Talent

Extensiones MSX2 Basic

Talent

Extensiones de MSX2 Basic

TELEMATICA S.A.

Chile 1347 – Tel: 37-0051 al 54
1098 – BUENOS AIRES – ARGENTINA
ISBN 950-9688-06-X

Impreso en Argentina
Printed in Argentina

Hecho el depósito que marca la Ley 11.723

Licencia Telemática S.A. de usuario final:

El otorgante concede al usuario final una licencia consistente en el derecho a utilizar el "software" de acuerdo a los siguientes términos:

1. El usuario final solo podrá emplear el "software" en un solo sistema de computadora en cada momento dado.
2. Se permite al usuario final la transferencia a terceros de esta licencia y del "software" siempre que:

(a) Dicho tercero se avenga a todos los términos del presente acuerdo

(b) El usuario final no retenga en su poder copia alguna del "software".

Telemática S.A. no asume ningún tipo de responsabilidad sobre el uso y aplicaciones de este software, o por errores en este manual o en su software. Puede existir, sin embargo soporte adicional y/o garantías extras de parte de Microsoft Corporation o ASCII Corporation sobre este software. Telemática no tiene ninguna relación en ello, y no asume ninguna responsabilidad o garantía relacionada con lo atendido.

Este manual está sujeto a cambios sin previo aviso y no constituye una obligación para Telemática S.A. informar sobre estos cambios.

INDICE

INTRODUCCION

La computadora personal MSX 2 es una versión avanzada del sistema MSX con características ampliadas, especialmente en la parte gráfica: resolución, cantidad de colores, velocidad, posibilidad de superposición y digitalización de imágenes de video. Asimismo se dispone de un sistema de disco RAM, con lo cual se aprovecha al máximo la RAM no utilizada por MSX BASIC, y un reloj permanente de tiempo real.

Todos los programas confeccionados a través del sistema MSX aún utilizando directamente código de máquina pueden ser ejecutados con total compatibilidad bajo el sistema MSX2 en todos sus medios de almacenamiento (discos, cassettes o cartuchos).

MSX-BASIC

Para poder soportar el nuevo VDP (Video Display Processor), la mayor cantidad de RAM y el reloj, el lenguaje BASIC fue ampliado de la version 1.0 a la versión 2.0, por supuesto manteniendo su compatibilidad, pero al utilizar las nuevas sentencias o las extensiones de las sentencias anteriores es necesario tener en cuenta que algunos parámetros pueden ser distintos.
La memoria de la MSX 2 está dividida en tres clases: ROM, RAM y VRAM.

ROM (Read Only Memory - Memoria de lectura solamente):

La MSX 2 admite 48 kbytes de ROM como parte de la norma. Como en MSX 1 se admitía 32 kbytes, la ROM del sistema admite 16 kbytes extra, utilizados para soportar todas las instrucciones expandidas MSX2.

RAM (Random Access Memory - Memoria de acceso aleatorio):

La MSX 2 puede manejar 128K bytes de memoria RAM, lo que permite una mayor capacidad de almacenamiento de datos y programas. Dicha memoria extra puede accederse como si fuera una

unidad de discos, permitiendo aprovecharla almacenando programas, datos, etc.

VRAM (Video RAM - RAM para video):

Al poseer capacidades ampliadas en la parte gráfica, se requiere una mayor capacidad de memoria de video, que puede variar de 64 kbytes a 128 kbytes o más, según sea el modelo de MSX2. Gracias a esta capacidad extra de memoria gráfica, se puede utilizar la computadora en un sinnúmero de aplicaciones audio-visuales, como puede ser superposición de imágenes y digitalización de las mismas.

NUEVAS POSIBILIDADES.

1) Gráficas.

Mayor resolución.

Las nuevas capacidades gráficas de la MSX2 permiten obtener una alta resolución en gráficos, ya sea en cantidad de puntos por pantalla como cantidad de colores y tonalidades posibles. Esta mayor flexibilidad le da un alto realismo a los gráficos obtenibles en su computadora, permitiendo con unas pocas sentencias BASIC copiar sectores de pantalla, cambiar colores utilizando operaciones lógicas, 80 columnas en el modo texto, etc. La mayor capacidad de VRAM permite, asimismo, "paginar" esta memoria haciendo posible almacenar simultáneamente más de una pantalla en memoria, pudiéndose conmutar cualquier página para convertirla en la activa. Esto permite simular movimiento y agilizar las visualizaciones por pantalla, lográndose un máximo aprovechamiento de la memoria VRAM.

Ochenta columnas

MSX2 incorpora en forma estándar la presentación en pantalla de texto en 80 columnas, ya sea en MSX-BASIC o en el sistema operativo de discos MSX-DOS, posibilitando el uso directo de sistemas CP/M o similares.

Sprites multicolores.

Ahora los sprites poseen múltiples colores, con lo que se logra mayor realismo en la animación de imágenes por computadora.

Superposición de imágenes de video.

Con el nuevo procesador de video de la MSX2, se incorpora la posibilidad de superposición de imágenes, es decir, mezclar imágenes de video o televisión con lo que muestra la computadora, permitiendo subtitular películas o crear efectos especiales, todo ello utilizando la interfaz adecuada y programando en MSX- BASIC con las sentencias expandidas correspondientes.

Digitalización de imágenes.

Asimismo, la nueva norma prevee que la computadora con los periféricos adecuados pueda hacer digitalización de imágenes, es decir, tomar una fuente de video externa y almacenarla, digitalizada, en la memoria de video de la computadora, haciendo de su Talent MSX2 la herramienta idónea para publicidad, arte, video, etc.

2) Memoria.

Al decidir ampliar la capacidad de memoria de la norma MSX en su versión 2, se le brinda al usuario la capacidad de una computadora profesional, pudiendo almacenar mayores volúmenes de datos y programas. Por otra parte, se ha incorporado la posibilidad de acceder a la memoria que MSX-BASIC no utiliza sin tener que recurrir al lenguaje de máquina, mediante el nuevo dispositivo denominado "MEM:", que permite simular la utilización de un cassette o disco en la memoria, con la consiguiente facilidad de manejo y velocidad.

3) Reloj Interno Incorporado.

Se ha incorporado un reloj calendario que mantiene la fecha (día y hora) de operación de la máquina actualizada, permitiendo a la vez fechar los trabajos y almacenar datos del sistema sin que se pierdan cuando se apaga la computadora.

DIFERENCIAS Y AGREGADOS AL MSX BASIC QUE CONFORMAN LA VERSION 2.0

En este Manual se presentan todas las diferencias entre las sentencias para MSX Basic versión 1.0 y la versión 2.0, así como las sentencias que se han agregado al lenguaje a partir de esta nueva versión.

Las sentencias ya existentes en la versión 1.0 se vuelven a explicar agregando las ampliaciones ocurridas en la versión 2.0 y las nuevas sentencias se explican con detalle.

SENTENCIAS PARA GRAFICOS

BASE

BASE(<n>) (tipo: enterro)

Devuelve la primer dirección de las tablas de la memoria de procesador de pantalla. <n> varia de 0 a 44.

INT(<n>/5) determina el modo de pantalla. <n> MOD 5 determina el tipo de tabla.

EJEMPLO:

0-19 Idem MSX v.1
…
40 - base de tabla de imagen de pantalla para modo 8.
41 - base de tabla de color del modo 8.
42 - base de la tabla de generador de diseños para el modo 8.
43 - base de la tabla de atributos de sprite para el modo 8.
44 - base de la tabla de diseños de sprites para el modo 8.

Todos los modos de pantalla son válidos cuando se leen las direcciones de una tabla.

Cuando se coloca una dirección base, los valores válidos para <n> son de 0 a 19. Es decir, sólo los modos de pantalla 0 a 3 son válidos. Cuando se cambia la dirección de la base de alguno de los modos 2 o 4, la dirección base del otro modo también cambia.

Cuando se desea calcular la dirección de video RAM física para los modos 5 a 8 se debe utilizar un algoritmo según la siguiente tabla:

SCREEN	algoritmo de cálculo
5	No.Página X 8000H +(dirección base)
6	No. Página X 8000H + (dirección base)
7	No. Página X 10000H + (dirección base)
8	No.Página X 10000H + (dirección base)

Nota:

Las variables BASE(< n>) corresponden a variables del sistema que serán evaluadas o asignadas como cualquier variable común. Se han incluido para programadores avanzados únicamente.

CIRCLE

CIRCLE <espec. coord.>, <radio> [,<color> [,<ángulo inic.> [,<ángulo final>[, <excentricidad>]]]]

Para dibujar una elipse con un centro y un radio como se indica en los dos primeros argumentos.

<espec. coord.> puede tomar una de las siguientes formas:

STEP (offset x, offset y) o
(x absoluto, y absoluto)

La primera forma es una coordenada relativa al último punto referenciado. La segunda forma es la más común y son coordenadas absolutas, referidas al punto (0,0). Ejemplos:

(10,10) forma absoluta
STEP (10,0) offset: 10 en x y 0 en y.
(0,0) origen.

Nótese que cuando Basic analiza los valores de coordenadas permite que éstos estén fuera de rango, sin embargo los valores que se encuentren fuera del rango entero (-32768 a 32767) causarán un error de desborde (overflow). Los valores fuera de la pantalla serán sustituidos con el valor más próximo posible.

Tenga en cuenta que la coordenada (0,0) es el borde superior izquierdo. La coordenada X puede tomar valores de 0 a 255 para los modos de pantalla 2,3,4,5 y 8, y 0 a 511 para modos de pantalla 6 y 7. La coordenada Y puede tomar valores de 0 a 191 para los modos de pantalla 2,3 y 4, y 0 a 211 para los modos 5, 6, 7 Y 8.

La descripción que se ha hecho de las coordenadas es aplicable para todas las sentencias gráficas.

El <color> por defecto es el color de frente.

El <ángulo inic.> y el <ángulo final> están descriptos en radianes y varían entre 0 y 2xPI, lo que permite especificar dónde comienza y termina la sección de elipse. Si el ángulo inicial o final es negativo, la elipse será conectada con el punto central con una línea, y los ángulos serán tratados como si fueran positivos. (Nótese que es diferente a sumar 2xPI).

La <excentricidad> es el cociente entre los radios horizontales y verticales de la elipse.

COLOR

FORMATO 1:

`COLOR [<color de frente>, [<color de fondo>[,<color de bordes>]]]`

Para definir los colores de pantalla. Los valores por defecto son 15,4,4. El argumento es un número en el rango de 0...15. Sólo si el modo de pantalla es el 8, el rango puede ser de 0...255. Los verdaderos colores correspondientes a cada valor ya se han descripto en el manual de MSX BASIC.

Cuando dibujamos, utilizamos una paleta cromática para mezclar colores y obtener diferentes tonalidades. La MSX2 posee 16 paletas cromáticas, En cada paleta cromática es posible exponer los colores primarios (rojo, verde y azul) con 8 grados distintos de brillo. Cuando se mezclan, se pueden generar hasta 512 colores diferentes (8x8x8).

El número de color utilizado por la instrucción COLOR no refleja el color exacto, sino el número de paleta cromática.
Se pueden mostrar 16 colores de los 512 cambiando el color de la paleta. En MSX2 BASIC, al iniciarse la ejecución del sistema o con el comando COLOR=NEW, se genera una mezcla de rojo. verde y azul que corresponde a los 16 colores codificados de la siguiente forma:

Número de paleta	**Color**	**Nivel Rojo**	**Nivel Verde**	**Nivel Azul**
0	transparente	0	0	0
1	negro	0	0	0
2	verde mediano	1	1	6
3	verde claro	3	3	7
4	azul oscuro	1	7	1
5	azul claro	2	7	3
6	rojo oscuro	5	1	1
7	cian	2	7	6
8	rojo mediano	7	1	1
9	rojo claro	7	3	3
10	amarillo oscuro	6	1	6
11	amarillo claro	6	4	6
12	verde oscuro	1	1	4
13	magenta	6	5	2
14	gris	5	5	5
15	blanco	7	7	7

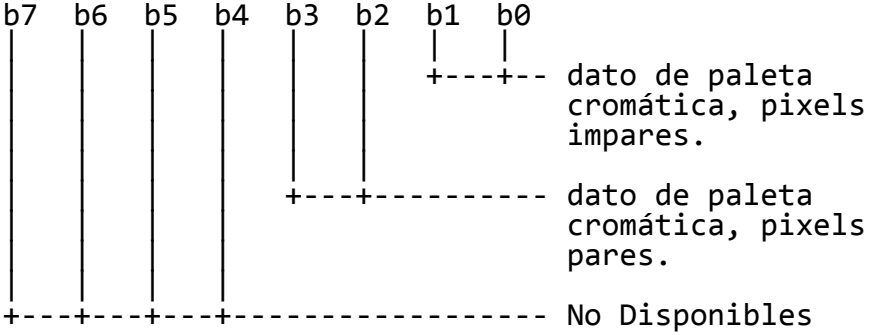
El número de paleta cromática que se utiliza varía de acuerdo al modo de pantalla. SCREEN 0, 1, 2, 3, 4, 5 y 7 utilizan 16 paletas, pero el modo 6 utiliza 4.

Para modos de pantalla "bit mapped" (mapeados por puntos, p.ej. modo 5 a 8), el valor especificado y el color realmente mostrado se corresponden de la siguiente forma:

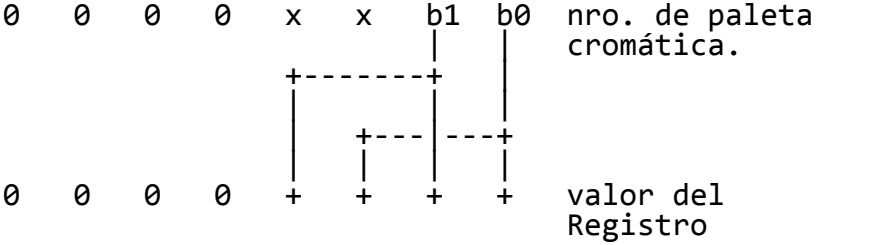
Modo de pantalla 6

El valor de color puede estar entre O y 3. El color realmente mostrado se define con el contenido de la tabla de búsqueda de color que se maneja con la sentencia "COLOR=" que se describe posteriormente. Los valores por defecto son los mismos que los que se encuentran en la tabla descripta anteriormente.

En el modo de pantal1a 6, los distintos valores de paleta cromática pueden utilizarse como color de borde y sprite, dependiendo de la coordenada X que se declara: si es par o impar. Esto se denomina "Hardware Tiling" (Tramado de Hardware). El contenido del registro de color de borde se corresponde con el color de la pantal1a de la siguiente manera:



Cuando el color de borde designado es de 0 a 15, los 2 bits menos significativos forman el número que se utiliza para los pixels pares e impares. Todos los pixels del borde tienen el mismo color.



Cuando el valor indicado de borde es desde 16 a 31, los 4 bits menos significativos se envían directamente al registro. Aparecen rayas verticales en el borde.

```
0   0   0   1   b3  b2  b1  b0   nro. de paleta
                 |   |   |   |   cromática.
                 |   |   |   |
0   0   0   0   +   +   +   +    valor del
                                 registro
```

El color de borde se cambia cuando se ejecuta la sentencia COLOR. El color de fondo cambia cuando se ejecuta la sentencia CLS.

Los registros de color de fondo y de borde no se inicializan con la sentencia SCREEN. Las rayas verticales aparecen cuando se ejecuta "SCREEN 6". Utilice la sentencia COLOR para inicializar los registros.

Modos de pantalla 5 y 7

El valor del color puede estar entre 0 y 15. El color que se muestra corresponde a la tabla de búsqueda de color que puede modificarse con la sentencia "COLOR=". Los valores por defecto son los mismos que los de la tabla descripta anteriormente.

La relación entre el color y su número es la que sigue:

Número de color (decimal)	Binario b7 b6 b5 b4 b3 b2 b1 b0	Verde b7 b6 b5	Rojo b4 b3 b2	Azul b1 b0
227	1 1 1 0 0 0 1 1	1 1 1	0 0 0	1 1

Modo de pantalla 8.

El color puede variar entre O y 255. El color que se mostrará por pantalla se determina con la siguiente fórmula:

<color>=32xG+4xR+B

Donde R, G y B son las intensidades de Rojo (Red), Verde (Green) y Azul (Blue), respectivamente. Y toman un valor de 0 a 7 para rojo y verde, 0 a 3 para azul.

FORMATO 2:

```
COLOR=(<nro. Paleta cromática>, <rojo>, <verde>, <azul>)
```

Para cambiar la tabla de búsqueda de colores. La tabla de búsqueda de colores traduce el código de color especificado por las sentencias graficas a los colores mostrados realmente por pantalla. Esta función está disponible para todos los modos de pantalla excepto el modo 8 (modo 256 colores por punto). El número de paleta croml1tica toma un valor de 0 a 15 y cada uno corresponde al código de color. En el modo de pantalla 6, que tiene sólo 4 colores por pixel, el número de paleta cromática toma un valor de 0 a 3 únicamente.

```
<rojo>  = 0 a 7
<verde> = 0 a 7
<azul> = 0 a 3
```

El número de paleta croml1tica O se asigna usualmente al color transparente. Este número de paleta cromática puede utlizarse como los otros cuando el bit TP del registro de VDP 8 está habilitado. Tenga en cuenta que no debe modificarse los otros bits de este registro.

```
VDP(9)=VDP(9)   OR   &H20   'habilita   paleta
cromática 0
VDP(9)=VDP(9)  AND  &HDF  'deshabilita  paleta
cromática 0
```

FORMATO 3:

```
COLOR [=NEW]
```

Para restaurar los valores por defecto de la tabla de búsqueda de colores descripta anteriormente.

FORMATO 4:

```
COLOR = RESTORE
```

Para enviar a la tabla de búsqueda de colores los valores almacenados en la VRAM. Esta sentencia es útil cuando los datos de paleta cromática han sido cargados con la sentencia "BLOAD ,S" y Ud. desea cambiar la paleta cromática a estos valores.

FORMATO 5:

```
COLOR SPRITE$ (<nro, plano>)= expression alfanumérica>
```

Para modificar el color de cada línea de un sprite. La <expresión alfanumérica> puede tener de 1 a 16 caracteres. Cada carácter corresponde con cada línea del sprite, y toma los valores que se describen a continuación en su codificación binaria. Esta sentencia tiene validez para los modos de pantalla 4 a 8.

b7	Cuando vale 1, corre el sprite 32 pixels a la izquierda
b6	Cuando vale 1, ignora la prioridad y coincidencia de sprites y utiliza operaciones "or" para los códigos de color cuando los sprites coinciden.
b5	Cuando vale 1, ignora la coincidencia de sprites.
b4	sin uso
b3-b0	código de paleta cromática.

Ejemplo

```
COLOR SPRITE$(0)=CHR$(1)+CHR$(7)
```

Coloca el color de la primer linea del plano O como el código de paleta cromática 1, y el color de la segunda línea del mismo plano paleta cromática 7. Las otras líneas del plano permanecen sin cambios.

FORMATO 6:

```
COLOR SPRITE(<nro. Plano>)=<va1or>
```

Para colocar el valor del plano completo especificado en el número de paleta cromática indicado por <valor>. Esta sentencia tiene validez en los modos 4, 5, 6, 7 y 8.

El <valor> debe estar en el rango de 0 a 127 y se almacena en el registro que se describió anteriormente. El bit 1 (MSB) no puede especificarse con esta sentencia. Cuando el valor de C% varía de 0 a 127, las siguientes sentencias tienen el mismo efecto:

```
COLOR SPRITE$(P%)STRING$(16,C%)
COLOR SPRITE(P%)=C%
```

Nota:

Los colores de los sprites pueden definirse con las sentencias COLOR SPRITE$, COLOR SPRITE o PUT SPRITE. El color que se establece es el indicado en la última de estas sentencias que se haya ejecutado. Si desea utilizar la sentencia "COLOR SPRITE$" para definir los colores de un sprite, omita el número de color en la sentencia "PUT SPRITE".

COPY

```
COPY (X1,Y1)-(X2,Y2) [,<página origen>] TO
(X3,Y3) [,<página destino> [,<operación
lógica>]]

COPY (X1,Y1)-(X2,Y2) [,<página origen>] TO
<variable dimensionada I espec. arch.>
```

COPY <variable dimensionada | espec. arch. > [, <dirección>] TO (X3, Y3) [, <página destino>, [<operación lógica>]]

COPY <variable dimensionada> TO <espec. arch.>

COPY <espec. arch.> TO <variable dimensionada>

Para transferir datos de video entre VRAM, variables dimensionadas y archivos.

Las coordenadas (X1,Y1)-(X2,Y2) especifican el área de pantaIla origen en forma análoga al comando LINE. La coordenada (X3,Y3) especifican el punto inicial del área de destino.

La coordenada (X1,Y1) especifica el punto inicial de transferencia y (X2,Y2), el punto final. Existen 4 tipos de direcciones de transferencias aún cuando (X1,Y1) y (X2,Y2) especifiquen la misma área. Cuando copia a VRAM desde un ordenamiento (array) o archivo, <dirección> especifica la dirección de transferencia.

0	desde arriba izquierda	hacia abajo derecha.
1	desde arriba derecha	hacia abajo izquierda.
2	desde abajo izquierda	hacia arriba derecha.
3	desde abajo derecha	hacia arriba izquierda.

Cuando la <página destino> es diferente de la <página origen>, se transfiere una imagen gráfica desde una página a la otra. Vea también la sentencia SET PAGE.

Si no se especifica un número de página, se asume la página activa.

<variable dimensionada> es un nombre de variable. Debe ser un ordenamiento de variables numéricas y debe tener suficiente espacio de memoria para que se puedan almacenar los datos. El tamaño de los datos de video es:

INT((<tamaño pixel> * (ABS(X2 - X1) + 1) * (ABS((Y2 - Y1) + 1) + 7) / 8) + 4) bytes

<tamaño pixel> es el número de bits utilizado por un pixel. Su valor es 4 en los modos 5 y 7. 2 en el modo 6, 4 en el modo 8. El formato de los datos de video es:

Offset	Contenido
0	Byte menos significativo del tamaño horizontal
1	Byte más significativo del tamaño horizontal
2	Byte menos significativo del tamaño vertical
3	Byte más significativo del tamaño vertical
4-	Dato de imagen de bit

Se agregan 0 si los bits permanecen en el último byte.

<operación lógica> es una de las siguientes:

```
XOR     :CP=NOT(C)*CP+C*NOT(CP)
OR      :CP=C+CP
AND     :CP=C*CP
PSET    :CP=C
PRESET  :CP=NOT(C)
TXOR    :-+Es lo mismo que lo
TOR     : | anterior, pero el
TAND    : | color transparente
TPSET   : | no tiene ninqdn efecto. Es decir,
TPRESET :-+ el pixel modificado por el color
           transparente permanece con el color
           original.
```

Donde:

```
C   :  <color>
CP  :  color pantalla
```

Si no se especifica ninguna operación. se aplica PSET.

LINE

LINE [<espec. coord.>]-<espec. coord. > [, <color>[, <B|BF>[, <operación lógica>]]]

Para dibujar una línea conectando dos puntos de coordenadas especificadas. Para más detalle de <espec. coord.>. véase la sentencia CIRCLE.

Si se especifica 'B', se dibuja un rectángulo. Si se especifica 'BF', se pinta el rectángulo.

Para <operación lógica>, véase la sentencia COPY.

PAD

PAD (<n>) (tipo: entero)

Devuelve el estado del touch pad. lápiz óptico y mouse/trackball.

Cuando <n> varia de 0 a 7, se obtiene el estado del touch pad. Los valores de 0 a 3 seleccionan el pad conectado en la entrada de joystick 1, 4 a 7 seleccionan la entrada de joystick 2.

Cuando <n>=0 ó 4 el estado del touch pad se obtiene de la siguiente forma: -1 = al presionar el touch pad, 0 = cuando está liberado.

Cuando <n>= 1 ó 5, se obtiene la coordenada X. Cuando <n>= 2 ó 6, se obtiene la coordenada Y.

Cuando <n>= 3 o 7, se obtiene el estado de los botones del touch pad: -1 cuando se pulsan, en caso contrario 0.

Nótese Que las coordenadas tienen validez solamente cuando se evalúan PAD (0) o PAD (4). Cuando se lee PAD (0), PAD (5) y PAD (6) quedan

afectados, y con PAD (4) se afectan PAD (1) y PAD(2) .

Cuando <n> toma valores entre 8 y 11, se obtiene el estado del lápiz óptico. Asegúrese que PAD (8) devuelva -1 antes de leer cualquier dato.

8	-1 si es dato válido, 0 si no lo es. Leer los valores de X e Y.
9	Posición X.
10	Posición Y.
11	-1 si se pulsa el botón, 0 si no está pulsado.

Cuando <n> se encuentra entre 12 y 15, se obtiene el estado del mouse/trackball conectado al pórtico de entrada/salida 1 (entrada de joystick 1). Se obtienen datos cuando se ejecuta PAD (12).

Cuando <n> varia de 16 a 19, se obtiene el estado del mouse/trackball conectado al pórtico de entrada/salida 2. Se obtienen datos cuando se ejecuta PAD (16).

12,16	Devuelve siempre -1. Leer los valores de X e Y.
13,17	Posición X.
14,18	Posición Y.
15,19	Devuelve siempre 0.

Utilice la función STRIG para leer el estado de los botones del mouse o del trackball.

Nota:

Los datos del mouse y del trackball se leen solamente cuando se ejecutan las instrucciones PAD (12) o PAD (16). Si el intervalo de tiempo para lectura es muy largo, el contador interno del mouse/trackball se desborda y la función devolverá un valor incorrecto.

La longitud .del contador interno es de 8 bits para el mouse y de 4 para el trackball.

El software del sistema (BIOS) reconoce automáticamente que dispositivo (mouse/trackball) está conectado.

PAINT

PAINT <espec. coord.>[,<color de superficie>] [,<color del borde>]

Para pintar una figura gráfica arbitraria con el color de superficie especificado, comenzando en la <espec. coord.>. Para más detalles del <espec. coord.>, véase la sentencia CIRCLE. PAINT no permite que <espec. coord.> esté fuera de rango.

Tenga en cuenta que no se puede especificar <color de borde> para el modo de pantalla 2, sólo se permite para los modos 3, 4, S, 6, 7. En el modo alta resolución (modo 2), se asume que el color de borde coincide con el color de superficie.

PSET, PRESET

PSET <espec. coord>[,<color>][,<operación lógica>]
PRESET <espec. coord>[, <color>] [, <operación lógica>]

Para prender/apagar la coordenada especificada. Para más detalles del <espec. coord.>, véase la sentencia CIRCLE.

La única diferencia entre PSET y PRESET es que si no especifica <color> en la sentencia PRESET, se selecciona el color de fondo.

Cuando se indica el argumento <color>, PRESET es idéntico a PSET.

SCREEN

SCREEN [<modo>[, <tamaño sprite>[, <interruptor click de tecla> [, <velocidad baudios cassette>[, <opción impresora> [,<modo display>]]]]]]

Para asignar el modo de pantalla, tamaño de los sprite, click de las teclas, velocidad en baudios del cassette, la opción de una impresora y el modo de display.

Existen 9 modos diferentes de pantalla en la Talent MSX2, dos son modos para textos y los siete restantes son modos gráficos.

<modo> puede tener los siguientes valores correspondientes a cada modo de pantalla:

Modo 0:

modo texto 40 col x 24 filas u 80x24 (véase también la sentencia WIDTH)

Sus caracteres se forman con una matriz de 5x7 puntos que se obtienen de un formato de 6x8 puntos. Por lo tanto, existe un espacio de un punto entre las letras y entre las líneas. En este modo no se pueden utilizar las sentencias y comandos de gráficos.

Es el modo por defecto cuando se enciende la computadora.

Modo 1:

modo texto 32x24

Sus caracteres se forman con 8x8 puntos.

Todos los comandos y sentencias que utilizan gráficos no funcionan salvo las referidas a sprites.

Modo 2:

modo alta resolución

Se pueden mostrar puntos de colores en una matriz de 256 puntos en horizontal por 192 puntos en vertical.

Se pueden utilizar 16 colores diferentes en este modo.

Modo 3:

modo multicolor
Posee 64x48 bloques de color consistentes de 4x4 puntos.
Cada bloque puede tener cualquiera de los 16 colores.

Modo 4:

modo gráfico 3
Es idéntico al modo alta resolución con las funciones para sprites mejoradas. Se utiliza el modo 2 de las funciones de sprites.

Modo 5:

modo gráfico 4 (bitmap 256x212, 16 colores elegibles de 512 colores)
Por cada pixel, se pueden elegir 16 colores de los 512 disponibles para señalar el color. Si se utiliza el modo entrelazado, el mapa de bits se expande a 256x424 puntos.
Se utiliza el modo 2 de funciones de sprites. La memoria de video queda dividida en 4 páginas (números 0-3) y se utilizan. Es posible copiar pantallas a la memoria RAM.

Modo 6:

modo gráfico 5 (bitmap 512x212, 4 colores elegibles de 512 colores).
Por cada pixel, se pueden elegir 4 colores de los 512 disponibles para señalar el color. Si se utiliza el modo entrelazado, el mapa de bits se expande a 512x424 puntos.
Se utiliza el modo 2 de funciones de sprites.
La memoria de video queda dividida en 4 páginas (números 0-3) y se utilizan. Es posible copiar pantallas a la memoria RAM.

Modo 7:

modo gráfico 6 (bitmap 512x212, 16 colores elegibles de 512 colores).
Por cada pixel, se pueden elegir 16 colores de los 512 disponibles para
señalar el color. Si se utiliza el modo entrelazado, el mapa de bits se expande a 512x424 puntos.
Se utiliza el modo 2 de funciones de sprites.
La memoria de video queda dividida en 2 páginas (números 0-1) Y se utilizan. Es posible copiar pantallas a la memoria RAM.

Modo 8:

modo gráfico 7 (bitmap 256x212, 256 colores)
Por cada pixel, se pueden utilizar 256 colores para pintarlo.
Si se utiliza el modo entrelazado, el mapa de bits se expande a 512x424 puntos.
Se utiliza el modo 2 de funciones de sprites.
La memoria de video queda dividida en 2 páginas (números 0-1) Y se utilizan. Es posible copiar pantallas a la memoria RAM.

Se requieren 128 kbytes de VRAM para los modos de pantalla 7 y 8.

Nótese que en el modo texto 32x24, todos las sentencias gráficas excepto 'PUT SPRITE' generan un error "Illegal function call" (llamado a función incorrecto). Además, todos los modos de pantalla se fuerzan a modo texto cuando se encuentra una sentencia 'INPUT' o BASIC regresa al nivel comando.

Además, los modos 0 a 3 (excepto cuando se tiene texto de 80x24) son totalmente compatibles con MSX v.l. Véase el Manual del MSX BASIC para mis detalle.

Las opciones <tamaño de sprite>, <interruptor click de tecla>, <velocidad baudios cassette> y <opción de impresora> son idénticas a las de la sentencia SCREEN en MSX v.l. Para mis detalles de estas opciones, véase el Manual del MSX BASIC

<modo display> determina el modo de VDP para pantallas entrelazadas de la siguiente manera:

0	Normal
1	Entrelazado
2	No entrelazado, pantallas par/impar alternadas.
3	Entrelazado, pantallas par/impar alternadas.

Para el modo par/impar, p.e. modos 2 y 3, el número de página de pantalla debe ser impar, y la página de pantalla indicada con la de número inferior en uno se alternarán en pantalla. (véase la instrucción SET PAGE para más detalles).

La sentencia SCREEN borra la pantalla y los sprites en la página 0 pero los diseños de sprites no se alteran.

SET PAGE

SET PAGE <página de pantalla>, <página activa>

Para seleccionar una página de pantalla como activa y que debe ser mostrada por pantalla. Esta sentencia está disponible para los modos de pantalla 5 a 8. SET PAGE no modifica la VRAM.

Una página activa es la página donde los comandos de entrada/salida tienen efecto. Una página de pantalla es la página que se muestra en la imagen de la pantalla. Los números máximos de pantalla varían de acuerdo al modo de pantalla y al tamaño de VRAM de la siguiente forma:

RANGO DE NUMEROS DE PAGINA

MODO DE PANTALLA	64KB VRAM	128KB VRAM
5	0 a 1	0 a 3
6	0 a 1	0 a 3
7	NO DISPONIBLE	0 a 1
8	NO DISPONIBLE	0 a 1

VDP

VDP (<n>) (Tipo de función: byte sin signo)

Para obtener el valor actual de un registro de VDP o para enviar un valor a un registro.

n	Número de registro
0-8	0-8 (idem MSX v.l)
9-24	8-23
33-47	32-46
-1- -9	registros de estado 1-9.

El valor <n> corresponde a un registro de VDP como se describe a continuación. Si se intenta escribir en el registro 8 se obtiene un error, ya que es un registro de lectura solamente.

Ejemplos:

```
PRINT VDP(3)
VDP(l)=&HE1
```

Nota:

Las variables VDP (<n>) corresponden a variables del sistema que serán evaluadas o asignadas como cualquier variable común. Se han incluido para programadores avanzados únicamente.

VPEEK

```
VPEEK(<direcci6n VRAM>)
```

Devuelve el valor almacenado en la dirección especificada de VRAM. <dirección VRAM> puede variar de 0 a 65535. Cuando el modo de pantalla es de 5 a 8, la dirección absoluta de VRAM que se accede es (<dirección VRAM> + dirección comienzo de página activa).

Ver también PEEK en MSX-BASIC

VPOKE

VPOKE <dirección>,<valor a escribir>

Envía el valor especificado en <valor a escribir> a la dirección especificada de VRAM.

<dirección VRAM> puede variar de 0 a 65535. Cuando el modo de pantalla es de 5 a 8, la dirección absoluta de VRAM que se accede es (<dirección VRAM> + dirección comienzo de página activa).

Ver también POKE en MSX-BASIC

WIDTH

WIDTH <ancho de pantalla en modo texto>

Para seleccionar el ancho de la pantalla (cantidad de caracteres por línea) para una pantalla de texto. Los valores admitidos son de 1 a 80 en el modo 0, y de 1 a 32 en el modo 1.

En el caso del modo de pantalla 0, la sentencia WIDTH automáticamente selecciona el modo de VDP. Cuando <ancho> es menor o igual a 40, se selecciona el modo de 40 caracteres por 24 líneas y se muestran caracteres grandes (modo análogo a MSX v.1). Cuando WlDTH es mayor que 40, se selecciona el modo de 80 caracteres por 24 lineas y se muestran caracteres pequeños.

SENTENCIAS PARA SUPERPOSICION y DIGITALIZACION DE IMAGENES.

COPY SCREEN

COPY SCREEN [<modo>]

Para digitalizar una fuente de video externa, usando un equipo periférico adecuado.

Cuando <modo> es 0, se digitaliza un campo y se escribe a la página de memoria de pantalla.

Cuando <modo> es 1, dos campos (= un cuadro) se digitalizan y se escriben en la <número página de pantalla-l> y en la página de pantalla. En este caso, el número de página de pantalla debe ser impar.

El valor por defecto de <modo> es 0.

SET VIDEO

SET VIDEO <modo> [, <ym> [, <cb> [, <sync> [,<audio> [, <entrada video> [, <control AV>]]]]]]

Para colocar el modo de superposición de imágenes (super imposition), usando un equipo periférico adecuado.

<modo> toma valores de 0 a 3 de acuerdo a lo siguiente:

MODO	S1	S0	TP	FUNCION
0	0	0	0	COMPUTADORA
1	0	1	1	COMPUTADORA
2	0	1	0	IMPOSE
3	1	0	0	TV

S1, S0 y TP son los registros de VDP que son variados por esta sentencia. SYNC indica si la VDP utiliza sincronización interna o externa. FUNCION describe lo que se muestra por pantalla.

Cuando el modo es 0, la sincronización externa no está disponible. Cuando los modos son 1, 2 ó 3, la salida compuesta de la VDP no está disponible.

<ym> debe ser 0 ó 1. Cuando <ym> es 1, la intensidad de la TV se reduce a la mitad. Cuando <ym> es 0, la intensidad es completa.

<cb> debe ser 0 ó 1. Cuando <cb> es 0, el bus de color de la VDP se coloca en salida (output). Cuando <cb> es 1, el bus de color se coloca en entrada (input).

Se obtiene sincronización externa cuando <sync> es 1 e interna cuando es 0.

<audio> selecciona si se permite la mezcla de sonido externo con el sonido del PSG (Programmable Sound Generator) de la siguiente manera:

0	Sonido de la computadora solamente.
1	Mezcla entrada externa canal derecho.
2	Mezcla entrada externa canal izquierdo.
3	Mezcla entrada externa ambos canales.

<entrada video> selecciona la entrada desde una entrada de video externa. Con 0 se selecciona un conector RGB múltiple y 1 selecciona un conector RCA.

<control AV> selecciona la salida de control A V del conector múltiple RGB. Con 0 se lo desconecta, 1 lo conecta.

SENTENCIAS PARA ENTRADA/SALIDA

BSAVE ,S
BLOAD ,S

BSAVE <espec. archivo>, <dirección inicial>, <dirección final>,S

BLOAD <espec. archivo>, S

Para grabar/cargar el contenido de VRAM desde/hacia un archivo en disco.

Esta sentencia estd disponible para todos los modos de pantalla. En los modos 5 a 8, el contenido de la página activa es el que se graba/carga. El <espec. archivo> debe ser un archivo de disco, no puede ser un archivo de cassette. <dirección> puede variar de 0 a 65534 (&HFFFE).

OPEN

OPEN "<descr. dispositivo> [<nombre archivo>]" [FOR <modo>] AS [#] <nro. archivo>

Para reservar un área de memoria para I/O (INPUT/OUTPUT = ENTRADA/SALIDA) e indicar el modo en que se utilizará dicha área.

Esta sentencia abre (inicializa) un dispositivo para procesos posteriores. Actualmente, se pueden abrir los siguientes dispositivos:

CAS: Cassette
CRT: Pantalla
GRP: Pantalla gráfica
LPT: Impresora
MEM: Disco de memoria

Para utilizar el disco de memoria deberá ejecutarse la sentencia MEMINI. Véase la sección "Sentencias para el Disco de Memoria".

<modo> es uno de los siguientes:

OUTPUT: Especifica modo secuencial de salida
INPUT: Especifica modo secuencial de entrada
APPEND: Especifica modo secuencial de agregado

<nro. archivo> es una expresión entera cuyo valor se encuentra entre uno y el máximo número de archivos especificados por la sentencia MAXFlLES=.

<nro. archivo> es el número que será asociado al archivo a partir de ese momento y mientras se encuentre abierto, y es utilizado en las otras sentencias de I/O para referirse al mismo.

Se debe ejecutar una sentencia OPEN antes de poder utilizar cualquier sentencia de I/O de archivos (o cualquiera que requiera número de archivo):

PRINT#, PRINT# US1NG
INPUT#, UNE INPUT#
INPUT$, GET, PUT

SENTENCIAS PARA EL DISCO DE MEMORIA

MSX-BASIC 2.0 puede utilizar el área de RAM desde 0000H hasta 7FFFH como un disco de memoria.

Se debe ejecutar la sentencia MEMINI antes de utilizar el disco de memoria. Las sentencias del disco de memoria son generalmente las mismas que para el Dísk-BASIC.

El especificador de archivo <espec> para el disco de memoria es:

MEM:<nombre archivo>[, <extensi6n>]

"MEM" es el nombre del dispositivo asignado al disco de memoria. Un nombre de archivo puede tener de 1 a 8 caracteres de largo. La extensión puede tener hasta 3 caracteres de largo. Si se omite la extensión, se debe omitir asimismo el "." (punto). El nombre de archivo y la extensión no pueden incluir ":" (dos puntos), "." (punto) o un carácter de control, p.e. los caracteres desde 01H a lFH, o los caracteres gráficos de 2 bytes.

MEMINI

CALL MEMINI [(<tamaño>)]

Para inicializar y borrar la memoria de disco. Esta sentencia debe ejecutarse antes de utilizar el disco de memoria.

<tamaño> es un número cuyo rango válido es desde 6655 hasta 98303. El valor por defecto es 98303. En el primer caso se asignan 256 bytes y en el segundo, 91904. Tenga en cuenta que cuando se inicializa el disco de memoria, se borran todos los archivos de disco.

"CALL MEMINI (0)" cancela la función del disco de memoria.

MFILES

```
CALL MFILES
```

Para mostrar los nombres de los archivos almacenados en el disco de memoria y el tamaño de la memoria libre.

MKILL

```
CALL MKILL (<espec>)
```

Para eliminar un archivo del disco de memoria.

MNAME

```
CALL MNAME (<espec anterior>) AS <nuevo espec>)
```

Para cambiar el nombre de un archivo del disco de memoria.

OTRAS SENTENCIAS QUE SE PUEDEN UTILIZAR CON EL DISCO DE MEMORIA.

Se pueden utilizar las siguientes sentencias y funciones con el disco de memoria. La sintaxis es idéntica a la de las sentencias del Disk-BASIC, excepto que el nombre del dispositivo del disco de memoria es "MEM". No está disponible el acceso directo.

SAVE
LOAD
RUN
MERGE
OPEN
CLOSE
PRINT#
PRINT# USING
INPUT#
LINE INPUT#
INPUT$
EOF
LOC
LOF

SENTENCIAS PARA MANEJO DEL RELOJ DE TIEMPO REAL.

GET DATE

GET DATE X$[, A]

Para obtener la fecha del circuito integrado del reloj. El formato del alfanumérico X$ es:

AA/MM/DD

Donde "AA" son los últimos dos dígitos del año, p.e. "19XX". "MM" es el mes y "DD", el día. Cuando se especifica el parámetro opcional "A", X$ se supone que es la fecha de alarma.

GET TIME

GET TIME X$ [, A]

Para obtener la hora del circuito integrado del reloj. El formato del alfanumérico X$ es:

HH:MM:SS

Donde "HH" es la hora, "MM" son los minutos, "SS" son los segundos. Cuando se especifica el parámetro opcional "A", se supone que X$ es la hora de alarma.

SET DATE

SET DATE <fecha>[, A]

Para cambiar la fecha del circuito integrado del reloj. El formato de la expresión <fecha> es:

AA/MM/DD

Donde" AA" son los últimos dos dígitos del año, p.e. "19XX", "MM" es el mes y "DD", el día. Cuando se especifica el parámetro opcional "A", <fecha> se supone que es la fecha de alarma y se coloca en los registros de fecha de alarma sin borrarlos. El sistema no soporta la función de alarma para el reloj, simplemente provee el modo para colocar la fecha de alarma. Sólo son válidos los días para la función de alarma del circuito integrado.

"0" no puede omitirse de la <fecha>. Por ejemplo:

"85/06/03"es válido

"85/6/3"no es válido

SET TIME

SET TIME <hora> [, A]

Para colocar la hora en el reloj. El formato de la expresión alfanumérica <hora> es:

HH:MM:SS

Donde "HH" es la hora, "MM" son los minutos, "SS" son los segundos. Cuando se especifica el parámetro opcional "A", se supone que <hora> es la hora de alarma y se coloca en los registros de fecha de alarma borrándolos. El sistema no soporta la función de alarma para el circuito integrado del reloj, simplemente provee el modo para colocar

la fecha de alarma. Sólo son válidos las horas y minutos para la función de alarma del circuito integrado.

'0' no puede omitirse del parámetro <hora>

Nota:

El comando SET TIME X$,A borra los registros de alarma antes de colocar la hora de alarma, pero SET DATE X$,A, no. Por lo tanto, deberá ejecutar SET TIME antes de SET DATE. Si sólo se ejecuta la sentencia SET TIME, la fecha no generará una alarma.

SENTENCIAS PARA INDICADORES DE MEMORIA.

Las sentencias SET coloca diversos parámetros y los almacena en la RAM que posee el circuito integrado del reloj, que tiene una batería de respaldo.

Se puede utilizar solamente una de las siguientes sentencias por vez: SET TITLE, SET PASSWORD y SET PROMPT. Si dos o más sentencias se ejecutan, solamente la última sentencia es válida. Esto se debe a que estas sentencias utilizan la misma área de memoria RAM del reloj.

SET ADJUST

SET ADJUST(<posición x>,<posición y>)

Para cambiar la posición de la pantalla. <posición x> y <posición y> pueden tomar valores desde -7 a 8 y cambian la posición de la imagen de pantalla. Los valores iniciales son (0,0). Para la coordenada X, los valores positivos mueven la pantalla hacia la derecha y los negativos, hacia la izquierda. Para la Y, los valores positivos mueven la pantalla hacia arriba, y los valores negativos, hacia abajo.

SET BEEP

SET BEEP <tono>,<volumen>

Para indicar el tipo de sonido y volumen del sonido "bip". <tono> debe variar de 1 a 4 para seleccionar el tipo de sonido del "bip". Con 1 selecciona el sonido como en la versión MSX v.1. <volumen> debe variar entre 1 y 4 para seleccionar el volumen del sonido "bip". El valor 4 genera el volumen máximo.

SET PASSWORD.

SET PASSWORD <contraseña>

Para colocar una contraseña. <contraseña> es un alfanumérico de hasta 255 bytes de longitud.

Si se encuentra un "cartucho llave" colocado, su "valor llave" también se almacena.

Si está colocada la contraseña, la palabra "Password:" se muestra en la pantalla de presentación cuando se enciende el sistema, y espera que el operador ingrese la contraseña. Si el texto ingresado no coincide con la contraseña almacenada con esta sentencia, se la requerirá nuevamente.

Si no está colocado el "cartucho llave" cuando se ejecuta esta sentencia, el sistema verifica las teclas <GRAPH> y <STOP>, y si ambas se encuentran presionadas, no se pedirá la contraseña. Si está colocado el "cartucho llave" con el valor correcto de contraseña, tampoco se requerirá el ingreso de la misma. Si está colocado el "cartucho llave" y se almacena una contraseña nula con esta sentencia, se necesita un "cartucho llave" válido para poder proseguir.

SET PROMPT

SET PROMPT <texto>

Para almacenar la expresión alfanumérica <texto> como prompt en la RAM del reloj.

Este alfanumérico se utiliza como texto a mostrar reemplazando al standard "Ok". La longitud máxima permitida para <texto> es de 6 caracteres.

SET SCREEN

SET SCREEN

Para almacenar en la memoria permanente del reloj la configuración actual de la pantalla. Una vez ejecutada, cada vez que se encienda la máquina la pantalla se configura con los parámetros presentes en el momento que se ejecutó por última vez.
Los parámetros almacenados por esta sentencia son los siguientes:

Parámetro	Rango	Se fija con
Modo de pantalla	0 ó 1	SCREEN
Ancho	1 a 80	WIDTH
Color frente	1 a 15	COLOR
Color fondo	1 a 15	COLOR
Color borde	1 a 15	COLOR
Visualización de funciones	ON/OFF	KEYON/KEYOFF
Click de tecla	SI/NO	SCREEN
Tipo de Impresora	MSX/stansard	SCREEN
Velocidad del cassette	1200 ó 2400	SCREEN
Modo de display	0 a 4	SCREEN

Ejemplo:

```
SCREEN 0,,1,1,0,0
WIDTH 80
COLOR 15,1,1
KEYOFF
SET SCREEN
```

Después de ejecutar esta secuencia de instrucciones cada vez que se encienda la máquina la pantalla se va a configurar en screen 0, con líneas de 80 columnas, color blanco sobre fondo y borde negro, sin visualizar teclas de función, con click de tecla, cassette a 1200 bauds, impresora standard y modo de display normal.

SET TITLE

SET TITLE <titu1o>[,<color>]

Para almacenar la expresión alfanumérica <título>, que será mostrada en pantalla cuando se encienda el sistema.

Si la longitud del alfanumérico <título> es igual a 6, el sistema espera que se pulse una tecla luego de mostrar la pantalla inicial. <color> varía entre 1 y 4 y permite seleccionar el color del título. La longitud máxima de <título> es de 6 caracteres.

PROGRAMAS DE EJEMPLO.

1. SET ADJUST:

Utilizando la sentencia SET ADJUST podrá regular la posición general de la pantalla en las cuatro coordenadas. Veamos un ejemplo:

```
10 COLOR 15,0,0:SCREEN 5
20 FOR I=0 TO 80 STEP 4
30 C1RCLE(128,105),I
40 NEXT
50 DATA 0,-7,1,-7,2,-7,3,-6,4,-6
60 DATA 5,-5,6,-4,6,-3,7,-2,7,-1
70 DATA 7,0,7,1,7,2,6,3,6,4
80 DATA 5,5,4,6,3,6,2,7,1,7
90 DATA 0,7,-1,7,-2,7,-3,6,-4,6
100 DATA -5,5,-6,4,-6,3,-7,2,-7,1
110 DATA -7,0,-7,-1,-7,-2,-6,-3,-6,-4
120 DATA -5,-5,-4,-6,-3,-6,-2,-7,-1,-7
130 FORI=1 TO 40
140 READ X,Y
150 SET ADJUST(X,Y)
160 NEXT:RESTORE
170 GOTO 130
```

2. COLOR SPRITE.

En los sprites, a partir de SCREEN 4 se puede colocar colores por cada línea del mismo; es más, si coloca en uno el bit 6 del registro de color se puede llegar a utilizar sprites sumamente complicados. Como demostración, se presenta este programa; refiérase a la explicación de COLOR SPRITE.

```
100 '
110 'EJEMPLO COLOR SPRITE BIT6
120 '
130 ON STOP GOSUB 450:STOP ON
140 COLOR 15,15,15:SCREEN 5,3
150 T=85:S$="":FOR 1=1 TO 32:GOSUB 420:NEXT
160 SPRITE$(0)=S$
170 T=51:S$="":FOR 1=1 TO 32:GOSUB 420:NEXT
```

```
180 SPRITE$(l)=S$
190 T=15:S$="":FOR I=1 TO 32:GOSUB 420:NEXT
200 SPRITE$(2)=S$
210   T=1:S$=CHR$(0):FOR   I=1   TO   7:S$=S$+CHR$(T):
T=T*2+1:NEXT
220 FOR I=1 TO 8:S$=S$+CHR$(T):T=(T*2)MOD256:NEXT
230 T$=R1GHT$(S$,8)+LEFT$(S$,8):S$=S$+T$
240 SPRITE$(3)=S$
250 COLOR SPRITE$(0)=STR1NG$(16,1)
260 COLOR SPRITE$(1)=STR1NG$(16,66)
270 COLOR SPRITE$(2)=STR1NG$(16,68)
280 COLOR SPRITE$(3)=STR1NG$(16,72)
290 COLOR=(0,0,0,0):COLOR=(8,1,1,1)
300 COLOR=(1,0,0,7):COLOR=(9,0,0,4)
310 COLOR=(2,0,7,0):COLOR=(10,0,4,0)
320 COLOR=(3,0,7,7):COLOR=(11,0,4,4)
330 COLOR=(4,7,0,0):COLOR=(12,4,0,0)
340 COLOR=(5,7,0,7):COLOR=(13,4,0,4)
350 COLOR=(6,7,7,0):COLOR=(14,4,4,0)
360 COLOR=(7,7,7,7):COLOR=(15,4,4,4)
370 PUT SPRITE 0,(32,100) :A$=INPUT$(l)
380 PUT SPRITE 1,(64,100) :A$=INPUT$(l)
390 PUT SPRITE 2,(96,100) :A$=INPUT$(l)
400 PUT SPRITE 3,(128,100) :A$=INPUT$(l)
410 GOTO 370
420 S$=S$+CHR$(T)
430 T=(T*2)MOD256+T\128
440 RETURN
450 STOP OFF
460 COLOR 15,4,7
470 END
```

3. PAGE.

Para computadoras MSX2 con Video RAM de capacidad de 128 kbytes o más, se puede almacenar varias páginas por cada pantalla y se los puede intercambiar.

```
10 '
20 ' EJEMPLO DE PAGE
30 '
```

```
40 COLOR 15,1,1:SCREEN 5
50 DEFINT A-Z:DIM A(258)
60 FOR I=0 TO 3:SET PAGE I,I
70 CLS:NEXT:SET PAGE 0,0
80 CIRCLE(48,48),15,15,-1,-3.14
90 PAINT(40,40),8,15
100 COPY(32,32)-(63,63),0 TO A:CLS
110 FOR I=0 TO 3:SET PAGE I,I
120 COPY A,I TO (32,32),I
130 FOR J=0 TO 999:NEXT
140 NEXT:GOTO 110
```

4. RGB.

Con la capacidad de paleta cromática se pueden elegir 16 colores entre 512. Este programa podrá mostrar todas las combinaciones posibles de colores.

```
100 '
110 'PRUEBA RGB
120 '
130 SCREEN 0
140 COLOR 15,0
150 CLS
160 LOCATE 15,8:PRINT "R   G   B"
170 FOR R=0 TO 7
180 FOR G=0 TO 7
190 FOR B=0 TO 7
200 LOCATE 14,10:PRINT R;G;B
210 LOCATE 14,12
220 R$_RIGHT$("000"+BIN$(R),3)
230 G$_RIGHT$("000"+BIN$(G),3)
240 B$_RIGHT$("000"+BIN$(B),3)
250 PRINTR$+G$+B$:COLOR=(0,R,G,B )
260 FOR I=0 TO 500:NEXT
270 NEXT B,G,R
280 GOTO 170
```

5. SCREEN 5.

Asignando a cada paleta cromática colores con una diferencia mínima, se puede ver un efecto de graduación de colores.

```
10 '
20 ' DEMO PALETA SCREEN 5
30 '
40 COLOR 15,0,0:SCREEN 5
50 FOR I=0 TO 7;READ A,B,C,D
60 GOSUB 90:GOSUB 150
70 NEXT:RESTORE:GOTO 50
80 '*** ENVIA COLOR ***
90 FOR J=0 TO 7
100 COLOR=(J+A*8,J*B,J*C,J*D)
110 NEXT:RETURN
120 DATA ,,,1,1,,1,,,1,,,1,1,1,1
130 DATA ,,1,1,1,1,,1,,1,1,,1,,,
140 '*** LINEA ***
150 E=A*8:F=E:N=0:M=1
160FOR X=0 TO 255
170 IF X>211 THEN N=N+1
180 LINE (X,255*A)-(N,ABS(A*211-X+N)),E
190 E=E+M:IF E=8+F THEN E=E-1:M=-M
200 IF E<F THEN E=F:M=-M
210 NEXT
220 FOR Y=209*A+1 TO 211*(1-A) STEP 1-A*2
230 N=N+1:LINE(255,Y)-(N,211*(1-A)),E
240 E=E+M:IF E=8+F THEN E=E-1:M=-M
250 IF E<F TREN E=F:M=-M
260 NEXT:RETURN
```

6. SCREEN 8.

En el modo SCREEN 8 se tiene una resolución de 256 colores, y se pueden hacer dibujos con una óptima resolución.

```
100 '
110 'COPIA MARIPOSA REDUCIDA EN
120 '
130 DEFINT A-Z:DIM A(2697)
140 DEFFNP=INT(RND(1)*2)+2 OR INT(RND(1)*3+3)*32
```

```
150 '*** MARIPOSA ***
160 SCREEN 8 :COLOR 227,0,0:CLS
170 PSET(128,100):FOR J=0 TO I
180 FOR I=0 TO 14:READ X,Y
190 IF J THEN X=-X:IF I=14 THEN 210
200 LINE-STEP(X*2,Y*2)
210 NEXT:RESTORE:NEXT
220 LINE-(128,100)
230 DATA 30,-30,10,-5,5,0,2,10,0,9
240 DATA -2,15,-3,10,-1,9,-2,6,-3,3
250 DATA -5,2,-12,-2,-11,-7,-8,-11,0,-9
260 PAINT(130,100),227,227
270 PAINT (120,100),227,227
280 FOR X=128 TO 255 STEP 2:P=FNP
290 LINE (128,100)-(X,0),P,,AND
300 LINE (128,100)-(256-X,0),P,,AND
310 NEXT
320 FOR Y=0 TO 211 STEP 2:P=FNP
330 LINE (128,100)-(255,Y),P,,AND
340 LINE (128,100)-(0,Y),P,,AND
350 NEXT
360 FOR X-255 TO 128 STEP-2:P=FNP
370 LINE (128,100)-(X,211),P,,AND
380 LINE (128,100)-(255-X,211),P,,AND
390 NEXT
400 PSET(138,80),227
410 LINE-(128,100)
420 LINE-(128,130)
430 PSET (127,124),227
440 LINE-(127,130)
450 LINE-(127,100)
460 LINE-(118,80)
470 '*** REDUCCION ***
480 FOR Y=0 TO 52
490 FOR X=0 TO 63
500 PSET(X,Y),POINT(X*4,Y*4)
510 NEXT X,Y
520 '*** COPIAR ***
530 SET PAGE 1,0
540 COPY(0,0)-(63,52),0 TO A :CLS
550 FOR Y=0 TO 211 STEP 53
560 FOR X-64 TO 255 STEP 128
570 COPY A,1 TO (X-1,Y)
```

```
580 COPY A,2 TO (X,Y+52)
590 NEXT X,Y
600 LINE (0,0)-(255,211),255,BF,XOR
610 SET PAGE 0,1:CLS
620 FOR Y=0 TO 211 STEP 53
630 FOR X=64 TO 255 STEP 128
640 COPY A,1 TO (X-1,Y)
650 COPY A,2 TO (X,Y+52)
660 NEXT X,Y
670 FOR A=1
680 SET PAGE 1,1:FOR W=1 TO 200:NEXT
690 SET PAGE 0,0:FOR W=1 TO 200:NEXT
700 NEXT
```

EXTENSIONES MSX2 BASIC

Se presentan en esta publicación las extensiones MSX BASIC 2.0 al lenguaje BASIC, creadas para soportar la ampliada capacidad gráfica de las nuevas computadoras personales Talent MSX2.
La versión MSX BASIC 2.0 incluye importantes mejoras en estas funciones, manteniendo una total compatibilidad ascendente con las versiones MSX BASIC 1.0. Se han agregado también comandos y funciones especiales para manejar el reloj calendario interno permanente, y el emulador a de disco en RAM.
Con las computadoras Talent MSX2 y el MSX BASIC 2.0 es posible acceder a un nuevo espectro de posibilidades que permiten:

- 5 nuevos modos de pantalla, incluyendo texto en 80 columnas y alta resolución de 512x212 pixels.
- 256 colores disitntos en modo multicolor.
- 512 colores disponibles en modo alta resolución.
- 8 sprites por línea.
- Sentencias para el manejo de reloj calendario interno.
- Sentencias para emular un disco en RAM, similares al Disk-BASIC.

Este Manual incluye solamente las extensiones de la versión MSX-BASIC 2.0. Para tener una referencia completa del lenguaje, debe utilizarse en conjunto con el Manual MSX-BASIC.

Producido en Argentina por TELEMATICA S.A.