

**MSX**  
**2.0**

PREZADO CLIENTE

GOSTARIAMOS DE FELICITA-LO PELA AQUISICAO DE SEU NOVO "KIT"MSX 2.0.

COM ESTE SENSACIONAL PROJETO DE ADEMIR CARCHANO, VOCE TERA EM SUAS MAOS UM NOVO MICRO, COM UMA CAPACIDADE INCRIVELMENTE MAIOR DO QUE A QUE VOCE ESTA ACOSTUMADO A ENCONTRAR EM SEU EQUIPAMENTO ABITUAL.

NO MANUAL QUE VOCE RECEBEU JUNTAMENTE COM SEU KIT, VOCE ENCONTRARA, ALEM DE TODOS OS COMANDOS DO BASIC MSX, NOVOS COMANDOS ACOMPANHADOS DE PROGRAMAS, ONDE VOCE PODERA TESTA-LOS, E VERIFICAR SUA EFICIENCIA.

APOS PREENCHER OS DADOS PEDIDOS NA FOLHA EM ANEXO, DESTAQUE-A E LEVA-A JUNTAMENTE COM SUA CPU E SEU KIT PARA NOSSO ENDERECO DE INSTALACAO E ASSISTENCIA TECNICA QUE E:

AV. PAULISTA, 2001, SOBRELOJA 08

FONES: 285-38-75

285-60-98

CASO SEU MICRO SEJA O EXPERT 1.0, ENVIE-NOS TAMBEM O ADAPTADOR PARA TELEVISAO QUE ACOMPANHA O EQUIPAMENTO (TA-1), PARA QUE SEJA FEITA A DEVIDA ADAPTACAO.

QUANDO RECEBER SEU MICRO DE VOLTA, PECA SEU DOCUMENTO DE GARANTIA, QUE DEVERA ESTAR DEVIDAMENTE CARIMBADO E DATADO, PARA QUE VOCE POSSA DESFRUTAR DA GARANTIA TOTAL QUE OFERECEMOS PELO PERIODO DE TRES(3) MESES, A PARTIR DA ENTREGA DO EQUIPAMENTO.

MAIS UMA VEZ FELICITANDO-OS PELA NOVA AQUISICAO, DESPEDIMO-NOS NA CERTEZA DE MAIS UMA VEZ SERVI-LO COM UM PRODUTO DE BOA QUALIDADE.

M. P. O. SOFTVIDEO

## OPERAÇÃO LÓGICA A CORES:

Com as instruções COPY, LINE, PRESET e PSET, pode usar uma operação lógica a cores, o que significa uma operação lógica entre o número da cor da paleta da instrução (SC) e o número da cor da paleta da zona de destino do screen (DC).

O resultado dessa operação lógica será o número da paleta de cores (C) da zona de destino depois da execução da instrução.

São possíveis as seguintes operações lógicas a cores:

XOR	$C = \text{NOT} (SC) \times DC + SC \times \text{NOT} (DC)$	
OR	$C = SC + DC$	
AND	$C = SC \times DC$	
PSET	$C = SC$	
PRESET	$C = \text{NOT} (SC)$	
TXOR	mesmo que XOR	mas a cor transparente não tem efeito
TOR	mesmo que OR	mas a cor transparente não tem efeito
TAND	mesmo que AND	mas a cor transparente não tem efeito
TPSET	mesmo que PSET	mas a cor transparente não tem efeito
TPRESET	mesmo que PRESET	mas a cor transparente não tem efeito

## CAPÍTULO 1 DESCRIÇÃO DA SINTAXE

Este manual contém descrições de todos os comandos, instruções e funções de "MSX2-BASIC". Estão colocadas em ordem alfabética para vossa convivência.

As descrições foram divididas da seguinte maneira:

**FORMATO** Mostra o formato próprio das instruções (ver notas sobre o formato mais adiante).

**FINALIDADE** Explica o fim da ordem, instrução ou função ou, por outras palavras, como é que se utiliza.

**CATEGORIA** Indica a que categoria pertence a declaração: ordem, instrução ou função.

**OBSERVAÇÕES** Explica como devem ser efectivamente usadas as instruções.

**EXEMPLO** É dado um exemplo para ilustrar a aplicação de cada instrução.

**NOTAS** 1. Itens dentro do formato, que são mostrados em letras maiúsculas, devem ser escritos exactamente como se indica.

2. Itens entre parêntesis < > têm que ser definidos pelo utilizador.

3. Itens entre parêntesis retos [ ] são opcionais.

4. Itens seguidos por três pontos (...) podem ser repetidos tantas vezes quantas o comprimento da linha do programa lhe permita.

5. Todas as pontuações com excepção dos [< >] devem ser escritas cuidadosamente tal como são mostradas. Isto aplica-se a pontos finais, vírgulas, parêntesis, dois pontos, ponto e vírgula, etc. (., ():;....).

6. "X", "Y" e "Z" representam expressões numéricas.

7. "X\$", "Y\$" e "Z\$" representa expressões alfanumérica.

=====

**BASE**

-----

**FORMATO:** BASE (< X >)

**FINALIDADE:** Contém o primeiro endereço das tabelas VDP.

**CATEGORIA:** Variável do sistema.

**OBSERVAÇÕES:** < X > deve ser sempre uma expressão de numeros inteiros de 0 a 44. Os numeros têm o seguinte significado:

0 = Primeiro endereço da tabela de nomes no modo de texto 1.

1 = sem significado.

2 = primeiro endereço da tabela padrão no modo de texto 1.

3 = sem significado.

4 = sem significado.

5 = primeiro endereço da tabela de nomes no modo de texto 2.

6 = primeiro endereço da tabela de cores no modo de texto 2.

7 = primeiro endereço da tabela padrão no modo de texto 2.

8 = primeiro endereço da tabela de "sprites" padrão no modo de texto 2.

9 = primeiro endereço da tabela de "sprites" padrão no modo de texto 2.

10 = primeiro endereço da tabela de nomes no modo gráfico 1.

11 = primeiro endereço da tabela de cores no modo gráfico 1.

12 = primeiro endereço da tabela padrão no modo gráfico 1.

13 = primeiro endereço da tabela de "sprites" no modo gráfico 1.

14 = primeiro endereço da tabela padrão de "sprites" no modo gráfico 1.

15 = primeiro endereço da tabela de nomes no modo gráfico 2.

16 = primeiro endereço da tabela de cores no modo gráfico 2.

17 = primeiro endereço da tabela padrão no modo gráfico 2.

18 = primeiro endereço da tabela de atributos de "sprites" no modo gráfico 2.

19 = primeiro endereço da tabela de "sprites" padrão no modo gráfico 2.

20 = primeiro endereço da tabela das etiquetas no modo gráfico 3.

21 = primeiro endereço da tabela das cores no modo gráfico 3.

22 = primeiro endereço da tabela das formas no modo gráfico 3.

23 = primeiro endereço da tabela dos atributos do "sprite" no modo gráfico 3.

24 = primeiro endereço da tabela das formas "sprite" no modo gráfico 3.

25 = primeiro endereço da tabela das etiquetas no modo gráfico 4.

26 = primeiro endereço da tabela das cores no modo gráfico 4.

27 = primeiro endereço da tabela de formas no modo gráfico 4.

28 = primeiro endereço da tabela de atributos do "sprite" no modo gráfico 4.

29 = primeiro endereço da tabela de formas do "sprite" no modo gráfico 4.

30 = primeiro endereço da tabela de etiquetas no modo gráfico 5.

31 = primeiro endereço da tabela das cores no modo gráfico 5.

32 = primeiro endereço da tabela de formas no modo gráfico 5.

33 = primeiro endereço da tabela de atributos do "sprite" no modo gráfico 5.

34 = primeiro endereço da tabela de formas do "sprite" no modo gráfico 5.

35 = primeiro endereço da tabela de etiquetas no modo gráfico 6.

36 = primeiro endereço da tabela de cores no modo gráfico 6.

37 = primeiro endereço da tabela de formas no modo gráfico 6.

38 = primeiro endereço da tabela de atributos do "sprite" no modo gráfico 6.

39 = primeiro endereço da tabela de formas do "sprite" no modo gráfico 6.

40 = primeiro endereço da tabela de etiquetas no modo gráfico 7.

41 = primeiro endereço da tabela de cores no modo gráfico 7.

42 = primeiro endereço da tabela de formas no modo gráfico 7.

43 = primeiro endereço da tabela de atributos do "sprite" no modo gráfico 7.

44 = primeiro endereço da tabela de formas de

"sprite" no modo gráfico 7.

Quando se lê um endereço de base, o valor válido de < X > varia entre 0 e 44.

Ao escrever um endereço de base, o valor válido de < X > varia entre 0 e 19. Isso significa que só são válidos os modos de texto 1 e 2 e os modos gráficos 1 e 2. Quando um endereço de base do modo gráfico 1 é alterado, os endereços de base dos modos gráficos 3,4,5,6, e 7 também são alterados.

USEM ESTA VARIÁVEL APENAS QUANDO JÁ ESTIVEREM FAMILIARIZADOS COM O MODO DE TRABALHO DO VDP.

Exemplo:     10 SCREEN 0  
              20 PRINT BASE(2)  
              30 END  
              RUN

=====  
BEEP  
-----

FORMATO: BEEP

FINALIDADE: Produzir um som "beep".

CATEGORIA: Instrução.

OBSERVAÇÕES: Esta instrução tem o mesmo efeito que "PRINT CHR\$(7)".

O som pode também ser obtido apertando-se as teclas CTRL e G simultaneamente.

Exemplo:       10 BEEP  
                  20 FOR I=0 TO 1000:NEXT  
                  30 PRINT "AGORA COM CHR\$(7)"  
                  40 PRINT CHR\$(7);  
                  50 END  
                  RUN

O canal de comunicações "RS232C" tem de ser fechado com a instrução "CLOSE" antes de se poder executar este comando.

As teclas de funções F6, F7 e F8 desempenha funções especiais no modo emulador de terminal, como se descreve abaixo:

F6: liga/desliga o modo literal

No modo literal, os códigos de caracteres de 0 a 31 são exibidos com uma seta no alto e o caracter que pertence aos códigos de caracteres de 0 a 31 deslocado de 64.

Por exemplo:

^A. O código de caracteres 0 (zero) sera exibido como

A falta é o modo literal desligado.

F7: ativa o modo "half/full duplex" (meio duplex/duplex total).

No modo "half duplex" (meio duplex), os caracteres teclados são repercutidos ("ecoados") para o screen, bem como enviados através do canal de comunicações "RS232C".

A falta é "full duplex" (duplex total).

F8: liga/desliga o eco da impressora.

Quando o eco da impressão se encontra ligado, todos os caracteres enviados ao screen também são enviados à impressora.

A falta é "eco da impressora desligado".

Exemplo: CALL CONTERM

NOTA: O COMANDO "CALL CONTERM" SÓ PODE SER UTILIZADO QUANDO UMA OU MAIS INTERFACES "RS232C" SE ENCONTRAM INCORPORADAS NO COMPUTADOR "MSX2", OU A ELE LIGADA(S).

=====

## CALL MEMINI

-----

**Formato:** CALL MEMINI [ (<X> ) ]

**Finalidade:** Preparar uma área de memória para o disco de memória.

**Categoria:** Instrução.

**Observações:** < X > é o tamanho do disco de memória e tem de ser um número inteiro maior do que 1023, ou 0 zero). O tamanho real do disco de memória calcula-se como se segue:

$$(INT((<X>-1023)/256)+1)X256$$

O valor de erro de <X> é 32767.

CALL MEMINI (0) anula a função do disco de memória.

O conteúdo do disco de memória será limpo quando for executada a instrução "CALL MEMINI".

Em vez da palavra também se pode usar o sublinhado (    ).

A instrução CALL MEMINI tem de ser executada antes de o disco de memória poder ser utilizado. Após a execução de CALL MEMINI, podem utilizar-se as "declarações" CALL MFILES, CALL MKILL e CALL MNAME.

Após a execução da instrução CALL MEMINI, podem utilizar-se as seguintes "declarações" para ler arquivos que se encontram no disco de memória, ou para escrever arquivos no mesmo:

SAVE, LOAD, RUN, MERGE, OPEN, CLOSE, PRINT#, PRINT #  
USING, INPUT#, LINE INPUT #, INPUTS #, EOF, LOC, LOF. A maneira de usar é a mesma que para um "diskette drive", só que o nome do dispositivo é MEM. Não é possível o acesso aleatório aos arquivos armazenados

no disco de memória.

Exemplo:     10 \_MENINI  
              20 REM  
              30 REM  
              40 REM  
              50 END  
              SAVE "MEM:PROG.BAS"

=====  
CALL MFILES  
-----

**Formato:** CALL MFILES

**Finalidade:** Exibir os nomes dos arquivos armazenados no disco de memória, bem como o tamanho do espaço livre.

**Categoria:** Comando

**Observações:** Em vez da palavra CALL, também se pode usar um sublinhado (  ).

Este comando só pode ser executado se a instrução CALL MENINI tiver sido executada uma vez antes. Se a instrução CALL MENINI não tiver sido já executada ou se o disco de memória for anulado por meio instrução CALL MENINI (0), o comando "CALL MFILES" produzirá a mensagem de erro "disk offline" (disco fora de circuito).

**Exemplo:**   MFILES

=====

## CALL MKILL

-----

**Formato:** CALL MKILL ("<nome do arquivo>")

**Finalidade:** Retirar um arquivo armazenado no disco de memória.

**Categoria:** Instrução.

**Observações:** Em vez da palavra CALL, também se pode usar um sublinhado (  ).

Nome do arquivo é o nome do arquivo que tem de ser retirado.

Esta instrução só pode ser utilizada se instrução "CALL MENINI" tiver sido executada uma vez antes. Se a instrução CALL MENINI não tiver já sido executada ou se o disco de memória for anulado por meio da instrução "CALL MENINI" (0), a instrução "CALL MFILES" produzirá a mensagem de erro "disk offline" (disco fora do circuito).

**Exemplo:**

```
10 _MKILL ("PROG.BAS")
20 PRINT "PROG.BAS FOI APAGADO"
30 END
```

=====

## CALL MNAME

-----

**Formato:** CALL MNAME("<file name-1>"AS"<file name-2>")

**Finalidade:** Alterar o nome de um arquivo armazenado no disco de memória.

**Categoria:** Instrução.

**Observações:** <file name-1> é o nome original do arquivo. ao passo que <file name-2> é o novo nome do arquivo.

Em vez da palavra CALL, pode usar-se o sublinhado (  ):

Esta instrução só pode executar-se se a instrução CALL MENINI não tiver já sido executada ou se o disco de memória tiver sido anulado por meio de instrução CALL MENINI (0), a instrução CALL MNAME produzirá a mensagem de erro "disk offline" (disco fora do circuito).

**Exemplo:**

```
10 MNAME ("PROG.BAS"AS"MYFILE.BAS")
20 MFILES
30 END
```

=====  
CLS  
-----

Formato: CLS

Finalidade: Limpar o Screen

Categoria: Instrução.

Observações: Esta instrução é a mesma que "PRINT CHR\$(12)".

Obtém-se o mesmo efeito apertando-se simultaneamente as teclas SHIFT e HOME ou apertando-se, também simultaneamente, as teclas CTRL e L.

Quando a instrução tiver sido executada, o cursor mover-se-á para a primeira posição da primeira linha.

Exemplo: 10 CLS  
20 PRINT "AGORA COM CHR\$(12)"  
30 FOR I=0 TO 1000:NEXT  
40 PRINT CHR\$(12);  
50 END  
RUN

=====

COLOR

-----

Formato: COLOR [<X>][,<Y>[,<Z>]]

Finalidade: Definir a cor do Screen

Categoria: Instrução.

Observações:<X> é um número inteiro e define a cor do primeiro plano.

<Y> é um número inteiro e define a cor de fundo.

<Z> é um número inteiro e define a cor das áreas limítrofes no modo de texto 2 e em todos os modos gráficos.

<X>,<Y> e <Z> têm de estar compreendido entre: 0 e 15 em todos os modos de texto e nos modos gráficos 1,2,3,4 e 6.

0 e 3 no modo gráfico 5

0 e 255 no modo gráfico 7

A cor real correspondente a cada número da paleta de cores, exceto no modo gráfico 7, é indicada no seguinte quadro da paleta de cores:

=====

nº paleta cor intensidade: vermelho verde azul

-----

nº	cor	Vm	Vd	Az
0	transparente	0	0	0
1	preto	0	0	0
2	verde médio	1	6	1
3	verde claro	3	7	3
4	azul escuro	1	1	7
5	azul claro	2	3	7
6	vermelho escuro	5	1	1
7	ciano	2	6	7
8	vermelho médio	7	1	1
9	vermelho claro	7	3	3
10	amarelo escuro	6	6	1
11	amarelo claro	6	6	4
12	verde escuro	1	4	1
13	magenta	6	2	5

14	cinzento	5	5	5
15	branco	7	7	7

A intensidade do vermelho, do verde e do azul pode ser alterado por meio de instrução COLOR=.

Modo gráfico 5.

Neste modo, podem fazer-se riscas com as cores limítrofes.

Se o número das cores limítrofes se encontrar entre 0 e 3, toda a borda ficará da cor que se encontra definida no quadro da paleta do cores.

Mas se o número das cores limítrofes ficar entre 16 e 31, a borda transforma-se-á em riscas de cores misturadas como se segue:

=====  
 nº das cores      limites das cores da paleta  
 -----

cor      limites

=====  
 16      0 e 0  
 17      0 e 1  
 18      0 e 2  
 19      0 e 3  
 20      1 e 0  
 21      1 e 1  
 22      1 e 2  
 23      1 e 3  
 24      2 e 0  
 25      2 e 1  
 26      2 e 2  
 27      2 e 3  
 28      3 e 0  
 29      3 e 1  
 30      3 e 2  
 31      3 e 3

Modo gráfico 7

Neste modo, o número das cores para o primeiro plano, fundo e bordas pode encontrar-se entre 0 e 255. A cor real é determinada como segue:

**Número da cor = 4XR+32XG+B,**

em que R, G e B são, respectivamente, a intensidade do vermelho, do verde e do azul.

O vermelho e o verde têm de ter um valor entre 0 e 7.

O azul tem de ter um valor entre 0 e 3. Assim, uma cor com intensidade de vermelho, verde e azul de, respectivamente, 7,5 e 2, darão o número de cor

$$190 (=4 \times 7 + 32 \times 5 + 2)$$

Quando não for dado nenhum valor para <X>, <Y> e <Z>, o "MSX2-BASIC" usará as últimas instruções sobre cores para o primeiro plano, fundo e áreas limítrofes (bordas de cores).

O valores de erro <X>, <Y> e <Z> são 15,4 e 7.

Exemplo:

```
10 SCREEN 8
20 OPEN "GRP:" FOR OUTPUT AS #1.
30 PRESET (0,0)
40 FOR I = 0 TO 255 : COLOR I
   :PRINT#1,CHR$(219); : NEXT
50 FOR I = 0 to 2000 : NEXT
60 CLOSE
70 END
```

=====

## COLOR

-----

**Formato:** COLOR=(**<X>**,**<XX>**,**<YY>**,**<ZZ>**)

**Finalidade:** Alterar a intensidade do vermelho, verde e azul do quadro da paleta de cores.

**Categoria:** Instrução.

**Observações:** **<X>** é o número da paleta de cores e tem de ser um número inteiro compreendido entre 0 e 15.

**<XX>** é a intensidade do vermelho e tem de ser um número inteiro compreendido entre 0 e 7.

**<YY>** é a intensidade do verde e tem de ser um número inteiro compreendido entre 0 e 7.

**<ZZ>** é a intensidade do azul e tem de ser um número inteiro compreendido entre 0 e 7.

Esta instrução pode ser utilizada em todos os modos do Screen, exceto no modo gráfico 7.

O número de cor da paleta 0 (zero) é normalmente atribuído ao transparente. Significa que esta cor não é visível, sejam quais forem as intensidades do vermelho, verde e azul. Para usar o número de cor da paleta 0 (zero), tal como quaisquer outros números de cores da paleta, sirva-se da seguinte instrução:

```
VDP (9)=VDP(9)OR&H20
```

Para reajustar o número da cor da paleta 0 (zero) para transparente, utilizar a seguinte instrução:

```
VDP (9)=VDP(9)AND&HDF
```

**Exemplo:**

```
10 SCREEN 4
20 LINE(10,10)-(100,100),2,BF
30 FOR I=0 TO 2000:NEXT
40 COLOR=(2,6,3,6)
50 FOR I=0 TO 2000:NEXT
60 END
```

=====

COLOR = NEW

-----

Formato: COLOR[=NEW]

Finalidade: Repor a intensidade do vermelho, do verde e do azul do quadro da paleta de cores nos valores de erro.

Categoria: Instrução.

Observações: Os valores de erro da intensidade do vermelho, do verde e do azul do quadro da paleta de cores são descritos juntamente com a instrução COLOR.

Exemplo:       10 SCREEN 0:WIDTH 40:COLOR 15  
                  20 COLOR=(15,1,1,2)  
                  30 PRINT "NEW COLOR"  
                  40 FOR I=0 TO 2000:NEXT  
                  50 COLOR=NEW  
                  60 PRINT "DEFAULT COLOUR"  
                  70 END

=====

**COLOR = RESTORE**

-----

**Formato:** COLOR=RESTORE

**Finalidade:** Ajustar a intensidade do vermelho, do verde e do azul do quadro da paleta de cores ao valor armazenando na memória de vídeo.

**Categoria:** Instrução.

**Observações:** Esta instrução é útil quando os dados referentes ao quadro da paleta de cores foram introduzidos pela instrução BLOAD...,S e a intensidade do vermelho, do verde e do azul do quadro da paleta de cores têm de ser alteradas pelos dados acabados de ser carregados.

O quadro da paleta de cores encontra-se armazenado na memória de vídeo como se segue:

=====

Modo da Screen	Área da VRAM
modo de texto 1 (40 caract.)	&H0400- &H0420
modo de texto 1 (80 caract.)	&H0F00- &H0F20
modo de texto 2	&H2020- &H2040
modo gráfico 1	&H2020- &H2040
modo gráfico 2	&H2020- &H2040
modo gráfico 3	&H2020- &H2040
modo gráfico 4	&H7680- &H76A0
modo gráfico 5	&H7680- &H76A0
modo gráfico 6	&HFA80- &HFAA0
modo gráfico 7	&HFA80- &HFAA0

-----

Nos modos gráficos 4,5,6 e 7 pode ajustar-se mais de uma página (ver instrução SET PAGE). Para conhecer o endereço da memória vídeo do quadro da paleta de cores das várias páginas, tem de ser efectuar o seguinte cálculo:

=====

Modo da Screen	Cálculo
----------------	---------

-----

Modo gráfico 4	Nº da página X &H08000+&H7680
Modo gráfico 5	Nº da página X &H08000+&H7680
Modo gráfico 6	Nº da página X &H10000+&HFA80
Modo gráfico 7	Nº da página X &H10000+&HFA80

---

Exemplo:

```
10 SCREEN 0:WIDTH 40:COLOR 15
20 COLOR=(15,0,7,0)
30 PRINT "COLOUR"
40 BSAVE "COL1.PIC",&H420,&H490,S
50 COLOR=NEW
60 FOR I=0 TO 2000:NEXT
70 BLOAD "COL1.PIC",S
80 COLOR=RESTORE
90 END
```

=====

## COLOR SPRITE

-----

-----

**Formato:** COLOR SPRITE (<X>)=<Y>

**FINALIDADE:** Ajustar a cor do "Sprite" especificado.

**Categoria:** Instrução.

**Observações:** Esta instrução só é válida nos modos gráficos 3,4,5,6 e 7

<X> é o número do "Sprite" tal como se encontra registado na variável SPRITE\$(X).

<Y> tem de ser um número inteiro compreendido entre 0 e 111.

Quando <Y> se encontrar entre 0 e 15, este números significam os números da paleta de cores.

Se acrescentar 32 ao número da paleta de cores de <Y>, significa o número da paleta de cores com a seguinte função:

"Ignore coincidence of Sprite". (despreze a coincidência de "Sprite").

Significa ainda que com a coincidência deste "Sprite" com outro "Sprite", o programa não saltará para a sub-rotina dada pela instrução ON SPRITE GOSUB.

Se somar 64 ao número da paleta de cores de <Y>, significa o número da paleta de cores com a seguinte função;

"Ignore priority and coincidence of sprite and use a logical OR operation of sprite colours". (Despreze a prioridade e a coincidência do "sprite" e utilize uma operação lógica OR das cores dos "sprites").

Significa ainda que, com coincidência deste "sprite" com outro "sprite", o programa não saltará para a sub-rotina dada pela instrução ON SPRITE GOSUB mas, nesse caso, as cores deste "sprite" serão a consequência de uma operação lógica OR

As cores dos "sprites" podem ser ajustadas por meio das instruções COLOR SPRITE, COLOR SPRITE\$ ou PUT SPRITE. É válida o número da paleta de cores da última instrução executada. Se desejar ajustar a cor do "sprite" por meio das instruções COLOR SPRITE OU COLOR SPRITE\$, queira omitir o número da paleta de cores da instrução PUT SPRITE.

```
Exemplo: 10 SCREEN 5,0
          20 B$=""
          30 FOR I=1 TO 8:READ A: B# = B$+CHR$(A):NEXT
          40 SPRITE$(0)=B$
          50 COLOR SPRITE (0)=12
          60 FOR I = 0 TO 212:PUT SPRITE 0,(I,I),,0
            :NEXT
          70 DATA 24,60,126,255,36,36,66,129
          80 END
```

=====

## COLOR SPRITES

-----

Formato: COLOR SPRITES(<X>)=<X\$>

Finalidade: Alterar a cor de uma linha do "sprite" especificada.

Categoria: Instrução.

Observações: Esta instrução só é válida nos modos gráficos 3,4,5,6 e 7.

<X> é o número do "sprite" tal como está registado na variável SPRITES(X).

<X\$> é uma expressão em "string" de 1 a 16 caracteres. Cada caracter corresponde a cada linha do "sprite". Quando o código de caracteres de <X\$> estiver entre 0 e 15, estes algarismos significam os números da paleta de cores.

Se somar 32 ao número da paleta de cores de <X\$>, obterá o número da paleta de cores com a seguinte função:

"Ignore coincidence of sprite". (Despreze a coincidência do "sprite").

Isto significa que, com a coincidência da linha deste "sprite" com outro "sprite", o programa não saltará para a sub-rotina dada pela instrução ON SPRITE GOSUB.

Se adicionar 64 ao número da paleta de cores de X\$, obterá o número da paleta de cores com a seguinte função:

"Ignore priority and coincidence of sprite and use a logical OR operation of sprite colours".

(Despreze a prioridade e a coincidência do "sprite" e use uma operação lógica OR das cores dos "sprites").

Isso significa que, com a coincidência da linha deste "sprite" com outro "sprite", o programa não saltará para a sub-rotina dada pela instrução ON SPRITE GOSUB, mas, nesse caso, as cores deste "sprite" serão consequência de uma operação lógica OR.

Se acrescentar 128 ao número da paleta de cores de X\$, isso significará que a linha deste "sprite" se deslocará 32 pontos para a esquerda.

As cores dos "sprites" podem ajustar-se por meio das instruções COLOR SPRITE ou COLOR SPRITES OU PUT SPRITE. É válida o número da paleta de cores da última instrução executada.

Se desejar ajustar a cor do "sprite" por meio das instruções COLOR SPRITE ou COLOR SPRITES, queira omitir o número da paleta de cores da instrução PUT SPRITE.

Exemplo:

```
10 SCREEN 5,0
20 B$=""
30 FOR I=1 TO 8:READ A:B$=B$+CHR$(A):NEXT
40 SPRITES(0)=B$
50 PUT SPRITE 0,(100,100),,0
60 FOR I=0 TO 2000:NEXT
70 COLOR SPRITES(0)=CHR$(12)+CHR$(1)+
  CHR$(130)
80 FOR I=0 TO 2000:NEXT
90 DATSA 24,60,126,255,36,36,66,129
100 END
```

=====  
CONT  
-----

**Formato:** CONT

**Finalidade:** Continuar a execução de um programa depois deste ter sido interrompido.

**Categoria:** Comando.

**Observações:** A execução do programa continuará, começando na linha do programa onde foi interrompida.

A execução de um programa pode ser interrompida com as instruções "STOP" ou "END", mas também apertando-se simultaneamente as teclas CTRL e STOP.

Quando o programa é interrompido depois de uma instrução "INPUT", a execução será retomada com a instrução "INPUT" depois da ordem "CONT".

"CONT" é muitas vezes usada em combinação com a instrução

"STOP" para detectar erros num programa. Quando a execução é interrompida pela instrução "STOP", o conteúdo dum variável pode ser investigado e alterado no modo directo.

**Exemplo:**

```
10 PRINT "PROVA"  
20 STOP  
30 PRINT "USOU-SE CONT"  
40 END  
RUN
```

=====

**COPY**

-----

-----

**Formato:** COPY (<X1>,<Y1>)-(<X2>,<Y2>)[,<XX>]  
TO (<X3>,<Y3>)[,<YY>[,<operação>]]

COPY (<X1>,<Y1>)-(<X2>,<Y2>)[,<XX>]  
TO <array>

COPY (<X1>,<Y1>)-(<X2>,<Y2>)[,<XX>]  
TO "<periférico>:<nome do arquivo>"

COPY <array>[,<sentido>]  
TO (<X3>,<Y3>)[,<YY>[,<operação>]]

COPY "<periférico>:<nome do arquivo>"[,<sentido>] do  
TO (<X3>,<Y3>)[,<YY>[,<operação>]]

COPY <array>  
TO "<periférico>:<nome do arquivo>"

COPY "<periférico>:<nome do arquivo>"  
TO<array>

COPY "<periférico>:<nome do arquivo-1>"  
TO "<periférico>:[<nome do arquivo-2>]"

**Finalidade:** Copiar uma parte de uma Screen gráfica para outra parte da Screen, para um "array" ou para um arquivo de disquete, e para copiar um arquivo numa disquete.

**Categoria:** Instrução.

**Observações:** Copiar uma parte da Screen gráfica só é válido nos modos gráficos 4,5,6 e 7.

<X1> representa a coordenada dos XX do ponto de partida da área-origem do Screen e tem de ser u numero inteiro compreendido entre 0 e 255 no mod gráfico 4 e 7, e entre 0 e 511 nos modos gráficos 5 e 6.

<Y1> representa a coordenada dos YY do ponto de partida da área-origem da Screen e tem de ser u número inteiro compreendido entre 0 e 211.

<X2> representa a coordenada dos XX do ponto terminal da área-origem do Screen e tem de ser u número inteiro entre 0 e 255 nos modos gráficos 4 e 7, e entre 0 e 511 nos modos gráficos 5 e 6.

<Y2> representa a coordenada dos YY do ponto terminal da área-origem da Screen e tem de ser u número inteiro compreendido entre 0 e 211.

<X3> representa a coordenada dos XX do ponto de partida da área de destino da Screen e tem de ser um número inteiro compreendido entre 0 e 255 nos modos gráficos 4 e 7, e entre 0 e 511 nos modos gráficos 5 e 6.

<Y3> representa a coordenada dos YY do ponto de partida da área do destino da Screen e tem de ser um número inteiro compreendido entre 0 e 211.

Nos modos gráficos 4,5,6, e 7, pode ajustar-se mais de uma página (ver instrução SET PAGE).

<XX> é o número da página-origem e <YY> é número da página-destino.

Se <XX> e/ou <YY> não for especificados, assumir-se-á a página ativa.

<sentido> especifica o sentido da transferência. Tem de ser um número inteiro compreendido entre 0 e 3 e tem o seguinte significado:

-----  
**No do sentido      Transferência**  
-----

0	da esquerda superior p/ a direita inferior
1	da direita superior p/ a esquerda inferior
2	da esquerda inferior p/ a direita superior
3	da direita inferior p/ a esquerda superior

-----

<array> é um nome de uma variável. Tem de ser uma fila (array) de variáveis numéricas e deve possuir espaço suficiente para conter os dados da parte da screen. A extensão do "array" pode calcular-se como se

segue:

$$\text{INT}(\langle \text{pixel} \rangle * (\text{ABS}(X2-X1)+1) * (\text{ABS}(Y2-Y1)+1) + 7) / 8 + 4$$

$\langle \text{pixel} \rangle$  é 4 nos modos gráficos 4, 6 e 7, e 2 no modo gráfico 5.

$\langle \text{operação} \rangle$  significa uma operação lógica e pode ser uma das seguintes palavras:

XOR, OR, AND, PSET, PRESET,  
TXOR, TOR, TAND, TSEP e TPRESET.

$\langle \text{periférico} \rangle$  pode ser:

A: = drive 1  
B: = drive 2  
C: = drive 3  
D: = drive 4  
E: = drive 5  
F: = drive 6

$\langle \text{nome do arquivo} \rangle$  é o nome do arquivo que encerrará os dados na Screen.

$\langle \text{nome do arquivo-1} \rangle$  é o nome do arquivo que tem de ser copiado.

$\langle \text{nome do arquivo-2} \rangle$  será o nome do arquivo copiado. Se se omitir  $\langle \text{nome do arquivo-2} \rangle$ , o nome do arquivo do arquivo-meta (arquivo-objectivo) terá o mesmo nome do arquivo-origem.

Exemplo:

```
10 SCREEN 6
20 X1=50:Y1=50:X2=100:Y2=0
30 CIRCLE (X1,Y1),40,3
40 PAINT (X1,Y1),3
50 AA=INT((4*(ABS(X2-X1)+1)*(ABS(Y2-Y1)+1))/8)+4
60 DIM A (AA)
70 COPY (X1,Y1)-(X2,Y2)TO A
80 FOR I=1 TO 2000:NEXT
90 COPY A,2 TO (150,150)
100 FOR I=1 TO 2000:NEXT
110 END
```

=====

**COPY SCREEN**

-----

**Formato:** COPY SCREEN [<X>]

**Finalidade:** Digitar o "input" interno de vídeo

**Categoria:** Instrução.

**Observações:** <X> é o modo de digitar. Deve ser ou 0 ou 1 e tem o seguinte significado:

0 = um campo é digitado e escrito na página exibida

1 = digitam-se e escrevem-se dois campos (1"frame") no número da página exibida menos 1, e na página exibida. Nesse caso, o número da página exibida tem de ser ímpar (ver instrução SET PAGE).

Quando não se dá <X>, utiliza-se o último modo de digitar dado.

O valor de erro de <X> é 0 (zero).

**Exemplo:** 10 COPY SCREEN 1  
20 END

**NOTA:** SÓ SE PODE USAR A SCREEN DE COPIA QUANDO SE ENCONTRA INCORPORADO OU ACOPLADO UM DIGITALIZADOR NO COMPUTADOR MSX2.

=====

## COS

-----

Formato: COS (<X>)

Finalidade: Dar o coseno de <X> em radianos.

Categoria: Função.

Observações: Não há

Exemplo:       10 X=5  
                 20 PRINT COS(X)  
                 30 END

```
50 INPUT "B!";B!  
60 INPUT "C#";C#  
70 INPUT "D$";D$  
80 RSET N1$=MKIS(A#):RSET N2$=MKSS(B!):  
   RSET N3$=MKD(C#):LSET N4$=D$  
90 PUT #1,1  
100 A#=0:B#=0:C#=0:D$=""  
110 PRINT A#;B#;C#;D$  
120 GET #1,1  
130 A#=CVI(N1$):B#=CVS(N2$):C#=CVD(N3$):  
   D$=N4$  
140 PRINT A#;B#;C#;D$  
150 CLOSE #1  
160 END
```

NOTA: "GET" SÓ PODE SER USADO QUANDO 1 OU MAIS DRIVES ESTIVEREM LIGADOS AO COMPUTADOR "MSX2".

=====

## GET DATE

-----

**Formato:** GET DATE <X\$>[,A]

**Finalidade:** Obter a data do "chip" do relógio.

**Categoria:** Instrução.

**Observações:** O formato da data no "string" <X\$> pode ser:

MM/DD/YY ou  
DD/MM/YY ou  
YY/MM/DD

consoante a versão do "MSX2".

DD é a data; MM o mês; YY é um décimo do ano.

Se se especificar o parâmetro facultativo A, presume-se que a data é de alarme. Nesse caso, o sistema só coloca a data (DD) no "string" <X\$>.

**Exemplo:** 10 GET DATE PR\$:PRINT PR\$  
20 END

=====

GET TIME

-----

Formato: GET TIME <X\$>[,A]

Finalidade: Obter as horas do "chip" do relógio.

Categoria: Instrução.

Observações: O formato das horas no "string" <X\$>  
é: HH:MM:SS

HH é a hora; MM os minutos; SS os segundos.

Se se especificar o parâmetro facultativo A, presume-se que as horas são de alarme. Nesse caso, só a hora (HH) e os minutos (MM) serão colocados pelo sistema no "string" <X\$>.

Exemplo: 10 GET TIME PR\$:PRINT PR\$  
20 END

=====

## GOSUB-RETURN

-----

**Formato:**       GOSUB<X>  
                  -  
                  -  
                  -  
                  RETURN [<Y>]

**Finalidade:** Saltar para uma sub-rotina e voltar ao programa principal de novo a partir da sub-rotina.

**Categoria:** Instrução.

**Observações:** <X> é o número da primeira linha do programa da sub-rotina.

Uma sub-rotina pode ser chamada tantas vezes quanto quiser um programa.

Uma sub-rotina pode também ser chamada a partir da outra sub-rotina.

A instrução "RETURN" faz o "MSX2-BASIC" voltar para a instrução seguinte à mais recente instrução "GOSUBB".

Quando uma instrução "RETURN" é seguida por u número de linha, (<Y>) o "MSX2-BASIC" voltará para o número de linha indicado.

**NOTA 2:** "OPEN CAS" SÓ PODE SER USADA QUANDO UM GRAVADOR DE DADOS ESTIVER LIGADO CO COMPUTADOR "MSX2".

**NOTA 3:** "OPEN A-F" SÓ PODE SER USADO QUANDO 1 OU MAIS DISKETTES DRIVE UNITS ESTIVEREM LIGADAS AO COMPUTADOR "MSX2".

**NOTA 4:** "OPEN MEM" SÓ PODE SER USADA QUAND INSTRUÇÃO CALL MENINI JA FOI USADA ANTERIORMENTE.

**NOTA 5:** "OPEN COM" SÓ PODE SER USADA QUANDO UMA OU MAIS INTERFACES RS232C ESTAO INSERIDAS OU LIGADAS AO COMPUTADOR "MSX2".

=====  
OUT  
-----

Formato: OUT <X>, <Y>

Finalidade: Enviar o valor <y> para um "port" de saída (<X>)

Categoria: Instrução.

Observações: <X> e <Y> devem ser numeros inteiros, de 0 a 255. Usa-se a função "INP" para ler um byte de um "port" de entrada.

Exemplo:       10 OUT &HA8, INP(&HA8)  
                  20 END  
                  RUN

=====

**PAD**

-----

**Formato:** PAD <X>

**Finalidade:** Dá o estado de uma placa sensora à caneta de escrita "LIGHTPEN, MOUSE OR TRACK BALL".

**Categoria:** Função.

**Observações:** <X> tem de ser um número inteiro, entre 0 e 19.

Quando <X> tem um valor entre 0 e 3, parte-se do princípio de que há um "touch pad" (placa sensora) ligado à tomada do "joystick" (comando manual).

Quando <X> tem um valor entre 4 e 7, parte-se do princípio de que há um "touch pad" ligado à tomada 2 do "joystick".

Quando <X> tem um valor entre 8 e 11, parte-se do princípio de que há uma "lightpen" (caneta de escrita) ligada. Certifique-se de que PAD (8) regressa a -1 antes de ler os dados da "lightpen".

Quando <X> tem um valor entre 12 e 15, parte-se do princípio de que há um "mouse" (rato) ou uma "track ball" (esfera marcadora) ligada à tomada 1 do "joystick". Certifique-se de que PADS (12) é executado antes de ler os dados do "mouse" ou da "track ball" ligada à tomada 1 do "joystick".

Quando <X> tem um valor entre 16 e 19, parte-se do princípio de que há um "mouse" ou uma "track ball" ligada à tomada 2 do "joystick". Certifique-se de que PAD (16) é executado antes de ler os dados do "mouse" ou da "track ball" ligada à tomada 2 do "joystick".

O estado do "touch pad" é dado quando <X> é 0 ou 4. O resultado dessa função é -1 quando o "pad" tenha sido tocado, e 0 quando ainda não foi tocado.

A coordenada dos XX é dada quando <X> é 1 ou 15.

A coordenada dos YY é dada quando <X> é 2 ou 6.

O estado do comutador de "touch pad" é dado quando <X> é 3 ou 7.

O resultado desta função é -1 quando o comutador tenha sido premido, e 0 quando não foi premido.

O estado da "lightpen" é dado quando <X> é 8. O resultado desta função é -1 quando se dispõe da coordenada dos XX e da coordenada dos YY da "lightpen", e 0 (zero) quando delas não se dispõe.

Quando <X> é 9, obtém-se a coordenada dos XX da "lightpen".

Quando <X> é 10, obtém-se a coordenada dos YY da "lightpen".

Quando <X> é 11, obtém-se o estado do comutador da "lightpen".

O resultado desta função é -1 quando o comutador foi premido, e 0 (zero) quando não tenha sido premido.

Quando <X> é 12 ou 16, obtém-se o estado de "mouse" da "track ball".

O resultado desta função é sempre -1.

Quando <X> é 13 ou 17, obtém-se a coordenada dos XX de um "mouse" ou de uma "track ball".

Quando <X> é 14 ou 18, obtém-se a coordenada dos YY de um "mouse" ou de uma "track ball".

Quando <X> é 15 ou 19, o sistema regressa sempre a 0 (zero).

Utiliza-se a função STRIG para ver o estado dos botões de um "mouse" ou de uma "track ball".

Exemplo:       10 SCREEN 2  
                  20 AA=0  
                  30 IF PAD (0)=0 THEN 20  
                  40 X=PAD(!):Y=PAD (2)  
                  50 IF AA=0 THEN PSET(X,Y) ELSE LINE-(X,Y)  
                  60 AA=1

70 GOTO 30  
RUN

NOTA: "PAD" SÓ PODE SER USADO SE ESTIVER LIGADO AO  
COMPUTADOR "MSX2".

=====  
PAINT  
-----

**Formato:** PAINT [STEP] (<X>,<Y>)[,<Z>[,<XX>]]

**Finalidade:** Preencher uma figura gráfica com uma determinada cor no modo gráfico 1 ou 2.

**Categoria:** Instrução.

**Observações:**<X> equivale à coordenada X do ponto de partida no Screen e deve ser um número inteiro de 0 a 255.

<Y> equivale à coordenada Y do ponto de partida no Screen e deve ser um número inteiro entre 0 e 191.

Quando a palavra "STEP" é usada, os valores <X> e <Y> são interpretados relativamente à posição do cursor. Neste caso <X> e <Y> também podem ser inteiros negativos.

<Z> é o número da cor usada para pintar a figura.

<XX> é o número da cor da linha limite da figura.

Nos modos gráficos 1 e 3 o número <Z> da paleta de cores das tintas tem de ser o mesmo que o da cor da linha limitrofe. Neste caso não se pode dar <XX>.

Nos modos gráficos 2,4,5,6 e 7, o número da paleta das cores das tintas pode ser diferente do número da paleta das cores da linha limitrofe.

<Z> e <XX> têm de ser números inteiros compreendidos entre 0 e 15 nos modos gráficos 1,2,3,4 e 6; entre 0 e 3, no modo gráfico 5, e entre 0 e 255 no modo gráfico 7 (ver instrução COLOR).

Quando não é dado nenhum número da paleta das cores das tintas, utilizar-se-á a última cor do primeiro plano dado.

O valor de erro de <Z> é 15.

Quando não é dado nenhum número da paleta de

cores da linha limítrofe, partir-se-á do princípio de que <XX> é o mesmo número da paleta de cores de <Z>.

Exemplo:      10 SCREEN7:COLOR 15,4,4  
                 20 CIRCLE (180,80),40,8  
                 30 PAINT (180,80),2,8  
                 40 FOR I=1 TO 2000:NEXT  
                 50 END  
                 RUN

=====  
PDL  
-----

**Formato:** PDL (<X>)

**Finalidade:** Dá o estado da placa sensora.

**Categoria:** Função.

**Observações:** Uma ou duas placas sensoras podem ser ligadas ao Computador através das fichas da ligação dos comandos manuais.

Esta função dá um valor entre 0 e 255.

<X> é um número inteiro de 1 a 12.

Quando <X> tem um valor de 1,3,5,7,9 ou 11, presume-se que a placa sensora está ligada à ficha 1 do comando manual.

Quando o valor é 2,4,6,8,10 ou 12, presume-se que a placa sensora está ligada à ficha 2 do comando manual.

<Y> determina o tamanho dos sprites que devem ser usados.

Deve ser um número inteiro entre 0 e 3 e tem o seguinte significado:

- 0= pequenos sprites (8X8 pixels).
- 1= pequenos sprites ampliados para 16X16 pixels.
- 2= grandes sprites (16X16 pixels).
- 3= grandes sprites alargados para 32X32 pixels.

Quando <Y> não é dado, o último tamanho de sprite é usado de novo.

Na falta de indicação o valor de <Y> é zero.

No modo gráfico 2 e 3 apenas se podem colocar 4 "sprites" numa linha da Screen.

No modo gráfico 4,5,6 e 7 o número de "sprites" por linha pode ser de 8.

<Z> determina se o som "click" é ouvido quando uma tecla é premida. Deve ser um 0 ou 1 e tem os seguintes significados:

0= nenhum som "click"  
1= som "click"

Quando <Z> não é dado, o valor do último som "click" é usado de novo.

Na falta de indicação o valor de <Z> é 1.

<XX> dá o "baud rate" para o gravador de dados. Deve ser 1 ou 2 e tem os seguintes significados:

1= 1200 baud  
2= 2400 baud

Quando <XX> não é indicado, o último "baud rate" é indicado.

Quando se não indica o valor de <XX> é 1.

<YY> diz ao Computador se um tipo específico de impressora é usada. Deve ser 0 ou 1 e tem o seguinte significado:

0= MSX printer (impressora MSX)  
1= outro tipo de impressora

Quando <YY> não é dado, mantém-se a última indicação de impressora.

Na ausência de indicação o valor de <YY> é 0.

<ZZ> determina o modo de exibição. Tem de ser um número inteiro entre 0 e 3, com o seguinte significado:

0= normal  
1= entrelaçado  
2= exibição normal com alternativa

paridade/imparidade

3= exibição entrelaçada com alternativa paridade  
imparidade

Nos modos de exibição 2 e 3, o número da página de exibição tem de ser ímpar (ver instrução SET PAGE). Nesses modos, a página de exibição (número ímpar) e a página (número ímpar menos 1) são exibidas alternadamente.

Quando <ZZ> não é dado, utilizar-se-á o último modo de exibição dado.

O valor por defeito de <ZZ> é 0 (zero).

Todo o "sprite" será apagado após a instrução "SCREEN".

Exemplo:      10 FOR I=2 TO 8  
                 20 SCREEN I:COLOR 15,4,4  
                 30 LINE (32,32) - (200,132),3  
                 40 FOR K=0 TO 3000:NEXT  
                 50 NEXT I  
                 60 END  
                 RUN

=====

## SET ADJUST

-----

**Formato:** SET ADJUST (<X>,<Y>)

**Finalidade:** Alterar a regulação inicial da posição anterior esquerda da Screen (0,0).

**Categoria:** Instrução.

**Observações:**<X> tem de ser um número inteiro compreendido entre -7 e 8.

Quando <X> for positivo, a posição superior esquerda da Screen desloca-se para a direita. Quando <X> for negativo, a posição superior esquerda da Screen é deslocada para a esquerda.

<Y> tem de ser um número inteiro compreendido entre -7 e 8.

Quando <Y> for positivo, a posição superior esquerda da Screen desloca-se para cima. Quando <Y> for negativo, a posição superior esquerda da Screen é deslocada para baixo.

Os valores de <X> e <Y> serão armazenados na memória do relógio. Isto significa que, quando se voltar a ligar o sistema, esta instrução será executada automaticamente.

**Exemplo:** 10 SET ADJUST (8,-7)  
20 END

=====

## SET BEEP

-----

**Formato:** SET BEEP <X>, <Y>

**Finalidade:** Alterar a tonalidade e o volume do som "bip".

**Categoria:** Instrução.

**Observações:** <X> representa a tonalidade e tem de ser um número inteiro compreendido entre 1 e 4.

<Y> representa o volume e tem de ser um número inteiro compreendido entre 1 e 4.

0 4 gera o volume máximo.

Os valores de <X> e <Y> serão armazenados na memória do relógio. Isto significa que, quando o sistema voltar a ser ligado, esta instrução será executada automaticamente.

**Exemplo:** 10 SET BEEP 3,2  
20 END

-----  
SET DATE  
-----

**Formato:** SET DATE <X\$>[,A]

**Finalidade:** Regular a data no "chip" do relógio.

**Categoria:** Instrução.

**Observações:** O formato da data no "string" <X\$> pode ser:

MM/DD/YY ou  
DD/MM/YY ou  
YY/MM/DD

consoante a versão do "MSX2".

DD á a data, MM é o mês e YY é o décimo de ano.

Se o parâmetro facultativo A for especificado, assume-se a data como data de alarme. Nesse caso, só se coloca no "clip" do relógio a data (DD).

Para regular uma hora e data de alarme, tem de ser executar a instrução SET TIME...,A antes da instrução SET DATE...,A. Se só se executar a instrução SET TIME...,A, despreza-se a data para gerar o alarme.

O "string" <X\$> será armazenado na memória do relógio. Isto significa que a data será actualizada pelo sistema.

**Exemplo:** 10 SET DATE "01/11/85  
20 END

=====

SET PAGE

-----

Formato: SET PAGE <X>, <Y>

Finalidade: Selecionar uma página da Screen que servirá de página activa e escolher uma página da Screen que será a página de exibição.

Categoria: Instrução.

Observações: Esta instrução só é valida nos modos gráficos 4, 5, 6 e 7.

<X> é o número da página de exibição e tem de ser um número inteiro compreendido entre 0 e 3, conforme o modo gráfico escolhido.

<Y> é o número da página activa e tem de ser um número inteiro compreendido entre 0 e 3, conforme o modo gráfico escolhido.

Uma página activa é aquela em que são activos os "statements" de "input/output".

Uma página de exibição é aquela que é exibida na Screen.

O número total de páginas varia de acordo com os seguintes modos gráficos (com memória vídeo de 128k):

=====

modo gráfico	numero total de páginas
4	0 a 3
5	0 a 3
6	0 a 1
7	0 a 1

-----

Na falta de <X> e <Y> o valor será 0.

Exemplo:

```
10 COLOR 15,4,4
20 OPEN "GRP:" FOR OUTPUT AS #1
30 SCREEN 5
40 SET PAGE 0,0:COLOR 15
50 PRESET (32,32)
60 PRINT #1,"THIS IS PAGENR. 1"
70 SET PAGE 0,1:COLOR 1
80 PRESET (32,16)
90 PRINT #1,"THIS IS PAGENR.1"
100 FOR I=1 TO 5
110 SET PAGE 1,1
120 FOR J=1 TO 1000:NEXT J
130 SET PAGE 0,0
140 FOR J=1 TO 1000:NEXT J
150 NEXT I
160 CLOSE
170 END
```

=====

## SET PASSWORD

-----

**Formato:** SET PASSWORD<X\$>

**Finalidade:** Estabelecer uma senha ou palavra de passe).

**Categoria:** Instrução.

**Observações:**<X\$> é o "string" de senha com máximo de 255 caracteres.

O "string" <X\$> será armazenado na memória do relógio, o que significa que, quando se voltar a ligar o sistema, este exibe a palavra "PASSWORD:" (senha) e aguarda até que o operador escreva a senha correcta. Se o "input" não for igual ao que está armazenado na memória do relógio, o sistema volta a pedir a senha.

Só se pode utilizar uma das instruções "SET PASSWORD", "SET PROMPT" ou "SET TITLE". Se se executar mais do que duas destas instruções, só a última será válida, já que essas instruções se servem da mesma área na memória do relógio.

**Exemplo:** 10 SET PASSWORD "SECRET"  
20 END

=====

## SET PROMPT

-----

**Formato:** SET PROMPT<X\$>

**Finalidade:** Alterar o "PROMPT" (ponto) de "MSX2-BASIC".

**Categoria:** Instrução.

**Observações:** <X\$> é o novo "string" de ponto com o máximo de 6 caracteres.

O novo ponto substituirá o ponto padrão do "MSX2-BASIC", que é "OK".

O "string" de <X\$> será armazenado na memória do relógio, o que significa que, quando se voltar a ligar o sistema, esta instrução será executada automaticamente.

Só se pode utilizar uma das instruções "SET PASSWORD", "SET PROMPT" ou "SET TITLE". Se se executar duas ou mais destas instruções, só será válida última já que essas instruções usam todas a mesma área na memória do relógio.

**Exemplo:**     10 SET PROMPT "HELLO"  
                  20 END

=====

SET SCREEN

-----

Formato: SET SCREEN

Finalidade: Armazenar vários parâmetros relativos ao Screen na memória do relógio.

Categoria: Instrução.

Observações: Os parâmetros armazenados na memória do relógio com esta instrução são:

modo da Screen 0 ou 1  
largura 1 a 80  
número da paleta do primeiro plano 0 a 15  
número da paleta de fundo 0 a 15  
paleta de cor das margens 0 a 15  
exibição das teclas de funções de funções ON ou  
OFF  
comutador do estalido das teclas ON ou OFF  
modo de impressora MSX ou non-MSX  
"baud rate" da cassete 1200 ou 2400  
modo de exibição 0 a 4

Da próxima vez que o sistema for ligado, ficam activos os valores dos parâmetros armazenados na memória do relógio.

Exemplo: 10 SCREEN 0:WIDTH 80:COLOR 1,15:KEY OFF  
20 SET SCREEN  
30 END

=====

## SET TIME

-----

**Formato:** SET TIME <X\$>[,A]

**Finalidade:** Acertar as horas no "chip" do relógio.

**Categoria:** Instrução.

**Observações:** O formato das horas no "string" <X\$> é:  
HH:MM:SS

HH é a hora, MM os minutos e SS os segundos.

Se se especificar o parâmetro facultativo A, as horas devem considerar-se como hora de alarme. Nesse caso, só as horas (HH) e os minutos (MM) são introduzidos na memória do relógio.

O "string" <X\$> será armazenado na memória do relógio, o que significa que o sistema actualizará as horas.

**Exemplo:** 10 SET TIME "10:53:06"

=====

SET TITLE

-----

**Formato:** SET TITLE <X\$>[,<Y>]

**Finalidade:** Estabelecer um título que será exibido na Screen quando o sistema for ligado.

**Categoria:** Instrução.

**Observações:**<X\$> é o "string" do título com u máximo de 6 caracteres. Se a extensão do "string" <X\$> for de 6 caracteres, o sistema aguardará qualquer entrada por meio do teclado após a exibição da Screen de título.

<Y> é o número da paleta de cores da Screen d título.

Tem de ser um número inteiro entre 1 e 4.

O "string" <X\$> e o valor de <Y> serão armazenados na memória do relógio, o que significa que, quando se voltar a ligar o sistema, o título será exibido.

Só se pode usar uma das instruções "SET PASSWORD", "SET PROMPT" ou "SET TITLE". Se se usarem duas ou mais dessas instruções, só a última ser válida, porquanto essas instruções servem-se da mesma área na memória do relógio.

M.P.O. SOFTGUIDED  
285-38-75