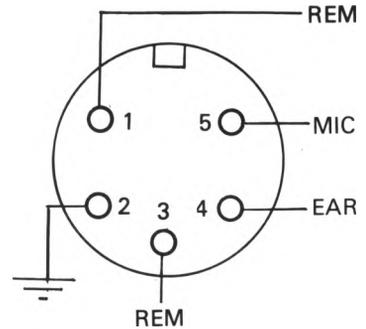
Código	Descripción	Indicación
LS	Cadena demasiado larga.	El máximo número de caracteres de una cadena es 255.
NF	Next sin FOR	Se ha ejecutado una sentencia NEXT sin haber pasado por una sentencia FOR.
NO	Fichero no abierto.	Ha intentado leer o escribir en un fichero que no ha sido abierto. Utilice la sentencia OPEN.
OD	Sin datos para la sentencia READ.	Ha intentado hacer una lectura con READ cuando ya ha leído todos los datos de las sentencias DATA. Utilice RESTORE o revise su programa.
OM	Sin memoria.	Ha utilizado demasiada memoria. Su programa o sus datos son demasiado grandes.
OS	Sin memoria para cadenas.	Ha ocupado todo el espacio reservado para cadenas. Utilice CLEAR para reservar más.
OV	Desbordamiento de una variable.	El valor de una variable se ha hecho demasiado grande.
RS	RETURN sin GOSUB.	Ejecución de una sentencia RETURN sin una sentencia GOSUB. Escriba sus subrutinas detrás de la sentencia END.
SN	Error de SINTAXIS	El formato de su sentencia o comando no es correcto. Consulte en el manual.
ST	Fórmula con cadenas demasiado largas.	Ha escrito una operación con cadenas demasiado larga. Realízela en varias líneas.
TM	Error de tipos.	Ha intentado asignar una variable numérica a una cadena o viceversa.
UF	Función desconocida.	Ha llamado a una función que no ha definido previamente con DEF FN.
UL	Línea no encontrada.	La línea a la que se hace referencia no está en su programa.

Apéndices

Apéndice A. Cassette

Con el Dragón 200 se puede utilizar el cassette como periférico de almacenamiento secuencial.

El esquema de la conexión es el siguiente:



PIN 1 — Control remoto para conectar el motor on/off.

PIN 2 — Masa.

PIN 3 — Control remoto.

PIN 4 — Entrada del cassette al Dragón.

PIN 5 — Salida del Dragón al cassette.

Todos los ficheros que se almacenan en cassette tienen la misma estructura.

- Primero un leader compuesto por 128 bytes con el valor \$55.
- Bloque de cabecera.
- 0,5 segundos en silencio para dar tiempo al BASIC para comprobar el nombre del fichero.
- Leader de sincronismo.
- Bloques de datos.
- Bloque de fin de fichero.

Todos los bloques contienen 256 bytes aunque no todos tienen por qué ser significativos.

Describamos el bloque de cabecera.

El bloque de cabecera tiene 15 bytes significativos cuyo contenido es:

1 — 8 : Nombre del fichero

9 — 11 : Tipo de fichero: \$ 00 00 00 programa en Basic
 \$ 02 00 00 programa en L/M
 \$ 00 FF FF Fichero de datos

12 — 13 : Dirección de carga para programas en L/M.

14 — 15 : Dirección de ejecución para programas en L/M.

Para el manejo del cassette el Dragón utiliza las siguientes posiciones de memoria:

\$ 7C	: Contiene el tipo del último bloque leído
\$00	bloque de cabecera.
\$01	bloque de datos.
\$FF	Fin de fichero.
\$7D	: Contiene el número de bytes significativos de un bloque.
\$7E,\$7F	: Contiene la dirección de lectura o grabación del siguiente bloque.
\$ 81	: Contiene el código de error.
0	sin error
1	I/O? ERROR
2	OM? ERROR

Para operar utiliza las siguientes subrutinas que nombraremos por su función.

MOTORON Parámetros necesarios: ninguno
Entrada: \$ 8015
Resultado: El motor en marcha.

MOTOROFF Parámetros necesarios: ninguno
Entrada \$ 8018
Resultado: El motor se para.

CSRDON Parámetros necesarios: ninguno
Entrada: \$ 8021
Resultado: MOTOR ON Y lee leader.

WRTLDR Parámetros necesarios: ninguno
Entrada: \$ 801B
Resultado: MOTORON y graba leader.

BLKIN Parámetros necesarios: MOTORON \$7E, \$7F dirección de carga.
Entrada: \$ B93E
Resultado: Lee un bloque y lo coloca en la dirección dada por \$ 7E, \$ 7F.

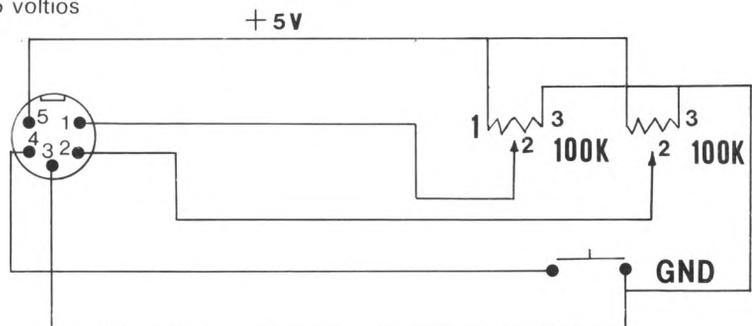
BLKOUT Parámetros necesarios: MOTOR ON \$7E, \$7F dirección de comienzo de grabación, \$7C tipo de bloque, \$7D longitud significativa del bloque.
Entrada: \$ B999
Resultado: Graba un bloque según los parámetros especificados.

Apéndice B. Esquemas

Controladores joystick

Además de los joysticks disponibles para Dragón, se le pueden conectar otros utilizando las puertas de joystick y conectores DIN de 5 pines. Las conexiones son las siguientes:

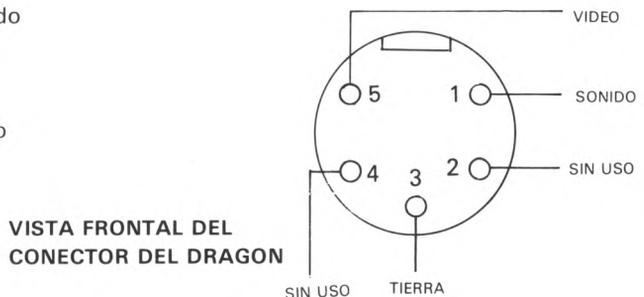
- PIN 1 Desplazamiento X
- PIN 2 Desplazamiento Y
- PIN 3 0 voltios
- PIN 4 Botón de disparo
- PIN 5 +5 voltios



Conexiones del monitor

El Dragón es compatible con monitores de señal compuesta de video. Para conectar un monitor PAL al Dragón 200 utilice un cable con conector DIN de 5 pines, como sigue:

- PIN 1 Señal de sonido
- PIN 2 Nada
- PIN 3 Tierra
- PIN 4 Nada
- PIN 5 Señal de video

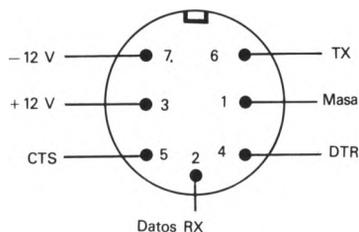


RS232 serie

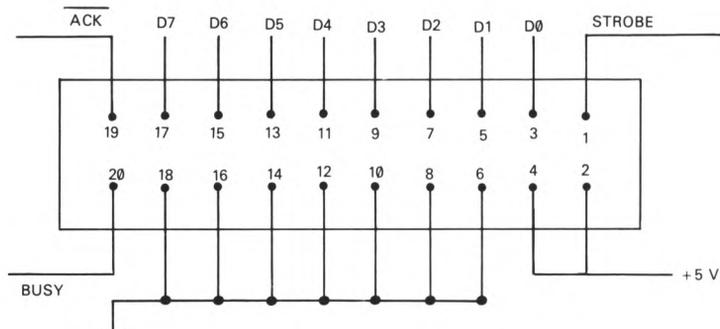
A la derecha de la conexión del joystick derecho se encuentra la salida serie RS232 I/O que es fácil localizar porque es la única que acepta un conector de 7 pin.

Esta entrada serie se puede utilizar para interconexiones entre ordenadores o para utilizar una impresora serie según se explica en el capítulo 4.

La configuración de este interface es la siguiente:



Esquema del interface centronics



Para construir un cable de impresora se conectan todos los puntos correlativamente, es decir, 1 a 1, 3 a 3, 5 a 5, 6 a 6, etc..., con las siguientes excepciones:

- Los puntos 2 y 4 libres.
- El punto 20 al punto 21 de la impresora.

Apéndice C. Código ASCII

TECLA	MINUSCULAS	MAYUSCULAS
[BREAK]	—	3
[CLEAR]	—	12
[ENTER]	—	13
[ESPACIO]	—	32
!	33	—
..	34	—
#	35	—
\$	36	—
%	37	—
&	38	—
'	39	—
(40	—
)	41	—
*	42	—
+	43	—
,	44	—
—	45	—
.	46	—
/	47	—
Ø	18	48
1	—	49
2	—	50
3	—	51
4	—	52
5	—	53
6	—	54
7	—	55
8	—	56
9	—	57
:	—	58
:	—	59
<	60	—
=	61	—
>	62	—

TECLA	MINUSCULAS	MAYUSCULAS
?	63	—
@	19	64
A	97	65
B	98	66
C	99	67
D	100	68
E	101	69
F	102	70
G	103	71
H	104	72
I	105	73
J	106	74
K	107	75
L	108	76
M	109	77
N	110	78
O	111	79
P	112	80
Q	113	81
R	114	82
S	115	83
T	116	84
U	117	85
V	118	86
W	119	87
X	120	88
Y	121	89
Z	122	90
↑	95	94
↓	91	10
←	21	8
→	93	9

Los siguientes caracteres en minúsculas están disponibles con la función CHR\$:

[CHR\$(123)
/ CHR\$(124)
] CHR\$(125)

↑ CHR\$(126)
← CHR\$(127)

Los caracteres del 127 al 255 son los caracteres gráficos siguientes:

CARACTERES GRAFICOS



128



129



130



131



132



133



134



135



136



137



138



139



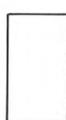
140



141



142



143

Para generar los caracteres anteriores utilice CHR\$ con el código correspondiente. Para obtener los demás colores, sume al código el número apropiado. Por ejemplo, PRINT CHR\$(142 + 112) produce el carácter 142 pero el área verde se convierte en naranja.

+16 amarillo

+32 azul

+48 rojo

+64 beige

+80 turquesa

+96 magenta

+112 naranja

Los códigos de caracteres de la VDG son:

CHARACTER	DECIMAL	HEX	CHARACTER	DECIMAL	HEX
@	0	00		32	20
A	1	01	!	33	21
B	2	02	"	34	22
C	3	03	#	34	23
D	4	04	\$	36	24
E	5	05	%	37	25
F	6	06	&	38	26
G	7	07	'	39	27
H	8	08	(40	28
I	9	09)	41	29
J	10	0A	*	42	2A
K	11	0B	+	43	2B
L	12	0C	,	44	2C
M	13	0D	-	45	2D
N	14	0E	.	46	2E
O	15	0F	/	47	2F
P	16	10	0	48	30
Q	17	11	1	49	31
R	18	12	2	50	32
S	19	13	3	51	33
T	20	14	4	52	34
U	21	15	5	53	35
V	22	16	6	54	36
W	23	17	7	55	37
X	24	18	8	56	38
Y	25	19	9	57	39
Z	26	1A	;	58	3A
[27	1B	;	59	3B
\	28	1C	<	60	3C
]	29	1D	=	61	3D
	30	1E	>	62	3E
	31	1F	?	63	3F

Acceso: Directo con la sentencia PRINT.

Modo 2

Tipo: Semigráfico.

Descripción: Divide la posición de cada carácter en 4 partes.

Cada carácter ocupa un byte de memoria.

Area de memoria: Desde 1024 hasta 1024 + 511.

Resolución: 64 columnas por 32 filas.

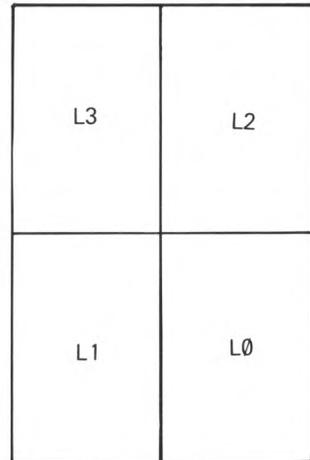
Configuración del byte:



color

Si es 0 el elemento negro
si es 1 del color especificado

carácter



Colores: 8

- 000 verde
- 001 amarillo
- 010 azul
- 011 rojo
- 100 beige
- 101 turquesa
- 110 Magenta
- 111 Naranja

Acceso: Desde el Basic con SET/RESET. Se superpone con el modo 1.

Modo 4

Tipo: Semigráfico.

Descripción: Divide cada carácter en 8 partes. Admite texto aunque no en modo directo.

Cada carácter ocupa 4 bytes.

Area de memoria: Del 1024 a 1024 + 2048.

Resolución: 64 columnas por 64 filas.

Configuración de los bytes:

1	C	C	C	L7	L6	X	X	+0
1	C	C	C	L5	L4	X	X	+32
1	C	C	C	X	X	L3	L2	+64
1	C	C	C	X	X	L1	L0	+96

1 – color
 0 – negro

L7	L6
L5	L4
L3	L2
L1	L0

Los bit señalados con XX no son usados por lo que recomendamos poner la misma configuración que en L₁ L₂ (o la pareja que corresponda).

Colores: 8 colores con los mismos códigos que en el modo 1.

Acceso: 100 POKE 65314, PEEK (65314) AND 7

110 POKE 65475, 0 : POKE 65475, 1: POKE 65472,0

Para escribir texto debe colocar el código de VDG del carácter en las 4 posiciones correspondientes.

Modo 5

Tipo: Semigráfico.

Descripción: Divide cada carácter en 12 partes.

Admite texto de la misma forma que el modo 4.

Cada carácter ocupa 6 bytes.

Area de memoria: desde 1024 a 1024 + 3072.

Resolución: 64 columnas por 96 filas.

Configuración de los bytes:

1	C	C	C	L11	L10	X	X	+ 0
1	C	C	C	L9	L8	X	X	+ 32
1	C	C	C	L7	L6	X	X	+ 64
1	C	C	C	X	X	L5	L4	+ 96
1	C	C	C	X	X	L3	L2	+ 128
1	C	C	C	X	X	L1	L0	+ 160

1 Color
 0 Negro

carácter

L11	L10
L9	L8
L7	L6
L5	L4
L3	L2
L1	L0

Los bit XX igual que en el modo 4.

Colores: Igual que en el modo 2

Acceso: 100 POKE 65314, PEEK (65314) AND 7

110 POKE 65477, 1: POKE 65474, 0: POKE 65472, 0

Texto igual que en el modo 4

Modo 6

Tipo: Semigráfico.

Descripción: Divide cada carácter en 24 partes.

Admite texto como en los modos anteriores.

Cada carácter ocupa 12 bytes.

Area de memoria: Del 1024 al 1024 + 6144.

Resolución: 64 columnas por 192 filas.

Configuración de los bytes:

1	C	C	C	L23	L22	X	X	+ 0
1	C	C	C	L21	L20	X	X	+ 32
1	C	C	C	L19	L18	X	X	+ 64
1	C	C	C	L17	L16	X	X	+ 96
1	C	C	C	L15	L14	X	X	+ 128
1	C	C	C	L13	L12	X	X	+ 160
1	C	C	C	X	X	L11	L10	+ 192
1	C	C	C	X	X	L9	L8	+ 224
1	C	C	C	X	X	L7	L6	+ 256
1	C	C	C	X	X	L5	L4	+ 288
1	C	C	C	X	X	L3	L2	+ 320
1	C	C	C	X	X	L1	L0	+ 352

1 — color
 0 — negro

carácter

L23	L22
L21	L20
L19	L18
L17	L16
L15	L14
L13	L12
L11	L10
L9	L8
L7	L6
L5	L4
L3	L2
L1	L0

XX como en los modos anteriores.

Colores: 8 colores como en los modos anteriores.

Acceso: 100 POKE 65314, PEEK (65314) AND 7
 110 POKE 65477,1 : POKE 65475,1: POKE 65472,0

Modo 7

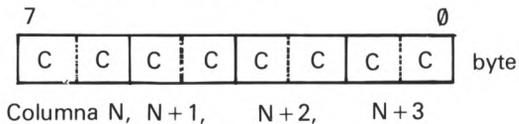
Tipo: Gráfico.

Descripción: Modo gráfico sin texto con 4 colores.

Area de memoria: del 1024 a 1024 + 1024.

Resolución: 64 columnas por 64 filas.

Configuración de los bytes:



CC es el código del color del elemento correspondiente.

Colores: 2 conjuntos de 4 colores.

C S S = 0

- 00 Verde
- 01 Amarillo
- 10 Azul
- 11 Rojo

C S S = 1

- 00 Beige
- 01 Turquesa
- 10 Magenta
- 11 Naranja

Acceso: 100 POKE 65314, (PEEK (65314) AND 7) + 128 + CSS
110 POKE 65473,1 : POKE 65474,0 : POKE 65476,0

Modo 8

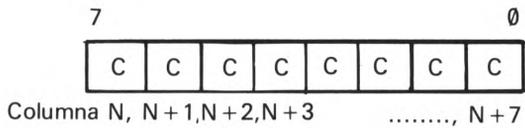
Tipo: Gráfico

Descripción: Modo gráfico con 2 colores.

Area de memoria: Del 1024 a 1024 + 1024.

Resolución: 128 columnas por 64 filas.

Configuración de los bytes:



C código del color.

Colores: 2 conjuntos de 2 colores.

CSS = 0 0 negro
 1 verde

CSS = 1 0 negro
 1 beige

Acceso: 100 POKE 65314 (PEEK (65314) AND 7) + 144 + CSS * 8
110 POKE 65473,1 ; POKE 65474,0 ; POKE 65476,0

Modo 9

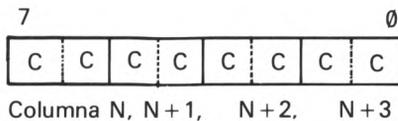
Tipo: Gráfico.

Descripción: Modo gráfico con 4 colores.

Area de memoria: Del 1024 al 1024 + 2048.

Resolución: 128 columnas por 64 filas.

Configuración de los bytes:



CC código de color.

Colores: Como en el modo 7.

Acceso: 100 POKE 65314, (PEEK (65314) AND 7) + 160 + CSS * 8
110 POKE 65472,0 : POKE 65475,0 : POKE 65476,1

Modo 10

Tipo: Modo gráfico con dos colores utilizables desde el Basic.

Area de memoria: 1536 Bytes.

Resolución: 128 columnas por 96 filas.

Configuración del byte: Como en el modo 8.

Colores: Como en el modo 8.

Acceso: PMODE 0.

Modo 11

Tipo: Modo gráfico con 4 colores utilizable desde el BASIC.

Area de memoria: 3072 bytes.

Resolución: 128 columnas por 96 filas.

Configuración del byte: Como en el modo 7.

Colores: Como en el Modo 7.

Acceso: PMODE 1.

Modo 12

Tipo: Gráfico.

Descripción: Modo gráfico con dos colores utilizable desde el BASIC.

Area de Memoria: 3872 bytes.

Resolución: 128 columnas por 192 filas.

Configuración del byte: Como en el modo 8.

Colores: Como en el modo 8.

Acceso: PMODE 2.

Modo 13

Tipo: Gráfico.

Descripción: Modo gráfico con 2 colores utilizable desde el BASIC.

Area de memoria: 6144 bytes.

Resolución: 128 columnas por 192 filas.

Configuración del byte: Como en el modo 7.

Colores: Como en el modo 7.

Acceso: PMODE 3.

Modo 14

Tipo: Gráfico.

Descripción: Modo gráfico de máxima resolución con 2 colores utilizable desde el BASIC.

Area de memoria: 6144 bytes.

Resolución: 256 columnas por 192 filas.

Configuración del byte: Como en el modo 8.

Colores: Como en el modo 8.

Acceso: PMODE 4.

Apéndice E. Mapas de Memoria

En el modo de 32L el mapa de memoria del Dragón 200 es el siguiente:

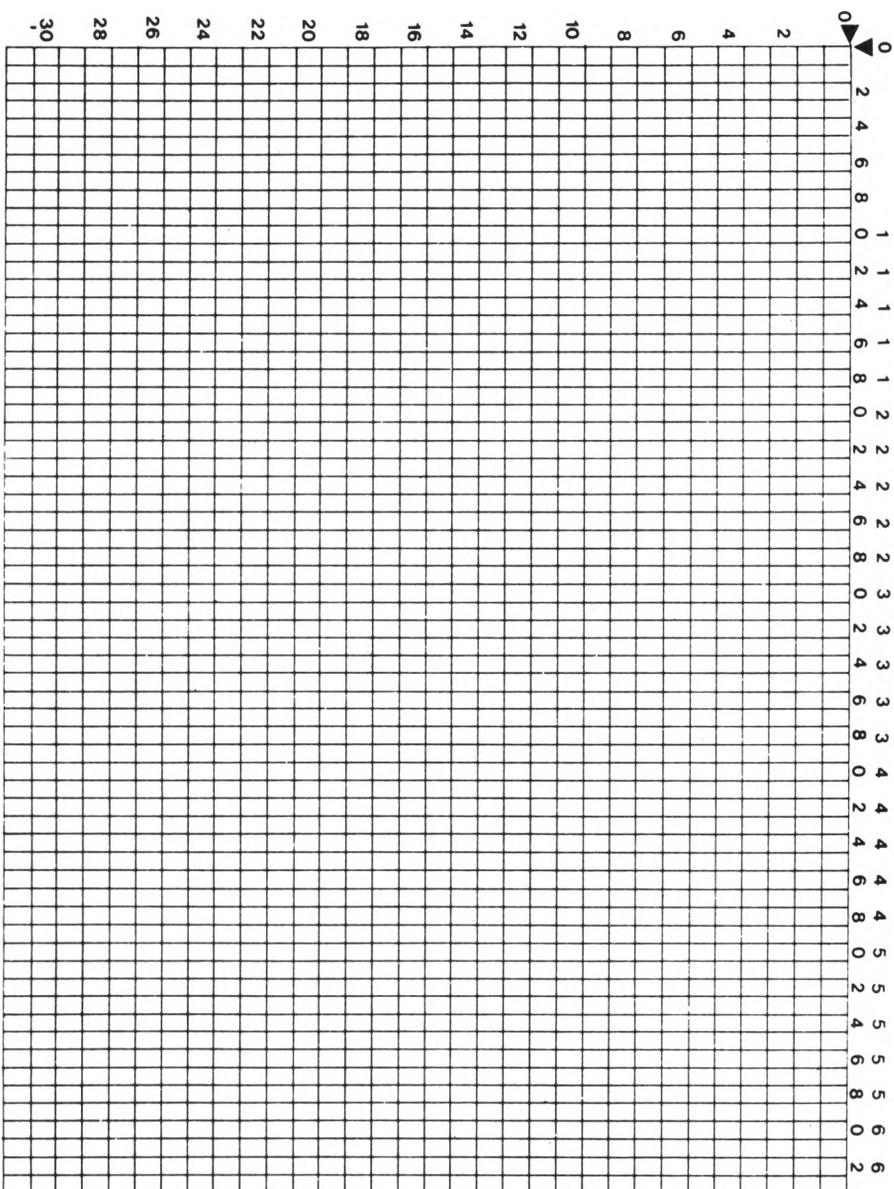
\$00	—	\$FF	: Página del sistema usada por el BASIC.
\$100	—	\$1FF	: Página RAM Extendida. Punteros de Drives I/O.
\$200	—	\$3FF	: Buffers de cassette, teclado, etc...
\$400	—	\$5FF	: Pantalla de texto.
\$600	—	\$BFF	: 1. ^a Página de Gráficos.
\$C00	—	\$11FF	: 2. ^a Página de Gráficos.
\$1200	—	\$17FF	: 3. ^a Página de Gráficos.
\$1800	—	\$1DFF	: 4. ^a Página de Gráficos.
\$1E00	—	\$23FF	: 5. ^a Página de Gráficos.
\$2400	—	\$29FF	: 6. ^a Página de Gráficos.
\$2A00	—	\$2FFF	: 7. ^a Página de Gráficos.
\$3000	—	\$35FF	: 8. ^a Página de Gráficos.
\$3600	—	\$7FFF	: Programas y variables.
\$8000	—	\$BFFF	: Intérprete BASIC en ROM.
\$C000	—	\$FEFF	: Cartuchos ROM.
\$FF00	—	\$FFBF	: Entrada/Salida.
\$FFC0	—	\$FFDF	: Bits de control SAM.
\$FFE0	—	\$FFFF	: Vectores MPU.

Al ejecutar la subrutina contenida en la dirección \$BB80 (48000 en decimal) y pasar al modo de 48K el mapa de memoria se ve afectado desde la dirección \$3600 a la \$FEFF, conservando el resto de su estructura. En esta sección se configura como sigue:

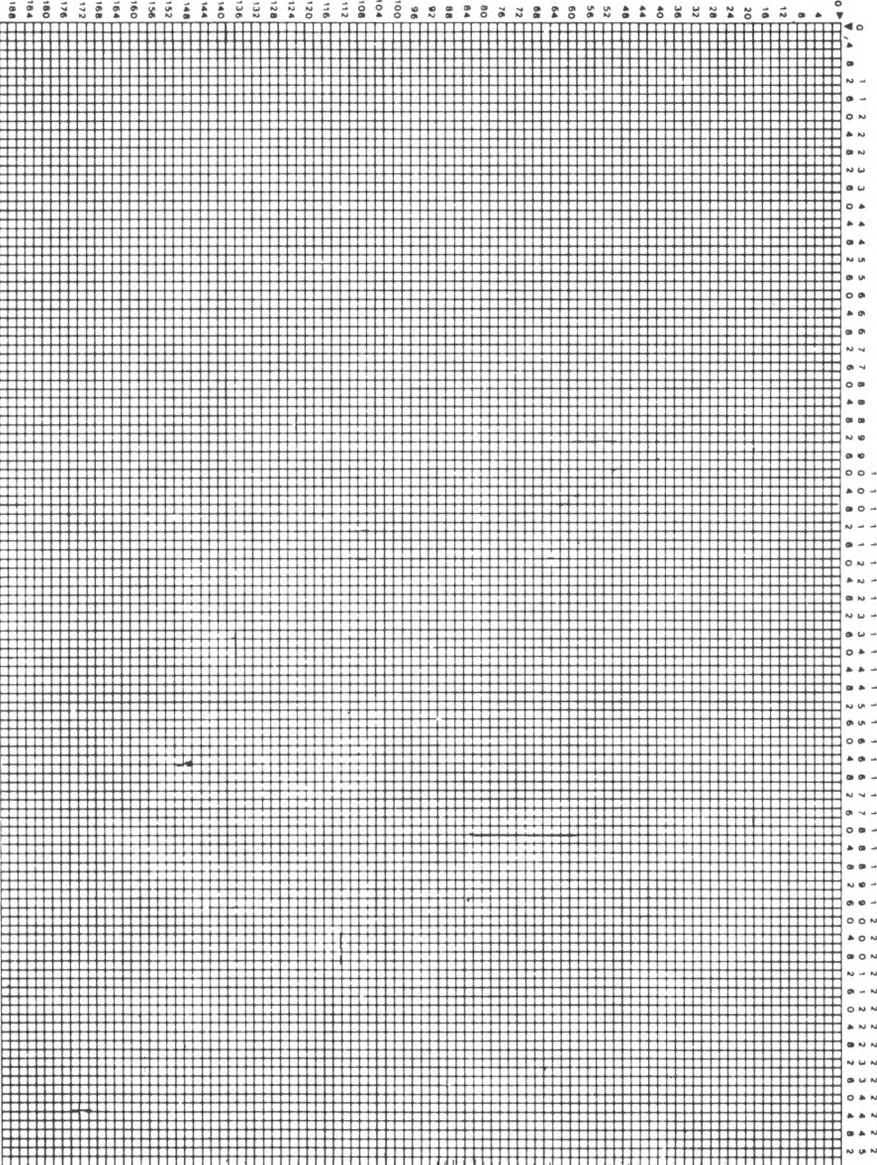
\$3600	—	\$BFFF	: Programas y variables.
\$C000	—	\$FEFF	: Intérprete de BASIC en RAM.

Apéndice F. Retículas

Retícula de Baja Resolución



Retícula de Alta Resolución



Notas



Eurohard

Españoleto, 25

Teléfs. (91) 410 30 64 - 410 31 96 - 410 34 98

Télex 45845 ICSG E

28010 MADRID