

forth

COMPILER
COMPILER

**CON
SPRITES**

AMSTRAD

ACE
Software

FORTH COMPILER

Cualquier reproducción total o parcial de este programa sin consentimiento por escrito del propietario del copyright será considerada como una violación de la ley. La distribución y la venta de este producto es unicamente para el uso del comprador.

ACE software está autorizada para enmendar o alterar la información incluida en este programa sin previo aviso. No obstante, a pesar de que han sido tomadas todas las medidas necesarias para la verificación total del programa, no será tenida en cuenta ninguna reclamación de su exactitud, ya que un examen exhaustivo del mismo sería imposible, bajo cualquier condición de trabajo.

© ACE SOFTWARE S.A.

Producido en exclusiva para España por ACESSA. Depósito Legal B-20.095-86

Mod. AM-18

AM 18/D

INTRODUCCION

Nos encontramos ante un compilador de lenguaje FORTH que ha sido diseñado para cualquiera de los modelos AMSTRAD 464, 664, 6128.

Este compilador posee las definiciones que podemos encontrar en cualquier FORTH-79, más un conjunto de ampliaciones que están orientadas a la explotación del sistema sobre el que trabajamos.

No pretende este manual enseñar al usuario a programar en FORTH, sino realizar un estudio sobre las definiciones que incorpora así como su correcta sintaxis.

Para quien no ha programado nunca en FORTH, quizás sería importante señalar el uso de la NOTACION POLACA INVERSA por parte del compilador. Este tipo de notación se diferencia de la INFIJA en el orden de los operandos y las funciones.

En polaca inversa los operandos se introducen antes que la función lo que hace que el uso de parentesis sea anulado totalmente, pero veamos algunos ejemplos de equivalencia entre ambas notaciones:

INFIJA	POLACA INVERSA
$A + B$	$A B +$
$C * A$	$C A *$
X/Y	$X Y /$
$(A + B) / (C * A)$	$A B + C A * /$

A diferencia de otro tipo de compiladores, en FORTH podemos trabajar por programa o en MODO DIRECTO lo que nos facilitará en un principio su aprendizaje, ya que podremos ejecutar sentencias sin necesidad de tener que realizar su posterior compilación.

Gran parte de los ejemplos que contiene este manual serán introducidos en MODO DIRECTO para poder comprobar de una forma rápida el resultado de una determinada sentencia FORTH.

MANEJO DEL EDITOR

El programa fuente de FORTH se almacena en forma de líneas al igual que si se tratará de un programa BASIC.

La relación de los comandos que nos permite tratar el texto es la siguiente:

CLS	Borra el área de trabajo.
LIST	Lista el programa desde la línea inicial.
LIST n	Lista el programa desde la línea n.
EDIT n	Edita la línea n para su modificación.
AUTO	Inserta automáticamente líneas empezando por la número 10 con incrementos de 10.
AUTO n	Idem pero empezando en la línea n.
AUTO n m	Idem pero con incrementos de m.
RENUM	Renumeración de todas las líneas.
DELETE n	Borra del texto la línea n.
DELETE n m	Borra el grupo de líneas comprendido entre n y m.
SAVE nombre	Salvaguarda el fichero de texto Forth.
LOAD nombre	Recupera un fichero de texto Forth.
PRON	Activa la salida por impresora.
PROFF	Desactiva la salida por impresora.
Control + W	Actualizará la última entrada en MODO DIRECTO.
ESC	Borra la línea actual y/o desactiva el modo AUTO.
WARM	Restablece la disposición de la ventana de trabajo.
MEMORY	Nos indica la memoria disponible

Habrás observado que en la parte inferior del área de trabajo existe una zona en donde están referenciadas las teclas de función con sus respectivas asignaciones. Todos los comandos directos

del compilador y del editor se pueden encontrar en alguno de los niveles de las teclas de función, para pasar de un nivel al siguiente pulse la tecla de función "F0".

Si lo cree más conveniente puede teclear los comandos directamente lo que producirá el mismo efecto que la utilización de las teclas de función.

El signo "#" al inicio de una entrada indicará al compilador que el contenido de esa línea debe ejecutarse en MODO DIRECTO, esta opción nos será útil para realizar pruebas antes de la compilación o para verificar el funcionamiento de nuestras definiciones una vez compiladas y libres de errores.

100 1 do . "Soy su compilador de FORTH" loop;

El planteamiento del FORTH es el de un lenguaje compacto y de gran rapidez. Esto implica que muchas de las tareas que realizan en la sombra otros lenguajes, en FORTH deben ser tenidas en cuenta por parte del programador.

Es muy importante salvaguardar el código fuente antes de cualquier prueba ya que un error de diseño poco aparente puede destruir en su totalidad el área de trabajo que estamos utilizando.

Cuanto más fragmentado sea su programa mayor control y poder de verificación tendrá sobre él. En FORTH la definición de procedimientos independientes asegura una mayor fiabilidad del programa ya que cada parte de él puede ser verificada independientemente hasta lograr su perfecto funcionamiento.

Todas las definiciones FORTH empiezan y terminan igual. Su inicio viene determinado por ":" seguido del nombre que asignamos al procedimiento, este procedimiento se termina con el carácter ";". En FORTH se utiliza el espacio " " como delimitador, así todas las definiciones que integran una palabra se encuentran separadas por un espacio.

10 :cuadrado dup * ;

20 :espacios 1 do," " loop ;

En las dos líneas anteriores hemos definido dos nuevas palabras FORTH:

CUADRADO y ESPACIOS

La primera de ellas calcula el cuadrado de un número dejando su valor en la pila mientras que la segunda imprimirá tantos espacios (" ") como le indiquemos.

Después de su compilación mediante **RUNCOM** podremos utilizarlas en MODO DIRECTO.

4 cuadrado . IMPRIME: 16

8 espacios 4 cuadrado . IMPRIME: 16

Este compilador de FORTH posee aritmética entera lo que le permite el manejo de enteros de 16 bits en el rango de -32768 a 32767 si se trabaja con ellos como enteros con signo, en caso contrario puede trabajarse como si fueran enteros sin signo dentro del rango de 0 a 65535.

En **FORTH** las cadenas alfanuméricas se almacenan con su primer carácter igual a su longitud esto significa que la cadena siguiente "BUENOS DIAS" se almacenaría como:

chr \$ (11) + "BUENOS DIAS"

ocupando 12 bytes de memoria (la longitud de la cadena más el primer byte de la misma que indica su longitud).

La reserva de espacio para cadenas realizada con el comando **CHAR** permite el dimensionado de 0 a 65535 caracteres que deberán ser tratados posteriormente dependiendo del modo de almacenamiento de las cadenas en ese espacio de memoria.

A su vez la dimensión de matrices de enteros se realiza en el rango de 0 65535 elementos que pueden ser manejados mediante los comandos **ASK** (leer) y **PUT** (guardar).

La única limitación en la cantidad de variables y matrices es que en su totalidad no pueden sobrepasar de 200 ya que ese es el espacio reservado en la tabla de símbolos para la definición de las mismas.

Lo mismo sucede con el número de palabras compiladas que en su totalidad no podrán ser más de 300.

El editor permite el manejo de 200 líneas de programa para su posterior compilación con **RUNCOM**. Si fueran necesarias más líneas de programa el comando **FIX** borrará todo el texto del editor

para aceptar más líneas de texto que serán compiladas a continuación de las anteriores definiciones que permanecerán en el diccionario.

COMANDOS DEL COMPILADOR

RUNCOM

Este comando procederá al proceso de compilación del código fuente almacenado en el editor. El proceso de compilación se detendrá en el caso de que el texto contenga errores.

Una vez finalizado el proceso las palabras compiladas formarán parte del sistema hasta una nueva compilación.

WLIST

Nos facilita una lista de todas las palabras disponibles en el compilador de forth. Tanto las residentes como las de nuestra propia creación.

VLIST

Nos facilita una lista de todas las variables definidas por el sistema o por nosotros mismos.

FIX

Este comando se utilizará después de una compilación satisfactoria en el caso de que deseemos seguir compilando palabras y carezcamos de espacio para ello.

El comando **FIX** asume como palabras del sistema todas aquellas que se encuentren compiladas en el momento de su uso, esto quiere decir que una posterior compilación no destruirá las palabras que tengamos ya compiladas. A su vez borrará **TOTALMENTE** el texto que contenga en ese momento el editor por lo que es aconsejable realizar una grabación del mismo antes de utilizar esta opción.

OBJECT

Este es uno de los comandos más potentes del compilador ya que nos permitirá la grabación de un programa en formato binario

autoejecutable conteniendo todas las definiciones compiladas.

Este programa tendrá como dirección de inicio la dirección de la última palabra compilada.

Para hacerlo funcionar bastará con ir al **BASIC** y teclear el clásico

RUN "nombre del programa".

Este programa no necesita la presencia del compilador en el disco o en la memoria del ordenador por lo que funcionará en su ausencia.

Todos estos programas pueden cargarse también desde el BASIC con

MEMORY &6fff:LOAD "nombre del programa": CALL &7000
pudiéndose entremezclar el BASIC con el FORTH.

DIRECT

Nos proporcionará las direcciones de compilación de todas las palabras y variables una vez compilado el texto que contenga el editor.

Esto puede ser útil si pensamos utilizar algunas de las palabras junto con un programa en código máquina o BASIC e incluso para poder obtener el valor de ciertas variables del programa.

Anotemos también la posibilidad de utilizar toda la zona de memoria comprendida entre &40 y &6fff para definir bancos de sprites, tablas auxiliares, buffers de trabajo, etc. con la única condición de que sólo podrá utilizar esa zona un programa grabado con la opción OBJECT, pero que de ninguna manera se podrá testear ese programa en MODO DIRECTO desde el compilador ya que eso produciría la destrucción del mismo.

RESET

Reinicializará el sistema totalmente BORRANDO toda la información que pudiera tener el editor así como todas las palabras que se pudieran haber compilado.

- I Devuelve el valor del índice del bucle en el que se halla
100 1 DO I . CR LOOP
- J Devuelve el valor del índice del bucle exterior al que se halla
10 1 DO 10 1 DO J . " " I .
CR LOOP LOOP

-Operadores de Relación-

Todos los operadores de relación devuelven un 1 ó un 0 en el PAP dependiendo de la veracidad o no de la comparación.

< # 4 5 < . 4 < 5 ; = 1

= # 9 4 = . 9 = 4 ; = 0

> # 10 8 > . 10 > 8 ; = 1

◇ # 4 4 ◇ . 4 ◇ 4 ; = 0

U < Comprueba la relación "menor que" tomando los valores del PAP como enteros sin signo (0 65535)

23000 45678 U < .
23000 < 45678 ; = 1

-Operadores Booleanos-

Los operadores Booleanos al igual que los anteriores devuelven en el PAP el valor 0 ó 1.

AND # 1 0 AND . 1 AND 0 = 0

OR # 2 0 OR . 2 OR 0 = 2

XOR # 1 0 XOR . 1 XOR 0 = 1

NOT Esta definición devuelve 0 si el valor es distinto de 0 y devuelve 1 en el caso de que sea 0
234 NOT . 234 NOT = 0

-Estructuras de Control-

-Selección-

condición IF si verdadera THEN

```
# 2 3 < IF ."es menor" THEN CR ."O.K"
```

condición IF si verdadera a ELSE si falsa THEN

```
# 4 5 < > IF ."distintos" ELSE ."iguales" THEN CR ."O.K."
```

-Repetición-

final inicio DO código LOOP

```
# 1000 1 DO I DUP DUP * . ."es el cuadrado de" . CR LOOP
```

final inicio DO código incremento + LOOP

```
# 100 2 DO I . ." " 4 + LOOP
```

LEAVE Fuerza la salida de un bucle DO LOOP

```
# 10 1 DO I DUP . ." " 5 = IF LEAVE THEN LOOP
```

BEGIN condición WHILE código REPEAT

Mientras se cumpla la condición se ejecutará el código situado entre el WHILE y el REPEAT

```
# 1 DUP BEGIN 10 < WHILE ."Menor que 10" 1+ DUP REPEAT
```

BEGIN código condición UNTIL

Se ejecutará el código hasta que la condición sea falsa

```
# 0 BEGIN 1 + DUP ."Menor que 10 ?" CR 10 < UNTIL
```

-Salida-

. Imprime el valor situado en el PAP como un entero con signo (-32768, 32767)

```
# 2345 .
```

U. Imprime el valor situado en el PAP como un entero sin signo (0,65535)

```
# 45000 U.
```

."	Imprime la cadena alfanumérica situada entre comillas # ."Esta cadena está entre comillas"
CR	Situa el cursor al principio de la siguiente línea # ."1 línea" CR ."2 Línea" CR ."3 Línea"
EMIT	Imprime el carácter cuyo código ASCII tiene el PAP # 72 79 76 65 EMIT EMIT EMIT EMIT
TYPE	Imprime la cadena cuya dirección de almacenamiento se encuentra en el PAP # INPUT CR PAD TYPE (PAD es la dirección de una área intermedia de trabajo de 256 bytes de longitud que puede utilizarse para cualquier propósito. La palabra INPUT del FORTH sitúa en esta área la cadena que recibe por el teclado y es por eso que nosotros le facilitamos esa dirección a la palabra TYPE)
HEX	Establece la base hexadecimal para salida de datos # 23456 HEX .
DEC	Establece la base decimal para salida de datos # 12345 DEC .
<i>-Entrada-</i>	
GET	Se mantiene a la espera hasta la pulsación de una tecla, cuyo código ASCII coloca en el PAP. # 10 1 DO GET EMIT LOOP
KEY	Testea si una tecla determinada es pulsada o no devolviendo para ello un 1 ó un 0. El código de la tecla que se quiere examinar corresponde al de los "Números de las teclas y los Joysticks" de su manual del AMSTRAD # 13 KEY .

INPUT Espera la entrada de una cadena alfanumérica que almacenará en el PAD y cuyo primer carácter será igual a su longitud.

CONVERT Devuelve el valor numérico de una cadena (si lo representa) cuya dirección le facilitamos.
INPUT CR PAD CONVERT .

-Operaciones con la memoria-

! Coloca en dos posiciones de memoria consecutivas un entero de 16 bits.

valor dirección ! # 1234 PAD !

@ Lee el entero de 16 bits contenido en dos posiciones de memoria consecutivas

dirección @ # PAD @ .

C! Coloca en una posición de memoria un entero de 8 bits

valor dirección C! # 23 PAD C!

C@ Lee el entero de 8 bits almacenado en una dirección de memoria

dirección C@ # PAD C@ .

(Utilizamos en los ejemplos el PAD para guardar y recuperar valores por ser una zona segura de trabajo, sin el riesgo de una posible corrupción del compilador de FORTH. Siempre y cuando no sobrepasemos el límite de 256 bytes ya que si esto sucediera lo más probable es que perdiéramos el control del compilador)

-Gráficos-

MOVE Situa el cursor gráfico en unas determinadas coordenadas absolutas

x y MOVE # 100 100 MOVE

MOVER Situa el cursor gráfico en unas coordenadas re-

	lativas a la actual posición
	x y MOVER # 100 100 MOVER 10 10 MOVER
PLOT	Ilumina un pixel de pantalla x y PLOT # 100 100 PLOT
PLOTR	Idem en coordenadas relativas x y PLOTR # 100 100 PLOT 10 10 PLOTR
DRAW	Traza una línea desde la posición actual del cursor hasta las coordenadas absolutas indicadas x y DRAW # 0 0 MOVE 100 100 DRAW
DRAWR	Traza una línea desde la posición actual del cursor hasta las coordenadas relativas a este indicadas x y DRAWR # 100 100 MOVE 10 10 DRAWR
TEST	Deja en el PAP el lápiz con que se ha iluminado el pixel referenciado por las coordenadas absolutas x y TEST # 10 10 PLOT 10 10 TEST .
TESTR	Idem en coordenadas relativas x y TEST # 10 10 PLOT 10 10 MOVE 0 0 TEST.
GCURSOR	Devuelve en el PAP las coordenadas del cursor de gráficos # 100 100 MOVE GCURSOR . " " .
GPEN	Selecciona el lápiz utilizado para los gráficos # 2 GPEN
GPAPER	Selecciona el papel utilizado para los gráficos # 0 GPAPER

-Texto-

MODE	Asigna el modo de visualización deseado modo MODE # 1 MODE
CLS	Borra la ventana en la que estemos trabajando
BORDER	Asigna un par de colores al borde de la pantalla. Para dejar un color fijo sólo hay que asignarlo por duplicado color 1 color 2 BORDER # 0 0 BORDER
INK	Asigna un par de colores a un determinado lápiz lápiz color 1 color 2 INK # 1 24 24 INK
PEN	Determina el lápiz a utilizar para la escritura lápiz PEN # 2 PEN
PAPER	Determina el lápiz a utilizar para el color del fondo lápiz PAPER # 2 PAPER
FLASH	Establece la velocidad de parpadeo entre las dos tintas que posee cada lápiz velocidad 1 velocidad 2 FLASH # 10 5 FLASH
INVERSE	Intercambia los lápices de escritura # . "Hola" INVERSE . "Hola" INVERSE . "Hola"
FRAME	Sincroniza la impresión con el barrido del moni- tor # CLS 79 1 DO FRAME 1 10 LOCATE . "*" LOOP
WINDOW	Define una ventana de texto xi xf yi yf num WINDOW # 10 20 1 6 1 WINDOW

STREAM	<p>Selecciona una ventana de texto para las sucesivas instrucciones de impresión en pantalla</p> <p>ventana STREAM # 1 STREAM</p>
LOCATE	<p>Situa el cursor de texto en una posición determinada</p> <p>x y LOCATE # 10 20 LOCATE . "Hello"</p>
CURSOR	<p>Situa en el PAP el valor de las coordenadas del cursor</p> <p># CLS 24 4 LOCATE CURSOR . . " " .</p>
COPY	<p>Devuelve el código ASCII del carácter situado en la posición del cursor de texto (si lo hay)</p> <p># CLS . "FORTH" 3 1 LOCATE COPY 1 10 LOCATE EMIT</p>
TAG	<p>Desvia la impresión de texto hacia el cursor gráfico</p> <p># TAG 200 100 DO I DUP MOVE. "FORTH" LOOP TAGOFF</p>
TAGOFF	<p>Redirecciona al cursor de texto la impresión</p>
DEVICE	<p>Actúa como un interruptor que desvia el texto hacia la impresora o la pantalla alternativamente</p> <p># . "Conecte la impresora" DEVICE. "OK" CR DEVICE</p>
PRON	<p>Activa la posibilidad de mandar texto hacia la pantalla y hacia la impresora simultáneamente.</p> <p># PRON 100 1 DO I . . " " LOOP PROFF</p>
PROFF	<p>Desactiva esa posibilidad</p>
-Sonido-	
ENV	<p>Define una envolvente de volumen que viene determinada por varias secciones de tres parámetros, seguidas del número de secciones utili-</p>

zadas y del número de envolvente
Sólo se permiten 5 secciones como máximo
[s01 s02 s03] [s11 s12 s13] ... sec número
ENV
s01=No. de pasos | s02=altura | s03=dura-
ción
1 2 3 2 4 8 2 1 ENV

ENT Define una envolvente de tono utilizando la mis-
ma sintaxis que en la de volumen
1 - 2 3 2 3 4 1 ENT

SOUND Emite un sonido dependiente de los parámetros
asignados
canal env ent período ruido volumen dura-
ción SOUND
1 00 100 0 15 300 SOUND

–SPRITES–

INISPRT Inicializa un sprite en unas coordenadas
x y num INISPRT
100 100 1 INISPRT

PRNSPRT Visualiza un sprite si no está presente en panta-
lla o lo borra en caso contrario
num PRNSPRT # 1 PRNSPRT

PUTSPRT Mueve un sprite a una nueva posición en panta-
lla borrando la posición que ocupaba anterior-
mente
x y num PUTSPRITE
104 104 1 PUTSPRITE

COLSPRT Comprueba la posible colisión entre dos sprites
dejando en el PAP el valor lógico correspon-
diente (1 ó 0) en caso afirmativo o negativo
numa numb COLSPRT # 1 2 COLSPRT

POOSPRT Devuelve en el PAP las coordenadas de posi-

ción de un sprite

número POOSPRT # 2 POOSPRT . . " " .

Todas las coordenadas referentes a los SPRITES deberán tomarse en el rango (0,159) para las X, (0,199) para las Y.

Asimismo recuerde que los SPRITES sólo funcionarán correctamente en modo 0 (160 x 200 puntos en 16 colores)

I/O

CAT Presenta en pantalla el catálogo de programas existentes en el disco/cinta.

CAT

LOAD Carga un bloque de bytes en una posición determinada de la memoria.

El nombre del fichero ".BIN" debe situarse en el PAD (con un INPUT o la asignación PAD "nombre").

dirección LOAD

PAD "prueba" 10000 LOAD

SAVE Salva un bloque de bytes desde una posición determinada en memoria al disco/cinta.

El nombre del fichero debe situarse en el PAD.

direc. long. SAVE

PAD "nada" 1000 3500 SAVE

-Variables y Matrices-

VAR Inicializa una variable entera

VAR nombre # VAR inicio

CHAR Dimensiona una tira de caracteres (espacios) de longitud determinada

CHAR nombre longitud

CHAR mensaje 40

" "

Coloca en la dirección especificada la cadena

entrecorrida precedida por la longitud de la misma.

(En FORTH las cadenas alfanuméricas se almacenan con su longitud como primer carácter [0 255] colocándose a continuación la cadena)

mensaje "Esto es un mensaje" mensaje TYPE

ARRAY Dimensiona un matriz unidimensional de enteros

ARRAY nombre elementos

ARRAY teléfono 234

PUT Almacena el valor del PAP en un elemento de la matriz designada

nombre elemento PUT

25 teléfono 12 PUT

ASK Lee el valor de un elemento de una matriz entera depositándolo en el PAP

nombre elemento ASK

teléfono 12 ASK

-Varios-

RANDOM Inicializa la serie aleatoria mediante un valor aleatorio o no

valor RANDOM # 13450 RANDOM

RND Devuelve un valor aleatorio en el rango (-32768 , 32767)

RND . ." RND . ." RND .

USR Realiza una llamada a una subrutina en código máquina tomando los valores de los registros del sistema para actualizarlos al retornar

&9f00 DE ! &bc9b USR

! Permite la inclusión de código máquina en el código compilado en forma de cadena hexadecimal

```
# :11009FC59BBCC9    LD DE, 9F00H
                        CALL 0BC9BH
                        RET
```

CLEAR Inicializa el puntero de pila a su posición de partida

```
# 1 2 3 4 5 6 7 CLEAR .
```

ABORT Aborta el programa en cualquier nivel retornando al compilador o al programa que lo llamó.

VARIABLES DEL SISTEMA

AF BC DE HL IX IY

Hacen referencia a los respectivos registros del Z80 que serán inicializados a los valores que posean esas variables antes de realizar una función **USR** del **FORTH**, asignando a esas mismas variables los valores de sus mismos registros una vez terminada la ejecución de la rutina.

SPRITES

Esta variable contiene la dirección donde se almacena en memoria un archivo de sprites. Eso implica que si modificamos esa dirección las palabras que gobiernan esos sprites se dirigirán a la nueva dirección a buscar la información gráfica. Esto nos permite de hecho definir cuantos bancos de 8 sprites queramos, teniendo en cuenta que debemos almacenarlos en posiciones libres de memoria y que antes de utilizar un determinado banco deberemos colocar en la variable **SPRITES** su dirección.

PAD

Ya hemos explicado antes que es una zona de trabajo de 256 bytes de longitud en la que podemos almacenar temporalmente información para su posterior procesado o reubicación.

MENSAJES DE ERROR DEL COMPILADOR

Bad line

La cadena entrada no es interpretable por el compilador

Syntax error

Error de sintaxis. Alguna parte de la entrada no es reconocible como sentencia FORTH

Word missed

No existe nombre para una definición o no existe el “;” de final de definición.

Word not defined

Palabra que no se halla definida en el diccionario

System word

El nombre de palabra utilizado pertenece a las palabras del sistema

Name too long

Se ha intentado asignar un nombre demasiado largo a una definición

Invalid name

Se ha intentado definir el nombre de una palabra con la cadena vacía o con espacios

Word already defined

Se ha utilizado el mismo nombre para dos palabras distintas

Definition missed

No existe definición de procedimientos entre el nombre de una palabra y el “;” final

Number out of range

Número fuera del rango 0 65535, -32768 32767

No room for VAR

Se han intentado definir más de 200 variables

NIBBLE missed

Hay algún error en la cadena hexadecimal que precede al “;” en una definición de lenguaje máquina.

No room for WORDS

Se han intentado definir más de 300 palabras FORTH

Memory full

No hay más espacio para el programa

Division by 0

Se ha intentado dividir un número por 0

Sprite error

Se ha utilizado un número de SPRITE incorrecto o las coordenadas facilitadas están fuera de rango

MANEJO DEL EDITOR DE SPRITES

El manejo del EDITOR DE SPRITES comporta la utilización de las teclas del cursor para el movimiento del lápiz en el área de dibujo y de la barra espaciadora para pintar o borrar.

Esto se complementa con la utilización de las teclas de función que quedan definidas de la siguiente forma:

- F1** Edita un SPRITE determinado en el área de trabajo
- F2** Almacena el área de trabajo en un número de SPRITE
- F3** Graba la información de los SPRITES en Disco o Cinta
- F4** Recupera un archivo de SPRITES para su edición
- F5** Realiza la compilación de los 8 SPRITES, grabando el código resultante en Disco o Cinta. Este fichero será el que deberemos utilizar en el compilador de FORTH.
- F6** Realiza una rotación de 180 grados del área de trabajo.
- F7** Realiza una inversión horizontal del área de trabajo.
- F8** Permite la asignación de tintas a los lápices.
Nótese que se requieren dos valores de tinta por si se desean tintas parpadeantes, en otro caso coloque el

mismo valor por duplicado

Para activar una tinta pulse la tecla correspondiente a la trama de colores situada en la pantalla.

Téngase presente que la información del área de trabajo debemos asignarla a algún SPRITE determinado para poder salvarla o compilarla posteriormente.