



STEVE WEBB

**ROTINAS GRÁFICAS
EM ASSEMBLER**



EDITORA
MANOLE
LTDA.



M S X

ROTINAS GRÁFICAS EM ASSEMBLER



M S X

ROTINAS GRÁFICAS EM ASSEMBLER

STEVE WEBB



EDITORA MANOLE LTDA.
1987

Tradução do Original Francês
MSX - Routines Graphiques en Assembleur
Copyright © 1985, Virgin Books Ltd.

Tradução:
Angelo Eugênio Peccioli

Revisão de Texto:
Orlando Parolini

Capa:
Jerardo Cofré

Editor
Roberto Manole

Não é permitida a reprodução total ou parcial deste livro sem a autorização expressa dos editores.

Direitos reservados para a língua portuguesa adquiridos pela Editora Manole Ltda.

EDITORA MANOLE LTDA
Rua 13 de Maio, 1.026 - C.E.P. 01327
Caixa Postal 1489 - Telefone 287-0746
São Paulo - Brasil

Impresso no Brasil
Printed in Brazil

INTRODUÇÃO

Este livro contém 30 rotinas gráficas (subprogramas escritos em linguagem de máquina) prontas para serem inseridas nos programas de jogos em BASIC.

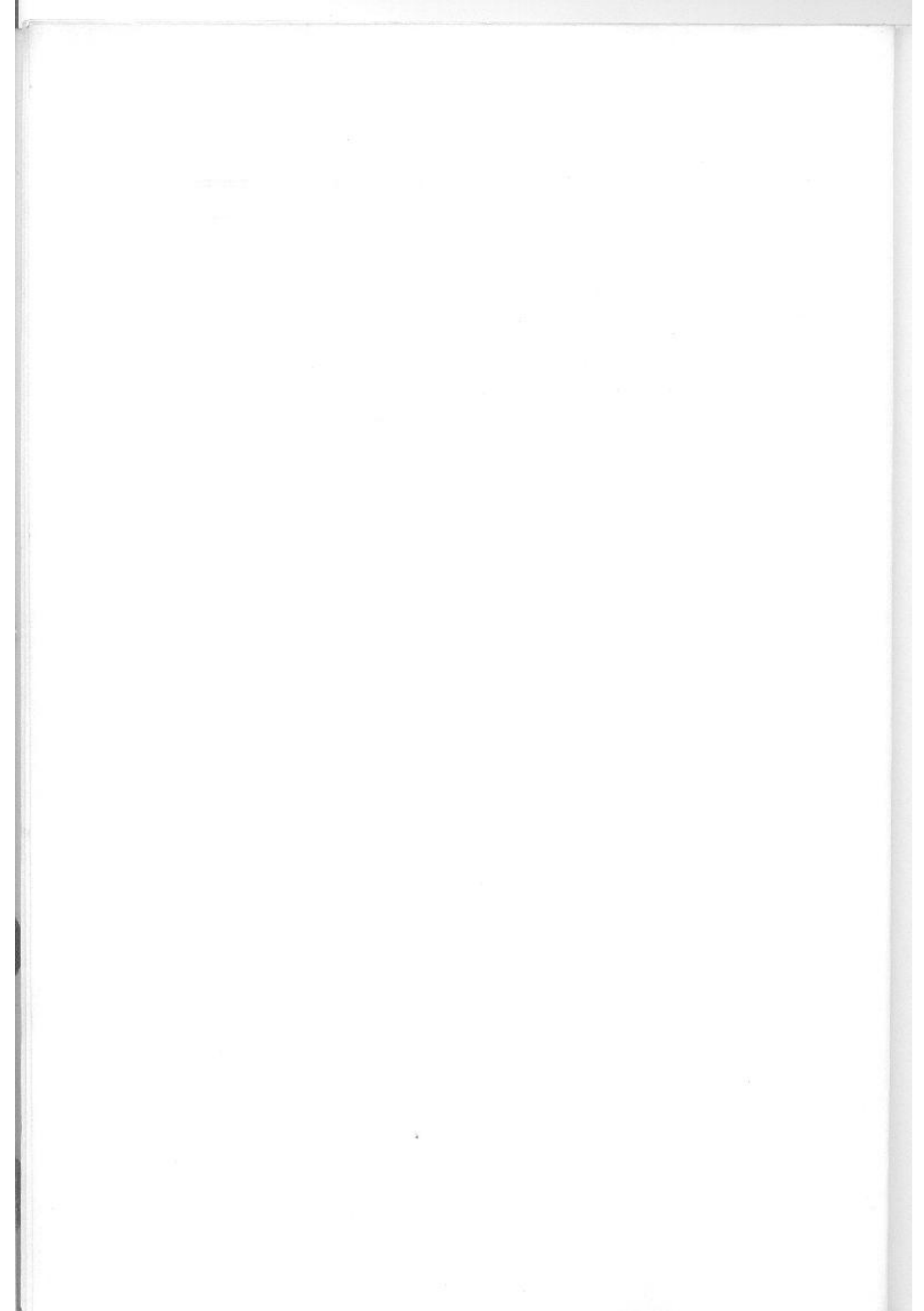
Estas rotinas permitem o deslocamento de caracteres no vídeo a uma velocidade cem vezes superior à do BASIC.

Com essas rotinas, podemos criar programas de jogos muito mais interessantes, pois não ficamos limitados à lentidão do BASIC.

A utilização desses subprogramas em linguagem de máquina não exige nenhum conhecimento especial dessa linguagem. Mais adiante, e de maneira bastante clara, mostraremos como implantar esses subprogramas em programas BASIC.

Apenas a descrição da eficiência dessas rotinas não é o suficiente; é preciso vê-las em funcionamento para que se possa apreciar todos os seus efeitos.

Bem, agora, ao teclado e mãos à obra!



A MEMÓRIA

A MEMÓRIA

Eis aqui um breve esquema da estrutura e do funcionamento da memória de um computador MSX.

Examinando o diagrama abaixo, podemos perceber que a memória está dividida em "casas" numeradas de 0 a 65535. Isto representa o número total de lugares de memória de que dispõe um computador MSX 64K. Esta memória está estruturada em duas partes principais.

A primeira parte vai do endereço 0 ao endereço 32767 e chama-se ROM (memória morta). A ROM contém as informações necessárias ao funcionamento do computador. Ela é uma espécie de gerente do sistema, e possibilita a partida inicial de um programa BASIC, a verificação da sintaxe dos comandos e a execução desses comandos.

Endereço 0

ROM

Endereço 32767

Endereço 32768

RAM

Endereço 65535

A segunda parte vai do endereço 32768 ao endereço 65535, e se chama RAM (memória viva). A RAM serve para armazenar os programas BASIC. Se digitarmos um programa BASIC de duas ou três linhas, ele começará no endereço 32768 e terminará mais ou menos no endereço 32820. Quanto maior for o tamanho do programa, tanto mais ele tenderá em direção ao maior endereço da memória.

Cada endereço da memória corresponde a um byte. Um byte é composto de oito bits e permite o armazenamento de um caractere ou de um valor.

Suponhamos que para armazenar nossos programas, dispomos de 32768 lugares de memória. Na realidade, o computador utiliza apenas uma parte desse espaço para a memória (entre os endereços 61500 e 65535). Esta parte da memória é chamada BIOS. A memória acessível ao usuário se estende, pois, do endereço 32768 ao endereço 61499. Isto explica porque a mensagem "28815 BYTES FREE" aparece no vídeo quando o computador é ligado.

O endereço 61500 é chamado pico da memória (ou seja o endereço mais alto, acessível ao usuário). Se digitarmos um programa tão longo que chegue a atingir esse endereço, a mensagem "OUT OF MEMORY" aparecerá no vídeo. Isto significa que não há mais lugar na memória para introduzir novas linhas de comando.

Quando digitamos um programa BASIC, não nos preocupamos em saber em que lugar da memória as linhas serão armazenadas. Quando programamos em linguagem de máquina, é essencial sabermos em que endereço começa o programa. A cada rotina deve ser alocado um endereço de partida específico. A rotina 1, por exemplo, começa no endereço 58000 e ocupa 60 bytes na memória. A rotina 2 poderá começar no endereço 58060. Se introduzirmos os códigos de máquina na memória, sem as precauções prévias, os programas BASIC podem sobrepor-se aos subprogramas em linguagem de máquina. É preciso, pois, que encontremos um meio de organizar a memória a fim de que os dois tipos de programa possam conviver sem riscos.

Um programa BASIC não pode ultrapassar o endereço mais alto da memória. O BASIC MSX nos permite deslocar esse limite. Digita-se o comando.

CLEAR 200,56000

O endereço mais alto acessível pelo BASIC é agora o endereço 56000, significando que os programas BASIC não poderão ultrapassar esse limite.

Podemos utilizar a zona da memória que começa no endereço 56000 para armazenar nossos subprogramas em linguagem de máquina. Teoricamente, pode-se colocar o pico da memória em qualquer endereço. O comando

```
CLEAR 200,45321
```

colocará o pico da memória no endereço 45321.

Não podemos introduzir os códigos de máquina na memória simplesmente digitando-os no teclado. Abaixo vamos mostrar um programa BASIC que permite a entrada dos códigos de máquina na memória. Após termos digitado esse programa, devemos certificar-nos de que ele não possui nenhum erro. Em seguida, guardamos o programa digitando:

```
SAVE"CAS:ENTHEX"
```

```
10 CLEAR 200,56000
20 CLS
30 LOCATE 0,0
40 PRINT "Aperte CAPS LOCK"
50 LOCATE 0,4
60 PRINT "Tecla E para entrar um código hex"
70 LOCATE 0,8
80 PRINT "Tecla C para verificar um código hex"
90 LOCATE 0,12
100 PRINT "Tecla X para verificar a soma total"
110 LOCATE 0,16
120 PRINT "Tecla Q para parar"
130 A$=INKEY$
140 IF A$="E" THEN 360
150 IF A$="C" THEN 650
160 IF A$="X" THEN 190
170 IF A$="Q" THEN STOP
180 GOTO 70
190 CLS
200 LOCATE 0,0
210 PRINT "Entre o endereço inicial"
220 INPUT SA
230 LOCATE 0,5
240 PRINT "Entre o endereço final"
250 INPUT EA
260 LET D=0
```

```

270 FOR C=SA TO EA
280 LET D=D+PEEK(C)
290 NEXT C
300 CLS
310 PRINT "Total da soma=";D
320 LOCATE 0,20
330 PRINT "Tecla M para retornar o menu"
340 IF INKEY# <> "M" THEN GOTO 340
350 GOTO 20
360 CLS
370 LOCATE 0,0
380 PRINT "A parte CAPS LOCK"
390 LOCATE 0,4
400 PRINT "Entre o endereco inicial"
410 INPUT S
420 IF S<56000 THEN GOTO 630
430 LET A$=""
440 LOCATE 0,23
450 LET ET=S
460 IF A$="" THEN INPUT A$
470 LET BAD=0
480 IF A$="M" THEN 20
490 LET C=VAL("&H"+A$)
500 IF C=0 THEN GOTO 610
510 IF BAD=1 THEN GOTO 580
520 LOCATE 0,21:PRINT S;" ";A$
530 LET C=VAL("&H"+A$)
540 POKE (S),C
550 LET S=S+1
560 GOTO 430
580 LOCATE 0,22:PRINT "Entrada incorreta.
Tente de novo."
590 LET S=ET
600 GOTO 430
610 IF A#<>"00" THEN LET BAD=1
620 GOTO 510
630 PRINT "O endereco inicial deve ser
maior ou igual a 56000"
640 GOTO 400
650 LET ND=0
660 CLS
670 LOCATE 0,0
680 PRINT "Entre o endereco inicial"
690 INPUT SA
700 LOCATE 0,5
710 PRINT "Entre o endereco final"

```

```

720 INPUT EA
730 CLS
740 IF BA+20>EA THEN GOTO 870
750 FOR C=BA TO BA+20
760 IF PEEK(C)<1& THEN GOTO 850
770 PRINT C;" ";HEX$(PEEK(C))
780 NEXT C
790 IF ND=1 THEN GOTO 900
800 IF C>EA THEN GOTO 505
810 PRINT "Tecla M para mais"
820 LET BA=C
830 IF INKEY$ <> "M" THEN GOTO 830
840 GOTO 740
850 PRINT C;"0";HEX$(PEEK(C))
860 GOTO 780
870 FOR C=BA TO EA
880 LET ND=1
890 GOTO 760
900 PRINT "Tecla M para voltar ao menu"
910 IF INKEY$ <> "M" THEN GOTO 910
920 GOTO 20

```

Para introduzir códigos de máquina na memória, carregamos, primeiramente, o programa BASIC abaixo:

LOAD"CAS:ENTHEX",R

O programa se inicia automaticamente após ter sido carregado, e envia as seguintes instruções:

**DIGITAR E PARA ENTRAR UM CÓDIGO DE MÁQUINA
DIGITAR C PARA VERIFICAR O CÓDIGO
DIGITAR X PARA O CÓDIGO DE VERIFICAÇÃO
DIGITAR Q PARA PARAR O PROGRAMA**

Vejamos agora como introduzir um código de máquina na memória, verificá-lo e guardá-lo em fita. Utilizamos a rotina 1 para ilustrar este exemplo:

Digitamos a tecla E para introduzir um código de máquina. No vídeo aparece uma mensagem:

**CERTIFIQUE-SE DE QUE A TECLA CAPS LOCK ESTÁ
ATIVADA**

O programa pede, em seguida, o endereço inicial da rotina. Este endereço é dado ao início de cada rotina. No caso da rotina 1, o endereço inicial é 56000.

Agora o programa está pronto para receber os códigos de máquina. Ao observar a listagem da primeira rotina, notamos que ela é composta de várias linhas, cada uma com oito valores hexadecimais de dois algarismos. O primeiro valor é 21. Digitamos 21 e pressionamos a tecla ENTER. O segundo valor é 00. Digitamos 00 e pressionamos a tecla ENTER. Procedemos desta maneira até o último valor que é AF. A seguir, introduzimos os valores da segunda linha da mesma maneira. Este procedimento se repete até que todos os valores tenham sido digitados. Todas as rotinas terminam com a letra M, para indicar ao programa que todos os valores foram introduzidos e que ele deve retornar à rotina principal.

Verificamos então se todos os dados foram introduzidos corretamente digitando a tecla C. O programa pede o endereço inicial e o endereço final da rotina. Esses endereços são apresentados em cada rotina. No caso da rotina 1, são respectivamente 56000 e 56088. Os códigos de máquina aparecem no vídeo e nós os comparamos cuidadosamente com a listagem original. Quando esta operação estiver terminada, voltamos à rotina principal apertando a tecla M.

A seguir, pressionamos a tecla X para obter o código de verificação. O programa pede novamente os endereços inicial e final da rotina. Após um curto espaço de tempo, um número aparece no vídeo. Esse número corresponde ao resultado da soma do valor de todos os bytes dos códigos introduzidos na memória. No caso da rotina 1, o número é 10572. Esse número é apresentado em cada rotina.

As verificações são necessárias porque, se os códigos contêm erros, os dados da memória correm o risco de serem afetados quando do início do programa. Quando, no curso das verificações, um erro é detectado, voltamos à rotina principal e recomeçamos a entrada dos códigos um a um. Quando temos certeza de que todos os códigos digitados estão corretos, podemos guardá-los em fita. Digitamos a tecla Q para parar o programa. O comando utilizado para guardar esses códigos toma a forma seguinte:

BSAVE"CAS:NOME", Endereço Inicial, Endereço Final.

No caso da rotina 1, digitamos:

BSAVE"CAS:ROT1", 56000,56088

Para maior segurança, guardamos as rotinas em duas fitas diferentes.

UTILIZAÇÃO DAS ROTINAS

UTILIZAÇÃO DAS ROTINAS

Vejamos agora como utilizar uma rotina em um programa BASIC. Desligamos e religamos o computador, para termos a certeza de que a memória está totalmente vazia. Antes de carregar uma rotina, digitamos o comando:

```
CLEAR 200,56000
```

Podemos agora carregar na memória uma ou várias rotinas. Para carregar a rotina 1, digitamos:

```
BLOAD"CAS:ROT1"
```

A utilização da rotina exige a gravação de um curto programa BASIC. Como a rotina 1 serve para deslocar caracteres sobre o vídeo, vamos gravar um programa que apresente os caracteres no vídeo. Eis o programa BASIC:

```
5 SCREEN 0
10 LET X = 65
20 FOR A = 6144 to 6719
30 VPOKE (A), X
40 LET X = X + 1
50 X = 90 THEN LET X = 65
60 NEXT A
70 POKE 62000, 10
75 POKE 62001, 5
80 POKE 62002, 11
85 POKE 62003, 6
90 DEFUSR = 56000
100 B =USR(0)
110 GOTO 100
```

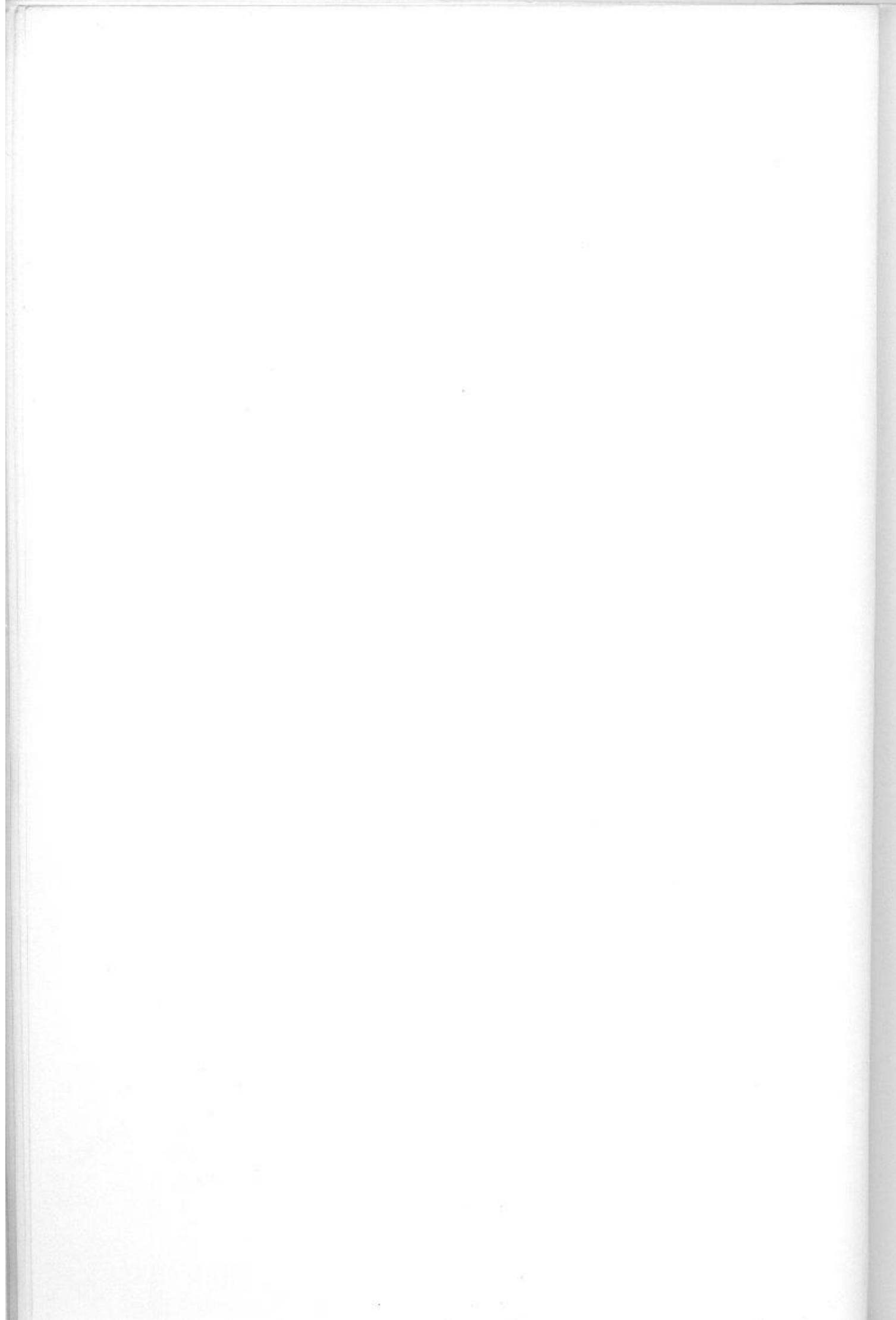
As linhas de 10 a 60 enchem o vídeo de caracteres. As linhas 70 a 85 informam os subprogramas sobre as zonas do vídeo, no interior das quais os caracteres serão deslocados. As linhas 90 a 100 contêm as instruções que permitem o acesso ao subprograma em linguagem de máquina. A linha 110 é um loop que permite modificar o vídeo continuamente.

Podemos guardar o programa BASIC em fita. Iniciamos o programa e vemos os caracteres aparecerem no vídeo. Em seguida, o conteúdo das zonas especificadas pelo programa é deslocado. Podemos tornar o programa mais lento através da inserção da linha:

```
105 FOR C=1 TO 1000:NEXT C
```

Podemos modificar os valores das linhas 70 a 85 a fim de deslocar o conteúdo de outras zonas do vídeo.

O funcionamento e a maneira de testar cada rotina estão descritos com detalhes. É importante seguir a ordem do livro, introduzir as rotinas uma a uma e, em seguida, guardá-las. Com um pouco de prática você poderá até mesmo escrever seus próprios programas BASIC e utilizar, por exemplo, essas rotinas em programas de jogos.



AS ROTINAS

Função:

Deslocamento de caracteres para cima.

Endereço inicial:	56000
Endereço final:	56088
Código de verificação:	10572

Descrição:

O endereço 62000 contém a coordenada vertical correspondente ao canto superior esquerdo da zona no interior da qual os caracteres são deslocados. No modo vídeo 1, este número está compreendido entre 1 e 32.

O endereço 62001 contém a coordenada horizontal correspondente ao canto superior esquerdo desta mesma zona. No modo vídeo 1, este número está compreendido entre 1 e 24.

O endereço 62002 contém um número compreendido entre 1 e 24 que representa a altura da zona.

O endereço 62003 contém um número compreendido entre 1 e 32 que representa a largura da zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	D0	DA	01	28	00
3A	31	F2	3D	CA	DB	DA	09
C3	D3	DA	11	00	00	3A	30
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	33	F2	F5	CD
4A	00	32	40	F2	3A	32	F2
3D	F5	09	CD	4A	00	ED	42
CD	4D	00	09	F1	3D	C2	F9
DA	3A	40	F2	CD	4D	00	13
D5	E1	F1	3D	C2	EE	DA	C9
01	M						

Função:

Deslocar os caracteres para baixo.

Endereço inicial:	56100
Endereço final:	56191
Código de verificação:	10442

Descrição:

O endereço 62004 contém a coordenada vertical correspondente ao canto inferior esquerdo da zona no interior da qual os caracteres são deslocados.

O endereço 62005 contém a coordenada horizontal desta mesma zona.

O endereço 62006 contém um número correspondente à altura da zona.

O endereço 62007 contém um número correspondente à largura desta zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	34	DB	01	28	00
3A	35	F2	3D	CA	3F	DB	09
C3	37	DB	11	00	00	3A	34
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	37	F2	F5	CD
4A	00	32	40	F2	3A	36	F2
3D	F5	A7	ED	42	CD	4A	00
09	CD	4D	00	A7	ED	42	F1
3D	C2	5D	DB	3A	40	F2	CD
4D	00	13	D5	E1	F1	3D	C2
52	DB	C9	02	M			

Função:

Deslocar os caracteres para a esquerda.

Endereço inicial: 56200

Endereço final: 56289

Código de verificação: 10551

Descrição:

O endereço 62008 contém a coordenada vertical correspondente ao canto superior esquerdo da zona no interior da qual os caracteres são deslocados.

O endereço 62009 contém a coordenada horizontal desta mesma zona.

O endereço 62010 contém um número que representa a altura desta zona.

O endereço 62011 contém um número relativo à largura desta zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	98	DB	01	28	00
3A	39	F2	3D	CA	A3	DB	09
C3	9B	DB	11	00	00	3A	38
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	3A	F2	F5	CD
4A	00	32	40	F2	3A	3B	F2
3D	F5	23	CD	4A	00	2B	CD
4D	00	23	F1	3D	C2	C1	DB
3A	40	F2	CD	4D	00	C5	E1
19	E5	D1	F1	3D	C2	B6	DB
C9	03	M					

Função:

Deslocar os caracteres para a direita.

Endereço inicial:	56300
Endereço final:	56389
Código de verificação:	10311

Descrição:

O endereço 62012 contém a coordenada vertical correspondente ao canto superior direito da zona no interior da qual os caracteres são deslocados.

O endereço 62013 contém a coordenada horizontal desta zona.

O endereço 62014 contém um número que representa a altura desta zona.

O endereço 62015 contém um número relativo à largura desta zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	FC	DB	01	28	00
3A	3D	F2	3D	CA	07	DC	09
C3	FF	DB	11	00	00	3A	3C
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	3E	F2	F5	CD
4A	00	32	40	F2	3A	3F	F2
3D	F5	2B	CD	4A	00	23	CD
4D	00	2B	F1	3D	C2	25	DC
3A	40	F2	CD	4D	00	C5	E1
19	E5	D1	F1	3D	C2	1A	DC
C9	04	M					

Utilizamos o programa BASIC abaixo para testar as rotinas 1, 2, 3 e 4. Em primeiro lugar, esse programa mostra os caracteres no vídeo, e, em seguida, desloca o conteúdo de 4 zonas distintas do vídeo.

```
10 SCREEN 1
20 LET X = 65
30 FOR A = 6144 TO 6911
40 VPOKE (A),X
50 LET X = X+1
60 IF X=90 THEN X=65
70 NEXT A
80 POKE 60000,3
90 POKE 60001,2
100 POKE 60002,5
110 POKE 60003,7
115 POKE 60004,20
120 POKE 60005,10
130 POKE 60006,7
140 POKE 60007,3
150 POKE 60008,16
160 POKE 60009,15
170 POKE 60010,4
180 POKE 60011,9
190 POKE 60012,25
200 POKE 60013,22
210 POKE 60014,2
230 POKE 60015,6
240 DEF USR = 56000
250 A = USR (0)
260 DEF USR = 56100
270 A = USR(0)
280 DEF USR = 56200
290 A = USR(0)
300 DEF USR =56300
310 A = USR(0)
330 GOTO 240
```

ROT5

Função:

Esta rotina permite que se faça uma rotação de 90 graus dos caracteres situados em uma zona determinada. Cada vez que se utiliza esta rotina, os caracteres efetuam uma nova rotação de 90 graus. Ela possibilita também a afixação de texto em todos os lados do vídeo.

Endereço inicial:	56400
Endereço final:	56471
Código de verificação:	8491

Utilização:

Somente no modo 1.

21	00	00	3E	FF	F5	E5	3E
08	F5	CD	78	DC	F1	23	3D
C2	59	DC	E1	E5	13	EB	01
08	00	CD	5C	00	E1	01	08
00	09	F1	3D	C2	55	DC	C9
11	62	F2	CD	4A	00	47	3E
08	F5	CD	8F	DC	F1	3D	C2
81	DC	C9	CD	4A	00	47	1A
CB	10	CB	17	12	1B	C9	05
M							

Função:

Com esta rotina, todos os caracteres mostrados visualmente com uma cor de segundo plano específico tomarão uma nova coloração. Quando esta rotina é utilizada, os caracteres mostrados sobre um fundo verde aparecerão sobre um fundo amarelo.

Endereço inicial:	56500
Endereço final:	56535
Código de verificação:	4710

Descrição:

O endereço 62017 contém um número que representa a cor de segundo plano a ser substituída.

O endereço 62018 contém um número que representa a nova cor.

Os códigos de cores são dados no Apêndice deste livro.

Utilização:

Somente no modo 1.

ED	4B	41	F2	21	00	20	3E
20	F5	CD	4A	00	E6	0F	B8
C2	D0	DC	CD	4A	D0	E6	F0
B1	CD	4D	00	23	F1	3D	C2
BD	DC	C9	06	M			

O programa abaixo é utilizado quando do emprego das duas rotinas anteriores. O programa mostra, primeiramente, os caracteres sobre diferentes cores de segundo plano. Após alguns instantes, todos os caracteres são mostrados sobre um fundo verde.

```

10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911

```

```
40 VPOKE (A), X
50 LET X = X+1
60 IF X = 256 THEN LET X=0
70 NEXT A
80 FOR X = 17 TO 31
90 VPOKE (8175+X),X
100 NEXT X
110 FOR X = 17 TO 31
120 VPOKE (8190+X),X
130 NEXT A
140 FOR X = 1 TO 1000: NEXT X
150 POKE 62017,12
160 POKE 62018,15
170 DEF USR = 56500
180 A = USR (0)
```

Função:

Esta rotina permite a modificação da cor dos caracteres mostrados no vídeo.

Endereço inicial:	56550
Endereço final:	56585
Código de verificação:	4556

Descrição:

O endereço 62019 contém um número que representa a cor de tinta a ser substituída.

O endereço 62020 contém um número que representa a nova cor de tinta.

Os códigos de cores são descritos no Apêndice deste livro.

Utilização:

Somente no modo 1.

ED	4B	43	F2	21	00	20	3E
20	F5	CD	4A	00	E6	F0	B8
C2	02	DD	CD	4A	00	E6	0F
B1	CD	4D	00	23	F1	3D	C2
EF	DC	C9	07	M			

O programa BASIC seguinte permite a utilização desta rotina. Em primeiro lugar, o programa mostra os caracteres de diferentes cores no vídeo. Após alguns instantes, todos os caracteres de cor vermelha se tornam brancos.

```

10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
50 LET X = X+1

```



```
60 IF X = 256 THEN LET X=0
70 NEXT A
80 LET X = 21
90 FOR A = 0 TO 14
100 VPOKE (8192+A),X
110 LET X = X + 16
120 NEXT A
130 LET X = 21
140 FOR A = 0 TO 14
150 VPOKE (8208+A),X
160 LET X = X + 16
170 NEXT A
180 FOR X = 1 TO 1000: NEXT X
190 POKE 62019,240
200 POKE 62020,96
210 DEF USR = 5655
220 A = USR (0)
```

Função:

Esta rotina permite substituir um caractere por outro. Por exemplo, substituir a letra "E" pela letra "X".

Endereço inicial:	56600
Endereço final:	56629
Código de verificação	3047

Descrição:

O endereço 62021 contém o código ASCII do caractere a ser substituído.
 O endereço 62022 contém o código ASCII do caractere de substituição.
 A lista dos códigos ASCII é apresentada no Apêndice deste livro.

Utilização:

No modo 0 e no modo 1.

21	00	18	ED	4B	45	F2	11
00	03	CD	4A	00	B8	C2	2D
DD	79	CD	4D	00	23	1B	7B
B2	C2	22	DD	C9	08	M	

O programa BASIC abaixo é utilizado para se empregar a rotina precedente. Em primeiro lugar, o programa mostra os caracteres no vídeo. Após alguns instantes, cada letra "A" é substituída pela letra "X".

```

10 SCREEN 1
20 LET X = 65
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
    
```

```
50 LET X = X+1
60 IF X = 90 THEN LET X=65
70 NEXT A
80 FOR X = 1 TO 1000: NEXT X
90 POKE 62021,88
100 POKE 62022,65
110 DEF USR = 56600
120 A = USR (0)
```

Função:

No modo 1, é possível colocar-se no vídeo simultaneamente 32 combinações de cores. Estas combinações estão armazenadas em 32 locais de memória da memória do vídeo. O primeiro endereço determina a cor dos caracteres de 0 a 7; a segunda, a dos caracteres de 8 a 15, e assim por diante até a última, que determina a cor dos caracteres 248 a 255. Esta rotina permite um desfile de cores. O primeiro bloco de oito caracteres toma a cor do segundo, o segundo toma a cor do terceiro e assim por diante.

Endereço inicial:	56650
Endereço final:	56686
Código de verificação:	3150

Utilização:

Somente no modo 1.

```

21 00 20 01 1F 00 CD 4A
00 32 40 F2 23 CD 4A 00
2B CD 4D 00 23 23 0B 7B
B1 C2 57 DD 2B 3A 40 F2
CD 4D 00 C9 09 M
    
```

O programa BASIC abaixo permite o emprego da rotina precedente. Caracteres de diferentes cores aparecem no vídeo; após alguns instantes, produz-se um desfile de cores.

```

10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
    
```

```
50 LET X = X+1
60 IF X = 256 THEN LET X = 0
70 NEXT A
80 LET X = 17
90 FOR A = 0 TO 31
100 VPOKE (8192+A),X
110 X = + 3
120 NEXT A
130 FOR X = 1 TO 1000: NEXT X
140 DEF USR = 56650
150 A = USR (0)
160 GOTO 150
```

Função:

Esta rotina inverte as cores de primeiro e segundo plano de todos os caracteres. Se um caractere aparece em branco sobre um fundo preto, após a execução da rotina esse mesmo caractere aparecerá em preto sobre um fundo branco.

Endereço inicial:	56700
Endereço final:	56744
Código de verificação:	4757

Utilização:

Somente no modo 1.

21	00	20	3E	20	F5	CD	4A
00	E6	0F	47	CD	4A	00	CB
0F	CB	0F	CB	0F	CB	0F	E6
0F	CB	00	CB	00	CB	00	CB
00	B0	CD	4D	00	23	F1	3D
C2	81	DD	C9	0A	M		

O programa BASIC apresentado abaixo permite o emprego desta rotina. O vídeo se enche, primeiramente, de caracteres de diferentes cores. Em seguida, as cores de primeiro e segundo plano se invertem.

```

10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
50 LET X = X+1
60 IF X = 256 THEN LET X=0
70 NEXT A
80 LET X = 17
90 FOR A = 0 TO 31
100 VPOKE (8192+A),X
    
```

```
110 X = + 3
120 NEXT A
130 FOR X = 1 TO 1000: NEXT X
140 DEF USR = 56700
150 A = USR (0)
160 FOR X = 1 TO 300: NEXT X
170 GOTO 150
```


ROT11

Função:

Em sua concepção, esta rotina se assemelha à rotina 10. Ao invés de inverter as cores de todos os caracteres, ela inverte somente a cor dos caracteres que tenham uma cor de fundo específica.

Endereço inicial:	56750
Endereço final:	56807
Código de verificação:	6823

Descrição:

O endereço 62024 contém um número correspondente à cor de fundo da zona no interior da qual as cores serão invertidas.

Utilização:

Somente no modo 1.

21	00	20	ED	4B	47	F2	3E
20	F5	CD	4A	00	E6	0F	B8
C2	E0	DD	CD	4A	00	E6	0F
5F	CD	4A	00	CB	0F	CB	0F
CB	0F	CB	0F	E6	0F	CB	03
CB	03	CB	03	CB	03	B3	CD
4D	00	23	F1	3D	C2	B7	DD
C9	0B	M					

O programa BASIC apresentado abaixo permite o funcionamento desta rotina. Primeiro, o vídeo se enche de caracteres de cores diferentes. Após alguns segundos, as cores de fundo e de impressão dos caracteres mostrados sobre um fundo branco são invertidas.

```
10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
50 LET X = X+1
60 IF X = 256 THEN LET X=0
70 NEXT A
80 LET X = 17
90 FOR A = 0 TO 31
100 VPOKE (8192+A),X
110 X = + 1
120 NEXT A
130 FOR X = 1 TO 1000: NEXT X
140 POKE 62024, 15
150 DEF USR = 56750
160 A = USR (0)
```

Função:

Em sua concepção, esta rotina se assemelha à rotina 10. Ao invés de inverter as cores de tinta e as cores de fundo, ela inverte apenas a cor dos caracteres que têm uma cor de tinta específica.

Endereço inicial: 56820

Endereço final: 56877

Código de verificação: 6935

Descrição:

O endereço 62023 contém um número que representa a cor de tinta que deve ser invertida com a cor de fundo.

Utilização:

Somente no modo 1.

21	00	20	ED	4B	47	F2	3E
20	F5	CD	4A	00	E6	FD	B9
C2	26	DE	CD	4A	00	C6	0E
5F	CD	4A	00	CB	0F	CB	0F
CB	0F	CB	0F	E6	0F	CB	03
CB	03	CB	03	CB	03	B3	CD
4D	00	23	F1	3D	C2	FD	DD
C9	0C	M					

O programa BASIC apresentado abaixo permite o emprego desta rotina. O vídeo se enche de caracteres de diferentes cores. Após alguns segundos, os caracteres mostrados em branco têm sua cor de tinta e de fundo invertida.

```
10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
```

```
40 VPOKE (A), X
50 LET X = X+1
60 IF X = 256 THEN LET X=0
70 NEXT A
80 LET X = 1
90 FOR A = 0 TO 15
100 VPOKE (8192+A),X
110 X = - 1
120 NEXT A
130 FOR X = 1 TO 1000: NEXT X
140 POKE 62023,240
150 DEF USR = 56820
160 A = USR (0)
```

Função:

Apaga o conteúdo de uma zona do vídeo.

Endereço inicial: 56900

Endereço final: 56969

Código de verificação: 7829

Descrição:

O endereço 62025 contém um número relativo à coordenada vertical do canto superior esquerdo da zona a ser apagada.

O endereço 62026 contém um número representando a coordenada horizontal do canto superior esquerdo desta mesma zona.

O endereço 62027 contém um número representando a altura da zona.

O endereço 62028 contém um número representando a largura desta zona.

Utilização:

Em modo 0 e em modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	54	DE	01	28	00
3A	4A	F2	3D	CA	5F	DE	09
C3	57	DE	11	00	00	3A	49
F2	3D	5F	19	ED	5B	22	F9
19	3A	4B	F2	F5	3A	4C	F2
E5	F5	3E	20	CD	4D	00	23
F1	3D	C2	75	DE	E1	09	F1
3D	C2	70	DE	C9	0D	M	

O programa BASIC apresentado abaixo permite o emprego desta rotina. Primeiro, são mostrados caracteres no vídeo, e, após um curto espaço de tempo, o conteúdo de uma certa zona do vídeo é apagado. Esta zona é definida entre as linhas 100 e 115.

```
10 SCREEN 1
20 LET X = 0
30 FOR A = 6144 TO 6911
40 VPOKE (A), X
50 LET X = X+1
60 IF X = 256 THEN LET X=0
70 NEXT A
100 POKE 62025,10
105 POKE 62026,15
110 POKE 62027,8
115 POKE 62028,5
120 DEF USR = 56900
130 A = USR (0)
```

ROT14

Função:

Mostrar os caracteres em um tamanho oito vezes superior ao normal.

Endereço inicial: 57000

Endereço final: 57147

Código de verificação: 15560

Descrição:

O endereço 62029 contém um número correspondente ao número da coluna na qual o caractere é mostrado.

O endereço 62030 contém o número da fila onde está o caractere.

O endereço 62031 contém o código ASCII do caractere a ser mostrado.

Utilização:

No modo 0 e no modo 1.

21	F0	07	01	08	00	3E	FF
CD	56	00	01	08	00	3A	4F
F2	3C	21	00	00	3D	CA	C5
DE	09	C3	BD	DE	11	5B	F2
CD	59	00	3A	4E	F2	01	20
00	21	00	00	3D	CA	DC	DE
09	C3	D4	DE	11	00	00	3A
4D	F2	5F	19	ED	5B	22	F9
19	11	5B	F2	3E	08	F5	1A
47	3E	FE	CB	78	C4	4D	00
23	3E	FE	CB	70	C4	4D	00
23	3E	FE	CB	68	C4	4D	00
23	3E	FE	CB	60	C4	4D	00
23	3E	FE	CB	58	C4	4D	00
23	3E	FE	CB	50	C4	4D	00
23	3E	FE	CB	48	C4	4D	00

23	3E	FE	CB	40	C4	4D	00
13	01	19	00	09	F1	3D	C2
EE	DE	C9	0E	M			

O programa BASIC apresentado abaixo permite o emprego desta rotina.

```
5 SCREEN 1
10 DEF USR = 57000
20 POKE 62030,5
30 POKE 62029,3
40 POKE 62031,77
50 A = USR (0)
60 POKE 62029,13
70 POKE 62031,83
80 A = USR (8)
90 POKE 62029,23
100 POKE 62031,88
110 A = USR (0)
```


ROT15

Função:

Efetuar a rotação de 90 graus de um caractere específico.

Endereço inicial: 57150

Endereço final: 57222

Código de verificação: 8135

Descrição:

O endereço 62032 contém o código ASCII do caractere a ser modificado.

Utilização:

Somente no modo 1.

01	08	00	21	00	00	3A	50
F2	3C	3D	CA	50	DF	09	C3
48	DF	E5	3E	08	F5	CD	67
DF	F1	23	3D	C2	53	DF	E1
13	EB	01	08	00	CD	5C	00
C9	11	62	F2	CD	4A	00	47
3E	08	F5	CD	7E	DF	F1	3D
C2	70	DF	C9	CD	4A	00	47
1A	CB	10	CB	17	12	1B	C9
0F	M						

O programa BASIC apresentado abaixo permite o emprego desta rotina. O programa mostra primeiro a palavra "COMPUTER", e então efetua uma rotação nas letras C, P, U e R.

```
10 SCREEN 1
20 DEF USR = 57150
30 LOCATE 10,10
40 PRINT "COMPUTER"
```

50 POKE 62032,67
60 A = USR (0)
70 POKE 62032,80
90 A = USR (0)
100 POKE 62032,85
110 A = USR (0)
120 POKE 62032,82
130 A = USR (0)

ROT16

Função:

Esta rotina guarda na memória o conteúdo do vídeo. Após o emprego desta rotina, é possível guardar esta imagem do vídeo em fita a fim de recarregá-la mais tarde com a ajuda da rotina 17. No momento em que esta rotina é posta a funcionar, a imagem do vídeo é automaticamente copiada na memória. Ela pode ser guardada com o comando:

57250, 57273.
BSAVE "CAS:VIDEO",61000,61999

Esta rotina é muito útil quando se deseja utilizar frequentemente uma certa imagem do vídeo.

Endereço inicial:	57250
Endereço final:	57273
Código de verificação:	2509

Utilização:

No modo 0 e no modo 1.

01	00	03	3A	AF	FC	3D	CA
AF	DF	01	C0	03	2A	22	F9
11	48	EE	CD	59	00	C9	10
M							

20-22

ROT17

Função:

Esta rotina permite mostrar no vídeo uma imagem de vídeo e guardá-la com a rotina 16. Para carregar uma imagem de vídeo na memória digita-se:

BLOAD "CAS:VIDEO"

Endereço inicial:	57300
Endereço final:	57324
Código de verificação:	2865

O emprego desta rotina permite a afixação instantânea de uma imagem de vídeo. As rotinas 16 e 17 são interessantes porque podem ser utilizadas juntas.

Utilização:

No modo 0 e no modo 1. Atenção para não tentar afixar uma imagem guardada no modo 0, no modo 1 e vice-versa.

01	00	03	3A	AF	FC	3D	CA
E1	DF	01	C0	03	ED	5B	22
F9	21	48	EE	CD	5C	00	C9
11	M						

22-24

Função:

Em sua concepção, esta rotina assemelha-se à rotina 16. Entretanto, ao invés de copiar a totalidade do vídeo, ela somente copia uma zona específica.

Endereço inicial:	57350
Endereço final:	57424
Código de verificação:	8130

Descrição:

O endereço 62033 contém a coordenada vertical correspondente ao canto superior esquerdo da zona a ser copiada na memória.

O endereço 62034 contém a coordenada horizontal correspondente ao canto superior esquerdo desta mesma zona.

O endereço 62035 contém um número relativo à altura da zona.

O endereço 62036 contém um número relativo à largura da zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	16	E0	01	28	00
3A	52	F2	3D	CA	21	E0	09
C3	19	E0	11	00	00	3A	51
F2	3D	5F	19	ED	5B	22	F9
19	11	48	EE	3A	53	F2	F5
C5	D5	E5	01	00	00	3A	54
F2	4F	CD	59	00	E1	D1	C1
09	EB	09	EB	F1	3D	C2	35
E0	C9	12	M				

ROT19

Função:

Esta rotina é empregada juntamente com a rotina 18. Ela permite mostrar uma imagem de vídeo guardada com a rotina 18.

Endereço inicial: 57450

Endereço final: 57525

Código de verificação: 8786

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	7A	E0	01	28	00
3A	52	F2	3D	CA	85	E0	09
C3	7D	E0	11	00	00	3A	51
F2	3D	5F	19	ED	5B	22	F9
19	EB	21	48	EE	3A	53	F2
F5	C5	D5	E5	01	00	00	3A
54	F2	4F	CD	5C	00	E1	D1
C1	09	EB	09	EB	F1	3D	C2
9A	E0	C9	13	M			

O programa BASIC apresentado abaixo permite o emprego dessas duas rotinas anteriores. Este programa contém comentários que indicam ao usuário a função das rotinas.

```
5 SCREEN 1
10 REM AFIXAÇÃO DA MENSAGEM
20 LOCATE 10,9
30 PRINT "*****"
40 LOCATE 10,10
50 PRINT "*ESTA ZONA*"
60 LOCATE 10,11
70 PRINT "**SERÁ**"
80 LOCATE 10,12
```

```
90 PRINT "**GUARDADA**"
100 LOCATE 10,13
110 PRINT "**NA MEMÓRIA**"
120 LOCATE 10,14
130 PRINT "*****"
140 REM ESPECIFICAÇÃO DA ZONA
150 POKE 62033,13
160 POKE 62034,10
170 POKE 62035,6
180 POKE 62036,11
190 REM ESPERA DA LEITURA DA MENSAGEM
200 FOR X=1 TO 1000:NEXT X
210 REM BIFURCAÇÃO PARA A ROTINA
    GUARDADA
220 DEF USR=57350
230 A=USR(0)
240 REM ESCRITA DA MENSAGEM
250 LET X=0
260 FOR A=6144 TO 6911
270 VPOKE A,X
280 LET X=X+1
290 IF X=256 THEN LET X=0
300 NEXT X
310 REM CICLO DE ESPERA
320 FOR X=1 TO 1000:NEXT A
330 CLS
340 REM BIFURCAÇÃO PARA A ROTINA DE
    AFIXAÇÃO
350 DEF USR=57450
360 A=USR(0)
```

Função:

Esta rotina mostra uma tabela dos melhores escores. Este tipo de tabela é encontrado com freqüência nos jogos comercializados. A rotina mostra um escore e decide se ele é suficientemente bom para figurar na lista. Se for o caso, o jogador inscreve seu nome e ele aparece na lista.

Endereço inicial:	57550
Endereço final:	58412
Código de verificação:	95996

Descrição:

Durante o jogo, a variável SC (escore) é utilizada para guardar o escore na memória. No início da partida ela deve ser colocada no 0. No decorrer do jogo, é aumentada ou diminuída conforme seja o caso. No fim do jogo deve haver uma bifurcação que permita o acesso ao programa BASIC apresentado abaixo:

```

10 X = INT (SC/10000)
20 POKE 57640,X
30 X = SC - (X*10000)
40 SC = INT (X/1000)
50 POKE 57641,SC
60 SC = X - (SC*1000)
70 X = INT (SC/100)
80 POKE 57642,X
90 X = SC - (X*100)
100 SC = INT (X/10)
110 POKE 57643,SC
120 SC = X - (SC*10)
130 POKE 57644,SC
140 CLS
150 DEF USR = 57550
160 A = USR (0)
170 RETURN
    
```


Utilização:

Somente no modo 1.

C3	58	E1	53	4F	4E	59	20
20	20	20	20	20	2E	2E	54
4F	53	48	49	42	41	20	20
20	2E	2E	53	41	4E	59	4F
20	20	20	20	20	2E	2E	59
41	4D	41	48	41	20	20	20
20	2E	2E	47	4F	4C	44	53
54	41	52	20	20	2E	2E	30
30	30	30	39	30	30	30	30
38	30	30	30	30	37	30	30
30	30	36	30	30	30	30	35
00	00	00	00	00	00	09	00
48	49	47	48	20	53	43	4F
52	45	53	2E	59	4F	55	20
53	43	4F	52	45	44	20	31
32	33	34	35	2E	00	00	00
00	00	00	00	00	00	00	00
00	00	21	28	E1	7E	C6	30
77	23	7E	C6	30	77	23	7E
C6	30	77	23	7E	C6	30	77
23	7E	C6	30	77	21	28	E1
11	45	E1	01	05	00	ED	B0
01	10	00	11	A8	18	21	3A
E1	CD	5C	00	3E	14	11	29
E1	CD	5D	E3	22	26	E1	3A
28	E1	21	0D	E1	46	90	CA
A6	E1	FA	B9	E1	C3	56	E2
11	0E	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	56	E2
CA	63	E2	3A	28	E1	21	12
E1	46	90	CA	CA	E1	FA	DD
E1	C3	63	E2	11	13	E1	CD
5D	E3	ED	4B	26	E1	A7	ED
42	FA	63	E2	CA	70	E2	3A
28	E1	21	17	E1	46	90	CA
EE	E1	FA	01	E2	C3	70	E2
11	18	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	70	E2
CA	7D	E2	3A	28	E1	21	1C
E1	46	90	CA	12	E2	FA	25
E2	C3	7D	E2	11	1D	E1	CD
5D	E3	ED	4B	26	E1	A7	ED
42	FA	7D	E2	CA	8A	E2	3A

28	E1	21	21	E1	46	90	CA
36	E2	FA	46	E2	C3	8A	E2
11	22	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	8A	E2
3E	14	F5	3E	C8	CD	EB	E3
F1	3D	C2	48	E2	C3	8A	E3
01	14	00	CD	94	E2	11	0D
E1	E5	C3	C1	E2	01	0F	00
CD	94	E2	11	12	E1	E5	C3
C1	E2	01	0A	00	CD	94	E2
11	17	E1	E5	C3	C1	E2	01
05	00	CD	94	E2	11	1C	E1
E5	C3	C1	E2	11	21	E1	21
00	E1	E5	C3	C1	E2	C5	21
20	E1	11	25	E1	ED	B8	E1
29	7D	01	02	00	09	FE	0A
CA	B6	E2	09	FE	14	CA	B6
E2	09	FE	1E	CA	B6	E2	09
E5	C1	21	00	E1	11	0C	E1
ED	B8	C9	21	28	E1	01	05
00	ED	B0	01	2E	00	21	4B
E1	3E	0C	71	23	3D	C2	D1
E2	11	43	19	21	F8	E3	01
19	00	CD	5C	00	11	63	19
21	11	E4	01	1B	00	CD	5C
00	3E	0A	32	2D	E1	2E	0F
26	07	CD	C6	00	21	4B	E1
CD	9F	00	FE	0D	CA	4F	E3
F5	3A	AB	FC	3C	CA	1E	E3
F1	FE	2E	CA	1F	E3	FE	20
CA	1F	E3	D6	20	C3	1F	E3
F1	FE	41	FA	2C	E3	FE	5B
FA	40	E3	C3	31	E3	FE	2E
CA	3B	E3	3E	1C	CD	A2	00
3E	00	C3	40	E3	3E	5C	C3
40	E3	77	23	CD	A2	00	3A
2D	E1	3D	32	2D	E1	C2	FE
E2	E1	23	EB	01	0C	00	21
4B	E1	ED	B0	C3	8A	E3	21
00	00	1A	01	E8	03	CD	80
E3	13	1A	01	64	00	CD	80
E3	13	1A	01	0A	00	CD	80
E3	13	1A	01	01	00	CD	80
E3	C9	D6	2F	3D	C8	09	3D
C2	84	E3	C9	CD	A7	E3	21
2E	E1	01	0C	00	11	CA	1A
CD	5C	00	3E	32	F5	3E	C8
CD	EB	E3	F1	3D	C2	9B	E3

C9	3E	20	21	00	18	01	00
03	CD	56	00	01	0C	00	11
28	19	21	D1	E0	3E	05	F5
CD	5C	00	01	40	00	09	EB
01	0C	00	F1	3D	C2	BD	E3
01	05	00	11	34	19	21	0D
E1	3E	05	F5	CD	5C	00	01
40	00	09	EB	01	05	00	F1
3D	C2	D9	E3	C9	F5	3E	FF
3D	C2	EE	E3	F1	3D	C2	EB
E3	C9	45	4E	54	45	52	20
59	4F	55	52	20	4E	41	4D
45	20	55	53	49	4E	47	20
54	48	45	4B	45	59	42	4F
41	52	44	2E	20	54	48	45
4E	20	50	52	45	53	53	20
45	4E	54	45	52	2E	14	M

Função:

Já vimos duas rotinas que permitem efetuar rotações, seja de um caractere específico ou de todos os caracteres. A principal limitação dessas duas rotinas é que elas não autorizam mostrar o caractere em seu estado original e, simultaneamente, a sua nova versão. Para certas aplicações pode ser útil mostrar, por exemplo, uma linha de caracteres normais na parte superior do vídeo e uma linha de caracteres que passavam por uma rotação na parte inferior do vídeo. Esta rotina permite isto. Observando a tabela dos códigos ASCII que se encontra no Apêndice deste livro, percebe-se que a parte principal do jogo de caracteres se situa entre os códigos 48 (número 0) e 90 (letra Z maiúscula). Esta rotina cria três novos jogos de caracteres; um jogo comportando caracteres que tenham passado por uma rotação de 90 graus, outro, uma rotação de 180 graus, e um outro, uma rotação de 270 graus. O inconveniente desta rotina reside no fato de que ela não permite o emprego de letras minúsculas.

Endereço inicial:	59200
Endereço final:	59353
Código de verificação:	16826

Utilização:

Somente no modo 1.

21	80	01	11	48	EE	01	58
01	CD	59	00	21	48	EE	11
D8	02	01	58	01	CD	5C	00
21	48	EE	11	30	04	01	58
01	CD	5C	00	21	48	EE	11
88	05	01	58	01	CD	5C	00

21	D8	02	CD	95	E7	21	30
04	CD	95	E7	21	30	04	CD
95	E7	21	88	05	CD	95	E7
21	88	05	CD	95	E7	21	88
05	CD	95	E7	C9	3E	2B	F5
E5	3E	08	F5	CD	BA	E7	F1
23	3D	C2	9B	E7	E1	E5	13
EB	01	08	00	CD	5C	00	E1
01	08	00	09	F1	3D	C2	97
E7	C9	11	62	F2	CD	4A	00
47	3E	08	F5	CD	D1	E7	F1
3D	C2	C3	E7	C9	CD	4A	00
47	1A	CB	10	CB	17	12	1B
C9	15	M					

ROT22

Função:

Esta rotina permite redefinir uma parte do jogo de caracteres do seu computador. Ela redefine as formas dos caracteres cujo código ASCII esteja compreendido entre 48 e 90.

Endereço inicial:	59400
Endereço final:	59757
Código de verificação:	46374

Utilização:

Somente no modo 0. Não esqueça de travar o teclado em maiúsculas (CAPS LOCK) para utilizar seu novo jogo de caracteres.

21	15	E8	01	58	01	11	80
01	CD	5C	00	C9	38	4C	CE
D6	E6	64	38	00	78	18	18
18	18	18	7E	00	7C	C6	66
1C	30	62	FE	00	7E	8C	18
3C	06	88	7C	00	1C	3C	6C
CC	FE	0C	0C	00	7E	C0	C0
FC	06	C6	7C	00	1C	30	60
FC	C6	C6	7C	00	FE	86	0C
18	30	60	C0	00	7C	C6	C6
7C	C6	C6	7C	00	7C	C6	C6
7E	06	8C	78	00	38	44	92
A2	A2	92	44	38	00	DF	6F
37	39	0F	03	01	70	60	18
38	70	60	18	38	FF	FF	FF
FF	FF	F9	E7	83	FF	FF	FD
F3	E7	CF	8F	04	FF	FF	FF
FF	FF	FE	ED	FF	FF	FF	FF
FF	FF	FF	FF	FF	3A	46	C6
C6	C6	CE	76	00	DC	E6	C6
FC	C6	C6	FC	00	3C	42	C0
C0	C0	66	3C	00	0E	76	CE
C6	C6	46	3C	00	3C	66	C6

FC	C0	66	3C	00	5C	66	60
F8	60	60	60	80	3A	66	C6
C6	4E	76	86	7C	C0	FC	E6
C6	C6	CC	DE	00	7C	B8	30
30	30	34	FC	00	7F	06	06
46	C6	6C	38	00	C6	CC	D8
F0	F0	D8	CE	00	F8	30	60
60	60	66	7C	00	AC	D6	D6
D6	D6	D6	D4	00	DC	F6	E6
C6	C6	CC	DE	00	38	4C	C6
C6	C6	64	38	00	DC	EE	C6
C6	C6	EC	D8	C0	38	4C	C6
C6	C6	64	38	1E	DC	E6	C6
C4	F8	CC	C8	00	7C	C2	C0
7C	06	C6	7C	00	FE	18	30
30	30	30	18	00	F6	66	46
C6	C6	EE	76	00	E6	46	C6
C6	66	3C	18	00	CC	46	D6
D6	D6	D6	6C	00	86	CC	78
38	3C	66	C2	00	C6	66	66
66	3C	8C	CC	78	3E	46	0C
18	30	62	FC	00	16	M	

ROT23

Função:

Esta rotina cria um novo jogo de caracteres utilizável no modo 1.

Endereço inicial:	59800
Endereço final:	60157
Código de verificação:	1188

21	A5	E9	01	58	01	11	80
01	CD	5C	00	C9	78	84	84
84	84	84	84	78	20	60	A0
20	20	20	20	70	78	84	04
08	10	20	40	FC	78	84	04
1C	1C	04	84	78	0C	14	24
44	84	FC	04	04	7C	80	80
F8	04	04	04	F8	7C	80	80
F8	84	84	84	78	FC	04	04
08	10	20	40	80	78	84	84
E4	9C	84	84	78	78	84	84
7C	04	04	84	78	00	10	10
00	00	10	10	00	00	10	10
00	00	10	30	00	00	0C	30
C0	C0	30	0C	00	00	00	FC
00	00	FC	00	00	00	C0	30
0C	30	C0	00	00	78	84	04
08	10	10	00	10	7C	82	BA
A2	BA	82	7C	00	78	84	84
84	FC	84	84	84	78	84	84
F8	84	84	84	78	7C	80	80
80	80	80	80	7C	70	88	84
84	84	84	88	70	FC	80	80
F8	80	80	80	FC	7C	80	80
F0	80	80	80	80	7C	80	80
80	9C	84	84	78	84	84	84
FC	FC	84	84	84	FC	30	30
30	30	30	30	FC	FC	10	10
10	10	90	90	60	88	90	A0
C0	C0	A0	90	88	80	80	80
80	80	80	80	F8	84	CC	B4

B4	B4	84	84	84	84	C4	A4
A4	94	94	8C	84	78	84	84
84	84	84	84	78	78	84	84
F8	80	80	80	80	78	84	84
84	A4	94	8C	78	78	84	84
F8	A0	90	88	84	78	84	80
60	1B	04	84	78	FC	30	30
30	30	30	30	30	84	84	84
84	84	84	84	78	84	84	48
48	48	30	30	30	84	84	84
84	84	B4	B4	48	84	48	48
30	30	48	48	84	84	84	48
48	30	30	30	30	FC	04	08
10	20	40	80	FC	17	M	

ROT24

Função:

Esta rotina gera o ruído de uma explosão.

Endereço inicial: 60200

Endereço final: 60301

Código de verificação: 9080

3E	07	1E	F5	CD	93	00	21
71	EB	22	6A	EB	01	D0	07
0B	79	B0	C2	38	EB	3A	6E
EB	D6	01	38	06	32	6E	EB
C3	35	EB	2A	6A	EB	7E	FE
FF	C8	32	6E	EB	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	6A	EB	C3
35	EB	70	EB	70	EB	00	00
FF	09	14	13	05	12	0B	0A
0F	0C	02	02	0D	08	09	0E
0C	06	0C	07	04	09	03	02
05	00	00	00	FF	18	M	

Função:

Esta rotina gera o ruído da queda de um objeto na água.

Endereço inicial: 60350

Endereço final: 60472

Código de verificação: 8588

3E	07	1E	F5	CD	93	00	21
07	EC	22	00	EC	01	D0	07
0B	79	B0	C2	CE	EB	3A	04
EC	D6	01	38	06	32	04	EC
C3	CB	EB	2A	00	EC	7E	FE
FF	C8	32	04	EC	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	00	EC	C3
CB	EB	37	EC	06	EC	00	00
FF	00	05	02	00	05	04	00
05	06	01	05	08	01	04	0A
00	04	0C	00	04	0E	02	03
0F	05	03	0E	06	02	0C	08
02	0A	08	01	08	09	01	06
09	00	04	09	00	02	00	00
00	FF	19	M				

ROT26

Função:

Esta rotina gera o som de um tiro de revólver.

Endereço inicial: 60500

Endereço final: 60601

Código de verificação: 9581

3E	07	1E	F5	CD	93	00	21
9D	EC	22	96	EC	01	D0	07
0B	79	B0	C2	64	EC	3A	9A
EC	D6	01	38	06	32	9A	EC
C3	61	EC	2A	96	EC	7E	FE
FF	C8	32	9A	EC	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	96	EC	C3
61	EC	B8	EC	9C	EC	00	00
FF	01	14	0A	01	12	0B	01
0F	0C	02	0C	0D	02	09	0E
02	06	0C	02	04	09	03	02
05	00	00	00	FF	1A	M	

ROT27

Função:

Esta rotina gera o ruído de um tiro de canhão laser.

Endereço inicial: 60650

Endereço final: 60725

Código de verificação: 7620

3E	07	1E	F5	CD	93	00	3E
08	1E	0F	CD	93	00	3E	07
1E	FE	CD	93	00	3E	00	1E
6E	CD	93	00	3E	01	1E	00
CD	93	00	1E	6E	3E	00	CD
93	00	3E	C8	F5	3E	0A	3D
C2	19	ED	F1	3D	C2	16	ED
7B	C6	06	CA	2D	ED	3D	5F
C3	0F	ED	3E	07	1E	FF	CD
93	00	C9	1B	M			

ROT28

Função:

Esta rotina gera o ruído emitido por uma rã mecânica.

Endereço inicial: 60750
Endereço final: 60872
Código de verificação: 11141

3E	07	1E	F5	CD	93	00	21
9E	ED	22	99	ED	01	D0	07
0B	79	B0	C2	5E	ED	3A	9C
ED	D6	01	38	06	32	9C	ED
C3	5B	ED	2A	99	ED	7E	FE
FF	C8	32	9C	ED	23	3E	02
5E	CD	93	00	23	3E	03	5E
CD	93	00	23	3E	09	5E	CD
93	00	23	22	99	ED	C3	5B
ED	9D	ED	BE	ED	00	00	FF
01	00	04	0F	01	96	03	0F
01	32	03	0F	00	C8	02	0F
00	64	02	0F	00	C8	01	0F
00	32	01	0F	00	00	00	00
FF	00	00	0C	0F	00	00	00
00	FF	1C	M				

Função:

Esta rotina dá a impressão de que o conteúdo do vídeo desaparece em um buraco no centro do mesmo vídeo.

Endereço inicial: 58900

Endereço final: 59128

Código de verificação: 16986

Descrição:

A rotina 16 permite guardar a imagem de vídeo gerada por essa rotina.

Utilizamos a rotina 17 para afixar novamente o vídeo original.

Utilização:

Somente no modo 1.

DD	21	88	E6	FD	21	B8	E6
3E	0C	F5	DD	6E	00	DD	23
DD	66	00	DD	23	3E	20	01
20	00	CD	56	00	DD	6E	00
DD	23	DD	66	00	DD	23	3E
20	01	20	00	CD	56	00	FD
6E	00	FD	23	FD	66	00	FD
23	01	20	00	3E	18	F5	3E
20	CD	4D	00	09	F1	3D	C2
52	E6	FD	6E	00	FD	23	FD
66	00	FD	23	01	20	00	3E
18	F5	3E	20	CD	4D	00	09
F1	3D	C2	8D	E6	01	10	27
0B	79	B0	C2	7C	E6	F1	3D
C2	1E	E6	C9	00	18	E0	1A
20	18	C0	1A	40	18	A0	1A
60	18	80	1A	80	18	80	1A
A0	18	40	1A	C0	18	20	1A
E0	18	00	1A	00	19	E0	19

20	19	C0	19	40	19	A0	19
60	19	80	19	00	18	1F	18
01	18	1E	18	02	18	1D	18
03	18	1C	18	04	18	1B	18
05	18	1A	18	06	18	19	18
07	18	18	18	08	18	17	18
09	18	16	18	0A	18	15	18
0B	18	14	18	0C	18	13	18
0D	18	12	18	0E	18	11	18
0F	18	10	18	1D	M		

ROT30

Função:

Esta rotina gera o efeito contrário da rotina 29. Ela produz o efeito de uma imagem emergindo de um buraco situado no centro do vídeo.

Endereço inicial: 58700

Endereço final: 58869

Código de verificação: 15234

Descrição:

Primeiro, utilizamos a rotina 16 para guardar a imagem de vídeo na memória. Esta rotina provoca, em seguida, o efeito de uma imagem que parece sair de um buraco.

Utilização:

Somente no modo 1.

3E	0C	11	C1	E5	F5	1A	32
F0	E0	13	1A	00	5A	60	93
36	E5	03	1A	40	BF	F0	80
80	00	00	80	80	3A	AF	FC
3D	CA	77	E5	01	28	00	3A
F2	E5	3D	CA	82	E5	09	03
DC	F9	84	80	00	3A	F1	E5
3D	5F	19	ED	5B	22	F9	19
EB	21	48	F4	9D	F3	C5	F5
C0	D4	E5	01	00	00	3A	F4
E5	4F	CD	5C	00	E1	D1	C1
09	EB	09	EB	F1	3D	C2	97
E5	01	10	27	0B	79	B0	C2
B4	E5	D1	F1	3D	C2	51	E1
81	08	0C	02	0A	0B	0B	04
0C	0A	0A	06	0E	09	09	08
10	08	08	0A	12	07	07	0C
14	06	06	0E	16	05	05	10

18	04	04	12	1A	03	03	14
1C	02	02	16	1E	01	01	18
20	01	01	18	20	1E	M	

Função:

Esta rotina inverte o conteúdo de uma zona específica do vídeo. Se a zona contém, por exemplo, a palavra "COMPUTER", o emprego desta rotina produzirá no vídeo a palavra "RETUPMOC".

Endereço inicial: 58550

Endereço final: 58657

Código de verificação: 14068

Descrição:

O endereço 62039 contém a coordenada vertical correspondente ao canto superior esquerdo da zona.

O endereço 62040 contém a coordenada horizontal desta mesma zona.

O endereço 62041 contém um número representando a altura da zona.

O endereço 62042 contém um número representando a largura desta zona.

Utilização:

No modo 0 e no modo 1.

21	00	00	01	20	00	3A	AF
FC	3D	CA	C6	E4	01	28	00
3A	58	F2	3D	CA	D1	E4	09
C3	C9	E4	11	00	00	3A	57
F2	3D	5F	19	ED	5B	22	F9
19	C5	01	00	00	3A	5A	F2
3D	4F	E5	D1	09	C1	3A	59
F2	F5	D5	E5	3A	5A	F2	CB
0F	CB	BF	F5	CD	4A	00	F5
EB	CD	4A	00	EB	CD	4D	00

EB	F1	CD	4D	00	EB	13	2B
F1	3D	C2	F9	E4	E1	D1	09
E5	EB	09	EB	E1	F1	3D	C2
EF	E4	C9	1F	M			

Bem, agora já digitamos e testamos todas as rotinas apresentadas neste livro. No entanto, é pouco prático guardar essas 30 rotinas em fita e tentar recarregá-las na memória sem precisar o endereço inicial e o endereço final de cada uma delas. Abaixo apresentamos um método que nos permitirá economizar muito tempo. Primeiro digitamos:

```
CLEAR 200,56000
```

Agora carregamos todas as rotinas, digitando:

```
BLOAD"CAS:ROT1"  
BLOAD"CAS:ROT2"  
"  
etc.  
"  
"  
BLOAD"CAS:ROT31"
```

A partir disso, é possível guardar todas as rotinas num só bloco, digitando:

```
BSAVE"CAS:TUDO",56000,60872
```

Ao invés de carregar as rotinas uma a uma, é mais simples carregar todas as rotinas na memória, digitando:

```
BLOAD"CAS:TUDO"
```

Já mostramos, neste livro, a maneira de chamar uma rotina com o auxílio das instruções DFF USR e A=USR(0). Se nós utilizarmos um programa longo com chamadas freqüentes de diferentes rotinas, a utilização repetida dessas instruções empregará uma grande parte da memória. Neste caso, é preferível utilizar um outro método de programação.

Este método pressupõe o acréscimo de algumas linhas de BASIC no fim do seu programa. Vamos supor que você utilize regularmente três das rotinas propostas. Basta acrescentar as linhas seguintes:

```
10000 DEF USR = 56000
10005 A = USR (0)
10010 RETURN
10015 DEF USR = 56600
10020 A = USR (0)
10025 RETURN
10030 DEF USR = 57000
10035 A = USR (0)
10040 RETURN
```

Se você deseja acesso a uma dessas três rotinas, será preciso prudência quando da utilização das instruções GOSUB. A instrução GOSUB 10015 permite acesso à segunda rotina.

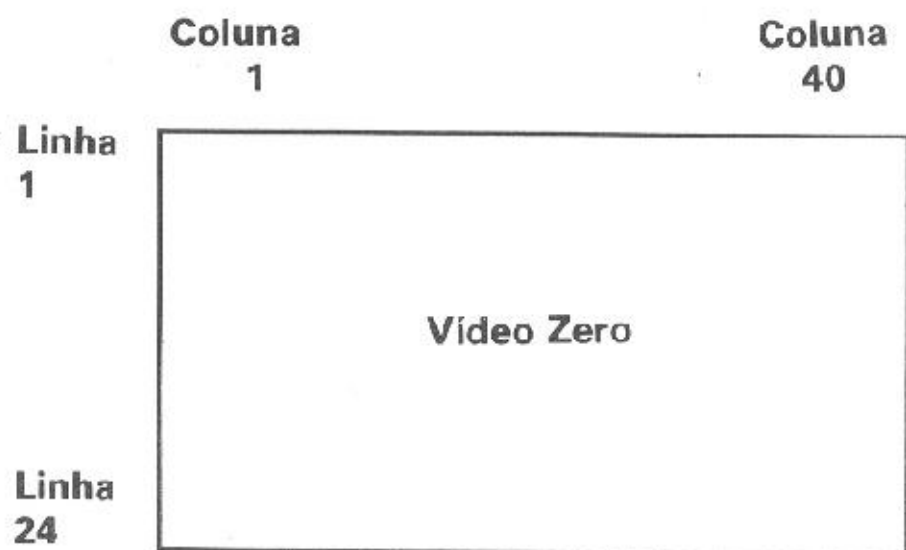
TABLE
N° XVI

APÊNDICE

TABELA DOS CÓDIGOS INK E PAPER

Cores	Código Paper	Código Ink
Preto	1	16
Verde-médio	2	32
Verde-claro	3	48
Azul-escuro	4	64
Azul-claro	5	80
Vermelho-escuro	6	96
Ciano	7	112
Vermelho-médio	8	128
Vermelho-claro	9	144
Amarelo-escuro	10	160
Amarelo-claro	11	176
Verde-escuro	12	192
Magenta	13	208
Cinza	14	224
Branco	15	240

Modo vídeo 0



Modo vídeo 1

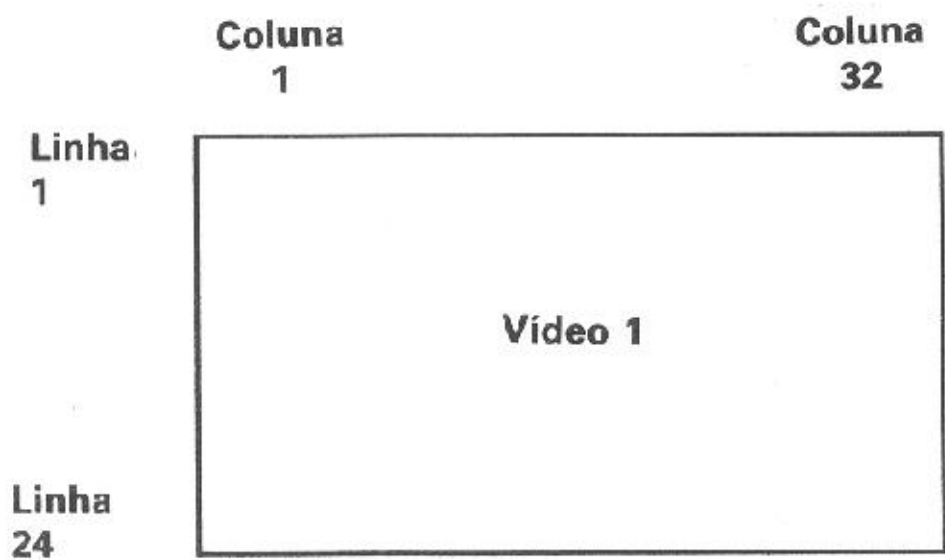


Tabela dos códigos ASCII

Código	Caractere	Código	Caractere	Código	Caractere
32	espaço	63	?	94	↑
33	!	64	@	95	
34	..	65	A	96	a
35	#	66	B	97	b
36	\$	67	C	98	c
37	%	68	D	99	d
38	&	69	E	100	e
39	'	70	F	101	f
40	(71	G	102	g
41)	72	H	103	h
42	.	73	I	104	i
43	-	74	J	105	j
44	,	75	K	106	k
45	_	76	L	107	l
46	~	77	M	108	m
47	/	78	N	109	n
48	0	79	O	110	o
49	1	80	P	111	p
50	2	81	Q	112	q
51	3	82	R	113	r
52	4	83	S	114	s
53	5	84	T	115	t
54	6	85	U	116	u
55	7	86	V	117	v
56	8	87	W	118	w
57	9	88	X	119	x
58	..	89	Y	120	y
59	^	90	Z	121	z
60	~	91	[122	
61	=	92	\		
62	>	93]		

Outros títulos publicados pela Editora Manole :

- **BASIC Básico**
David Munro
- **Voce Conhece os Computadores ?**
Karen Drilling
- **Entendendo os Computadores**
Dyles Walsh
- **MSX Jogos em Assembler**
Eric Ravis
- **MSX Jogos de Acao**
Pierre Nonsaut
- **Tecnicas de Programacao de Jogos em Assembler** (Incluiem 15 superjogos !!)
Georges Fagot-Barreix
- **MSX Programas em Linguagem de Maquina**
Steve Mehh
- **Mais Jogos e Graficos para o MSX**
Graham Carter
- **SPECTRUM Jogos de Acao**
Pierre Nonsaut
- **Jogos de Acao para o APPLE**
Faca seus proprios jogos
Zimmerman & Zimmerman
- **250 Regras de Boa Programacao**
Ledin & Ledin

Caso nao encontrar os livros na sua livraria, procurar na editora que fica na Rua 13 de Maio , 1026 - Bela Vista - SP
(travessa da Av. Brig. Luis Antonio)
Fone: 207-0746

Impresso nas oficinas da
EDITORIA PARMA LTDA.
Fone: 209-5077
Av. Antônio Bardella, 280
Guarulhos - São Paulo - Brasil
Com filmes fornecidos pelo Editor



Este livro contém mais de 30 rotinas gráficas que permitirão a você realizar, rapidamente e sem esforço, programas eficazes e de bom desempenho.

A primeira parte contém as rotinas básicas para o uso da memória de vídeo e dos sprites. A segunda parte apresenta aplicações mais complexas: desdobramento do vídeo nas quatro direções, o desdobramento de uma só linha, rotação dos caracteres e escrita vertical, etc.

Como os programas gráficos são geralmente sonoros, várias rotinas possibilitam a obtenção de efeitos surpreendentes.

