

+2

+3

Manual do Utilizador
por EURICO DA FONSECA

+2 +3



Manual do utilizador
por EURICO DA FONSECA

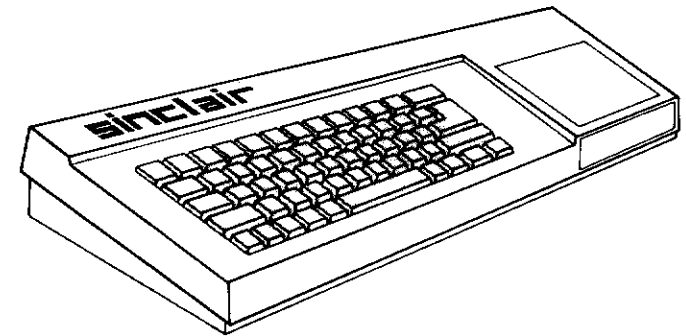
TRIUDUS

TRIUDUS

TRIUDUS

Eurico da Fonseca

Manual do Utilizador



**SPECTRUM
Plus 2 / Plus 3**

Título: "MANUAL DO UTILIZADOR"
Compilado por: EURICO DA FONSECA
© TRIUDUS 1988
Editado por: TRIUDUS
Fotocomposição: Ramos Nogueira
Execução Gráfica:
GRAFILÂNDIA — Artes Gráficas, Lda.
Olival Basto — ODIVELAS
Dep. Legal n.º 24282/88

ÍNDICE

INTRODUÇÃO	
Spectrum + 2 e + 3 — Que diferenças?	3
— Os jogos de computador	4
— As características	5
— Os cuidados	7
Como pôr a funcionar o seu computador	10
— O que pode falhar. O que se deve fazer	13
Como carregar um programa	14
Como gravar programas em cassette	22
O BASIC	
O que é o BASIC?	25
Aprenda a usar o teclado	26
Aprenda a programar em 48 BASIC	32
O BASIC e a Matemática	43
Os gráficos	48
As funções	60
As matrizes	67
Os caracteres	73
Listas de dados — “Read”, “Data” e “Restore”	79
As cadeias de caracteres	82
A organização do écran	86
A lógica	92
Tempo e movimento	95
O som no 48 BASIC	97
Programação estruturada e Subrotinas	102
A GRAVAÇÃO DE PROGRAMAS EM CÓDIGO E O REGISTO DE IMAGENS E DADOS	105
O SPETRUM + 2 E O 128 BASIC	109
O SPECTRUM + 3 E O + 3 BASIC	115
O GERADOR DE SOM	126
AS IMPRESSORAS	134
A MEMÓRIA	139
VIA E CANAIS, ENTRADAS E SAÍDAS	142
MENSAGENS	147
PALAVRAS RESERVADAS	157
COMO USAR A CALCULADORA	158
LIGAÇÃO A PERIFÉRICOS	160

Spectrum + 2 e + 3 Que Diferença?

Os microcomputadores do tipo "Spectrum" são os mais usados como máquinas de jogos, mas podem também ser auxiliares preciosos nos estudos, na profissão, na empresa.

Os primeiros "Spectrum", apesar do sucesso que tiveram, não eram efectivamente concebidos para os jogos ou para as aplicações sérias. Tal como o ZX-80 e o ZX-81, que os haviam antecedido, destinavam-se acima de tudo à aprendizagem da informática. Não dispunham de interfaces para "joysticks" nem de saídas para monitores ou impressoras profissionais, e o teclado de borracha tinha de ser premido com alguma lentidão.

O teclado do Spectrum + era melhor, mas não tanto que mudasse significativamente a situação. O Spectrum 128K dispunha já de saída para monitor RGB ou monocromático, de um gerador de sons programável e de uma interface para impressora profissional, além de uma memória suplementar de 64 Kb. Portanto permitia a concepção de jogos de qualidade superior a utilização de programas de processamento de texto, folhas de cálculo, bases de dados, e linguagens como o Pascal, o C, etc.. Mas o teclado continuava a ser um problema.

Com a absorção da Sinclair, pela Amstrad, tudo mudou: quanto aos jogos, a inclusão de fichas para "joysticks", de um disco de memória e de um gravador-leitor de cassetes no Spectrum + 2 tornou a sua utilização muito mais prática. O novo teclado eliminou por sua vez o maior obstáculo às aplicações profissionais. Mas havia ainda um problema: os programas que tiravam um proveito máximo da capacidade da memória demoravam quase o dobro do tempo a carregar. Daí o lançamento do Spectrum + 3, com um leitor de disquetes de 3 polegadas.

O presente manual descreve todos os aspectos práticos da utilização dos Spectrum + 2 e + 3. **UM MANUAL COM OS DADOS NECESSÁRIOS PARA A PROGRAMAÇÃO AVANÇADA EM CÓDIGO DE MÁQUINA PODERÁ SER OBTIDO A UM PREÇO NOMINAL ATRAVÉS DO DISTRIBUIDOR.**

E agora:

Se o seu interesse são os videojogos, passe à página seguinte, mas não se esqueça de que os Spectrum + 2 e + 3 têm muitas outras aplicações.

Se o seu interesse são as aplicações profissionais ou educacionais, passe à página 5.

Os Spectrum + 2 e + 3 E Os Jogos de Computador

Para usar os Spectrum + 2 ou + 3 não são necessários quaisquer conhecimentos de informática.

O Spectrum + 2 não precisa de ser ligado a um gravador-reprodutor de cassetes. O "Datacorder" nele incorporado assegura o carregamento e gravação de programas e dispõe de regulação automática. O carregamento dos jogos é automático.

O Spectrum + 3 possui um sistema de disquetes de 3 polegadas que permite carregar os jogos numa fracção do tempo exigido pelas cassetes e tem ainda a vantagem de ser praticamente infalível. Uma disquete pode conter 2 a 6 jogos e a passagem de um a outro pode ser feita em segundos.

Os Spectrum + 2 e + 3 dispõem de uma memória útil 2,25 vezes maior que a do Spectrum 48K e do Spectrum +, o que permite o desenvolvimento de "megagames", mas podem usar os milhares de programas de jogos concebidos para os seus antecessores.

Os Spectrum + 2 e + 3 podem ser ligados a um monitor RGB, proporcionando imagens muito mais firmes e nítidas que as obtidas através de qualquer televisor.

Quando os Spectrum + 2 e + 3 são ligados a um televisor, o som é emitido através deste, tal como a imagem.

O som dos Spectrum + 2 e + 3 pode ser programado através de três canais e de um gerador de ruídos — e pode ser ligado a um amplificador.

Os Spectrum + 2 e + 3 possuem duas interfaces para punhos de comando (*joystick*) compatíveis com o protocolo Sinclair.

E agora ...

Se nunca usou um computador, passe à página 7.

Se está familiarizado com os Spectrum e deseja conhecer mais a fundo as diferenças entre os modelos anteriores e os + 2 e + 3 passe à página 5.

Spectrum + 2 e + 3 As Características

O Spectrum + 2 possui um "Datacorder" incorporado, o qual dispensa o uso dos vulgares gravadores-reprodutores de cassetes.

O Spectrum + 3 dispõe de um sistema de disquetes de 3 polegadas que pode ser suplementado por um segundo leitor.

Os Spectrum + 2 e + 3, tal como os seus antecessores, têm por base o microprocessador Z80A trabalhando a 3,5469 MHz. Dispõem de uma memória permanente (ROM) de 32 kB no Spectrum + 2 e 64 Kb no Spectrum + 3, e de uma memória volátil (RAM) de 128 Kb. O Spectrum 48K tinha 16 Kb de ROM e 48K de RAM. Note-se que 16 Kb da RAM eram ocupados pelo BASIC Sinclair (e pela imagem), no Spectrum 48K.

Os Spectrum + 2 e + 3 dispõem de um teclado profissional e de dois BASIC — o 48 BASIC, semelhante ao BASIC Sinclair, e um BASIC adicional que é também semelhante ao Sinclair, mas no qual os comandos são inseridos letra a letra. Denomina-se 128 BASIC no Spectrum + 2, e possui comandos suplementares para a gestão de um disco de memória e de um gerador de som. O do Spectrum + 3 denomina-se + 3 BASIC e dispõe de comandos adicionais para o comando do leitor de disquetes.

Os Spectrum + 2 e + 3 dispõem de um gerador de som de 3 canais, comandado por um processador AY-3-8912. Os seus antecessores de 48 Kb tinham um simples gerador de "beeps".

Os Spectrum + 2 e + 3 têm uma saída para vídeo e som. O cabo de vídeo transporta o som, que é ouvido através do altifalante do televisor. Existe também uma saída para monitor a cores. O cabo do monitor não é fornecido com o computador, mas o do televisor é-o.

As saídas para o monitor têm ligações diferentes no + 2 e no + 3. No Spectrum + 2 o cabo do monitor, ao contrário do que acontece com o do televisor, não conduz o som. No Spectrum + 3 o cabo do monitor conduz o som. Se o monitor não dispuser de som ou tiver uma entrada separada de som, não há problema: os Spectrum + 2 e + 3 dispõem de uma saída de som separada que pode ser ligada por outro cabo ao monitor ou a um amplificador.

A saída para o monitor no Spectrum + 2 pode transmitir o sinal de BRIGHT. Se o monitor o aceitar, apresentará as 16 cores disponíveis (8 de base e 8 mais intensas).

Os Spectrum + 2 e + 3 dispõem de duas fichas tipo D para punhos de comando (*joysticks*) do tipo Sinclair SJS1 — os dos outros tipos não funcionam porque as suas ligações são diferentes. Se um programa oferece várias opções de "joysticks", deve-se escolher a de "Interface One" ou "Sinclair".

Pode-se montar ou retirar os "joysticks" mesmo que o computador esteja ligado.

Os Spectrum + 2 e + 3 dispõem de uma interface RS232 que permite a ligação a uma impressora profissional do tipo série. Para isso é necessário um cabo especial, não fornecido com o computador, e semelhante aos usados nos antigos Sinclair QL.

O Spectrum + 3 dispõe também de uma interface paralela, que permite a ligação às impressoras profissionais compatíveis com a norma Centronics.

No Spectrum + 2 e no + 3 o interface RS232 pode também trabalhar como uma interface MIDI, em ligação a um sintetizador de som ou a um simulador de instrumentos de percussão (*drum machine*), mas apenas transmite sinais — não os recebe. O cabo especial, não fornecido com o computador, é diferente do da impressora.

O Spectrum + 2 possui uma saída denominada KEYPAD, em que devia ser inserido o cabo de um teclado suplementar, inicialmente concebida na versão espanhola do Spectrum 128K, mas que não chegou a ser produzido na Grã-Bretanha. No Spectrum + 3 essa saída é designada por AUX (de *AUXiliary interface*) e pode ter uma diversidade de aplicações, incluindo o comando de robots, mas carece de “software” para tais fins — entende-se que isso respeita ao utilizador.

Tal como os modelos anteriores, os Spectrum + 2 e + 3 possuem na traseira um conector de expansão para montagem de acessórios e periféricos.

O Spectrum + 2 não dispõe de conectores para ligação a um gravador reprodutor externo. O Spectrum + 3 possui um, mas o respectivo cabo não é fornecido com o computador. No entanto é fácil modificar os cabos antigos para os adaptar ao + 3. Veja-se o capítulo PERIFÉRICOS.

Nem todos os periféricos e acessórios dos Spectrum 48K funcionam com os Spectrum + 2 ou + 3.

As ligações dos conectores de expansão dos Spectrum + 2 e + 3 não são completamente iguais, pelo que nem todos os periféricos concebidos para aqueles podem ser usados no + 3.

O Spectrum + 2 pode funcionar com a Interface I e as “microdrives”, o Spectrum + 3, não.

CUIDADO!

Antes de tentar trabalhar com um computador leia a página seguinte.

Spectrum + 2 e + 3 Os Cuidados

LEIA !

- * A fonte de alimentação deve ser ligada à corrente da rede, alterna, de 220V e 50 Hz, e não outra.
- * Quando acabar de trabalhar com o computador, desligue sempre da corrente da rede a fonte de alimentação.
- * Dentro do computador não existe qualquer componente que possa ser ajustado ou substituído pelo utilizador — NÃO TENHA ABRIR A FONTE DE ALIMENTAÇÃO — HÁ ALTAS TENSÕES NO INTERIOR. A abertura do computador ou da fonte de alimentação implica a perda da garantia. As reparações devem ser feitas unicamente por pessoal qualificado.
- * Não tape os rasgos de ventilação.
- * Não use nem guarde o computador em áreas excessivamente quentes, frias, húmidas ou poeirentas.
- * Nunca monte no conector traseiro ou desmonte dele qualquer periférico ou acessório sem desligar o computador, pois pode avariar tanto o computador como o dispositivo.
- * Nunca ligue ou desligue o Spectrum + 3 enquanto tiver uma disquete inserida no leitor. De outro modo pode corromper a disquete e perder programas e dados.
- * Quando desligar o televisor ou monitor, não desligue imediatamente o computador. Aguarde uns segundos.
- * Enquanto houver na memória do computador um programa ou dados que deseje conservar, não desligue o computador nem ligue ou desligue qualquer periférico a ele ligado — pode provocar uma bloqueio (*crash*) e perder o programa ou os dados.
- * No Spectrum + 2 mantenha sempre o Datacorder e as cassetes longe de campos magnéticos, nomeadamente do televisor ou monitor ou de qualquer fonte de interferências eléctricas.
- * No Spectrum + 3 mantenha sempre o leitor ou leitores de disquetes e estas, e o gravador-reprodutor e as cassetes, longe de campos magnéticos, nomeadamente do televisor ou monitor ou de qualquer fonte de interferências eléctricas.
- * No Spectrum + 3, se dispuser de um segundo leitor de disquetes, mantenha o cabo de fita afastado do cabo de alimentação.
- * Se usar o Spectrum + 3 em fins profissionais, faça sempre que possível duplicados das disquetes que contenham programas ou dados importantes.
- * No Spectrum + 3 nunca toque na superfície magnetizada da disquete.
- * No Spectrum + 3 nunca ejecte uma disquete enquanto a luz vermelha indicar que o leitor está a funcionar.
- * No Spectrum + 3 nunca esqueça de que a formatação de uma disquete apaga tudo quanto ela contém.

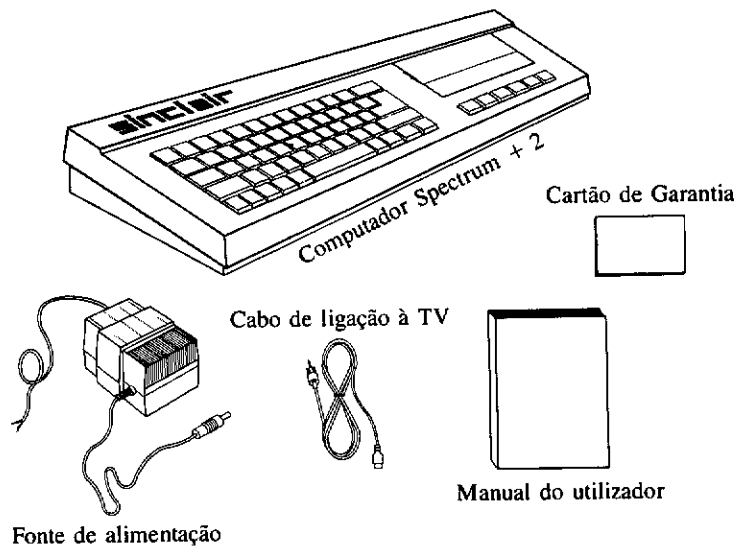


Fig. 1 — O QUE ENCONTRARÁ DENTRO DA CAIXA DO SPECTRUM + 2

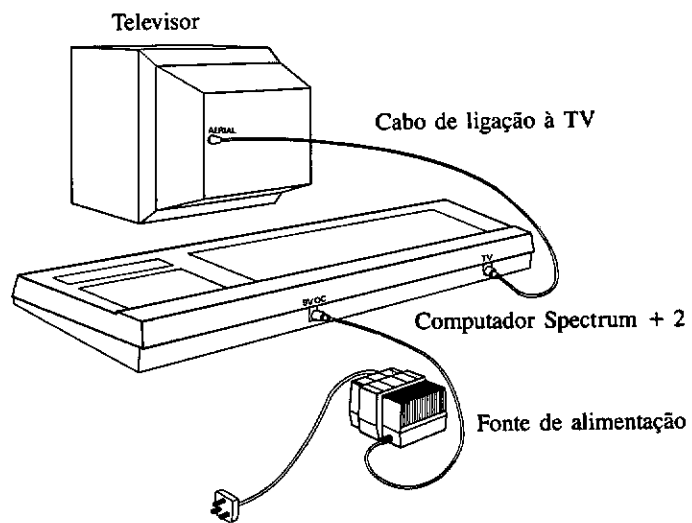


Fig. 2 — COMO INSTALAR O SPECTRUM + 2

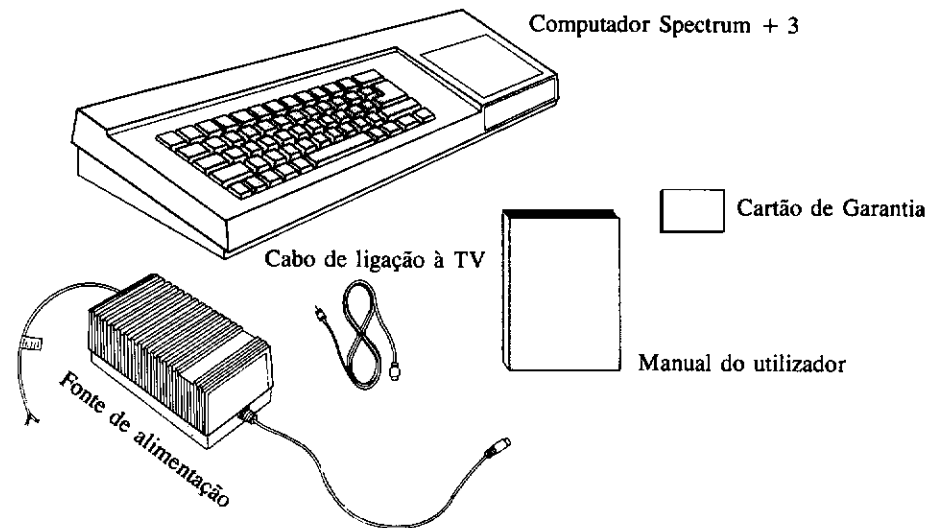


Fig. 3 — O QUE ENCONTRARÁ DENTRO DA CAIXA DO SPECTRUM + 3

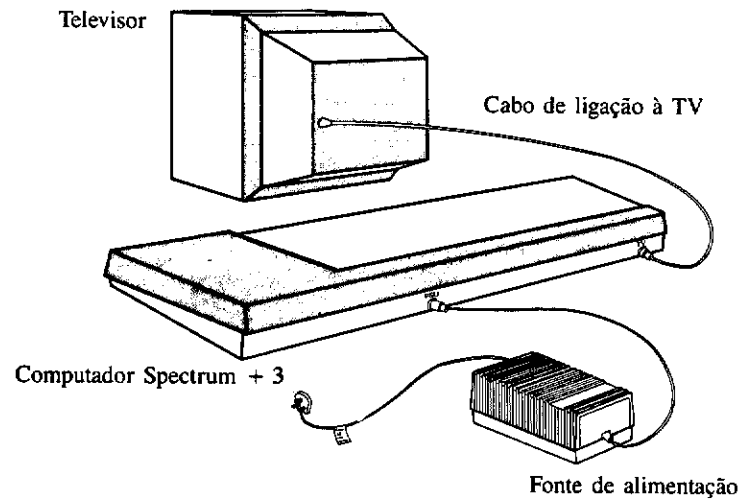


Fig. 4 — COMO INSTALAR O SPECTRUM + 3

Como Pôr a Funcionar O Seu Computador

Quando abrir a caixa do seu Spectrum + 2 ou + 3 deverá encontrar:

- 1 — O computador;
- 2 — A fonte de alimentação;
- 3 — O cabo de ligação ao televisor;
- 4 — Uma cassette ou disquete de demonstração;
- 5 — O cartão de garantia;
- 6 — O manual.

Por enquanto, tudo quanto lhe falta é um televisor, a cores ou a preto e branco, com UHF. Em vez do televisor pode usar um monitor: a imagem será então mais nítida e mais firme. Veja o que diz o capítulo anterior e, antes de adquirir o monitor e o respectivo cabo, certifique-se de que ele funciona com o computador. Note que se o monitor não dispuser de som, terá de usar um amplificador separado.

O Spectrum + 2 pode ser ligado a impressoras profissionais série. O Spectrum + 3 pode ser ligado a impressoras profissionais série ou paralelo. Os respectivos cabos não são fornecidos com o computador.

Convém adquirir uma extensão de tomadas de corrente para 4 fichas — uma para a fonte de alimentação, outra para o televisor ou monitor, outra para o gravador no caso do caso do Spectrum + 3, outra ainda, eventualmente, para a impressora. Não é aconselhável usar as vulgares fichas triplas ou quadruplas porque não garantem ligações firmes.

Entendido?

Vire a parte traseira para si e encontrará nela, sucessivamente, os seguintes conectores:

No Spectrum + 2

9VDC	— entrada da corrente vinda da fonte de alimentação;
EXPANSION I/O RS232/MIDI	— conector para periféricos e acessórios; — conector da interface para comunicações, impressora série e comando de instrumentos musicais;
KEYPAD	— conector para teclado auxiliar (não fornecido);
RGB	— saída de imagem para monitor a cores RGB;
TV	— saída de som e imagem para televisor;
SOUND	— saída de som para monitor ou amplificador.

No Spectrum + 3

DISK B	— conector para o segundo leitor de disquetes;
PRINTER	— conector da interface para impressora paralela;
PSU	— entrada da corrente vinda da fonte de alimentação;
EXPANSION I/O RS232/MIDI	— conector para periféricos e acessórios; — conector da interface para comunicações, impressora série e comando de instrumentos musicais;
AUX	— conector auxiliar para comando (de robots, por exemplo);
RGB/PERITEL	— saída de som e imagem para monitor a cores RGB;
TV	— saída de som e imagem para televisor;
TAPE/SOUND	— conector para gravador-reprodutor de cassetes e saída de som.

No Spectrum + 2 e no + 3

Rode o computador de 90 graus. Encontrará dois conectores para “joysticks” Sinclair SJS1 e um botão de RESET.

Coloque o computador sobre a mesa, na posição normal, a um mínimo de 30 cm do televisor ou do monitor.

Ligue o cabo do televisor à saída assinalada por TV ou o cabo do monitor à designada por RGB no Spectrum + 2 e por RGB/PERITEL no + 3.

Se usar um televisor, ligue a outra ponta do cabo a ele em substituição da antena. Note que não é necessário desligar a antena sempre que se quiser ligar o cabo do computador. Existem selectores que permitem que o computador e a antena estejam permanentemente ligados ao televisor. Em alternativa poderá obter nas casas de material de TV uma ficha em T para antena de televisor. Ligue a base do T ao televisor e depois um lado ao computador e outro à antena. Se usar um monitor, ligue a outra ponta do cabo a ele. Se o monitor tiver uma entrada separada para audio e quiser dispôr de som, precisará de um cabo para ligar a saída de som (SOUND no Spectrum + 2 e TAPE/SOUND no Spectrum + 3) à tomada audio do monitor.

Como se disse no capítulo anterior, o cabo do monitor não é fornecido com o computador e o mesmo acontece no Spectrum + 3 com o do gravador. No capítulo PERIFÉRICOS são fornecidos os elementos para o seu fabrico, no caso de não se encontrarem no mercado.

Ligue ao computador a impressora ou quaisquer outros periféricos ou acessórios que possuir. Como se disse, só os “joysticks” podem ser ligados com o computador sob corrente. Ligue o cabo da fonte de alimentação à entrada de corrente do computador, designada por 9VDC no Spectrum + 2 e por PSU no Spectrum + 3. Depois ligue a ficha da fonte à corrente eléctrica.

O computador tem um indicador visual — uma pequena luz vermelha — que permite saber se está ou não a receber corrente. Se a luz não se acender veja se as fichas estão devidamente ligadas. Se usar um monitor, bastará acendê-lo e regular a imagem através dos respectivos comandos. Se usar um televisor, haverá que sintonizá-lo num canal de UHF que não seja usado para recepção de programas. A maneira de o fazer depende do tipo do televisor, mas todos têm botões de sintonia que permitem escolher o canal a usar e dispositivos de sintonia fina que devem ser regulados para se obter de maneira firme e nítida a imagem gerada pelo computador. Se tiver dúvidas sobre a maneira de fazer essa regulação, consulte o manual do televisor.

Se não conseguir obter imagem alguma, verifique todas as ligações.

Se obtiver uma imagem com linhas oblíquas ou desfocadas, afine a sintonia.

Se mesmo assim não obtiver uma imagem correcta, consulte o distribuidor.

Se a imagem for correcta mas as cores não o forem, regule os comandos de cores do televisor, em conjunto com os de sintonia fina. Se mesmo assim não obtiver satisfação consulte o distribuidor.

Ao contrário do que muitas pessoas pensam, ligar um computador a um televisor em nada prejudica este. O único problema que poderia haver seria a de uma imagem ficar impressa como um "fantasma" no ecrã se se mantivesse fixa durante muitas horas, mas isso nunca acontece: nos computadores as imagens estão sempre em mutação.

Tanto o Spectrum + 2 como o + 3 podem gerar um sinal de teste para afinação de som e da imagem. Prima a tecla de **[BREAK]** (no canto superior direito do teclado). Mantenha-a assim e prima e solte o botão de **RESET**. Continue a premir a tecla de **BREAK** durante uns segundos e depois solte-a. No televisor (ou no monitor) surgirão 16 barras coloridas, com caracteres sobrepostos. (Nos televisores e monitores monocromáticos, as cores são substituídas por tons de cinzento). Simultaneamente deve-se ouvir um som repetido.

NOTE BEM:

Prima as teclas

Não lhes bata como se estivesse a trabalhar com uma máquina de escrever.

O Que Pode Falhar

O Que Se Deve Fazer

A luz vermelha do indicador de corrente não acende e não há imagem no televisor ou monitor

- * Verifique se a ficha que liga a fonte de alimentação ao computador está bem inserida.
- * Verifique se a ficha que liga a fonte de alimentação à tomada de corrente do sector está bem inserida.
- * Verifique se a corrente chega à tomada.
- * Verifique se a ficha que liga a fonte de alimentação ao sector está em ordem.

A luz vermelha acende mas não há imagem.

- * Verifique se o televisor ou o monitor estão devidamente afinados e a trabalhar correctamente.
- * Verifique se o televisor está a trabalhar em UHF.
- * Verifique se o cabo que liga o computador ao televisor está bem inserido.
- * Se usar um monitor, certifique-se de que o cabo é adequado.
- * Se o televisor tiver comando por teclas, verifique se foi premida a correspondente ao canal correcto.

A imagem obtida não é a do sinal de teste

- * Verifique se o sinal de teste foi chamado como atrás se indicou.

As barras coloridas aparecem, mas não há som.

- * Verifique se o volume de som está devidamente regulado.
- * Verifique se a afinação foi feita a fundo.
- * Se usar um monitor com entrada separada de som, verifique se o cabo de som está devidamente inserido.
- * Se o monitor receber o som pelo conector de imagem, só poderá ser usado no Spectrum + 3 e necessitará de um cabo especial.

O som do sinal de teste ouve-se, mas as barras coloridas não se vêem

- * Verifique os comandos de brilho e contraste do televisor ou monitor.
- * Verifique se a afinação foi feita a fundo.

As barras coloridas e o som do sinal de teste surgem mas os caracteres do texto não se vêem.

- * Verifique se a afinação foi feita a fundo.
- * Verifique se a regulação dos comandos de brilho, contraste e cor é correcta.

Se nenhum dos procedimentos indicados resultar, experimente o computador com outro televisor ou monitor. Se o problema se mantiver, consulte o distribuidor.

Como Carregar um Programa

Os procedimentos para o carregamento dos programas são diferentes no Spectrum + 2 e + 3, não apenas pelo facto de o primeiro possuir o Datacorder e o segundo um leitor de disquetes, mas também porque o uso das cassetes no Spectrum + 3 implica o recurso a um gravador-reprodutor exterior.

Cassete

Spectrum + 2

Na sua maior parte, os programas comercializados para os Spectrum são gravados em cassetes, porque se trata do meio de registo menos dispendioso, quanto à aquisição e à utilização, mas as cassetes não são muito fiáveis — longe disso.

Por norma, os gravadores de alta fidelidade não são os que dão melhores resultados: dispõem de filtros que são essenciais para a reprodução da música, mas que deformam os sinais que correspondem aos programas de computador. Um gravador de qualidade mediana pode dar resultados mais aceitáveis, mas nada poderá ser melhor do que um gravador especial como o "Datacorder" integrado no Spectrum + 2. Não é necessário regular o som ou o tom para assegurar gravação e o carregamento. Note-se, no entanto, que muito embora o "Datacorder" tenha uma larga tolerância, não pode fazer milagres. Se um programa tiver sido gravado deficientemente num gravador vulgar, pode não ser aceite pelo "Datacorder".

Note que as teclas do "Datacorder" são semelhantes às dos gravadores vulgares. Da esquerda para a direita, a primeira tecla, com um ponto vermelho, é a de gravação. A segunda, com uma aspa voltada para a direita, é a de "PLAY". A terceira, com duas aspás para a esquerda, é a de rebobinar. A quarta com duas aspás para a direita, é a de avanço rápido. A quinta, com uma aspa apontada para cima e um traço por baixo, é de "STOP" e ejeção. A última, com dois traços verticais, é a de pausa.

Se a imagem e o som estiverem devidamente afinados como se disse no capítulo precedente, retire a cassette de demonstração da respectiva caixa e leia com cuidado as instruções. Prima a quinta tecla do "Datacorder" (a de "STOP" e ejeção) e a tampa abrir-se-á. Coloque a cassette no alojamento de tampa. Feche esta. Se necessário recue a fita até ao começo premindo a terceira tecla, a de rebobinar.

Prima o botão RESET e depois solte-o. No écran deverá surgir um menu com as seguintes opções:

```
TAPE LOADER
128 BASIC
Calculator
48 BASIC
```

As opções são escolhidas deslocando uma barra por meio das teclas cursoras (assinaladas por setas na fila inferior do teclado, à direita).

Se a cassette colocada no Datacorder for a de um programa destinado ao Spectrum 128K ou ao Spectrum + 2, coloca-se a barra sobre a opção "TapeLoader" e prime-se a tecla de "PLAY" do Datacorder. Se tudo estiver em ordem o programa será automaticamente carregado.

Para interromper o carregamento basta premir a tecla BREAK.

Se se pretender carregar um programa no Spectrum 48K, coloca-se a barra sobre a opção 48 BASIC e prime-se ENTER. O menu inicial desaparece e no écran surge a mensagem:

```
@ 1982 Amstrad
```

Premindo a tecla [J], seguida por duas pressões na tecla das aspás [""] (na fila inferior do teclado, à esquerda) a mensagem será substituída pela instrução:

```
LOAD""
```

Prime-se ENTER, coloca-se a cassette no Datacorder e prima-se a tecla de PLAY deste. A ordem de LOAD desaparecerá. O computador ficará à espera do programa. Durante a espera, enquanto a fita corre, a borda (BORDER) do écran mudará de cor, alternando entre o azul pálido (cyan) e o vermelho (red).

Quando o programa é encontrado, começarão a passar na borda faixas horizontais vermelhas. Normalmente, deve surgir no canto superior esquerdo do écran a indicação

```
Program:
```

com o nome do programa a seguir, se ele tiver sido escrito em BASIC, ou

```
Bytes:
```

se tiver sido escrito em linguagem de máquina, e também, nesse caso com o título a seguir. Na prática, quando se usam programas comercializados, essas indicações raramente são vistas. Os produtores cuidam de as ocultar, por uma questão de estética ou para dificultar a execução de cópias. A seguir as faixas horizontais desaparecem e a borda volta a mudar de cor, alternando entre o azul e o vermelho. Depois uma série de linhas horizontais amarelas e azuis surge na borda, indicando que o carregamento está em curso. Se isso não acontecer, ou se o carregamento for interrompido, surgirá então a mensagem:

```
R. Tape loading error
```

Há que fazer voltar a fita ao princípio, verificar as ligações e, se estas estiverem correctas, afastar o gravador do televisor no caso de ele se encontrar demasiado próximo. Em último caso, alterar a regulação do gravador. Carregar de novo na tecla de PLAY.

Se o programa mesmo assim não for carregado, o problema pode ser da gravação ou do Datacorder. É conveniente experimentar a cassette noutro + 2, se possível ou num computador com gravador externo. Se então o programa for aceite pelo computador, o defeito será do Datacorder — é possível que a cabeça reprodutora esteja desalinhada. Se mesmo então o programa não puder ser carregado, o defeito deverá ser da cassette. Note também que **entre os vários tipos do Spectrum existem pequenas diferenças que afectam a aceitação de um pequeno número de programas comercializados.**

Em alguns casos, o computador indica que a memória se esgotou, apresentando a mensagem:

OUT OF MEMORY

Em regra, tratando-se de programas comercializados, isso acontece por a memória do computador não ter sido devidamente “limpa” antes da introdução do programa. Carregue no botão de RESET e volte ao princípio.

Quando o carregamento do programa termina, podem dar-se duas coisas: Ou o programa arranca automaticamente, como é de norma nos programas de jogos e outros, ou surge no fundo do écran a mensagem:

0 OK,0:1

Nesse caso será necessário carregar em RUN (tecla R) e ENTER.

Por vezes, os programas de arranque automático recusam-se a arrancar quando o carregamento chega ao fim. Experimente premir tecla por tecla, nomeadamente a [S] — frequentemente os produtores fazem depender o arranque da pressão sobre uma certa tecla mas não dizem isso nas instruções. Há também programas que exigem códigos para o carregamento ou o arranque e por vezes esse código não corresponde ao referido nas instruções. Noutros casos ainda, por deficiência de gravação ou reprodução, o programa foi corrompido, imobilizando o computador. Faça RESET e proceda a uma nova tentativa. No entanto, quando surge um problema dessa natureza, é raro que o programa possa vir a ser utilizado.

Spectrum + 3

Como se disse, no Spectrum + 3 o Datacorder foi substituído por um leitor de disquetes de 3 polegadas. O carregamento é portanto muito mais rápido e mais seguro. No entanto o Spectrum + 3 pode trabalhar com programas em cassette desde que seja ligado a um gravador-reprodutor.

O cabo é diferente dos usados para os modelos anteriores do Spectrum — a ponta que entra no computador, no conector denominado TAPE/SOUND tem os contactos de EAR e MICRO reunidos numa só ficha igual à dos “jacks” estereofónicos. O cabo não é fornecido com o computador — deve ser adquirido separadamente ou construído a partir de um cabo antigo e de um “jack” estereofónico como se mostra no capítulo PERIFÉRICOS.

Ligue o cabo por um lado ao conector TAPE/SOUND do computador e por outro aos conectores EAR e MICRO do gravador. No Spectrum + 3, ao contrário do que acontecia nos modelos anteriores, a gravação e o carregamento podem ser feitos mesmo que as fichas dos dois cabos estejam inseridas nos conectores.

Depois proceda como se indicou atrás para o Spectrum + 2. Note que a lista de opções é um pouco diferente:

Loader
+ 3 BASIC
Calculator
48 BASIC

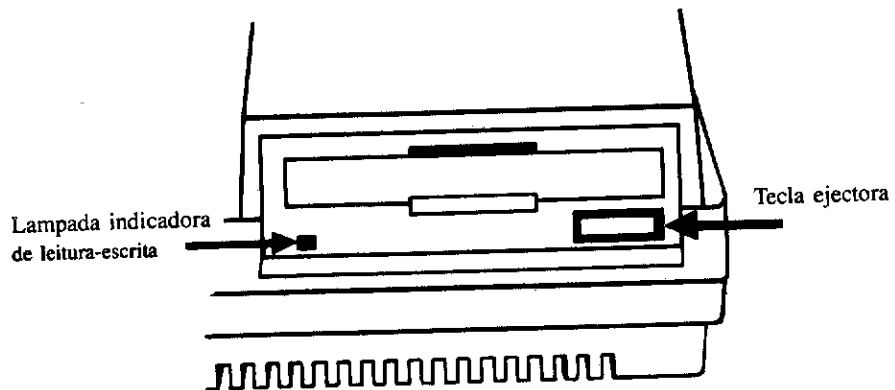
No entanto, na essência, a operação é a mesma. A única diferença é a de que, se uma disquete estiver inserida no respectivo leitor, o computador ignorará o gravador. Se não estiver qualquer disquete no leitor, o Spectrum + 3 comportar-se-á exactamente como o + 2, mas o facto de depender de um gravador-reprodutor externo implica cuidados suplementares.

O primeiro é o de verificar se as pilhas estão em ordem ou se o gravador está devidamente ligado e a funcionar. Note que é preferível ter o gravador ligado à corrente, porque desse modo há a certeza de a tensão de alimentação ser correcta e não se terem variações de volume de som, de tom ou da velocidade da fita. Verifique também se a cabeça de reprodução está devidamente limpa. É bom usar uma fita de limpeza de tempos a tempos e verificar o alinhamento da cabeça de longe em longe.

Regule o volume do som do gravador para cerca de $\frac{3}{4}$ do máximo. Se o gravador tiver comando de tonalidade, regule-o para o máximo de agudos. Para os programas do Spectrum 128K e do Spectrum + 2 use a opção de Loader. Tudo o mais como para o Spectrum + 2.

Disquete

As disquetes usadas no Spectrum + 3 são as de 3 polegadas, do tipo CF2. Note que embora haja outros computadores que usam disquetes semelhantes, só as gravadas no Spectrum + 3 funcionam nele.



Furo de protecção contra escrita

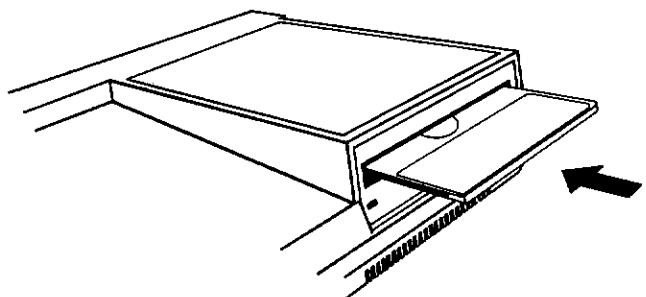
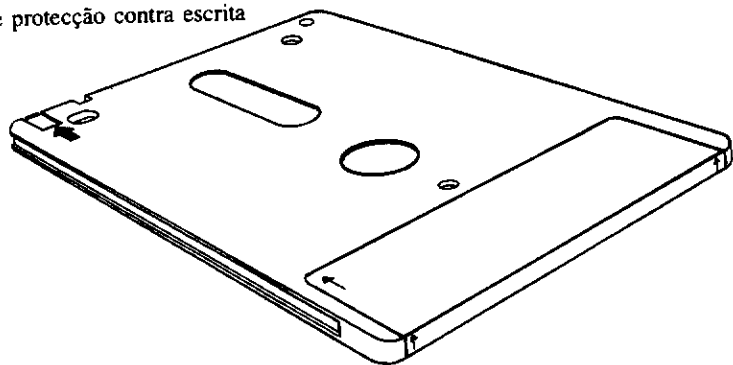


Fig. 5 — COMO INSERIR UMA DISQUETE NO SPECTRUM + 3

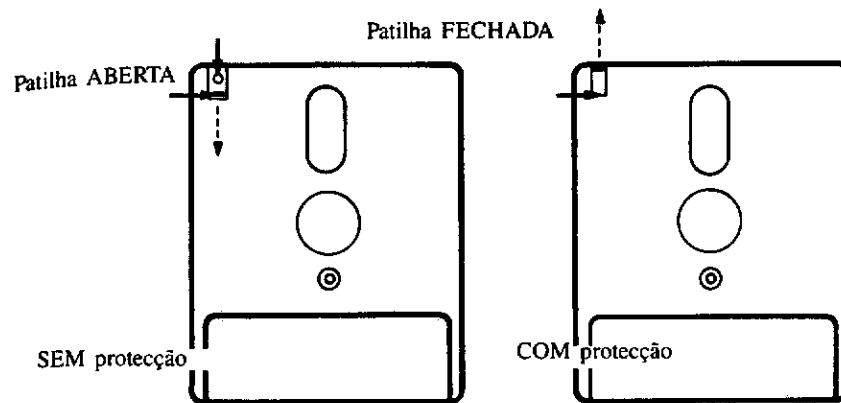


Fig. 6 — PROTECÇÃO POR PATILHA

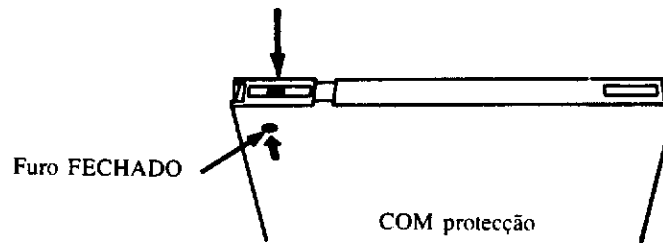
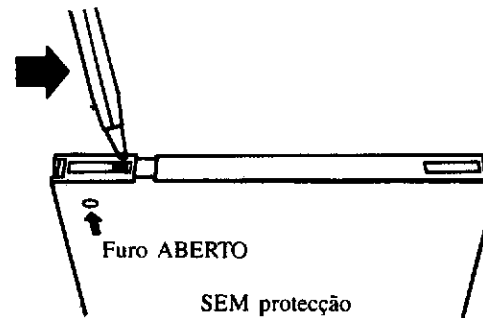


Fig. 7 — PROTECÇÃO POR ALAVANCA

Retire a disquete de demonstração da respectiva caixa. Verifique que ela tem em cada face um rótulo e uma fenda pela qual se vê a superfície magnetizada da disquete. **Não colocar os dedos sobre a superfície magnetizada, não expor a disquete a poeiras, cinzas, líquidos ou campos magnéticos.**

Note que no canto superior esquerdo de cada face da disquete há uma corrediça. Se ela estiver puxada **para cima** a disquete poderá ser lida e gravada. Se estiver puxada **para baixo**, a disquete poderá ser lida, mas não poderá ser gravada. Assegure-se portanto que ela está puxada para baixo, de um lado e outro da disquete, para impedir uma corrupção accidental dos jogos gravados nela. Alguns fabricantes de disquetes substituem a corrediça por uma pequena alavanca situada também no canto esquerdo e que pode ser movida pela ponta de uma esferográfica ou objecto semelhante. (Não é conveniente usar o lápis porque pode partir-se ou largar pó). Deslocando a alavanca **para fora**, o conteúdo da disquete fica protegido.

A disquete deve ser inserida no leitor com cuidado e por completo. A face voltada **para cima** é a que será lida.

Para carregar um programa basta inserir a disquete no leitor, fazer RESET, colocar a barra sobre a opção **LOADER** quando o menu aparecer no écran e premir a tecla **ENTER**.

Dada a capacidade das disquetes, é de esperar que cada uma contenha vários programas. É o que acontece com a disquete de apresentação, que mostra uma lista de opções. Nas disquetes em que essa lista não exista, haverá que correr a barra até à opção **+ 3 BASIC** e premir então **ENTER**. Introduce-se letra a letra a instrução **CAT** e prime-se de novo **ENTER**. No écran surgirá a lista ou catálogo dos programas existentes na disquete. Depois bastará introduzir o comando **LOAD**, letra a letra, seguido o nome do programa, entre aspas.

Formatação

O processo de formatação divide a disquete em 360 áreas distintas.

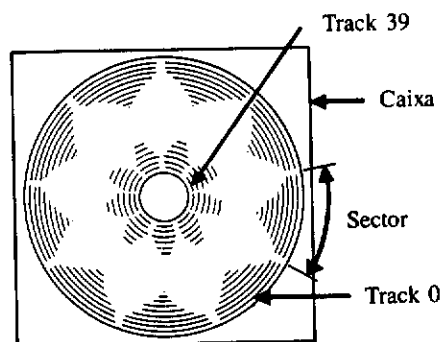


Fig. 8 — DISQUETE

Há 40 pistas (*tracks*) concêntricas da periferia da disquete (*track 0*) ao interior (*track 39*) e cada pista é dividida em 9 sectores.

Cada porção de um sector pode armazenar até 512 bytes de dados. Portanto o total disponível em cada face da disquete é de 180 quilobytes (180 K ou 180 kB). Como 7 Kb dos 180 Kb são usados pelo próprio computador, têm-se 173 Kb por face, para os programas.

Como Gravar Programas Em Cassete

As cassetes a usar na gravação dos programas devem ser virgens e de baixo ruído de fundo. O uso de cassetes já gravadas não é conveniente, mesmo que tenham sido apagadas, porque o ruído de fundo aumenta.

Note que a fita das cassetes têm no início e no fim uma parte não magnética. Coloque a cassette no Datacorder (no Spectrum + 2) ou gravador (no Spectrum + 3) e prima durante alguns segundos a tecla de PLAY para que a fita avance até a parte magnetizada. Para se certificar disso, abra a tampa do Datacorder (ou do gravador) e examine a cassette. Volte a fechá-la.

Spectrum + 2

Para gravar programas em 48 BASIC (o BASIC dos Spectrum 48K) prime-se a tecla [S]. No écran deve surgir a palavra SAVE. Prime-se a tecla das ["] (na fila inferior do teclado, à esquerda) e adiciona-se o nome do programa, o qual não deve ter mais que 8 caracteres. Prime-se de novo a tecla da aspa e a seguir ENTER.

No écran aparecerá a mensagem:

Press REC & PLAY, then any key

Isto é: Prima as teclas de RECORD e PLAY, e depois uma qualquer

Como se disse, as teclas de RECORD e PLAY são a primeira e segunda do gravador. Depois de elas terem sido premidas, prime-se uma tecla qualquer do computador. No écran surgirão faixas vermelhas e azuis a subirem lentamente: é o computador a dar informações ao regulador automático de volume do som. Em seguida surgirão faixas azuis e amarelas. É o nome do programa a ser enviado para a fita magnética. Depois de um breve intervalo, aparecerão de novo as faixas vermelhas e azuis, seguidas pelas azuis e amarelas enquanto o programa é gravado. Quando a gravação terminar, surgirá no écran a mensagem:

0 OK, 0:1

Para se verificar se o programa foi devidamente gravado, rebobina-se a fita premindo a terceira tecla do Datacorder e introduz-se VERIFY premindo as teclas de EXTEND MODE e SYMB SHIFT e depois a tecla [R]. Então prime-se a tecla PLAY do gravador. No écran deverão aparecer o nome do programa e as faixas coloridas como quando se faz LOAD. Se tudo estiver em

ordem, no final surgirá no écran a mensagem 0 OK, 0:1. Se isso não acontecer, deve-se repetir a operação de VERIFY. Se mesmo assim não se chegar a bons resultados, há que repetir a operação de SAVE.

Spectrum + 3

O procedimento no Spectrum + 3 para gravar programas em 48 BASIC em cassette é praticamente igual ao usado nos Spectrum 48K e + 3. A única diferença é a de que o cabo que liga o gravador-reprodutor ao computador pode permanecer inserido nas tomadas de MIC e EAR daquele.

Para gravar um programa, siga os conselhos gerais dados no início da presente secção. Ajuste o volume de som do gravador a dois terços do máximo. Introduza a palavra SAVE premindo a tecla [S], adicione o nome entre aspas (não mais de 8 caracteres) e prima ENTER. No écran aparecerá a mensagem:

Start tape, then press any key

Prima as teclas do gravador que iniciam a gravação (normalmente RECORD e PLAY). Em seguida prima uma tecla qualquer do computador. No écran surgirão faixas vermelhas e azuis a subirem lentamente: é o computador a dar informações ao regulador automático de volume do som que muitos gravadores possuem. Em seguida surgirão faixas azuis e amarelas. É o nome do programa a ser enviado para a fita magnética. Depois de um breve intervalo, aparecerão de novo as faixas vermelhas e azuis, seguidas pelas azuis e amarelas enquanto o programa é gravado. Quando a gravação termina, surge no écran a mensagem:

0 OK, 0:1

Para se verificar se o programa foi devidamente gravado, enrola-se de novo a fita e introduz-se o comando VERIFY. Prima a tecla PLAY do gravador. No écran deverão aparecer o nome do programa e as faixas coloridas como quando se faz LOAD. Se tudo estiver em ordem, no final surgirá no écran a mensagem 0 OK, 0:1. Se isso não acontecer, deve-se repetir a operação de VERIFY, alterando o volume do som e (ou) o de tom, se o houver. Se mesmo assim não se chegar a bons resultados, há que repetir a operação de SAVE. Verifique se as ligações estão correctas.

Para saber como gravar e carregar os programas em código de máquina e as fichas de dados e imagens leia o capítulo A GRAVAÇÃO DE PROGRAMAS EM CÓDIGO E O REGISTO DE IMAGENS E DADOS.

Para gravar e carregar quaisquer programas em cassete ou disquete usando o + 3 BASIC, leia o capítulo O SPECTRUM + 3 E O + 3 BASIC.

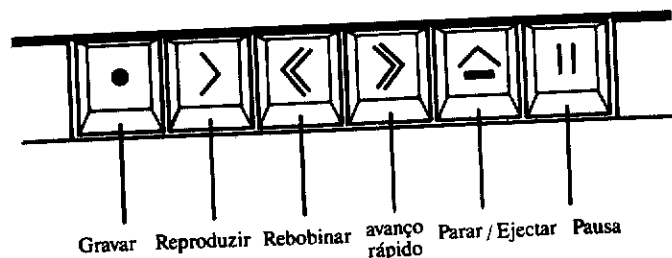


Fig. 9 — SPECTRUM + 2 — Comandos do "Datacorder"

O Que é o Basic?

Para uma pessoa comunicar com a outra tem de usar uma língua que ambas entendam — o Português, o Inglês, etc.. Do mesmo modo, para comunicar com um computador é necessário usar uma linguagem que tanto o utilizador como o computador entendam. A mais corrente é o BASIC, um nome que deriva das iniciais de:

Beginner's All-purpose Symbolic Instruction Code

ou seja:

Código de Instruções Simbólicas de Uso Geral, para Principiantes.

Há muitas outras linguagens que permitem comunicar com os computadores: FORTRAN, PASCAL, FORTH, LOGO, COBOL, C, etc. Todas elas têm a sua utilidade e as suas vantagens particulares, mas o BASIC é a mais usada porque se assemelha muito à linguagem (e à maneira de pensar) humana e tem aplicações gerais. O BASIC foi concebido com base numa série de instruções simples e lógicas na língua inglesa e, de uma maneira geral, assim se mantem. Têm sido feitas tentativas para o adaptar a outras línguas, mas sem grande sucesso, porque então, em regra, a sua simplicidade é prejudicada.

De resto não é necessário saber inglês para trabalhar com BASIC. Basta saber o que significam os seus comandos e instruções. Nem é necessário frequentar um curso para se aprender BASIC. Na realidade não existe um BASIC, mas sim vários. Praticamente, cada marca de computador utiliza uma versão particular do BASIC e por vezes mais do que uma.

Os Spectrum de 16K, 48K, 128K, +, + 2 e + 3, usam ou podem usar o BASIC Sinclair (denominado 48 BASIC nos Spectrum + 2 e + 3), uma versão muito simples e fácil, mas bastante completa, que se caracteriza pela possibilidade da introdução dos comandos e de algumas instruções premindo uma só tecla. Isso é muito útil para os principiantes e para as pessoas não habituadas ao teclado inglês (QWERTY) que equipa de origem a maior parte dos computadores, mas isso tem um preço — exige combinações de teclas que são difíceis de reter na memória. Daí que nos outros computadores (e nos modos 128 BASIC do Spectrum + 2 e + 3 BASIC do Spectrum + 3) seja necessário introduzir os comandos e as instruções letra a letra, por extenso.

No entanto, o que importa é praticar. O BASIC é tão intuitivo que se aprende sem se dar por isso.

Aprenda a usar o Teclado

Os Spectrum + 2 e + 3 possuem o que se chama um "teclado profissional" — isto é: têm teclas adequadas a um trabalho intensivo. Mas não confundam as suas teclas com as de uma máquina de escrever. As do Spectrum + 2 e + 3 são infinitamente mais sensíveis. Basta um leve toque para as fazer funcionar.

Como já foi dito, quando se liga o Spectrum + 2 ou o + 3 surge no ecrã uma lista de opções — um *menu*. Começemos pela de 48 BASIC, deslocando a barra com as teclas cursoras até coincidir com ela e premindo depois ENTER.

No ecrã surgirá a mensagem:

@ 1982 Amstrad

Isso não significa que o computador tenha sido fabricado em 1982, mas sim que essa é a data da primeira patente concedida em relação ao Spectrum.

Prima a tecla de ENTER. Em seguida verá surgir no canto inferior esquerdo do ecrã um quadrado relampejante ao qual se dá o nome de *cursor* e que contém então a letra K. É a inicial de "KEYWORD" — um termo que se aplica às palavras que correspondem às instruções básicas que determinam o funcionamento dos computadores e que em regra é traduzido por "palavra-chave". Mas na língua inglesa, "key" tanto designa "chave" como "tecla". Portanto, poderia também dizer-se "palavra de tecla", no modo 48 BASIC, em que as instruções, em vez de serem introduzidas letra a letra, como é de norma, o podem ser premindo uma determinada tecla. Mas o uso fez lei e a designação de "palavras-chave" é a normalmente usada.

Embora isso não seja essencial é interessante saber que há quatro tipos de palavras-chave:

- * **Comandos** — (*Commands*) — quando a acção por elas expressa é executada imediatamente. Por exemplo: RUN;
- * **Instruções** — (*Statements*) — quando a acção por eles expressa só é executada no momento em que o programa o determina. Por exemplo: DRAW;
- * **Funções** — (*Functions*) — quando produzem um valor de qualquer espécie. Por exemplo: RND;
- * **Operadores lógicos** — (*Logical operators*) — quando são usadas para exprimir a lógica numa instrução ou comando. Por exemplo: AND, OR, NOT.

A natureza da maior parte das palavras-chave é óbvia, mesmo para quem não conhece inglês. E a colocação é fácil de memorizar. Um exemplo é a instrução PRINT. Note que ela está impressa na tecla [P] e que alguma das instruções principais correspondem às letras das

teclas — RUN está na tecla [R], INPUT está na tecla [I], SAVE está em [S], DIM em [D], FOR em [F], GOTO em [G], LET em [L], CONT em [C], BORDER em [B], NEXT em [N].

Há cinco espécies de cursores.

- O "K", de "Keyword", ou "palavra chave", como se disse.
- O "E", de "Extended", indicando que se trata de uma "extensão".
- O "L", de "Letter", ou "letras", indicando as minúsculas.
- O "C", de "Capitals", ou "maiúsculas".
- O "G", de "Graphics", ou "gráficos".

Tome nota disto. Não é essencial, mas é útil, porque a letra do cursor indica-lhe o que vai acontecer quando premir uma tecla.

Note também que além das teclas CAPS SHIFT e SYMB SHIFT há uma com a indicação de EXTEND MODE.

Se no cursor estiver a letra "K", a tecla introduzirá uma "palavra-chave".

Se no cursor estiver a letra "E", isso indica que a tecla EXTEND MODE foi premeida.

Se no cursor estiver a letra "L", a tecla introduzirá letras minúsculas ou algarismos, conforme o caso.

Se no cursor estiver a letra "C", a tecla introduzirá letras maiúsculas ou algarismos.

Se no cursor estiver a letra "G", as teclas dos símbolos gráficos (as dos algarismos) introduzirão esses símbolos.

Muito fácil, portanto. Mas como mudar de cursor?

Tome nota:

- * O modo K surge automaticamente quando o computador espera um comando ou uma linha de programa, e portanto no início de uma linha ou depois de dois pontos [:] (excepto entre aspas) ou de THEN. Premindo nela as teclas indicadas tem-se acesso a:

[Q] — PLOT	[A] — NEW	[Z] — COPY
[W] — DRAW	[S] — SAVE	[X] — CLEAR
[E] — REM	[D] — DIM	[C] — CONTINUE
[R] — RUN	[F] — FOR	[V] — CLS
[T] — RANDOMIZE	[G] — GOTO	[B] — BORDER
[Y] — RETURN	[H] — GOSUB	[N] — NEXT
[U] — IF	[J] — LOAD	[M] — PAUSE
[I] — INPUT	[K] — LIST	
[O] — POKE	[L] — LET	
[P] — PRINT		

* O modo L é o normal. Premindo CAPS LOCK passa-se ao modo C.

Se no modo K ou L se premir simultaneamente SIMB SHIFT e a tecla indicada, tem-se acesso a:

[1] — !	[Q] — <=	[A] — STOP	[Z] — :
[2] — @	[W] — <>	[S] — NOT	[X] — £
[3] — #	[E] — >=	[D] — STEP	[C] — ?
[4] — \$	[R] — <	[F] — TO	[V] — /
[5] — %	[T] — >	[G] — THEN	[B] — =
[6] — &	[Y] — AND	[H] —	[N] — ,
[7] — '	[U] — OR	[J] — —	[M] — .
[8] — ([I] — AT	[K] — +	[.] — .
[9] —)	[O] — ;	[L] — =	
[0] — —	[P] — "		

* Premindo EXTEND MODE e as teclas indicadas, tem-se:

[1] — PAPER AZUL	[Q] — SIN	[A] — READ	[Z] — LN
[2] — "VERMELHO	[W] — COS	[S] — RESTORE	[X] — EXP
[3] — "MAGENTA	[E] — TAN	[D] — DATA	[C] — LPRINT
[4] — "VERDE	[R] — INT	[F] — SGN	[V] — LLIST
[5] — "TURQUESA	[T] — RND	[G] — ABS	[B] — BIN
[6] — "AMARELO	[Y] — STR\$	[H] — SQR	[N] — INKEY\$
[7] — "BRANCO	[U] — CHR\$	[J] — VAL	[M] — PI
[8] — BRIGHT OFF	[I] — CODE	[K] — LEN	
[9] — BRIGHT ON	[O] — PEEK	[L] — USR	
[0] — PAPER NEGRO	[P] — TAB		

* premindo SYMB SHIFT e as teclas indicadas, quando no modo E, tem-se:

[1] — DEF FN	[Q] — ASN	[A] — ~	[Z] — BEEP
[2] — FN	[W] — ACS	[S] — !	[X] — INK
[3] — LINE	[E] — ATN	[D] —	[C] — PAPER
[4] — OPEN #	[R] — VERIFY	[F] — {	[V] — FLASH
[5] — CLOSE #	[T] — MERGE	[G] — }	[B] — BRIGHT
[6] — MOVE	[Y] — [[H] — CIRCLE	[N] — OVER
[7] — ERASE	[U] —]	[J] — VAL\$	[M] — INVERSE
[8] — POINT	[I] — IN	[K] — SCREEN\$	
[9] — CAT	[O] — OUT	[L] — ATTR	
[0] — FORMAT	[P] — @		

* Premindo CAPS SHIFT e as teclas indicadas, quando no modo E, tem-se:

[1] — INK AZUL	[Q] — ASN	[A] — ~	[Z] — BEEP
[2] — "VERMELHO	[W] — ACS	[S] — !	[X] — INK
[3] — "MAGENTA	[E] — ATN	[D] —	[C] — PAPER
[4] — "VERDE	[R] — VERIFY	[F] — {	[V] — FLASH
[5] — "TURQUESA	[T] — MERGE	[G] — }	[B] — BRIGHT
[6] — "AMARELO	[Y] — [[H] — CIRCLE	[N] — OVER
[7] — "BRANCO	[U] —]	[J] — VAL\$	[M] — INVERSE
[8] — FLASH OFF	[I] — IN	[K] — SCREEN\$	
[9] — FLASH ON	[O] — OUT	[L] — ATTR	
[0] — INK NEGRO	[P] — @		

* Para ter acesso ao modo G e logo aos símbolos gráficos, prima GRAPHICS.

* Estando no modo G, para inverter as cores dos símbolos gráficos, prima CAPS SHIFT.

Pratique à vontade. Ensaie para cada tecla todos os procedimentos indicados, apagando depois o resultado com a tecla DELETE. Verá que é muito mais simples do que parece e que bem depressa tudo se automatiza.

Poderá encontrar no início alguma dificuldade em relacionar as palavras-chave com as teclas pois no teclado dos Spectrum + 2 e + 3 só estão indicados os que se relacionam com a execução de programas. Isso é propositado: tem por fim levar os utilizadores a adoptar o 128 BASIC ou o + 3 BASIC, que seguem as mesmas regras mas dispõem de mais comandos. Note que:

* Para passar do menu inicial ao 128 (ou + 3) BASIC mova a barra com as teclas cursoras e quando chegar à coincidência prima ENTER.

* Para passar do menu inicial ao 48 BASIC proceda de modo idêntico.

* Para passar do 128 (ou do + 3) BASIC ao 48 BASIC basta inserir a instrução SPECTRUM.

* Para passar do 48 BASIC ao 128 (ou + 3) BASIC, faça RESET e use o menu.

No que se segue será usado o 48 BASIC — ou seja (não será demais repetir), o BASIC dos Spectrum de 48K. Há uma boa razão para isso: como se disse, ele implica o conhecimento de uma série de combinações de teclas que são desnecessárias, no todo ou na maior parte, nos outros tipos de BASIC. Portanto, quem começar a sua aprendizagem pelo 128 BASIC ou pelo + 3 BASIC, terá dificuldade em usar o 48 BASIC. Mas quem conhecer o 48 BASIC depressa aprenderá a trabalhar com o 128 BASIC ou o + 3 BASIC, uma vez que, como também se disse, estes usam os mesmos comandos e as mesmas instruções.

Aprenda a Programar em 48 Basic

Faça RESET e escolha a opção 48 BASIC ou, se estiver em 128 BASIC ou + 3 BASIC, introduza a instrução SPECTRUM.

Prima a tecla [P]. No écran não aparecerá a letra P, mas sim a instrução correspondente à “palavra de tecla” PRINT, que significa “imprimir”. Na fila de teclas inferior, à esquerda, há uma com aspas [']. Prima-a. A seguir escreva o que quiser, o seu nome por exemplo. Use as teclas de CAPS SHIFT, nos extremos da fila inferior do teclado, para apor as maiúsculas, e use também a barra de espaços como se trabalhasse com uma máquina de escrever.

Se errou, não se preocupe. Carregue na tecla DELETE (à esquerda, a segunda a contar de cima), e as letras irão sendo apagadas. Quando terminar, prima de novo a tecla [']. E então prima a de ENTER — a grande, à direita.

Tudo quanto escreveu entre as aspas surgirá no alto do écran, a partir do canto superior esquerdo.

Carregue de novo na tecla ENTER. Nada acontecerá. Porque o que fez foi um **comando directo** — uma ordem dada directamente ao computador e que desapareceu dele logo que executada.

Procure na quarta fila do teclado a contar de cima a tecla [V]. No 48 BASIC é também o que introduz o comando CLS, que é abreviatura de CLEAR SCREEN (*Limpar écran*). Carregue nela e ao mesmo tempo em ENTER. A frase desaparecerá e o écran ficará de novo limpo.

Prima agora uma ou mais teclas numéricas — [1] e [0], por exemplo — no alto do teclado principal. Como seria de esperar, o número 10 aparecerá no écran. Escreva a seguir o que tinha escrito antes, isto é: prima a tecla [P] (PRINT), a tecla ['], as que formam o seu nome ou o que quiser, e prima de novo a tecla ['].

Suponha que se chama António Silva e que já o tinha escrito entre as aspas. Então, a partir do canto inferior esquerdo do écran ter-se-á formado a *linha*:

```
10 PRINT "ANTÓNIO SILVA"
```

Tal como antes, se se enganar poderá usar a tecla DELETE para corrigir o erro.

Note que até agora procedeu tal e qual como anteriormente. A única diferença foi a aposição de um número à ordem que se pretende ver executada.

Carregue na tecla ENTER. A linha passa da parte inferior do écran para a superior.

Procure na segunda fila do teclado a partir de cima a letra [R], e note que ela também designa RUN, uma palavra que significa literalmente “corra” ou ... “ponha-se a andar!”. Prima essa tecla e verá aparecer na ponta inferior do écran, à esquerda, a mesma palavra RUN. Prima agora ENTER: a palavra RUN desaparecerá e o mesmo acontecerá à linha que estava no alto do écran. No lugar dela, e como anteriormente, aparecerá aquilo que fora escrito entre as aspas. Tal como antes, prima CLS, e depois ENTER, e as palavras desaparecerão.

Dir-se-á que o *comando directo* é mais simples, mas há uma diferença extremamente importante. Carregue de novo nas teclas RUN e ENTER. As **palavras reaparecerão**.

Carregue outra vez em CLS e ENTER. As palavras desaparecerão. Carregue em RUN e ENTER e elas reaparecerão de novo. Faça isso tantas vezes quantas quiser. As palavras voltarão sempre a aparecer.

Note que as instruções dadas ao premir uma tecla só são cumpridas se seguidas por uma pressão na tecla ENTER (*entrar*). Desse modo pode-se verificar se uma instrução foi dada erradamente, antes de a introduzir no computador. A pressão da tecla ENTER é uma confirmação.

Note também — e principalmente — que o computador, sempre que tal lhe é determinado, cumpre as instruções contidas na linha, se elas forem correctas. Diz-se por isso que ele executa um **programa**.

Desligue o computador, carregando no respectivo interruptor. Torne a ligá-lo. Prima RUN e ENTER. As palavras não aparecerão. O programa foi apagado. Isso acontece porque os computadores têm duas espécies de memórias. Uma contém, por assim dizer, as “ordens permanentes” — não pode ser apagada e por isso é designada por “memória permanente” ou “memória só de leitura” (*Read Only Memory* ou ROM). Mas os programas introduzidos pelo utilizador, os dados por eles tratados e os resultados que vão sendo obtidos, são armazenados naquilo que se designa por “memória de acesso aleatório” (*Random Access Memory*, ou RAM), ou ainda por *memória volátil*, pois só funciona enquanto recebe corrente eléctrica. Quando a corrente é cortada, tudo quanto se encontrava armazenado na memória volátil desaparece.

Obviamente, não é necessário desligar o computador para mudar de programa. Para isso prime-se a tecla da letra [A], que corresponde ao comando NEW, o qual “apaga” o programa corrente, deixando o computador livre para receber um novo programa.

Note-se que só os programas em BASIC são apagados pela instrução NEW. Quando se usam programas em linguagem (ou “códigos”) de máquina — isto é: com instruções que podem ser compreendidas directamente pelo computador, e portanto são obedecidas mais rapidamente — o comando NEW não as faz desaparecer. Por isso, um bom computador deve possuir um interruptor ou um dispositivo de RESET para apagar todo o conteúdo da memória volátil. É o que acontece nos Spectrum + 2 e + 3.

Faça portanto NEW e depois ENTER. Escreva:

```
10 PRINT "BONS DIAS!"
```

Não se esqueça de que deve premir CAPS SHIFT simultaneamente com as tecla das letras, ou então carregar ao mesmo tempo em CAPS SHIFT e em CAPS LOCK antes de escrever a frase. Para que ela surja em maiúsculas. Se usar CAPS LOCK veja se o cursor passou a mostrar a letra “C”. Se isso não acontecer repita o procedimento.

Prima ENTER. Como anteriormente, a linha desaparecerá da parte inferior do écran, passando para a superior. Também como anteriormente, faça RUN e ENTER. No canto superior do écran surgirá a frase entre aspas:

BONS DIAS!

Suponha que quer alterar essa frase. Nada mais simples: prima simultaneamente as teclas CAPS SHIFT e EDIT. A linha surgirá em baixo, com o cursor "K" a relampejar logo a seguir ao número 10. Prima a tecla da seta que aponta para a direita. O cursor deslocar-se-á para a direita e mudará para "L", indicando que se pode inserir letras ou algarismos. Leve o cursor até onde deseja iniciar a inserção e comece a fazê-la. Se quiser fazer alguma correcção, leve o cursor para a direita com a tecla da seta que aponta para a direita ou para a esquerda com a que aponta para a esquerda, até imediatamente depois da letra ou letras ou números que quiser apagar e prima a tecla DELETE. Quando terminar as alterações prima ENTER.

Suponha que no fim a linha ficava com o aspecto de:

10 PRINT "BONS DIAS, MARIA!"

A nova linha irá substituir a primeira. faça RUN e ENTER, para confirmar as alterações. Se tiver errado, repita o procedimento.

Carregue de novo em SYMB SHIFT e EDIT. A linha voltará a surgir na parte inferior do écran. Mova o cursor por meio das teclas de setas e com a ajuda de DELETE apague uma das aspas. Prima ENTER. Note que o computador se recusa a aceitar a linha assim alterada **porque a falta da aspa constitui um erro.**

Mova o cursor sobre a linha para longe do ponto onde falta a aspa. Prima de novo ENTER. Verifique que a linha continua sem ser aceite **mas o cursor deslocou-se para o ponto onde se situa o erro.**

Essa é uma das características mais interessantes do 48 BASIC: a detecção automática dos *erros de sintaxe* e a indicação da sua localização.

Corrija o erro que introduziu propositadamente: reponha a aspa. E note um facto: a frase que colocou entre aspas surge sempre a partir do canto superior esquerdo do écran.

Como colocá-la noutra local do écran?

O écran tem dois modos de localização. Um destina-se aos caracteres e o outro aos pontos que formam os desenhos.

No primeiro, o écran divide-se em 32 colunas e 22 linhas. Há programas que permitem apresentar 64 colunas mas isso só é prático quando se usa um monitor. Com um televisor, a apresentação a 32 colunas é a mais legível.

No segundo, o écran divide-se em 256 "elementos de imagem" (*pixels*) ou "pontos" na horizontal e 176 na vertical. Note-se que há uma zona de duas linhas de caracteres ou de 16 filas de "pontos" na parte inferior do écran, normalmente reservados a mensagens sobre o funcionamento do computador, mas acessíveis mediante certos procedimentos.

Prima simultaneamente as teclas (I) e SYMB SHIFT verá surgir no écran a palavra AT, que corresponde a EM, ou NO. Portanto, se a PRINT se juntar a AT, teremos PRINT AT, significando IMPRIMA EM. A seguir a AT, escreva, por exemplo, 11,7. Junte um ponto e vírgula, premindo a tecla respectiva. Assim, no exemplo antes dado, ficará:

10 PRINT AT 11,7; "BONS DIAS, MARIA!"

Isso significa que a frase que se pretende imprimir no écran irá formar-se na linha 11, a partir da coluna 7, ou seja sensivelmente no centro da imagem.

Note a necessidade de colocar o ponto e vírgula (premindo a respectiva tecla) a seguir a AT. O ponto e vírgula funciona como um símbolo de união. Significa simplesmente que a frase deve iniciar-se na localização indicada. (No capítulo A ORGANIZAÇÃO DO ÉCRAN referem-se outros sinais e instruções que determinam a posição dos caracteres no écran).

Suponhamos agora que quéríamos dispôr da facilidade de colocar a frase em qualquer das 32 colunas e das 22 linhas, sem termos de alterar continuamente a linha.

É o momento de saber o que é uma "variável" e como se pode variar o seu valor.

Prima a tecla [L]. No écran surgirá a palavra LET, que no caso significa FAÇA ou SEJA.

Apague-a e introduza uma nova linha: escreva por exemplo:

3 LET a = 11

Prima ENTER, como habitualmente. A nova linha, como o seu número é o 3 e logo está primeiro, irá aparecer no alto do écran, acima da linha que já tínhamos introduzido e que era a 10.

Introduza mais uma linha:

5 LET b = 7

Como seria de esperar, quando carregar em ENTER, a linha 5 ir-se-á colocar entre a 3 e a 10. Note que a seguir ao número 5, na linha que acaba de fazer entrar, se vê o sinal >. Esse sinal indica que a linha em que se encontra pode ser editada — isto é, modificada.

Só para verificar essa possibilidade, prima simultaneamente CAPS SHIFT e EDIT (no extremo esquerdo do teclado). A linha surge de novo na parte inferior do écran, e poderia ser modificada, se necessário. Prima ENTER e a linha voltará ao programa.

Prima agora a tecla cursora que mostra a seta apontada para cima e note que o sinal que estava na linha 5 passa para a 3. Se carregar em CAPS SHIFT e EDIT, a linha 3 virá abaixo e poderá ser editada.

Prima a tecla cursora que mostra a seta voltada para baixo. O sinal que estava na linha 3 baixará para a linha 5 e depois para a linha 10.

A linha 10 é precisamente a que temos de corrigir para lhe introduzirmos as variáveis. Prima de novo CAPS SHIFT e EDIT, para a trazer abaixo, apague nela com DELETE os números 11 e 7 e substitua-os pelas letras correspondentes às variáveis, ou seja a e b.

Note bem: quando se trata de letras que designam variáveis, o computador não distingue entre maiúsculas e minúsculas: tanto faz usar a e b como A e B. Qualquer designação serve para uma variável desde que seja unicamente formada por letras ou por letras e espaços. O computador aceita também designações que contenham algarismos, desde que não sejam iniciadas por um deles. Assim:

```
x
t5
nome correcto
Nome Correcto
```

São designações aceites pelo computador.

```
1990
4 rodas
Sub/ramo
calcule-se
```

São designações não aceites.

Voltemos ao programa. Quando acabar de o introduzir, prima ENTER como sempre. Verá no alto do écran surgir o programa:

```
3 LET a = 11
5 LET b = 7
10 PRINT AT a,b; "BONS DIAS, BELA!"
```

Faça RUN e, como sempre ENTER. A frase entre aspas aparecerá como antes no meio do écran.

Vejamos agora como se podem variar as variáveis ...

A primeira coisa a fazer é carregar na tecla [V] (correspondente a CLS, como se disse) para limpar o écran. Não é indispensável mas torna tudo mais claro. Claro que depois terá de carregar em ENTER. De futuro não faremos essa recomendação. Deve ter ficado bem evidente que para fazer "entrar" instruções no computador se deve premir a tecla ENTER, e que isso só deve ser feito quando um conjunto de instruções está completo.

Depois de limpar o écran prima a tecla [K], que corresponde a LIST. Se não acontecer nada, lembre-se do que lhe dissemos: prima ENTER — não voltaremos a repetir.

No alto do écran voltará a surgir o programa — a listagem do programa. Note que o sinal > desapareceu, o que significa apenas que a primeira linha é a que está pronta a ser editada. Prima portanto CAPS SHIFT e EDIT e a linha 3 surgirá em baixo. Altere o valor de a, substituindo 11 por um número entre 0 e 31.

Quando acabar de fazer isso e a linha 3 tiver voltado ao seu lugar, note que o símbolo > apareceu a seguir ao 3 — está a indicar que essa linha foi a última a ser editada. Prima a tecla da seta com a ponta para baixo. O sinal baixará para a linha 5. Faça-lhe o mesmo que fez à linha 3 e substitua o valor de b por um número entre 0 e 21.

Quando terminar faça RUN. Verá que a frase aparece noutra local do écran — a partir de outra coluna e noutra linha, talvez dividindo-se por duas linhas. Se a linha for o número 21 — a última, porque a primeira é designada por 0 — então surgirá no écran a pergunta:

scroll?

Se quiser que a imagem no écran role para cima, prima qualquer tecla menos a dos espaços. Se quiser que ela não role, prima a dos espaços.

Se repetir as operações antes descritas — e é bom que o faça para praticar — aprenderá a colocar frases (texto) em qualquer ponto do écran. Mas por certo que não é prático alterar constantemente as linhas onde se encontram as variáveis.

Como seria de esperar, há uma maneira mais prática de imputar valores às variáveis. A tecla [I], a mesma de AT, é a mesma que determina a inserção da instrução INPUT. Um primeiro uso para ela pode ser exemplificado substituindo a linha 3:

```
3 LET a = 11
```

por

```
3 INPUT a
```

O mesmo quanto à linha 5. Em vez de:

```
5 LET b = 7
```

faça:

```
5 INPUT b
```

Faça RUN agora e verá que no canto inferior esquerdo surge o cursor com a letra L. Isso significa que o computador pede um dado para continuar a executar o programa. Introduza, no caso, um número entre 0 e 31. No canto inferior esquerdo aparecerá de novo o L a relampejar. Introduza, como antes, um número entre 0 e 21. Faça RUN e o texto será impresso no lugar que escolheu. Se não estiver correcto ou não gostar, volte a fazer RUN e introduza outros números — isto é *impute* outros valores às variáveis *a* e *b*.

Tudo bem.. menos um problema: como o cursor aparece sempre a relampejar a letra L, quer peça um número de coluna, quer de linha, é sempre possível um engano. Mas também para isso há uma solução. Basta alterar as linhas 3 e 5 para:

```
3 INPUT "Coluna ="; a
5 INPUT "Linha ="; b
```

Quando se fizer RUN no canto inferior do écran aparecerá primeiro:

```
Coluna = [L]
```

e depois:

```
Linha = [L]
```

E deixará de haver possibilidade de confusão.

O programa tem ainda uma possibilidade de refinamento. Não é muito prático ter de alterar a linha 10 para substituir a frase entre aspas, mas isso também tem solução.

No 48 BASIC como nos outros, tudo quanto é colocado entre aspas denomina-se "string", um termo que significa "corda", "cordão", e que é aplicado no sentido de considerar uma série de letras, algarismos ou símbolos como uma *cadeia de caracteres*. O computador trata a cadeia como se fosse uma unidade e aceita a sua representação por uma letra seguida de um cifrão. Escreva uma nova linha:

```
7 INPUT "Texto: "; a$
```

e altere a linha 10 para:

```
10 PRINT AT a,b; a$
```

O programa tomará, portanto, o seguinte aspecto:

```
3 INPUT "Coluna ="; a
5 INPUT "Linha ="; b
7 INPUT "Texto: "; a$
10 PRINT AT a,b; a$
```

Poderá assim colocar a partir de qualquer coluna ou linha o texto que quiser — com caracteres alfabéticos ou numéricos (*alfanuméricos*), ou quaisquer símbolos disponíveis no teclado. E muitas outras coisas poderá fazer com o texto. Por exemplo: se uma parte estiver contida na linha e outra ou outras numa cadeia, poderá simular um diálogo entre o computador e o utilizador.

Faça:

```
10 REM Conversar com o computador
20 INPUT "Como te chamas? "; a$
30 PRINT "Muito prazer. "; a$: ". Como vais?"
40 INPUT b$
50 PRINT b$
70 PRINT "Ora! Gostava de falar com outra pessoa."
```

No programa anterior há muita coisa a notar. Para começar, a palavra REM, que se introduz premindo a tecla [E] e significa REMEMBER (relembrar). O que se escrever a seguir a ela é (em princípio) ignorado pelo computador e exclusivamente dirigido ao utilizador. É um lembrete, sobre a finalidade do programa ou algumas das suas partes.

Note depois que entre cada linha há um espaço de dez. Não é indispensável mas é uma boa prática, pois, como se compreende, dá amplas possibilidades de introduzir outras linhas nesse espaço. Alguns tipos de BASIC dispõem da instrução AUTO que permite numerar automaticamente as linhas a espaços de 10; outros permitem determinar um espaço qualquer. Por vezes o BASIC inclui também a instrução RENUMBER que permite renumerar as linhas, com intervalos determinados. No 48 BASIC não existe essa facilidade, mas também é possível obtê-las através de programas auxiliares. No 128 BASIC + 3 BASIC existe um menu auxiliar que inclui a opção RENUMBER.

Note também o uso do ponto e vírgula como elemento de ligação de texto e tenha cuidado com os espaços em branco e sinais de pontuação.

O mais importante é o final: note que todos os passos do programa seguem uma ordem lógica. É o que há de mais fundamental nas regras da programação. Um exemplo clássico é o do "programa da manhã". Primeiro é preciso acordar (a menos que se seja sonâmbulo...). O segundo passo é o de saltar da cama. E os outros, pela ordem natural das coisas, são os de tomar banho,

vestir, tomar o pequeno-almoço. Ninguém toma banho antes de saltar da cama ou se veste antes de tomar banho. Se quiser, troque as linhas do programa que estamos a analisar. E veja o efeito... Há ainda um ponto a considerar: quando o programa chega ao fim, o computador "diz" que "gostava de falar com outra pessoa". Para isso basta fazer RUN. É fácil, mas se isso tiver de ser feito muitas vezes seguidas será melhor procurar uma maneira de voltar automaticamente ao princípio do programa. Isso pode ser feito com o auxílio de uma nova "palavra": GOTO, que significa "vá para" e que pode ser introduzida premindo a tecla [G]. Basta introduzir uma nova linha:

```
80 GOTO 20
```

A nova linha faz o programa voltar ao princípio "activo", à linha 20, dando ao computador a possibilidade de "conhecer" outras pessoas...

Note que mesmo que prima a tecla BREAK ou carregue simultaneamente em CAPS SHIFT e na barra dos espaços, o ciclo não se interrompe. É uma situação de "circuito fechado" ou *ciclo* — aquilo a que os ingleses chamam *loop*. No caso, só a intervenção do utilizador pode pôr fim ao "loop". Atente em que o cursor, quando pede que se introduza um dado, apresenta-se entre aspas. Com a tecla DELETE apague a primeira aspa (ou as duas). Faça STOP premindo CAPS SHIFT e a tecla [A]. Na parte inferior do écran surgirá a nota:

H STOP in INPUT

Se o utilizador mudar de ideias e decidir continuar, bastará que faça CONT (de CONTINUAR), premindo SYMBOL SHIFT e a tecla [C]. Se não, poderá retomar o comando do computador, fazendo NEW e apagando assim o programa, se quiser.

No entanto, há outras espécies de "loop" e outras maneiras de pôr fim a um "loop". No 48 BASIC, em vez de usar o "circuito fechado" pode-se recorrer ao ciclo FOR ... NEXT. Experimente o seguinte exemplo:

```
10 REM Circuito fechado limitado
20 FOR n= 1 TO 5
30 PRINT n
40 NEXT n
```

FOR introduz-se premindo [F] enquanto para NEXT é necessário premir [N]. Quanto a TO, obtém-se premindo [F] e SYMB SHIFT.

Como FOR significa "para" TO corresponde, no caso, a "até", e NEXT a "que se segue". O programa pode ser entendido como dizendo:

Para n = 1 até 5 imprima o n que se segue

No écran surgirão os números 1, 2, 3, 4 e 5 — e não mais. O "loop" imprimirá os sucessivos valores da variável mas somente tantas vezes quantas as determinadas.

No exemplo assumiu-se um acréscimo igual à unidade, mas ele pode ter um valor qualquer, como veremos a seu tempo.

Voltemos ao programa "Conversar com o computador" para lhe introduzirmos um "loop". Então ter-se-á:

```
10 REM Conversar com o computador
20 INPUT "Com quantas pessoas tenho de falar ?"; x
30 FOR n=1 TO x
40 PRINT "Número"; x ; ', como te chamas ?'
50 INPUT a$
60 PRINT "Muito prazer, "; a$ ; '. Como vais ?'
70 INPUT b$
80 PRINT b$
90 PRINT "Ah, sim ? Continuemos."
100 NEXT n
110 PRINT "Pronto ! Acabou-se !"
```

Note em primeiro lugar que o termo do "loop" pode ser uma variável — o número x de pessoas com as quais o computador vai conversar. O mesmo pode acontecer com o início. Note também a importância da linha 110: como o "loop" chegou ao fim, o programa deixou de voltar atrás e seguiu em frente.

Compliquemos um pouco mais as coisas, analisando o uso de palavras-chave como IF, AND e OR.

IF significa "se", AND é "e" e OR é "ou".

Ensaie o seguinte programa:

```
10 REM Condicionaismos
20 PRINT "E de dia ? (S/N)"
30 INPUT a$
40 IF a$ <> "S" AND a$ <> "s" OR a$ <> "N" AND a$ <> "n" THEN GOTO 30
50 IF a$ = "S" OR a$ = "s" THEN GOTO 70
60 IF a$ = "N" OR a$ = "n" THEN GOTO 90
70 PRINT "Entao muito bom dia !"
80 STOP
90 PRINT "Entao muito boa noite !"
```


Para começar, note na linha 20 a falta de acento no "É", e na 70 e 90 em "Então". Como o BASIC tem por base a língua inglesa, o texto escrito nele não tem acentos. Note também na linha 20 como o computador solicita ao utilizador uma resposta — S de "sim" ou N de "não" (ou Y de "yes" ou N de "not"). A linha 40 prevê a possibilidade de o utilizador introduzir uma letra diferente de S ou N. Note o sinal <> que significa "diferente de" e que se introduz premindo [W] e SYMBOL SHIFT. Note também a palavra THEN (então) antes de GOTO. A linha pode ser portanto traduzida por:

Se a\$ for diferente de "S" e a\$ for diferente de "s" ou a\$ for diferente de "N" e a\$ for diferente de "n" então vá para 30.

Portanto, se houver um engano, o programa regressa à linha anterior espera que o utilizador volte a responder. Do mesmo modo, na linha 50 estabelece-se que se a letra introduzida for um S maiúsculo OU minúsculo o programa ENTÃO saltará para a linha 70 e parará na 80 devido à ordem STOP. Na linha 60 estabelece-se que se a letra introduzida for um N maiúsculo ou minúsculo o programa ENTÃO saltará para a linha 90 e parará porque não há mais linhas.

Se quiser saber de imediato mais coisas sobre a impressão de caracteres no écran, passe à página 70. Se quiser continuar a tomar conhecimento do que é básico no BASIC em geral e no 48 BASIC em particular ... continue, e aprenda a tratar das expressões matemáticas.

O Basic e a Matemática

Nos computadores os sinais das operações básicas são diferentes dos usuais, e note também as teclas que lhes dão acesso.

- + (adição), na tecla [K], premida simultaneamente com SYMB SHIFT.
 - (subtração), na tecla [J], premida simultaneamente com SYMB SHIFT.
 - * (multiplicação), na tecla [B], premida simultaneamente com SYMB SHIFT.
 - / (divisão), na tecla [V], premida simultaneamente com SYMB SHIFT.
 - ↑ (potenciação) na tecla [H], premida simultaneamente com SYMB SHIFT.
- SQR (raiz quadrada) na tecla [H], premida simultaneamente com EXTEND.

Recorde-se de que para obter a raiz de um número basta elevá-lo a uma potência igual à inversa da raiz. Isto é: para calcular a raiz cúbica de um número, basta elevá-lo a uma potência de um terço. Portanto, muito embora nos computadores só esteja prevista a possibilidade do cálculo directo da raiz quadrada, é possível calcular as raízes de qualquer grau. Isto é:

Para se obter o quadrado de (por exemplo) o número 5, faz-se:

```
PRINT 5↑2
```

Para se obter a raiz quadrada de (por exemplo) 25, faz-se:

```
PRINT SQR 25
```

Para se obter a raiz cúbica de 25 far-se-á:

```
PRINT 25↑(1/3)
```

E para se obter uma raiz qualquer r de um número qualquer n far-se-á:

```
PRINT n↑(1/r)
```

No 48 BASIC, quando várias operações são incluídas na mesma expressão, a sua sequência obedece à seguinte **prioridade**:

Subscritos e "cortes"	12
Todas as funções excepto NOT e menos unário	11
↑	10
Menos unário (menos usado para negar)	9
*, /	8
+, - (menos usado para subtrair)	6
=, >, <, <=, =>, <> (operadores relacionais)	5
NOT	4
AND	3
OR	2

Quando numa expressão há várias multiplicações e divisões, são executadas a partir da esquerda. Seguem-se as adições e as subtrações, que também são executados por essa ordem. Mas a ordem pode ser contornada recorrendo-se aos parenteses. As expressões tornam-se então idênticas às algébricas.

Compare os resultados de:

```
10 INPUT a
20 INPUT b
30 INPUT c
40 LET d = a + b * c
50 PRINT d
```

e de:

```
10 INPUT a
20 INPUT b
30 INPUT c
40 LET d = (a + b) * c
50 PRINT d
```

O computador pode também calcular directamente uma expressão, actuando então como uma simples calculadora. Basta escrever, por exemplo:

```
PRINT (1323 + 28.03) * .1385
```

para que o computador dê a resposta, funcionando como uma calculadora. Note que, como nestas e como é costume nos países de língua inglesa, a vírgula é substituída por um ponto, e o zero no início dos números decimais pode ser dispensado.

Os Spectrum + 2 e + 3 trabalham com 14 caracteres (isto é 13 algarismos e o ponto decimal ou 12 algarismos, o sinal - e o ponto . se o número for negativo e decimal. No entanto, quando se faz PRINT somente são apresentados 8 algarismos significativos.

Quando um número tem mais de 14 caracteres, os Spectrum + 2 e + 3 usam a notação científica. Essa notação mostra os números sob a forma de uma potência de 10. Nos computadores é expressa por um número decimal com um só algarismo à esquerda do ponto decimal, seguido pela letra E (de expoente, por um sinal de + ou - e por outro número que representa o expoente da potência de 10.

Assim:

```
2.34E + 14
```

Significa 2.34 vezes 10 elevado a 14. Ou seja:

$$2.34 \times 10^{14}$$

Escreva:

```
PRINT 2345678923456789
```

e veja o que acontece.

Note, por outro lado, que se introduzir no computador em notação científica um número com menos de 14 caracteres, o computador convertê-lo-á à notação normal. Experimente com:

```
2.34E3
```

Note que, tal como acontece na notação normal, o sinal + poder ser suprimido se o número for positivo.

Passemos a outro aspecto do cálculo:

Todos os computadores têm problemas devidos aos erros resultantes dos arredondamentos. Isso é inerente ao seu próprio funcionamento e pode dar origem a situações aparentemente embaraçosas.

Experimente o seguinte programa:

```
10 LET A = 1.01 - 1
20 PRINT A
```

O computador responderá que A é igual a 0.009999999, em vez de .01, como seria de esperar. A solução está em usar uma linha correctora:

```
25 LET A = INT (A * 100 + .5)/100
```

INT é a abreviatura de "inteiro" e é obtido premindo a tecla R em conjunto com a de EXTEND. A linha 25 começa por aumentar o valor de A para 1.499999998, cuja parte inteira é então dividida por 100, fazendo A = .01.

Os Spectrum + 2 e + 3, como todos os outros computadores podem tomar decisões. comparando caracteres ou quantidades. (Na verdade é essa característica que estabelece a distinção principal entre os computadores e as calculadoras). No entanto os Spectrum + 2 e + 3 dispõem de uma opção que lhes permite trabalhar como calculadoras. Veja a pág. 158.

A instrução IF... THEN, já mencionada, funciona se uma determinada condição for verdadeira. As condições mais simples são as que comparam dois números ou duas cadeias de caracteres. O computador pode verificar se um número é igual, maior ou menor que outro, ou se uma cadeia

de caracteres é igual a outra ou a precede na ordem alfabética, usando as relações =, <, >, <=, >=, e <>.

- = significa "igual", como se sabe.
- < (SYMB SHIFT e [R]) significa que o que está à esquerda é "menor que" o que estiver à direita.
- > (SYMB SHIFT e [T]) significa que o que estiver à esquerda é "maior que" o que estiver à direita.
- <= (SYMB SHIFT e [Q]) significa "igual ou menor que".
- => (SYMB SHIFT e [E]) significa "igual ou maior que".
- <> (SYMB SHIFT e [W]) significa "diferente de".

Comprove isso com o jogo do programa seguinte, destinado obviamente a duas pessoas, uma que introduz um número no computador e outra que procura adivinhar esse número:

```

10 INPUT a: CLS
20 INPUT "Adivinhe o número", b
30 IF b=a THEN PRINT "Acertou !": STOP
40 IF b < a THEN PRINT "Demasiado pequeno ! Repita !"
50 IF b > a THEN PRINT "Demasiado grande ! Repita !"
60 GOTO 20

```

A capacidade dos computadores quanto a cálculos matemáticos vai muito mais além: eis um programa simples, para a resolução de equações simultâneas, que demonstra de imediato algumas das possibilidades dos computadores no domínio das matemáticas:

```

10 CLS
20 PRINT AT 0,0 INK 1; "EQUAÇÕES SIMULTÂNEAS"
30 PRINT AT 2,3; INK 2; "AX+BY=E   CX+DY=F"
40 INPUT INK 1; "A=";a;"B=";b;"E=";e
50 INPUT INK 2;"C=";c;"D=";d;"F=";f
60 LET g=(c*b)-(d*a)
70 LET h=(-d*e)+(b*f)
80 LET i=(c*e)+(-a*f)
90 LET x=h/g: LET y=i/g
100 PRINT AT 4,5; INK 1;a "X+";b;"Y=";e
110 PRINT AT 5,5; INK 2;c;"X+";d;"Y=";f
120 PRINT AT 10,8;FLASH1;"RESPOSTA:"; FLASH 0;
    BRIGHT 1; "X=";x;BRIGHT 0

```

```

130 PRINT AT 11,8; FLASH 1; "RESPOSTA:"; FLASH 0;
    BRIGHT 1; "Y=";y;BRIGHT 0
140 INPUT "OUTRA VEZ ? (S/N)"; a$
150 IF a$ <> "S" AND a$ <> "s" OR a$ <> "N" AND a$ <> "n" THEN
    GOTO 140
160 IF a$="s" OR a$="S" THEN RUN
170 IF a$="n" OR a$="N" THEN NEW

```

(Note nas linhas 120 e 130 o uso de FLASH 1 para fazer relampejar os caracteres impressos no écran entre essa instrução e a de FLASH 0. Note também o uso de BRIGHT 1 e BRIGHT 0 para pôr caracteres em destaque).

Para ter acesso a FLASH, prima simultaneamente EXTEND e SYMB SHIFT e depois a tecla [V]. Para ter acesso a BRIGHT, prima simultaneamente EXTEND e SYMB SHIFT e depois a tecla [B].

Nos Spectrum + 2 e + 3 é possível trabalhar com logaritmos de base decimal ou de base natural, e também fazer cálculos trigonométricos, gerar números aleatórios (ou antes: pseudo-aleatórios), usar expressões lógicas para além de AND e OR, definir funções, etc., etc. Do mesmo modo, é possível representar graficamente, equações algébricas, funções, etc. inclusivé em 3 dimensões. Continue e verá...

Os Gráficos

Os Spectrum + 2 e + 3 podem fazer a apresentação em 8 cores, ou 16 (segundo terminologia usada por outros construtores), uma vez que através da variação do brilho, ou seja de BRIGHT, a que se tem acesso premindo EXTEND e depois SYMB SHIFT e a tecla [B], obtém-se uma tonalidade mais clara de cada côr.

O écran possui uma cercadura ou borda designada por BORDER, na qual são apresentadas as indicações sobre o registo e a carga dos programas. O espaço ocupado pela imagem propriamente dita funciona como uma folha de papel, e é por isso designado por PAPER.

A cor ou "tinta" com a qual são impressos os caracteres ou marcados pontos sobre o PAPER é designado por INK.

As cores básicas são designadas por números:

- 0 = negro
- 1 = azul
- 2 = vermelho
- 3 = magenta (arroxeadado)
- 4 = verde
- 5 = turqueza (cyan)
- 6 = amarelo
- 7 = branco

Para que no écran surja uma cercadura de uma dada côr bastará introduzir BORDER através da tecla [B] e acrescentar o número referente à cor que se pretende (premindo depois ENTER, obviamente). Se o écran estiver "limpo" — isto é: sem nada presente — ter-se-á do mesmo modo, um "paper" de uma certa côr introduzindo PAPER (tecla C) e acrescentando o número da cor. Ainda do mesmo modo, para se usar (por exemplo) "tinta" amarela recorrer-se-á a INK, usando a tecla [X], e juntar-se-á 6.

Note que a côr de BORDER muda logo que se introduz a respectiva instrução, mas que isso não acontece quando se pretende mudar a côr de PAPER ou INK. Essas mudanças de côr, quando se usa BASIC, só se executam depois de fazer CLS. Sendo assim, convém definir as cores de PAPER e INK, antes de apresentar qualquer texto ou imagem, para que estes não sejam apagados ao fazer-se CLS.

Execute o seguinte programa:

```
19 PRINT "***";
20 GOTO 10
```

O écran ficará coberto de estrelinhas negras sobre fundo branco até aparecer a pergunta "scroll ?"

Faça BREAK e depois introduza, por exemplo, BORDER 4 seguido de PAPER 2 e de INK 7 (carregando evidentemente em ENTER depois de cada uma dessas instruções). Repita o programa: a cercadura tornar-se-á verde, o fundo vermelho e as estrelas brancas. Ensaie outras cores.

É possível empregar INK e PAPER depois de PRINT: o modo Spectrum admite duas das oito cores no espaço de um carácter. Mas então só o espaço de oito pontos de imagem correspondente a cada carácter mudará de cor. Faça CLS e experimente o seguinte programa

```
10 BORDER 1
20 INK INT(RND*8)
30 PAPER INT(RND*8)
40 PRINT "SPECTRUM";
50 GOTO 10
```

A função RND, descrita em pormenor na pág. 48, permite que o computador escolha ao acaso uma das oito cores básicas (de 0 a 7). Como esse arredondamento é feito pela unidade inferior, há que fazer RND*6 para se terem as cores entre 0 e 7.

Depois de preencher o écran com o nome do computador e o fundo em diferente combinação de cores enquanto a cor da "moldura" vai também mudando, o programa deter-se-á com a indicação "scroll ?". Premindo qualquer tecla excepto N, BREAK, SPACE ou STOP, a imagem subirá e o computador continuará a combinar cores. Para interromper o programa bastará carregar em BREAK. Para o continuar faz-se CONT (tecla C) e para o recomeçar faz-se RUN.

Experimente depois o seguinte programa, um pouco mais complexo:

```
10 FOR b=0 TO 7
20 BORDER b: PAPER b: CLS
30 PRINT AT 6, 12; INK 9; b
40 FOR p=0 TO 7
50 PRINT AT p+8,8; INK p; PAPER 9; p
60 b * p - 20 + p
70 FOR i=0 TO 7
80 PRINT INK i; PAPER p; " "; i;
90 i * 5
100 NEXT i
110 NEXT p
120 NEXT b
```

Note em primeiro lugar, na linha 20, a presença de três instruções diferentes na mesma linha. Isso é possível no BASIC sinclair desde que as instruções sejam separadas por dois pontos (:). Este sinal é acessível directamente através da Tecla [:], na fila inferior, no extremo esquerdo.

Note também que na linha 50 se inclui a instrução PAPER 9, da qual resulta que os caracteres terão a côr inversa da do fundo.

Finalmente note que no programa existem três ciclos (ou loops) FOR ... NEXT "aninhados" (nested) — isto é, colocados uns entre outros. Note que os NEXT são colocados na ordem inversa dos FOR.

Vejamos agora como fazer gráficos com os Spectrum + 2 e + 3.

Como se disse, a apresentação de 32 colunas é mais legível. Para fazer gráficos com esse tipo de apresentação — gráficos de bloco — usam-se os caracteres gráficos disponíveis nas teclas 1 a 8 e que são acessíveis através da tecla GRAPH. Podem ser impressos a preto ou a côr, ou a branco sobre fundo negro ou côr se se carregar primeiro na tecla INV VIDEO. Para voltar à situação anterior carrega-se na tecla TRUE VIDEO. Experimente fazer:

```
10 BORDER 1 : INK INT (RND * 8)
20 PRINT " ";
30 GOTO 10
```

Na linha 20, entre as aspas, insira o quadrado que se obtém premindo a tecla [8] no modo gráfico. Experimente em seguida um programa mais complexo:

```
10 LET a$ = " "
20 FOR x=1 TO 7
30 LET a$=a$ + CHR$ (RND * 14 + 129)
40 NEXT X
50 INK INT (RND * 8)
70 PRINT a$;
80 GOTO 70
```

O resultado é a formação de padrões de desenho e côr sempre diferentes — quando o écran fica cheio e aparece a pergunta "scroll ?" faz-se BREAK, muda-se na linha 20 o número 7 por outro até 8, faz-se depois RUN, e surge um novo padrão com novas côres.

Note na linha 30 a função CHR\$, que é uma abreviatura de "CHaracter String", ou seja de "cadeia de caracteres" e que permite introduzir os caracteres através de números colocados a seguir a ela. Esses números, entre 32 e 95 e entre 97 e 126 no caso do 48 BASIC, do 128 BASIC e do +3 BASIC, obedecem ao código ASCII (American Standard Code for Information Interchange) e constam da tabela publicada no fim do presente manual.

Outro exemplo:

```
10 BORDER 0: CLS
20 LET h=16: LET v=11
30 LET x=INT (RND * 3 - 1) : LET y = INT (RND * 3 - 1)
40 INK INT (RND * 8)
50 FOR z = 1 to 20
60 PRINT AT v,h; CHR$ 143
70 LET h = h + x
80 IF h < 0 THEN LET h = 31
90 IF h > 31 THEN LET h = 0
100 IF v < 0 THEN LET v = 21
110 IF v > 21 THEN LET v = 0
```

De cada vez que se reinicia o programa surgem padrões diferentes. Substituindo na linha 30 o número 143 (que corresponde a um quadrado) por outro entre 33 e 142, o padrão será formado pelo carácter correspondente.

É possível definir (desenhar) outros caracteres gráficos — um expediente muito usados nos jogos. Para isso veja-se a pág. 62.

Como também se disse, no segundo tipo de apresentação o écran divide-se em 256 "elementos de imagem" (pixels) ou "pontos" na horizontal e 176 na vertical. Note-se que há uma zona de duas linhas de caracteres ou de 16 filas de "pontos" na parte inferior do écran, normalmente reservados a mensagens sobre o funcionamento do computador, mas acessíveis mediante certos procedimentos.

A localização dos pontos faz-se por coordenadas, a primeira designando a abcissa x, ou seja a distância em pontos na vertical, a segunda designando a ordenada y, ou seja a distância em pontos na horizontal.

Note que, quando se traça um gráfico, usando abcissas e ordenadas, parte-se sempre do canto inferior direito — isto é: as abcissas da direita para a esquerda e as ordenadas de baixo para cima. Portanto, a localização dos pontos em altura no écran faz-se ao contrário da dos caracteres:

Para imprimir um ponto no écran usa-se a instrução PLOT, que se pode traduzir por MARCAR num quadro. Assim:

```
PLOT 0,0
```

marcará um ponto no canto inferior esquerdo do écran. (Note que as coordenadas são separadas por uma vírgula).

PLOT 255,0

marcará um ponto no canto inferior direito do écran.

PLOT 0,175

marcará um ponto no canto superior esquerdo do écran.

PLOT 255,175

marcará um ponto no canto superior direito do écran.

Premindo a tecla (W) obtém-se DRAW (desenhar) à qual se junta:

- o número de pontos que se pretende avançar na horizontal, seguido de uma vírgula.
- o número de pontos que se pretende subir na vertical.

Suponha-se que se introduziu:

DRAW 50,100

Então surgirá no écran uma linha que parte do último ponto marcado e se deslocará 50 pontos para a direita e 100 para cima. Note que quando se liga o computador, ou se faz CLS, esse ponto se situa no canto inferior esquerdo do écran, mas pode-se determinar outro através da instrução PLOT.

Assim, se depois de se traçar uma linha se introduzir uma instrução para o traçado de nova linha, ela partirá do ponto onde a primeira terminou, a menos que se faça novo PLOT.

Note que os números que se seguem a DRAW não indicam as coordenadas, mas sim os pontos que a linha deve deslocar-se. Se a deslocação se fizer para a esquerda ou para baixo, os números devem ser antecedidos do sinal (-). Experimente de novo:

10 DRAW 50,100

Note que a linha é traçada a partir do canto inferior esquerdo do écran, para a direita e para cima.

Faça NEW e introduza:

10 PLOT 30,20

20 DRAW 50,100

Note que a linha agora tem a mesma extensão e a mesma inclinação, mas é traçada a partir de um ponto mais à direita e para cima.

Se quiser comparar as duas linhas, reuna as instruções, fazendo NEW e introduzindo:

10 DRAW 50,100

20 PLOT 120,50

30 DRAW 50,100

Introduza em seguida, directamente:

DRAW 80,10

A linha correspondente surgirá a partir do extremo da anterior e dirigir-se-á para a direita e um pouco para cima, uma vez que só subirá 10 pontos.

Faça RUN e repita o exercício com a seguinte instrução directa:

DRAW -80,-0

A linha correspondente surgirá a partir do extremo da anterior, mas dirigir-se-á para a esquerda e para baixo.

Para desenhar um quadrado de lado 100 no centro do écran far-se-á:

10 PLOT 63,38

20 DRAW 100,0

30 DRAW 0,100

40 DRAW -100,0

50 DRAW 0,-100

O desenho pode ser feito a cores. Adicione-se ao programa anterior a linha:

5 BORDER 2: PAPER 1: INK 6

Faça-se RUN e o quadrado aparecerá desenhado a amarelo sobre um fundo azul com uma cerca-dura vermelha.

Apague a imagem, fazendo NEW, e experimente:

```

10 BORDER 1: PAPER 1: CLS
20 INK 2
30 PLOT 128,174
40 DRAW 70,-140
50 DRAW -152,80
60 DRAW 164,0
70 DRAW -150,-80
80 DRAW 70,140

```

Para tornar os Spectrum + 2 e + 3 em máquinas de desenhar faça:

```

10 INPUT "INK";i
20 BORDER 3: PAPER 0: INK i; CLS
30 PLOT 25,25
40 LET x=5
50 INPUT k$
60 IF k$="1" THEN DRAW 0,1
70 IF k$="2" THEN DRAW 0,-x
80 IF k$="3" THEN DRAW x,0
90 IF k$="4" THEN DRAW -x,0
100 GO TO 50

```

Quando o programa pedir um dado, prima as teclas 1, 2, 3 ou 4 e veja o resultado.

Vejam agora como desenhar círculos e arcos de círculo. Nem todos os computadores o podem fazer directamente, mas o 48 BASIC, tal como o 128 BASIC e o + 3 BASIC, dispõe de um comando especial para esse fim — o CIRCLE.

Para obter CIRCLE prima EXTEND e depois SYMB SHIFT e a tecla [H].

Introduza:

```
10 CIRCLE 125, 90, 50
```

Se quiser, adicione:

```

20 CIRCLE 125, 90, 20
30 CIRCLE 125, 90, 100

```

Os dois primeiros números a seguir a CIRCLE correspondem às coordenadas do centro do círculo. O segundo número corresponde ao raio.

Para desenhar um arco de círculo usa-se DRAW com um parâmetro adicional: o ângulo do arco em radianos. Na prática o ângulo é introduzido em frações de PI.

Comece por colocar um ponto num lugar conveniente do ecrã — no centro do ecrã, por exemplo:

```
PLOT 127,87
```

Introduza:

```
DRAW 50,0, PI
```

o resultado será a metade inferior de um círculo — a metade inferior porque o 48 BASIC, tal como o 128 BASIC e o + 3 BASIC, desenha os círculos no sentido inverso dos ponteiros do relógio.

Experimente fazer:

```
DRAW -50,0,PI
```

Comprove adicionando DRAW 50,0

Para traçar um círculo completo usando DRAW, faça NEW, coloque de novo no écran o ponto que pretende que sirva de centro, e faça:

```
DRAW 50,0, 2 * PI
```

Para traçar um quarto de círculo deve-se usar PI/2 ou .5 * PI como terceiro parâmetro.

Faça NEW e recapitule:

```

10 PLOT 127,87
20 DRAW 50,0,PI

```

Faça RUN e introduza sucessivamente as seguintes instruções directas, observando os resultados:

```

DRAW -50, 0, PI
DRAW 50, 0
DRAW 50,0,.8 * PI
DRAW -50,0,.5 * PI
DRAW 50, 0, .3 * PI
PLOT 125, 0
DRAW 0,175

```

Como se disse, é possível representar equações e funções nos Spectrum + 2 e + 3. Para isso há que dividir o ecrã em quatro partes, por uma linha horizontal e outra vertical cruzando-se a meio, ou seja na localização 127,87. Portanto há que adicionar essas coordenadas aos valores que se pretende representar. Vejamos:

```

10 REM Sinusoide
20 FOR x=0 to 255
30 PLOT x,87
40 NEXT x
50 FOR y = 0 TO 175
60 PLOT 127,y
70 NEXT y
80 FOR I = -20 TO 20 STEP .03
90 PLOT I * 6 + 127,87 + 20 * SIN I
100 NEXT I

```

As linhas 20 a 70 traçam os eixos no ecrã. A linha 80 determina os limites da curva, em conjunto com os factores 6 e 20 na linha 10, estabelecidos experimentalmente para que o aproveitamento do ecrã seja máximo e os respectivos limites não venham a ser excedidos. O valor de STEP, na linha 80, pode ser aumentado para acelerar o traçado da curva, mas esta ficará então representada por pontos mais afastados uns dos outros. SIN (seno), tal como COS (coseno) e TAN (tangente) obtêm-se através de EXTEND e [Q], [W], e [E], respectivamente.

O programa seguinte permite representar funções de x :

```

10 REM Representação de f(x)
20 CLS
30 PLOT 0,87: DRAW 255,0
40 PLOT 127,0: DRAW 0,175
50 INPUT "ESCALA=";s: INPUT "f(x)=";e$
60 LET t=0
70 FOR f=0 TO 255
80 LET x=(f-128)*s/128: LET y=VAL e$
90 IF ABS y>87 THEN LET t=0: GOTO 120
100 IF NOT t THEN PLOT f, y+88: LET t=1: GOTO 120
110 DRAW 1,y-anty
120 LET anty=INT(y+.5)
130 NEXT f

```

Note na linha 80 a função VAL, que se obtém premindo EXTENDED e [J] e introduz no computador o valor de uma expressão contida numa cadeia de caracteres — no caso designado por e\$. Note-se também na linha 100 a relação lógica NOT (que se obtém premindo SYMBOL SHIFT e [S]) e a variável t, esta servindo apenas para determinar uma alteração na sequência do programa quando uma condição for cumprida. Assim, enquanto i NÃO existir (isto é, se t for igual a zero), as instruções subsequentes serão cumpridas, no caso contrário passar-se-á à linha 110.

Um programa muitas vezes procurado é o que permite traçar um gráfico a partir de dados previamente obtidos. Introduza:

```

10 REM Gráfico
20 CLS
30 INPUT "Título? (max. 32 caracteres)"; a$
40 INPUT "valor maximo?"; v
50 LET e=v/150: CLS
50 PRINT AT 0,0; a$; AT 6, 0; v
60 REM Traçado dos eixos
70 PLOT 0,10
80 DRAW 240,0
90 PLOT 0,10
100 DRAW 0,100
110 REM Graduacao horizontal
120 FOR n=0 to 240 STEP 20
130 PLOT n,10: DRAW 0,3
140 NEXT n
150 REM Graduacao vertical
160 FOR n=10 to 110 STEP 10
170 PLOT 0,n: DRAW 3,0
180 NEXT n
190 REM Introducao de dados
200 PLOT 0,10
210 FOR a= 10 TO 240 STEP 20
220 INPUT "QUANTO?"; q
230 REM Tracado
240 LET qe = q/e
250 DRAW a-PEEK 23677, qe+10-PEEK 23678
260 NEXT a

```


Para traçar outros gráficos sobre os mesmos eixos, para comparação com o primeiro, basta adicionar uma linha:

```
270 GOTO 200
```

Para introduzir cor nos gráficos basta adicionar:

```
15 BORDER 1: PAPER 1: INK 7
205 INPUT "COR ?"; b
```

e alterar a linha 250:

```
250 DRAW INK b; a-PEEK 23677,qe+10-PEEK 23678.
```

A função PEEK (que significa literalmente "espreitar") permite verificar o conteúdo das localizações da memória. No exemplo, PEEK 23677 indica a coordenada x do último ponto marcado, e PEEK 23678 o da coordenada y. Assim, a linha 250 permite traçar linhas com a instrução DRAW em valores absolutos, o que em regra é mais fácil do que a introdução em valores relativos, como o 48, o 128 e o + 3 BASIC exigem.

A utilização da cor pode conduzir a resultados espetaculares, mesmo sem recorrer a técnicas especiais de programação. Introduza o seguinte programa e aprecie o resultado:

```
10 BORDER RND * 6
20 INK RND * 7
30 PAPER RND * 6: CLS
50 LET z=RND * 10 + 2
60 FOR x=0 TO 174 STEP Z
70 PLOT 128,0
80 DRAW -128,0
90 NEXT x
100 FOR x=-127 TO 127 STEP z
110 PLOT 128,0
120 DRAW x,175
130 Next x
140 FOR x=174 TO 0 STEP -z
150 PLOT 128,0
160 DRAW 127,x
170 NEXT x
180 PAUSE 200
190 GOTO 10
```

A instrução PAUSE determina uma pausa em que cada unidade corresponde a 1/50 de segundo: PAUSE 200 corresponde a uma paragem de $200/50 = 4$ segundos.

Finalmente faça:

```
10 BORDER 0: PAPER 1: INK 6
20 CLS
30 FOR y = 0 TO 20 STEP 2
40 PLOT 0,y
50 DRAW 255,0
60 NEXT y
70 FOR n = 100 TO 220 STEP 30
80 FOR x = -10-n/10 TO 10+n/10
90 PLOT n,35 + n/10
100 NEXT x: NEXT n
```

Veja o resultado, faça BREAK e junte:

```
120 LET x = RND * 255
130 LET y = RND * 104 + 71
140 LET b = INT (175 - y)/8
150 LET c = INT x/8
160 PLOT 0,0 : DRAW OVER 1; x,y
170 PAUSE 1
180 PLOT 0,0 : DRAW OVER 1; x,y
190 PRINT AT b, c; "*"
200 GOTO 120
```

OVER 1, na linha 160 e na linha 180, permite que uma linha seja apagada por outra com a cor do fundo. BRIGHT, INVERSE e FLASH são outras instruções que permitem obter interessantes efeitos.

As Funções

O BASIC, seja qual for, baseia-se no uso de expressões — aritméticas, condicionais ou de cadeias. Para tornar as expressões ainda mais úteis, o BASIC permite criá-las também através de *funções*, algumas das quais podem por sua vez ser criadas pelo utilizador.

As funções principais usadas no 48 BASIC dividem-se em três grupos:

Funções aritméticas:

ABS — Valor ABSoluto de um número

Indica o valor absoluto de um número ignorando o seu sinal e tratando-o como se fosse positivo. $ABS(-2) = 2$ e $ABS(2)$ é também 2.

EXP — Função EXPonencial

Destina-se à realização de operações de matemáticas superiores tendo por base o número $e = 2,7182818...$ EXP(x) eleva o número e à potência x . Note que o uso desta função conduz rapidamente a números muito altos: o expoente mais elevado que o 48 BASIC aceita para ela é 88.

INT — Valor da parte INTeira de um número

Torna um número decimal em inteiro arredondando-o para a unidade imediatamente inferior. Assim, $INT(3.256)$ é igual a 3 e $INT(-4.758)$ é igual a -5 (recorde-se que -5 é um valor menor que -4).

LN — Logaritmo Natural de um número

O logaritmo natural de um número é a potência a que é preciso elevar para obter esse número. Note que os logaritmos mais vulgarmente usados são os decimais. No BASIC Sinclair não existe uma função que dê directamente os logaritmos decimais, mas estes podem ser obtidos dividindo pelo logaritmo natural de 10 o logaritmo natural do número cujo logaritmo decimal se pretende. Parece complicado mas corresponde simplesmente a:

$$\text{Logaritmo decimal de } (x) = \text{LN}(x) / \text{LN}(10)$$

PI (π)

Fornece o valor de $p = 3,14159265$

SIGN — Sinal (SIGnal) de um número

Dá como resultado +1 se o número for positivo e -1 se for negativo. Portanto, $SGN(-232)$ é -1 e $SGN(3239)$ é +1.

SQR — Raiz quadrada (SQuare) de um número

Esta função não pode ser aplicada a números negativos. Nesse caso o computador responde com a mensagem de erro **A Invalid argument.**

Funções trigonométricas

O 48 BASIC funciona com radianos e não com graus. Recordar-se que um radiano é o ângulo que corresponde a um arco com o comprimento do raio do círculo. A razão porque o 48 BASIC usa radianos em vez de graus é simples:

Considere um círculo com o raio igual a 1. A circunferência será portanto igual a $2r$, pois o diâmetro é de 2. Se o raio girar 90 graus, ou seja um quarto de círculo, descreverá um arco a correspondente a $r/2$. Se girar de 180 graus o arco a corresponderá a r . E se girar um círculo o arco a corresponderá a $2r$. Portanto pode-se ignorar os graus e medir o ângulo pela extensão do arco a , ou seja em radianos.

Para se converter graus em radianos multiplicam-se os graus por 180 e divide-se o resultado por PI. Ou, em alternativa, multiplicam-se os graus por 57,2978. Para converter de novo radianos em graus multiplicam-se aqueles por PI e divide-se o resultado por 180 ou em alternativa multiplicam-se os radianos por 0,017453293.

Suponha-se um círculo cruzado por dois eixos, um vertical (yy) e outro horizontal (xx). A distância a que um ponto sobre a circunferência se encontra em relação ao eixo yy é o cosseno e aquela a que ele se encontra em relação ao eixo xx é o seno. O 48 BASIC dá o valor do cosseno através da função COS, e o do seno através da função SIN (de SINE).

Note que quando o ponto se situa à esquerda do eixo dos yy o cosseno torna-se negativo, e quando desce abaixo do eixo dos xx o seno torna-se negativo. Note também que quando o ponto descreve um ângulo maior que $2r$ o seno e o cosseno voltam aos valores iniciais.

Dividindo o seno pelo cosseno obtém-se a tangente. A função correspondente é TAN.

Para se obter o valor do arco correspondente aos valores de SIN, COS, e TAN recorre-se às funções ASN (Arc SiN ou arco de seno); ACS (Arc CoS ou arco de cosseno), e ATN (Arc TaNgent ou arco de tangente).

Funções de cadeia

CHR\$ — Cadeia de caracteres (CHaRacter string).

Fornece o caracter correspondente a um determinado número na lista dos caracteres do 48 BASIC. Note que na lista não há caracteres com números inferiores a 33, pelo que se se introduzir CHR\$, por exemplo, não se obterá resposta. Note também que no 48 BASIC as "palavras de tecla" como LET, LIST, etc., são tratadas como um único caracter. Consequentemente, e ao contrário do que acontece no 128 BASIC e no +3 BASIC, a função CHR\$ pode dar como resposta uma cadeia de mais de uma letra.

CODE — Código de um caracter (Character CODE).

É o inverso de CHR\$: fornece o número de código de um caracter. Assim, enquanto CHR\$(65) é A, CODE "A" é 65. Note que se CODE for aplicado a uma cadeia de mais de uma letra, só fornecerá como resposta o código da primeira letra. Se a cadeia não tiver caracteres, o resultado será zero.

LEN — Comprimento de uma cadeia (String LENght).
Fornece o comprimento de uma cadeia. Por exemplo: LEN "SPECTRUM" dará 8 como resultado porque a cadeia tem 8 caracteres.

STR\$ — Junção a cadeia (STRing\$).
Permite juntar a uma cadeia o número ou o resultado de uma expressão que deva ser impresso com ela. Por exemplo: "Julho" + STR\$ (31) terá por resultado a cadeia "Julho 31".

VAL — Valor de uma expressão (eVALuate).
É oposto de STR\$. Permite avaliar o resultado de qualquer expressão numérica correcta contida numa cadeia. Por exemplo: VAL "43" é 34 e VAL "3 * 5 + 2" é 17. Isto é: actua como se eliminasse as aspas que ladeiam a cadeia.

VAL\$ — Valor de uma cadeia (VALue string).
É a inversa de STR\$. Do mesmo modo que VAL, actua como se eliminasse as aspas que ladeiam uma cadeia, mas o resultado, obviamente, em vez de ser um valor numérico é uma cadeia de caracteres, que então poderá ser usada em operações com cadeias.

Funções especiais

RND — Número obtido aleatoriamente, ou seja ao acaso, numa lista. (RaNDom number).
O resultado é o número entre 0 e 1, pelo que, quando se pretende obter ao acaso um número, por exemplo, entre 0 e 7, como é o caso da atribuição aleatória de cores, tem de multiplicar RND pelo número de opções e extrair a parte inteira. Assim, no caso, ter-se-á:

```
INK INT(RND*8)
```

Como a função INT arredonda sempre os números por baixo será necessário multiplicar RND por 8 para obter números inteiros entre 0 e 7. Pela mesma razão, se se pretender obter números a partir de 1, será necessário multiplicar RND pelo número máximo (inteiro) que se pretende obter e juntar-se 1. Isto é, para obter números entre 1 e 20, far-se-á:

```
INT (RND*20) + 1
```

Note-se que os números não são realmente obtidos ao acaso, mas sim gerados numa sequência que se reinicia sempre que o computador é ligado ou se faz NEW e que tem por base as potências de 75 (75, 75*75, 75*75*75, etc.), dividindo cada resultado por 65537, usando o resto, subtraindo 1 deste e dividindo o resultado por 65536, e assim sucessivamente. Os números da lista assim obtida não parecem ter qualquer relação entre eles e a lista é tão extensa que ninguém a pode reter na memória. No entanto, pela maneira como ela é gerada, os números apresentam-se pela mesma ordem. Experimente fazer:

```
10 FOR N= 1 TO 6: PRINT INT(RND * 45) + 1 : NEXT N
```

Faça RUN repetidas vezes e verá que a sequência é a mesma. Para obstar a isso recorre-se à função RANDOMIZE.

RANDOMIZE — Origem de uma lista aleatória.
Determina o início da sequência RND. Se se fizer, por exemplo, RANDOMIZE 30, a lista de números aleatórios começará (ou recomeçará) no número que nela figura no trigésimo lugar. Mas se se fizer RANDOMIZE 0, então o começo da lista será determinado pelo tempo decorrido desde o momento em que o computador foi ligado. Assim se se fizer:

```
10 RANDOMIZE 0
20 PRINT RND
30 GO TO 10
```

O primeiro número da sequência gerada por RANDOMIZE 0 surgirá no ecrã sempre que um programa voltar à linha 10 e determinar o recomeço da função RANDOMIZE. Note que esse número vai aumentando lentamente à medida que o tempo vai passando. Em seguida faça:

```
10 RANDOMIZE 0
20 PRINT RND
30 GO TO 20
```

Neste caso o programa não obriga a função RANDOMIZE a recomeçar num número determinado, e a função RND terá assim um ponto de partida aleatório. Consequentemente, e como se poderá ver no ecrã, os números são gerados tão acaso quanto é possível.

INKEY\$ — Introdução de cadeia por tecla. (Input KEY)
É aparentada com INPUT A\$, mas enquanto nesse caso o computador espera pela introdução da cadeia, no de INKEY\$ uma cadeia contendo a letra correspondente a uma tecla é introduzida de imediato, quando essa tecla é premida. Assim, se se fizer:

```
10 INPUT a$
20 IF a$ <> " " THEN PRINT a$
30 GOTO 10
```

Ter-se-á de premir ENTER antes de ver algo escrito, no ecrã. Pare o programa apagando as aspas esquerdas e fazendo STOP. Altere a linha 10 para:

```
10 LET a$ = INKEY$
```

Então, quando se premir uma tecla, o caracter correspondente aparecerá no ecrã de imediato, sem que seja preciso premir ENTER. Tente premir mais do que uma tecla ao mesmo tempo, usando SYMBOL SHIFT, etc., enquanto o programa corre. Um dos usos principais de INKEY\$ são os jogos, pois permite "programar" as teclas de modo que elas determinem movimentos numa dada direcção, o que pode ser exemplificado pelo seguinte programa:

```
10 LET a$ = INKEY$
20 IF a$ = " " THEN GOTO 10
30 IF a$ = "5" THEN PRINT "esquerda"
40 IF a$ = "6" THEN PRINT "abaixo"
50 IF a$ = "7" THEN PRINT "acima"
60 IF a$ = "8" THEN PRINT "direita"
70 GO TO 10
```

Note que tal como nos Spectrum Plus e ao contrário do que acontece no Spectrum 48K, as teclas em que se situam as setas nos Spectrum + 2 e + 3 não têm uma acção correspondente nos jogos: movem somente o cursor nas listagens. Os comandos dos jogos continuam a ser feitos pelas teclas 5, 6, 7, 8, na fila superior do teclado, se outras não tiverem sido escolhidas pelo programador.

POINT — *Determinação da cor de um ponto.*

Indica se a cor de um determinado ponto do ecrã é a do PAPER ou a de INK. Fazendo:

```
POINT (X,Y)
```

tem-se como resultado, para as coordenadas x.y. o número 1 se a cor do ponto for a de INK e a de 0 se for PAPER. É uma função muito útil para a impressão de cópias de ecrã e para a programação de jogos.

Funções definidas pelo utilizador

DEF FN — *Definição de função (DEFine FuNction)*

Permite definir funções que não estejam disponíveis nas teclas. Funciona em conjunto com:

FN — *Função (FuNction)*

Permite usar as funções definidas por DEF FN.

Note-se que nos Spectrum há uma função que indica directamente a raíz quadrada de um número, mas não há nenhuma que dê directamente o quadrado. Essa falta pode ser suprida com DEF FN, fazendo:

```
DEF FN a(x)= x * x
```

e obtendo a função, para um número x qualquer, com:

```
FN a(x)
```

Experimente:

```
10 DEF FN a(x)= x * x
20 INPUT x
30 LET q = FN a(x)
40 PRINT x;"**";x;"="";q
50 GO TO 20
```

e

```
10 DEF FN a(x,y)=b+(x*y)
20 INPUT b,x,y
30 PRINT FN a(x,y)
40 GO TO 20
```

Note-se que a função definida é identificada por uma letra (no caso a letra a), tal como as variáveis. E tal como estas as funções podem ser numéricas ou de cadeia, neste último caso identificadas também por um caracter seguido por \$. Por exemplo:

```
10 DEF FN a$(x$)=x$(2 TO)
20 INPUT x$
30 LET q$ = FN a$(x$)
40 PRINT q$
50 GO TO 20
```

As cadeias introduzidas serão impressas sem o primeiro caracter. A função definida em 10 corresponde, consequentemente, à função TL\$, existente noutros tipos de BASIC. Do mesmo modo:

```
DEF FN a$(x$,n) = x$(TO n)
```

corresponde a LEFT\$,

```
DEF FN a$(x$,n) = x$(n TO)
```

corresponde a RIGHT\$, e

```
DEF FN a$(x$,n,m) = x$(n TO m)
```

corresponde a MID\$.

Note também que as funções podem ter mais de um parâmetro e podem combinar os numéricos com os de cadeia. O número máximo de parâmetros, numéricos ou de cadeia, é de 26 — ou seja o das letras do alfabeto.

Note ainda que é possível definir qualquer função desde que ela corresponda a uma expressão aceitável pelo BASIC. Portanto é possível usar expressões sem parâmetros, mas mesmo assim há que conservar os parênteses:

```
DEF FN d()= INT(RND*6)+1
```

Do mesmo modo, pode-se usar uma função sem parâmetros para introduzir numa expressão os valores atribuídos originalmente às variáveis. Experimente o seguinte programa:

```
10 LET x=0 : LET y=5 : LET a=10
20 DEF FN p(x,y) = a+x*y
30 DEF FN q() = a+x*y
40 PRINT FN p(2.5)
50 PRINT FN q()
```

Na linha 40 foram atribuídos a x e y os valores de 2 e 5, respectivamente — portanto ela fará imprimir no ecrã os resultados da expressão $10+2*5=20$. Mas a linha 50 fará imprimir o resultado de $10+0*5 = 10$, pois que os seus parênteses estão vazios e portanto em nada alteram os valores inicialmente atribuídos às variáveis.

As Matrizes

Uma matriz (*array*) é uma forma de ordenar valores de modo que a sua localização seja conhecida. Uma folha de calendário é um exemplo de matriz. Mas os valores atribuídos aos elementos de uma matriz não necessitam de seguir uma ordem, como os dias do mês. Os valores podem ser quaisquer, tal como no caso das variáveis — a matriz serve apenas para determinar a ordem dos valores.

Suponhamos que se tem uma lista de valores numéricos:

```
12 5 7 22 14
```

Óbviamente, esses valores podem ser atribuídos a outras tantas variáveis:

```
LET a = 12
LET b = 5
LET c = 7
LET d = 22
LET e = 14
```

Do mesmo modo, esses valores podem ser atribuídos a uma matriz:

```
LET a(1) = 12
LET a(2) = 5
LET a(3) = 7
LET a(4) = 22
LET a(5) = 14
```

O número entre parênteses é o índice ou subscripto (*subscrit*).

Para se formar uma matriz é necessário reservar no computador memória para ela, o que se faz através de DIM (DIMension), juntando-lhe a identificação das matrizes (uma letra para os valores numéricos e uma letra e \$ para as cadeias) e, entre parênteses, o número de valores ou cadeias a incluir na matriz.

Assim, no exemplo, bastaria fazer:

```
DIM a(5)
```

A introdução dos valores na matriz poderia ser feita com o seguinte programa:

```

10 DIM a(5)
20 FOR n = 1 TO 5
30 INPUT a(n)
40 NEXT n

```

A leitura dos valores de a(n) pode ser feita através de um simples comando directo, como:

```
PRINT a(4)
```

ou num passo posterior do programa. Por exemplo:

```
50 LET B = a(1) + a(2): PRINT B
```

Outro exemplo:

```

10 DIM a(10)
20 FOR n=1 TO 10
30 LET a(n)=n*n*n
40 PRINT n;"**";n;"**";n;"=""; a(n)
50 NEXT n

```

Depois, altere a linha 20 para:

```
20 FOR n=1 to 15
```

e veja o que acontece.

As matrizes podem ter duas ou mais dimensões, em teoria até 255, mas na prática isso é limitado pela capacidade do ecrã e da impressora e ainda pela memória ocupada, que cresce rapidamente. Essas características são de extrema utilidade para a elaboração de mapas, como se pode ver pelo exemplo seguinte:

```

10 DIM a(12,2)
20 PRINT AT 3,0;"MES RECEITA DESPESA"
30 FOR x=1 TO 12
40 PRINT TAB 0;x;
50 FOR y=1 TO 2
60 INPUT a(x,y)
70 PRINT TAB 7*y; a(x,y);
50 NEXT y
60 NEXT x

```

Os valores introduzidos alinhar-se-ão automaticamente em colunas de "Receitas" e "Despesas" para cada mês. Note-se o uso de TAB, abreviatura de TABulador, e que funciona exactamente como o tabulador de uma máquina de escrever. Os valores de TAB correspondem ao das colunas do ecrã, situando-se portanto entre 0 a 31. O valor de TAB pode ser fixo, como acontece na linha 40, ou determinado por uma expressão, como acontece na linha 70. Neste caso, se o valor calculado for decimal, o computador usará o inteiro imediatamente inferior.

A faculdade que as matrizes têm de reter os valores introduzidos nos seus elementos permite que o programa dado como exemplo seja refinado pela inserção do cálculo automático dos totais anuais. Junte a ele as seguintes linhas:

```

200 LET totr=0: LET totd=0
210 FOR x=1 to 12
220 LET pr=a(x,1)
230 LET pd=a(x,2)
240 LET totr=totr+pr
250 LET totd=totd+pd
260 NEXT x
270 PRINT: PRINT "_____ "
280 PRINT "TOTAL"; TAB 7; totr; TAB 14; totd

```

A linha 200 coloca a zero os totais de receitas (totr) e de despesas (totd). O ciclo FOR...NEXT iniciado na linha 210 determina a leitura em sucessão dos valores das parcelas mensais das receitas (pr) e das despesas (pd) e junta-as aos totais (totr) e (totd) nas linhas 240 e 250, respectivamente. Quando o ciclo chega ao fim, a linha 280 determina a impressão dos totais.

O programa pode ter ainda um outro refinamento. Como atrás se demonstrou, é possível realizar operações entre valores inseridos em matrizes. Portanto é possível calcular também as diferenças entre as receitas e as despesas. Feitas as alterações necessárias, teremos:

```

10 DIM a(12,2)
20 PRINT AT 3,0; "MES RECEITA DESPESA LUCRO"
30 FOR x=1 TO 12
40 PRINT TAB 0;x;
50 FOR y=1 TO 2
60 INPUT a(x,y)
70 LET pl=a(x,1)-a(x,2)
80 PRINT TAB 7*y; a(x,y): TAB 23;pl
90 NEXT y

```

```

100 NEXT x
200 LET totr=0: LET totd=0: LET totl=0
210 FOR x=1 TO 12
220 LET pr=a(x,1)
230 LET pd=a(x,2)
240 LET pl=pr-pd
250 LET totr = totr + pr
260 LET totd = totd + pd
270 LET totl + pl
280 NEXT x
290 PRINT: PRINT "_____ "
300 PRINT "TOTAL"; TAB 7; pr; TAB 14;pd; TAB 23;pl

```

É o princípio das folhas de cálculo, ou "spreadsheets".

Como se disse, as matrizes podem trabalhar com cadeias de caracteres. Mas no 48 BASIC, assim como no 128 BASIC e no + 3 BASIC têm uma particularidade importante faça:

```

10 DIM a$(5)
20 FOR n= 1 TO,5
30 INPUT a$(n)
40 PRINT a$(n)
50 NEXT n

```

Verificará que só é impressa uma letra das cadeias introduzidas. Substitua a primeira linha por:

```
10 DIM a$(5,3)
```

Faça correr de novo o programa e verificará que são impressas três letras, no máximo. Substitua o número 3 por 5 e verificará que são impressos até cinco letras, e assim por diante.

Introduza no programa a linha

```
45 PRINT LEN a$(n)
```

Verificará que o comprimento atribuído à cadeia é sempre o determinado pelo último índice, ainda que a cadeia tenha um número menor de caracteres.

O último índice pode também ser constituído por uma instrução de corte. Faça correr o programa depois de ter substituído a primeira linha por:

```
10 DIM a$(5,3 TO 8)
```

Esta possibilidade só existe no 48 BASIC, no 128 BASIC e no + 3 BASIC. Nos outros as cadeias de caracteres são tratadas como os valores numéricos.

Um exemplo de utilização de cadeias de caracteres é o conhecido jogo do carrasço (*hangman*). Eis um programa:

```

10 DIM p$(5,10)
20 FOR i=1 TO 5 : INPUT "Palavra a adivinhar=";p$(i) : NEXT i
30 FOR i=1 TO 5
40 LET g=0
50 LET t$=p$(i)
60 LET g=g+1
70 INPUT "Introduza uma letra =";a$
80 LET f=0
90 FOR j=1 TO 10
100 IF a$(1)=p$(i,j) THEN BEEP .05,30: PRINT "Acertou no ";INVERSE
; a$(1): LET p$(i,j)=" "
110 IF p$(i,j)<>" " THEN LET f=1
120 NEXT j
130 IF f=1 TEHN GO TO 60
140 BEEP .05,30: PRINT "A palavra era "; FLASH 1; t$
150 NEXT i
160 FOR n=1 to 20: BEEP .05. 30: NEXT n
170 PRINT: PRINT TAB 14; INVERSE 1; "FIM": PRINT: PRINT "Outro
jogo: S/N":
PAUSE 0
180 IF INKEYS <> "S" AND INKEY$ <> "s" OR INKEY$ <> "N" AND
INKEY "n" THEN GO TO 180
190 IF INKEY$="S" OR INKEY$="s" TEHN GO TO 10
200 IF INKEY$="N" OR INKEY$""n" THEN NEW

```

No jogo intervêm duas pessoas — a que introduz as palavras e a que as adivinha. Respeite o espaço entre aspas nas linhas 100 e 110. A instrução DIM p\$(5,10) permite a inserção de cinco palavras com um máximo de dez caracteres: note o espaço ocupado por INVERSE quando a palavra adivinhada é impressa.

Uma das vantagens do 48 BASIC, do 128 BASIC e do + 3 BASIC é a possibilidade de gravar os dados inseridos nas matrizes, como se descreve no capítulo A GRAVAÇÃO DE PROGRAMAS EM CÓDIGO E O REGISTO DE IMAGENS E DADOS.

Os caracteres

Como se disse, os Spectrum + 2 e + 3 utilizam 256 caracteres, cada qual com o seu número de código. Para conhecer esse número basta fazer:

```
PRINT CODE (caracter)
```

e para se conhecer o caracter correspondente a um determinado número de código basta fazer:

```
PRINT CHR$(número)
```

A lista dos caracteres que podem ser representados no ecrã obtém-se fazendo:

```
10 FOR a=32 TO 255: PRINT a;"="";CHR$ a: Next a
```

Caracteres de comando

Os primeiros 32 códigos de caracteres (de CHR\$0, a CHR\$ 31) são de comando. Podem ser muito úteis ao programador quando trabalha com cadeias, mas nem todos são directamente acessíveis: os caracteres CHR\$ 10 e CHR\$ 11, por exemplo, determinam a subida e a descida do cursor, mas fazer PRINT CHR\$ 10 ou CHR\$ 11 não tem efeito algum. O código CHR\$ 6 determina espaços tal como uma vírgula numa instrução Print, e CHR\$ 13 determina nova linha. Os códigos CHR\$ 8 e CHR\$ 9 permitem deslocar o cursor, respectivamente, para a esquerda (*backspace*) ou para a direita sem interromper um programa, como se pode comprovar fazendo:

```
10 LET A$= "RODA": PRINT AT 10,10. A$
20 PAUSE 50
30 LET A$=A$+CHR$+CHR8+"S": PRINT AT 10,10; A$
```

A relação entre esses caracteres e os comandos é a seguinte:

CÓDIGO	COMANDO
0 a 5	Não usados
6	PRINT vírgula
7	[EDIT]
8	Cursor esquerdo ←
9	Cursor direito →
10	Cursor abaixo ↓

11	Cursor acima ↑
12	[DELETE]
13	[ENTER]
14 e 15	Não usados
16	INK (comando)
17	PAPER (comando)
18	FLASH (comando)
19	BRIGHT (comando)
20	INVERSE (comando)
21	OVER (comando)
22	AT (comando)
23	TAB (comando)
24 a 31	Não usados

Caracteres alfanuméricos

Os caracteres CHR\$ 32 a CHR\$ 127 incluem os *alfanuméricos* — ou seja as letras do alfabeto, maiúsculas e minúsculas, e os algarismos — e ainda os sinais gráficos e de pontuação.

Caracteres gráficos

No modo Spectrum, o TC3256 dispõe de 16 caracteres gráficos, a que correspondem os códigos CHR\$ 128 a CHR\$ 143, e que são constituídos pelas combinações possíveis de blocos de 4 x 4 pontos, donde a designação, que também lhes é dada, de “caracteres de bloco” (*block graphics*). Um exemplo simples do que pode ser feito com esses caracteres é a representação simplificada de um automóvel, segundo o seguinte programa:

```
10 PRINT CHR$ 132 + CHR 140
20 PRINT CHR$ 139 + CHR$ 143 + CHR$ 139 + CHR$ 138
```

Os caracteres gráficos são acessíveis através das teclas 1 a 8 do teclado principal, depois de se premir a tecla GRAPH — as correspondentes aos códigos CHR\$ 128 a CHR\$ 135 quando a tecla de TRUEVIDEO está premida, e os dos códigos CHR\$ 136 a CHR\$ 143, que são a imagem em negativo daqueles, quando a tecla premida é a de INVERSE VIDEO.

Caracteres definidos pelo utilizador

Se se fizer:

```
10 FOR n= 144 a 164
20 PRINT CHR$ (n)
30 NEXT n
```

Ter-se-ão de novo as letras A a U, que correspondem a 21 caracteres gráficos que podem ser definidos pelo utilizador (*user definable graphics* ou UDG). É uma tarefa muito simples no 48 BASIC, tal como no 128 e no + 3 BASIC, pois pode-se dispôr das funções BIN e USR.

Um meio prático de definir um caracter consiste em tomar uma folha de papel quadriculado na qual se marca um quadrado com oito quadradinhos de lado, ou sejam sessenta e quatro no total, correspondendo aos pontos ocupados por um caracter em cada uma das 32 colunas e das 22 linhas do ecrã. Enchem-se a negro os quadradinhos necessários para formar o caracter que se pretende definir.

Traça-se depois um quadrado semelhante, mas no qual, linha por linha, se preenchem os quadradinhos em branco com um “0”, e os quadradinhos a negro com um “1”.

A escolha do “0” e do “1” não é feita por acaso. A “linguagem” usada realmente pelos computadores é binária — isto é: usa apenas dois caracteres: 0 e 1, aos quais correspondem os dois estados possíveis de cada um dos seus circuitos básicos: aberto e inactivo ou fechado e activo.

Sendo assim, se num lugar conveniente da memória do computador — no endereço de um caracter cujo o código se situe entre CHR\$ 144 e CHR\$ 164 — forem introduzidos os “0” e os “1” presentes em cada uma das oito linhas, o caracter assim definido irá substituir o que ali se encontrava.

Para introduzir valores na memória de um computador recorre-se à instrução POKE (*introduzir*), a que se tem acesso premindo (O), e à qual se adicionam dois números inteiros: o do endereço, que no modo Spectrum pode situar-se entre 16384 e 65535, e o do “conteúdo” desse endereço, que corresponde a um valor entre 0 e 255.

O valor atribuído ao conteúdo do endereço é o correspondente, na base decimal, a um número binário com oito algarismos. A função BIN, acessível através de EXTENDED e (B), converte os números binários em decimais. Assim, para introduzir, no endereço da letra “a” a primeira linha de um caracter definido pelo utilizador, programar-se-ia:

```
POKE USR “a” + (número da linha), BIN (número binário)
```

Note-se que é indiferente usar letras maiúsculas ou minúsculas, mas que estas são preferíveis porque dispensam a necessidade de premir CAPS SHIFT para utilizar os caracteres definidos.

USR — *Sub-rotina para o utilizador (User SubRoutine)* — é uma função acessível através de EXTENDED e (L), e (no caso) abre ao utilizador os endereços dos caracteres definíveis. Fazendo:

```
PRINT USR "a"
```

ter-se-á por resposta o número 23675, que é o do endereço do primeiro carácter gráfico definível.

Para definir um carácter faça, por exemplo:

```
10 POKE USR "a", BIN 00110000
20 POKE USR "a"+1,BIN 01110000
30 POKE USR "a"+2,BIN 01111110
40 POKE USR "a"+3,BIN 01011111
50 POKE USR "a"+4,BIN 10011111
70 POKE USR "a"+5,BIN 00010010
80 POKE USR "a"+6,BIN 00010010
90 POKE USR "a"+7,BIN 00000000
```

Prima GRAPH e (a) e no ecrã aparecerá um pequeno elefante.

Outro exemplo, apropriado aos programas utilitários:

```
10 POKE USR "b", BIN 00000000
20 POKE USR "b"+1,BIN 01111110
30 POKE USR "b"+2,BIN 00100100
40 POKE USR "b"+3,BIN 00100100
50 POKE USR "b"+4,BIN 00100100
70 POKE USR "b"+5,BIN 00100100
80 POKE USR "b"+6,BIN 00100100
90 POKE USR "b"+7,BIN 00000000
```

Prima GRAPH e (b) e no ecrã aparecerá a letra grega r (PI).

Pode-se também definir caracteres portugueses. Como o Ç, por exemplo:

```
10 POKE USR "c" BIN 00000000
20 POKE USR "c"+1,BIN 00111100
30 POKE USR "c"+2,BIN 01000010
40 POKE USR "c"+3,BIN 01000000
50 POKE USR "c"+4,BIN 01000000
```

```
70 POKE USR "c"+5,BIN 01000010
80 POKE USR "c"+6,BIN 00111100
90 POKE USR "c"+7,BIN 00001000
```

A definição pode ainda ser feita mais facilmente se em vez dos números binários forem usados números decimais. Ainda que, como se disse, a conversão possa ser feita através da função BIN; será bom saber, que o "1" binário correspondente na base decimal a:

$2^0 = 1$ quando na 1ª casa (da direita)
 $2^1 = 2$ quando na 2ª casa
 $2^2 = 4$ quando na 3ª casa
 $2^3 = 8$ quando na 4ª casa
 $2^4 = 16$ quando na 5ª casa
 $2^5 = 32$ quando na 6ª casa
 $2^6 = 64$ quando na 7ª casa
 $2^7 = 128$ quando na 8ª casa

Em referência ao caso da definição do Ç, ter-se-á linha a linha:

```
00000000 = 0
00111100 = 32 + 16 + 8 + 4 = 60
01000010 = 64 + 2 = 66
01000000 = 64
01000000 = 64
01000010 = 64 + 2 = 66
00111100 = 32 + 16 + 8 + 4 = 60
00001000 = 8
```

O programa poderia assim ser reduzido a:

```
10 FOR n=0 TO 7
20 READ a
30 POKE USR "ç" + n, a
40 NEXT n
50 DATA 0, 60, 66, 64, 64, 66, 60, 8
```

Note que pode colocar em qualquer local do ecrã o carácter definido, referindo-o como CHR\$ 146 (porque CHR\$ 144 corresponde a A, CHR\$ 145 a B e CHR\$ 146 a C, etc.), ou simplesmente

colocando-o entre aspas e juntando-o a uma cadeia ou inserindo-a nela. Junte no programa anterior a seguinte linha e experimente:

```
60 PRINT AT 10,13; "CABE" + CHR$ 146 + "A"
```

ou

```
60 PRINT AT 10,13; "CABE" + "Ç" + "A"
```

Os Spectrum + 2 e + 3 podem, conseqüentemente, apresentar texto com caracteres portugueses, mas deve-se notar que esses caracteres só poderão ser impressos por cópia do ecrã, através do comando COPY

Caracteres de "palavras"

Os caracteres CHR\$ 165 a CHR\$ 255, correspondem às "palavras-chave" usadas no 48 BASIC — tais como PRINT, LOAD, SAVE, etc., que são tratadas como um só caracter. Exceptuam-se os caracteres de CHR\$ 199 a CHR\$ 201 que correspondem a <=, => e <> .

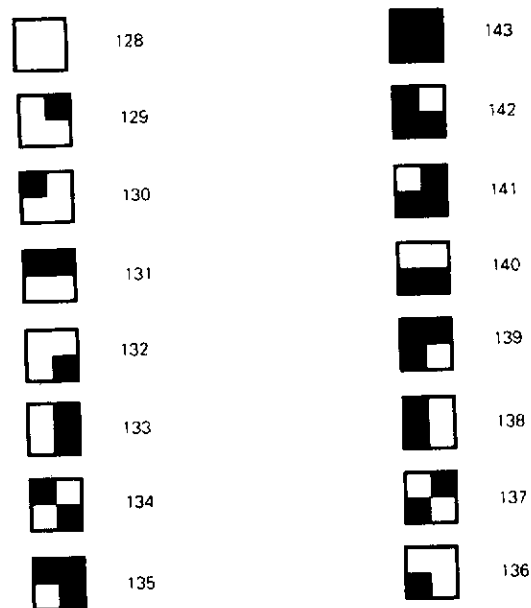


Fig. 18 — OS CARACTERES GRÁFICOS E OS SEUS CÓDIGOS

Listas de Dados

"Read", "Data" e "Restore"

Em muitos casos, os programas, durante o seu curso, necessitam de dados para além dos atribuídos de início às variáveis. Esses dados podem ser pedidos ao utilizador à medida que vão sendo necessários ou podem ter sido incluídos previamente no programa. O primeiro caso corresponde à instrução INPUT, já conhecida, perante a qual o programa pára à espera de lhe ser introduzida a informação pedida. Note que quando a instrução é do tipo INPUT a\$ e portanto pede um caracter ou uma cadeia de caracteres, o cursor surge entre aspas, mas se ela for do tipo INPUT a e portanto pedir um valor numérico, o cursor surge sem aspas. Se se introduzir uma letra ou letras quando o programa pede um número, o programa parará com a mensagem:

2 variable not found

Pode-se também usar a instrução INPUT para deter o programa enquanto o utilizador lê uma página de texto, por exemplo. Basta introduzir no lugar conveniente a linha:

```
INPUT "PRIMA ENTER PARA CONTINUAR"; a$
```

O programa pára e só continua quando se premir ENTER. A cadeia a\$ ficará vazia, uma vez que não se lhe atribuiu qualquer valor.

A inclusão prévia de dados num programa para que este proceda em tempo à sua leitura é conseguida através das instruções READ, DATA e RESTORE. O programa seguinte demonstra o uso de READ e DATA:

```
10 REM Capitais
20 READ a$
30 PRINT "QUAL A CAPITAL DA ";a$;" ?"
40 READ b$
50 INPUT c$
60 IF c$=b$ THEN PRINT "CERTO !"
70 IF c$ <> b$ THEN PRINT "ERRADO ! A CAPITAL DA ";a$" E ";b$
80 GOTO 20
90 DATA
"HOLANDA", "HAIA", "FRANÇA", "PARIS", "BELGICA",
"BRUXELAS", "ESPANHA", "MADRID", "R.F.A.", "BONA",
"URSS", "MOSCOVO", "INGLATERRA", "LONDRES"
```

As instruções READ (*ler*), em sucessão, permitem ler na mesma ordem os dados contidos em DATA sob a forma de cadeias de caracteres. Quando os dados acabarem, o programa pára. Mas nada impede que outros dados sejam adicionados, quer na mesma linha, quer noutras. Por exemplo:

```
100 DATA "DINAMARCA", "COPENHAGA", "GRECIA",
    "ATENAS", "NORUEGA", "OSLO"
110 DATA "POLONIA", "VARSÓVIA", "CHINA", "PEQUIM",
    "TURQUIA", "ANKARA"
```

Como já se disse, quando se pretende parar um programa que pede dados sob a forma de cadeias, há que apagar com DELETE as aspas que ladeiam o cursor antes de fazer STOP. De outro modo o computador interpreta a palavra STOP como uma cadeia inserida em resposta ao pedido de INPUT. Para evitar essa necessidade poder-se-ia inserir no programa anterior a linha:

```
75 IF c$="STOP" THEN STOP
```

Assim, como é óbvio, se se premir a tecla STOP, o programa pára.

Quando a leitura dos dados chega ao fim, se houver necessidade de a repetir importa fazer com que a instrução READ volte ao ponto de partida. Se isso não for feito, o computador tentará continuar a leitura dos dados e não os encontrará, pois que ultrapassou o fim da lista de DATA, e então surgirá no ecrã a mensagem:

E Out of data

Para fazer voltar READ ao ponto de partida de DATA ou se faz RUN ou se introduz o comando directo RESTORE, depois do que se faz GOTO 1.

Se os dados tiverem de ser de novo utilizados dentro do programa haverá que colocar a instrução RESTORE numa linha anterior ao início da nova leitura.

As linhas de DATA podem ser colocadas em qualquer altura do programa — no princípio, no meio ou no fim, mas se o comando RESTORE for acompanhado de um número de linha, então a leitura iniciar-se-á nessa linha. Se se introduzir o comando directo RESTORE 100, ou se a linha 15 for alterada em conformidade, a leitura iniciar-se-á por "DINAMARCA" (linha 100), e não por "HOLANDA" (linha 90).

Os dados incluídos em DATA tanto podem ser apresentados em cadeias como sob forma de valores numéricos. Neste caso a leitura faz-se com READ a em vez de READ a\$, tal como se faz para as variáveis. Nada impede que dados numéricos e de cadeia se juntem numa mesma linha de DATA. Um exemplo:

```
10 REM Meses
20 FOR m=1 TO 12
30 READ a$
40 READ a
50 PRINT " O MES DE "; a$," TEM "; a; " DIAS"
60 DATA "JANEIRO",31, "FEVEREIRO",28, "MARCO",31,
    "ABRIL",30, "MAIO",31, "JUNHO",30, "JULHO",31, "AGOSTO", 31,
    "SETEMBRO",30, "OUTUBRO",31, "NOVEMBRO",30,
    "DEZEMBRO",31.
```

Note que a ordem de leitura foi determinada por um ciclo FOR__NEXT — o caso mais usual — é que a linha 40 poderia desaparecer se a linha 30 passasse a ser:

```
30 READ a$,a
```

As Cadeias de Caracteres

No BASIC uma série de caracteres (alfabéticos, numéricos ou gráficos) contida entre aspas, constitui, uma cadeia (*string*) e pode constituir uma variável, sendo então designada por uma letra seguida por um cifrão. (Na realidade, o \$, nesse caso, não deve ser lido como "cifrão", mas sim como "string". Portanto, quando alguém se refere, por exemplo, à variável a\$, não deve dizer "a-cifrão", mas sim "a-string").

O uso das aspas para delimitar uma cadeia pode causar problemas quando dentro dela se pretende colocar uma frase também entre aspas. Se tentar introduzir no computador uma cadeia como a seguinte:

```
10 LET a$="Escreva 'NOME' do lado direito" : PRINT a$
```

A linha será rejeitada.

Muitos programadores usam por isso, a forma:

```
10 LET a$="Escreva 'NOME' do lado direito" : PRINT a$
```

No entanto a solução correcta é o uso de aspas dobradas:

```
10 LET a$="Escreva ""'NOME'"" do lado direito" : PRINT a$
```

Um dos principais pontos de interesse do uso das cadeias é a possibilidade de o computador trabalhar com textos e poder até dialogar com o utilizador. Alguns exemplos já foram dados, mas eis mais um:

```
10 REM Dialogo
20 INPUT "Como te chamas ?"; a$
30 PRINT "Muito prazer, ";a$;". Eu sou um Spectrum".
```

É possível adicionar (concatenar) cadeias, como se poderá ver pelo seguinte programa:

```
10 LET a$= "Primeiro"
20 LET b$= "—"
30 LET c$= "sargento"
40 LET d$= a$+b$+c$
50 PRINT d$
```

Supor-se-ia que também seria possível subtrair cadeias — isto é, fazer algo como

```
LET e$= d$ - b$
```

No entanto o computador rejeita uma linha como essa, uma vez que não é de aplicação geral. O tratamento das cadeias é feito por outro processo, que na língua inglesa se designa por "slicing" — literalmente "corte em fatias". O método de "corte" usado no BASIC Sinclair é muito mais simples que o usual. Consiste na indicação dos números de ordem dos caracteres em que o "corte" começa e acaba. A forma geral é:

cadeia de caracteres (começo TO fim)

de modo que, numa cadeia:

```
a$ = "abcdef"
```

se se fizer:

```
a$(2 TO 5)
```

Ter-se-á:

```
a$(2 to 5) = "bcde"
```

Quando se omite o começo, pressupõe-se que ele corresponde a 1:

```
"abcdef"(TO 5) corresponde a "abcdef"(1 TO 5) e logo a "abcde"
```

Quando se omite o fim, pressupõe-se que o "corte" vai até ao fim da cadeia:

```
"abcdef"(2 TO) corresponde a "abcdef"(2 TO 6) e logo a "bcdef"
```

Quando se omite o princípio e o fim, pressupõe-se que toda a cadeia é tomada:

```
"absdef" ( TO ) ou "abcdef" ( ) corresponde a "abcdef"(1 TO 6) e logo a "abcdef".
```

Do mesmo modo:

```
"abcdef"(3) é o mesmo que "abcdef"(3 TO 3) e logo "c"
```

Se o fim do "corte" for além do fim da cadeia, como em:

```
“abcdef”(5 TO 7)
```

o resultado será a mensagem de erro:

3 subscript wrong

Mas se o início corresponder a um número, qualquer que ele seja, maior que o do fim, como em:

```
“abcdef”(8 TO 7), ou  
“abcdef”(1 TO 0)
```

o resultado será uma "subcadeia" vazia.

No entanto não podem ser usados números negativos — o resultado é a mensagem de erro:

B integer out of range

As operações com cadeias de caracteres podem também ser condicionadas por parenteses. Assim, se se tiver:

```
“abc” + “def” (1 TO 2)
```

o resultado será "abcde", ou seja uma subcadeia que começa no caracter 1 e acaba no caracter 2 da segunda e última cadeia. Mas se se fizer:

```
(“abc” + “def”) (1 TO 2)
```

o resultado será "ab", ou sejam os dois primeiros caracteres, pois que as duas cadeias passam a ser tomadas como uma só.

Para uma melhor compreensão da utilidade do tratamento das cadeias de caracteres, experimente o seguinte programa:

```
10 LET a$ = "JanFevMarAbrMaiJunJulAgoSetOutNovDez"  
20 FOR n=1 TO 12  
30 PRINT a$(3*(n TO (n+3))  
40 NEXT n
```

Conhecer o comprimento de uma cadeia de caracteres pode ser muito útil. Para isso usa-se a função LEN (de LENght, *comprimento*). Assim, fazendo-se:

```
10 INPUT a$  
20 PRINT AT 10,((31-LEN a$)/2); a$
```

a cadeia a\$ será impressa a meio do ecrã. Experimente:

```
10 INPUT "Escreva uma mensagem"; a$  
20 LET S=50  
30 FOR b=1 TO LEN A$+33  
40 PRINT AT 5,0;("marcar 32 espaços" + a$ + " marcar 32 espaços")  
(B TO B+31)  
50 LET S=S-(5 AND INKEY$="R" AND S >1) + (5 AND  
INKEY$="D")  
60 IF INKEYS$="P" THEN GOSUB 110  
70 FOR a=1 TO S  
80 NEXT a  
90 NEXT b  
100 GOTO 20  
110 FOR a=1 TO 200 : NEXT a  
120 RETURN
```

A mensagem correrá através do ecrã como um letreiro luminoso. Premindo a tecla [R] depois de ter feito CAPS LOCK a mensagem correrá mais depressa. Premindo [D] correrá mais devagar. Premindo [P] parará. O programa pode servir de base a jogos: veja a pág. 96.

A Organização do Ecrã

Nas páginas 61 a 64 descreveu-se de uma maneira elementar como imprimir caracteres no ecrã. Vejamos agora os pormenores que facilitam essa impressão ou conduzem a uma melhor organização do ecrã.

Limpeza do ecrã

Já sabe que fazendo CLS, o ecrã fica limpo: todos os caracteres ou gráficos presentes nele desaparecem. Mas o mesmo pode acontecer com outros comandos. Escreva qualquer coisa no ecrã — mas não um programa — e depois faça RUN. Note que o ecrã fica limpo como se fizesse CLS. Repita, mas em vez de RUN faça CLEAR. O ecrã fica também limpo. CLEAR (tal como RUN) não tem por principal função limpar o ecrã, mas convém tomar em atenção esse efeito quando se elabora um programa, para que não se perca o que naquele está a ser apresentado.

Sinais de pontuação

; — depois de INPUT, ou PRINT, ou de uma variável, ou uma cadeia de caracteres, o **ponto** e **vírgula** determina que a ela fique unido o carácter ou a cadeia de caracteres cuja impressão se siga.

Faça:

```
10 PRINT AT 10,10: "SPECTRUM + 2";
20 PRINT "SPECTRUM + 3"
```

Compare com:

```
10 PRINT AT 10,10: "SPECTRUM + 2"
20 PRINT "SPECTRUM + 3"
```

, — depois de PRINT, a **vírgula** determina que o carácter ou a cadeia de caracteres que se sigam a ela sejam impressas na mesma linha mas a partir da coluna 16, ou na linha seguinte se aquele espaço já estiver ocupado.

Experimente:

```
10 PRINT AT 0,10; "SPECTRUM + 2", "SPECTRUM + 3"
```

e

```
10 PRINT AT 10,10; "SPECTRUM + 2", "SPECTRUM + 3"
```

' — depois de PRINT, o **apóstrofo** determina a passagem à linha seguinte.

Experimente:

```
10 PRINT "SPECTRUM + 2" ' "SPECTRUM + 3"
```

Varie o número de apóstrofos e note o resultado. É possível reunir vírgulas e apóstrofos. Faça:

```
10 PRINT "SPECTRUM + 2" , ' "SPECTRUM + 3"
```

: — os **dois pontos** permitem colocar várias instruções numa mesma linha, como em:

```
10 LET a=3*2 : PRINT a
```

Impressão nas últimas linhas do ecrã

Em princípio não é possível utilizar as duas últimas linhas do ecrã — a 22 e a 23 — pois são reservadas aos comandos, aos dados de INPUT e às mensagens. Por isso, considera-se em regra que a "última linha do ecrã" é a 21. Mas na prática, quando o programador necessita de espaço para uma mensagem e não pode, ou não deseja, colocá-la na parte normalmente utilizável do ecrã, bastará que junte a PRINT o sinal #, seguido de 0 ou 1, conforme se pretende escrever na linha 22 ou 23. No entanto, logo que o computador precisar desse espaço, irá utilizá-lo e apagará o que ali tiver sido impresso. Uma das aplicações mais usuais da impressão, nas linhas 22 e 23 é a paragem do programa para a leitura de instruções, através de subrotinas como a seguinte:

```
1000 PRINT # 0; "PARA CONTINUAR"
1010 PRINT # 1; "PRIMA QUALQUER TECLA"
1020 PAUSE 0
```

é acessível através de SYMBOL SHIFT e [3]. A linha 1020 determina a paragem do programa até que uma tecla seja premida. Faça RUN. Depois prima a tecla e veja o que acontece. Experimente também usar outros números, em vez de 0 e 1, a seguir a PRINT #.

Funções

AT e SCREEN\$ — Recordemos que a instrução AT é usada para imprimir um carácter numa determinada coluna e numa determinada linha, ou uma cadeia de caracteres a partir dessa localização no ecrã. A função SCREEN\$ é a inversa de PRINT AT. Indica (com algumas limitações) o carácter que se encontra num local do ecrã. Por exemplo: se na coluna 10, linha 5, se tiver o carácter "A", fazendo-se PRINT SCREEN\$(10,5) ter-se-á como resultado a letra A. Mas se o carácter for gráfico, ou tiver sido definido pelo utilizador, ou criado por sobreposição através

de OVER, ou se o espaço correspondente, de 8 x 8 pontos, for ocupado por uma linha traçada por PLOT, DRAW ou CIRCLE, a função considerá-lo-á vazio.

TAB — (de TABulador) actua exactamente como os tabuladores das máquinas de escrever. É acessível através de EXTENDED e [P].

Enquanto AT serve para alinhar a coluna e a linha em que a impressão se deve iniciar, TAB só pode indicar a coluna, pelo que só é acompanhada por um número, de 0 a 31. Mas isso não significa que seja menos útil — é particularmente interessante para a elaboração de tabelas e mapas. Um exemplo:

```
10 PRINT: PRINT TAB 6; "Mes"; TAB 18; "Receita"; PRINT
20 FOR N=1 TO 12
30 INPUT "Receitas="; r
40 PRINT TAB 7; n; TAB 18; r
50 NEXT n
```

Para substituir o número de ordem dos meses pelo respectivo nome, bastaria adicionar as linhas:

```
35 READ n$
60 DATA "Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago",
"Set", "Out", "Nov", "Dez".
```

e substituir na linha 40 a variável n por n\$.

Note que o alinhamento é sempre feito a partir da esquerda — da coluna indicada por TAB. Alguns BASIC permitem proceder directamente ao alinhamento pela direita, mas no 48 BASIC, tal como no 128 BASIC e no + 3 BASIC, há que transformar numa cadeia de caracteres o valor numérico a imprimir, através da função STR\$. Depois mede-se o seu comprimento através da função LEN e deduz-se o número restante do da coluna pela qual se pretende fazer alinhamento. Para isso, basta que o programa anterior tome a seguinte forma:

```
10 PRINT: PRINT TAB 6; "Mes"; TAB 18; "Receita"; PRINT
20 FOR N=1 TO 12
30 INPUT "Receitas="; r
40 LET i a = r * 100
50 LET i b = INT (r)
60 LET i c = i a - i b * 100
70 LET b$ = STR$ i b
80 LET c$ = STR$ i c
```

```
90 IF c$="0" THEN LET c$="00"
100 LET t = 25 - LEN b$
110 PRINT TAB 7; n; TAB t; "$" + c$
120 NEXT n
```

A linha 40 reduz a centavos a quantia introduzida pela linha 30.

A linha 50 separa os escudos — a parte inteira da quantia, a colocar à esquerda do cifrão.

A linha 60 determina os centavos — a parte da quantia a colocar à direita do cifrão.

O número 25, na linha 100, é o da coluna pelo qual, no exemplo, se faz o alinhamento das quantias.

Se em vez de quantias se tiverem números decimais, o processo é semelhante. Experimente:

```
10 INPUT "Quantos números na lista?"; q
20 FOR n=1 TO q
30 INPUT a
40 LET i = INT a
50 LET a$ = STR$ i
60 LET c = LEN a$
70 IF a$ = "0" THEN LET c = 0
80 LET t = 20 - c
90 PRINT TAB t; a
100 NEXT n
```

A linha 70 explica-se pelo facto de a função LEN considerar o "0" à esquerda do ponto decimal, num número inferior à unidade, como uma cadeia com um carácter de comprimento, embora não o imprima.

INPUT com AT e LINE

Por vezes — nomeadamente para evitar qualquer confusão na atribuição de valores às variáveis — não convém que INPUT surja como é de normal nas linhas 22 e 23, pois aí seria apagado pelo INPUT seguinte. A solução é a de usar INPUT em conjunto com AT. Experimente:

```
10 INPUT AT 6,2; "NOME:"; a$
20 INPUT AT 8,2; "MORADA:"; b$
30 INPUT AT 10,2; "CODIGO:"; c$
40 INPUT AT 12,2; "TELEFONE:"; d$
```


INPUT pode também ser usado com LINE, que é acessível através de EXTENDED, SYMBOL SHIFT e [3]. recorde-se que quando através de INPUT o programa solicita a introdução de uma cadeia de caracteres, o cursor aparece entre aspas. Como se disse, apagando as aspas pode-se interromper o programa e introduzir um comando directo. Nos programas educativos isso permite fraudes. Usando

INPUT LINE a\$

o cursor não aparece entre aspas e a interrupção e a fraude não serão possíveis. Além disso, se por engano se introduzir um valor numérico em vez de uma cadeia, o computador não o aceitará. Se esta particularidade for inconveniente, pode ser eliminada usando VAL\$ como se indica na página 52.

OVER, INVERSE, INVERSE VIDEO, FLASH, BRIGHT, INK 8 e 9

OVER regula a impressão de caracteres em sobreposição. Com OVER 0 os novos caracteres apagam os anteriores. Fazendo OVER 1 os novos caracteres sobrepõem-se aos anteriores. Desse modo podem-se obter muitos efeitos curiosos e alguns práticos. Um destes é o sublinhado de palavras ou frases. Faça:

```
10 PRINT AT 10,10; "SUBLINHADO"
20 PRINT AT 10,10; OVER 1; "_____"
```

INVERSE determina a inversão das cores de PAPER e INK. Com INVERSE 1 dá-se a inversão. Com INVERSE 0 volta-se à situação inicial. Consequentemente o texto entre as instruções de INVERSE 1 e INVERSE 0 é impresso em negativo no écran. Note-se que com PLOT, DRAW e CIRCLE, o uso de INVERSE 1 tem um efeito diferente: os pontos, linhas, arcos e círculos traçados a seguir a essa instrução tomam a cor do fundo e portanto desaparecem.

INVERSE VÍDEO tem uma acção semelhante a INVERSE. Pode ser usado nas listagens para destacar alguns dos seus passos, mas não pode ser inserido directamente na programação. A sua maior utilidade é a de permitir usar os caracteres gráficos em negativo.

FLASH permite alterar as cores de PAPER e INK, tal como INVERSE, mas numa sucessão contínua muito rápida. FLASH 1 inicia a alternância, FLASH 0 põe-lhe termo.

BRIGHT permite destacar o texto sobre o fundo, tornando ambos mais luminosos. BRIGHT 1 determina que os caracteres seguintes a PRINT ou INPUT sejam impressos no écran em tons de INK e PAPER mais brilhantes. BRIGHT 8 determina que os caracteres brilhantes sejam tam-

bém brilhantes, mas que os impressos nas posições dos caracteres normais continuem a ser normais. BRIGHT 0 faz cessar os efeitos de BRIGHT 1 e BRIGHT 8. Experimente:

```
10 PRINT AT 6,14; "VEJA"; AT 7,10; "COMO ACTUA O";
   BRIGHT 1; AT 8,11; "BRIGHT 1"
20 PAUSE 100, BRIGHT 8
30 PRINT AT 6,14; " "; AT 7,10; "E AGORA O "; AT 8,11;
   "BRIGHT 8"
40 PAUSE 50
50 PRINT AT 18,2; FLASH 1; "MUDA O TEXTO"
60 PRINT AT 19,2; FLASH 0; "CONSERVA O BRIGHT"
```

INK 8 corresponde a "transparente" — isto é, as cores mantêm-se inalteradas no ecrã, onde quer que se faça a impressão com INK 8.

INK 9 determina que INK seja branco ou preto, conforme o fundo, de maneira a destacar o que for impresso.

Experimente:

```
10 PAPER 6: PRINT AT 10, 10; INK 7; "SPECTRUM"
20 PAUSE 50
30 PRINT AT 10,10; INK 8; "      "INK 8"      "
40 PAUSE 50
50 PRINT AT 12,10; INK 9; "      "INK 9"      "
```

Os usos de CHR\$

A função CHR\$ pode ser usada para organizar o ecrã, ainda que em regra isso não seja prático.

```
CHR$ 22 + CHR$ a + CHR$ b corresponde a PRINT AT a, b
CHR$ 23 + CHR$ a + CHR$ b corresponde a PRINT TAB
                               a + 256 * b
CHR$ 16 + CHR$ a               corresponde a INK a
CHR$ 17 + CHR$ a               corresponde a PAPER a
CHR$ 18 + CHR$ 0 ou 1         corresponde a FLASH 0 ou 1
CHR$ 19 + CHR$ 0, 1 ou 8     corresponde a BRIGHT 0, 1 ou 8
CHR$ 20 + CHR$ 0 ou 1         corresponde a INVERSE 0 ou 1
CHR$ 21 + CHR$ 0 ou 1         corresponde a OVER 0 ou 1
```

A Lógica

Entre as palavras-chave usadas no 48 BASIC, assim como no 128 BASIC e no+ 3 BASIC, figuram os operadores lógicos AND, OR e NOT, assim chamados porque exprimem a lógica numa instrução ou comando. Podem também ser usados como funções.

AND, quando usado como um operador lógico tem por fim pôr à prova a verdade de uma combinação de condições. Se todas as condições forem cumpridas, a instrução será cumprida. Por exemplo:

```
10 FOR n=1 TO 10
20 INPUT "S/N"; s$
30 IF s$ = "S" AND n>5 THEN PRINT n
40 NEXT n
```

Os sucessivos valores da variável *n* só serão impressos se e quando *n* for maior que 4 e em resposta ao pedido de INPUT de introduzir um S maiúsculo.

Quando AND é usado como uma função, como em:

```
LET x = y AND z
```

O resultado é o valor de *y* se *z* não for igual a zero e será 0 se for igual a zero.

AND pode também ser usado como função em conjunto com cadeias de caracteres, mas o valor que se lhe segue deve ser numérico. Como em:

```
LET a$ = b$ AND c
```

O resultado só será igual a *b\$* se *c* não for 0 e será uma cadeia vazia (" ") se *c* for zero.

O 48 BASIC, tal como o 128 BASIC e o + 3 BASIC, não avalia as condições dos valores numéricos segundo as "tabelas de verdade" normais. A função AND atribui um valor 1 a uma condição verdadeira e um valor 0 a uma condição falsa, considera verdadeiro um valor não-zero e falso um valor zero. Na prática, e em suma:

x AND *y* significa *será x se y for verdade*

Note que o uso de AND como um operador lógico se faz em conjunto com o comando IF enquanto o uso de AND como função se faz em conjunto com LET.

OR obtém-se através de SYMBOL SHIFT e (U). Quando usado como um operador lógico tem por fim pôr à prova a verdade de uma combinação de condições. Se uma ou mais condições forem cumpridas, a instrução será cumprida. Por exemplo:

```
IF INKEY$ = "N" OR INKEY$ = "n" THEN STOP
```

Quer esteja a funcionar CAPS SHIFT ou CAPS LOCK, quer não, o programa parará, pois que uma das condições apresentadas como alternativa foi cumprida.

OR também pode ser usado como função. Nesse caso, e tal como AND, atribui um valor 1 a uma condição verdadeira e um valor 0 a uma condição falsa, considera verdadeiro um valor não-zero e falso um valor zero. Assim, se se pretendesse, por exemplo, estabelecer uma tabela de preços de bilhetes com 50 % de desconto para as crianças de menos de 10 anos, pelos processos normais de programação ter-se-ia:

```
10 INPUT "preço="; p ; " escudos"
20 INPUT "idade="; i
30 IF i<10 THEN PRINT "Paga "; .5 * p; " escudos"
40 IF i=>10 THEN PRINT "Paga "; p ; " escudos"
```

Usando OR como função ter-se-ia:

```
10 INPUT "preço="; p ; " escudos"
20 INPUT "idade="; i
30 PRINT "Paga "; p * (.5 OR i=>10); " escudos"
```

Se a idade for menor que 10 anos, a condição *i=>10* será falsa de modo que o preço é multiplicado por 0.5. Se a idade for igual ou superior a 10 anos então a condição será verdadeira e o preço será multiplicado por 1. A linha 30 prevê as duas alternativas, dispensando a linha 40 do programa anterior.

Na prática e em suma:

x OR *y* significa *será x a menos que y seja verdade*

Isto é, como função, OR não exprime alternativa, mas sim exclusão.

NOT pode também ser usado como um operador lógico ou uma função. No primeiro caso, e tal como AND, deve ser precedido pelo comando IF, invertendo a verdade de uma condição — isto é: tornando o verdadeiro em falso ou vice-versa.

Por exemplo:

```
10 INPUT y
20 INPUT z
30 PRINT y ; "+" ; z ; " = ?"
```

```
40 INPUT x
50 IF NOT x = y + z THEN GOTO 40
```

Enquanto o valor introduzido como sendo a soma de x e y não for verdadeiro, o programa exigirá que ele seja novamente introduzido.

Se NOT for seguido por uma condição verdadeira, designa para esta o valor de 1; se a condição for falsa, designa para ela o valor de 0. Nesse caso NOT actua como uma função, invertendo o valor produzido e permitindo que o inverso da condição seja comprovado.

Se a condição contiver as funções AND ou OR estas devem ser colocadas entre parênteses. Se NOT for seguido por um valor numérico que não seja zero, o valor atribuído é de 0, se for zero, o valor atribuído será 1.

Tempo e Movimento

O comando PAUSE, determina uma interrupção na execução do programa. Essa interrupção é medida em 1/50 de segundo nos países que dispõem de corrente alterna de 50 Hz e em 1/60 quando ela é de 60 Hz. A instrução correspondente é:

```
PAUSE n
```

Em que n pode ter um valor entre 0 e 65555. Se n=0 então o programa não continua enquanto não for premida uma tecla.

Teoricamente poder-se-ia estabelecer um programa como:

```
10 FOR n=1 TO 36000
20 PAUSE 49
30 PRINT AT 0,0; "    ";AT 0,0; n
40 NEXT n
```

e assim se teria um conta-segundos. Mas na prática há que contar com o tempo dispendido na execução do programa. O erro, ao fim de 3600 segundos, ou seja de uma hora, seria sensível. A solução seria substituir PAUSE 50 na linha 20 por PAUSE 49 ou outro número que na prática conduziisse à mínima diferença. Mas há uma maneira mais correcta de contar o tempo por intermédio do computador.

Na memória do computador, o contador do número de "quadros" enviados para o televisor ou monitor ocupa os endereços 23672, 23673 e 23674. Fazendo:

```
10 POKE 23672,255
20 POKE 23673,255
30 POKE 23673,255
40 LET t = (65536 * PEEK 23674 + PEEK 23672 + 256 * PEEK 23673
+ PEEK 23672)/50
50 LET t = INT (t *10)/10
60 PRINT AT 10,14; t
70 GOTO 40
```

Ter-se-ão os segundos decorridos desde que se fez RUN, contados como num cronómetro. As três primeiras linhas põem os contadores a zero. A linha 10 destina-se a evitar a impressão de fracções de segundo inferiores a 1/10. A precisão é muito razoável porque a contagem não depende do tempo que o programa leva a executar. Para se fazer a contagem em minutos e segundos, as últimas linhas devem ser substituídas por:

```

50 LET tm = INT (t/60)
60 LET ts = INT (t - INT (t/60) * 60)
70 PRINT AT 10,12; tm; " "; ts; " "
80 GOTO 40

```

Note-se que quando a corrente da rede funciona a 60 Hz, o número 50, no final da linha 40, deve ser substituído por 60.

INKEY\$ — *Tecla que introduz uma cadeia* (Input KEY String). Permite ler o teclado e distingue entre os caracteres maiúsculos e minúsculos, e ainda entre os caracteres introduzidos com ou sem SHIFT.

INKEY\$ é conseqüentemente muito útil para determinar acções mediante a pressão exercida numa ou mais teclas. É de regra nos programas regidos por "menus" ("menu driven") ou seja naqueles em que a pressão de certas teclas — "1", "2", "3", etc, ou "A", "B", "C", etc, determina a passagem a certas subrotinas. Outra aplicação frequente é a da determinação do sentido do movimento de imagens e elementos de imagens. Fazendo-se, por exemplo:

```
10 PAUSE 0: IF INKEY$ = "S" THEN PRINT "SIM"
```

Quando se premir a tecla [S] surgirá no ecrã a palavra SIM.
Note que um resultado semelhante pode ser obtido com:

```
10 INPUT a$: IF a$ = "S" THEN PRINT "SIM"
```

A diferença está em que com INPUT é necessário premir seguidamente ENTER, o que torna esse comando impraticável para determinar movimentos. Com INKEY\$ não é necessário premir ENTER — a acção segue-se imediatamente ao premir da tecla designada.

Eis um programa simples de aplicação de INKEY\$ ao movimento de um caracter gráfico no ecrã:

```

10 LET a=8: LET b=8
20 LET a1=8: LET b1=8
30 PRINT AT b, a; CHR$ 143
40 IF INKEY$ = "5" THEN LET=a-1
50 IF INKEY$ = "6" THEN LET=a+1
60 IF INKEY$ = "7" THEN LET=b-1
70 IF INKEY$ = "8" THEN LET=b+1
80 IF a>31 THEN LET a=31: LET a1=a
90 IF a<0 THEN LET a=0: LET a1=a

```

```

100 IF b>21 THEN LET b=21: LET b1=b
110 IF b<0 THEN LET b=0: LET b1=b
120 IF b1<>b OR a1<>a THEN PRINT AT b1, a1; " "; LET a1=a: LET
    b1 = b
130 GOTO 30

```

A animação é elementar, porque há um salto correspondente a um bloco de uma posição para outra. Elimine a linha 120 e veja o resultado. No programa seguinte a animação faz-se ponto por ponto:

```

10 FOR n=1 to 255
20 POKE 23606, n: POKE 23607,60
30 IF INKEY$<>" " THEN POKE 23606,0: POKE 23607,60:STOP
40 PRINT AT 0,0; "ABCDEFGHIJKLMNPOQRSTUVWXYZ"
50 PRINT "B": PRINT "C": PRINT "D": PRINT "E":
    PRINT "F"
60NEXT n
70 GOTO 10

```

O Basic

O Som no 48 BASIC

Como se disse, o som nos Spectrum + 2 e + 3 é emitido através do televisor ou do monitor, e pode-o ser também através de um amplificador externo, sendo portanto nítido e tão forte como se desejar. Usando o 128 BASIC ou o + 3 BASIC é possível programar um gerador de sons de três canais. No 48 BASIC tem-se a instrução BEEP, que determina a emissão de uma nota simples, de duração e tom expressos por dois números colocados a seguir àquela palavra e separados um do outro por uma vírgula.

O primeiro valor pode variar entre 0 e 10 e define a duração em segundos da nota emitida. O segundo pode variar entre -60 e +69. Os valores negativos situam-se abaixo do "dó" central e os valores positivos acima dele.

Para ter uma ideia da gama de sons que pode gerar com o 48 BASIC, introduza o seguinte programa:

```

10 FOR p=69 TO -60 STEP -1
20 BEEP 0.2, p
30 PRINT AT 0,0; " "; AT 0,0;p
40 NEXT p

```

A correspondência com a escala musical é:

```
BEEP 1,0 = Dó
BEEP 1,2 = Ré
BEEP 1,4 = Mi
BEEP 1,5 = Fá
BEEP 1,7 = Sol
BEEP 1,9 = Lá
BEEP 1,11 = Si
BEEP 1,12 = Dó
```

Note que, exceptuando entre Mi e Fá e entre Si e Dó, há sempre dois meios tons de intervalo entre as notas. Pode-se subir os sons de uma oitava ou baixá-los adicionando ou doze aos valores indicados (recorde que podem ser usados valores negativos). Adicionando 1 a uma nota, obtem-se o "sustenido"; subtraíndo-o obtem-se o "bemol".

A música oriental e a indiana também podem ser reproduzidas, usando valores decimais para os tons. Mas a reprodução das escalas dos instrumentos ocidentais podem também obrigar ao uso de tons decimais. Assim, enquanto para escala do piano se tem a mesma escala atrás indicada, mas com menor duração das notas:

```
10 BEEP .5,0: BEEP .5,2: BEEP .5,4: BEEP .5,5: BEEP .5,7:
BEEP .5,9: BEEP .5,11: .5,12: STOP
```

Para a escala natural, do violino, ter-se-á:

```
20 BEEP .5,0: BEEP .5,2.039: BEEP .5,3.86: BEEP .5,4.98: BEEP
.5,7.02: BEEP .5,8.84: .5,10.88: BEEP .5,12: STOP
```

Experimente o seguinte programa:

```
10 LET p= CODE INKEY$
20 IF p=0 THEN GOTO 10
20 BEEP .04, (p-30)/2
40 GOTO 10
```

Note a utilização de CODE e de INKEY\$. CODE introduz o código ASCII de um carácter, e INKEY\$ (INput KEY) indica a posição no teclado. O programa atribui um som diferente a cada tecla. Se se carregar simultaneamente em CAPS SHIFT o tom baixa.

Um programa simples, que reproduz um motivo musical bem conhecido:

```
10 BEEP .5,24: BEEP .5,26: BEEP .1,22: BEEP .1,10: BEEP .1,17
```

O 48 BASIC, apesar das suas limitações, permite ouvir Tchaikovsky:

```
10 BEEP .5,7: BEEP .5,4: BEEP .5,2: BEEP .5,0: BEEP .1,0
20 BEEP .5,16: BEEP .5,14: BEEP .5,17: BEEP .1,17: BEEP .5,16: BEEP .5,11:
BEEP .5,12: BEEP .5,2: BEEP .5,9: BEEP .1,9
30 BEEP .5,14: BEEP .5,16: BEEP .5,9: BEEP .1,9
40 BEEP .5,14: BEEP .5,11: BEEP .5,29: BEEP .5,7
50 BEEP .1,14: BEEP .5,16: BEEP .1,19: BEEP .5,19: BEEP .5,14: BEEP .5,14:
BEEP .5,16: BEEP .1,19: BEEP .5,19
60 BEEP .5,16: BEEP .5,21: BEEP .5,19: BEEP .1,14: BEEP .5,14
70 BEEP .25,11: BEEP .25,9: BEEP .25,7: BEEP .25,2: BEEP .25,2:
BEEP .1,0
```

Uma surpresa:

```
10 BEEP .5,9: BEEP .5,7: BEEP .5,0: BEEP .5,4: BEEP .5,7:
BEEP .1,5,9
20 BEEP .5,9: BEEP .5,7: BEEP .5,-1: BEEP .5,2: BEEP .5,5:
BEEP .1,5,9
30 BEEP .5,9: BEEP .5,7: BEEP .5,5: BEEP .5,7: BEEP .5,11:
BEEP .1,5,16
40 BEEP .5,14: BEEP .5,14: BEEP .5,12: BEEP .5,16: BEEP
.5,12: BEEP .1,5,7
50 BEEP .5,7: BEEP .5,12: BEEP .25,12: BEEP .5,12: BEEP
.25,12: BEEP .5,12: BEEP .1,5,11
60 BEEP .5,11: BEEP .5,10: BEEP .25,10: BEEP .5,10: BEEP
.25,10: BEEP .5,10: BEEP .1,5,9
70 BEEP .5,9: BEEP .5,8: BEEP .25,8: BEEP .5,8: BEEP .25,8:
BEEP .5,8: BEEP .1,5,7
80 BEEP .5,3: BEEP .5,2: BEEP .1,5,5
```

A geração de ruídos, para acompanhar as imagens dos jogos, por exemplo, pode ser feita do mesmo modo:

```

10 REM Laser 1
20 LET a=.0125
30 FOR b=1 TO 2
40 FOR c=4 TO 16 STEP 2
50 BEEP a,c
60 NEXT c
70 NEXT b

```

```

10 REM Laser 2
20 FOR a=10 TO 0
30 BEEP .0125,a
40 NEXT A
50 FOR b=0 TO -5 STEP -1
60 BEEP .0125,b
70 NEXT b

```

```

10 REM Passos
20 BEEP 0.02,30: BEEP 0.02,40
30 PAUSE 3
40 GOTO 20

```

```

10 REM Bomba
20 FOR x=69 TO 55 STEP -.3
30 BEEP .05,x
40 NEXT x
50 FOR y=0 TO 20
60 BEEP .01, -10: BEEP .01, -50: BEEP .01, -60
70 NEXT y

```

Um exemplo simples de coordenação de som e imagem:

```

10 REM Som e Imagem
20 BORDER 0: PAPER 0: CLS
30 FOR x=0 TO 31
40 LET a= RND*20
50 LET b= RND*.4

```

```

60 LET c= (RND*6) + 1
70 PRINT INK c; AT 20-a,x;''*''
80 BEEP b,a
90 NEXT x
100 CLS: GOTO 30

```

Note que enquanto o som é emitido, o computador pára. A partir do BASIC, não é possível gerar sons e mover imagens simultaneamente. Tudo quanto se pode fazer é emitir uma parte do som e deslocar uma imagem um pouco, outra parte do som e outro movimento, sucessivamente. O som que as teclas emitem pode ser alterado por um simples comando directo:

```
POKE 23609,100
```

Podem tentar qualquer número entre 0 e 255.

Programação Estruturada e Subrotinas

Chegou o momento de conhecer os “pontos finos” da programação em BASIC.

A execução dos programas em BASIC é relativamente lenta, mas com uma programação adequada — *estruturada* — pode tornar-se mais rápida.

O segredo da estruturação é o recurso às *subrotinas*, ou seja a pequenos programas auxiliares que são chamados a desempenhar certas funções uma ou mais vezes durante a execução do programa principal.

O programa seguinte contém um exemplo muito simples de subrotina, incluído numa outra versão do jogo de adivinhação de números.

```
10 INPUT a: CLS
20 INPUT "Adivinhe o numero ",b
30 IF a=b THEN PRINT "Certo !": STOP
40 IF a<b THEN GOSUB 100
50 IF a>b THEN GOSUB 100
60 GO TO 30
100 PRINT "Tente outra vez "
110 RETURN
```

Note nas linhas 40 e 50 a ordem GOSUB, que se obtém premindo a tecla (H) e que determinam a passagem ao pequeno programa (subrotina) da linha 100, que evita a repetição da mensagem “Tente outra vez” e que é completado na linha 110 pela ordem RETURN, perante a qual o programa regressa à linha de onde partira. Note também a diferença entre GOSUB e GOTO. Enquanto GOSUB volta obrigatoriamente à mesma linha através de RETURN, GOTO pode voltar a qualquer linha através de outro GOTO.

Eis um programa de cálculo mental que inclui uma subrotina para cada tipo de operação:

```
10 REM Calculo mental
20 LET a = INT (RND*9)+1
30 LET b = INT (RND*9)+1
40 PRINT AT 8,0; "PARA SOMAR PRIMA 1";
   AT 9,0; "PARA SUBTRAIR PRIMA 2"; AT 10,0; "PARA
   MULTIPLICAR PRIMA 3"; AT 11,0; "PARA DIVIDIR PRIMA 4"
50 INPUT D
100 IF D<1 OR C>4 THEN GOTO 50
110 CLS
```

```
120 GOSUB D*1000
130 INPUT E: PRINT AT 10,15; E
140 IF E=C THEN PRINT AT 15,10;"CERTO !"
150 IF E<>C THEN PRINT AT 15,10; "ERRADO !"
160 PAUSE 100 : CLS : GOTO 20
1000 LET c= a+b: PRINT AT 10,10; a; "+"; b;"= ?
1010 RETURN
2000 LET c= a-b: PRINT AT 10,10; a; "-"; b;"= ?
2010 RETURN
3000 LET c= a*b: PRINT AT 10,10; a; "*"; b;"= ?
3010 RETURN
4000 LET c= a/b: PRINT AT 10,10; a; "/"; b;"= ?
4010 RETURN
```

Note na linha 120 a utilização de uma expressão para dirigir o programa para as linhas 1000, 2000, 3000 e 4000 em que se iniciam as subrotinas.

Os programas anteriores foram organizados segundo a ordem lógica: primeiro as variáveis, depois o programa principal e por fim a subrotina. Isso é correcto para os compiladores, ou seja para os sistemas em que o programa em BASIC é convertido no seu conjunto em código de máquina, mas os Spectrum usam um interpretador, como é de regra nos microcomputadores. Nesse caso, quando o computador encontra um GOTO ou um GOSUB, verifica o programa linha a linha desde o início, o que resulta numa demora.

Portanto, se as primeiras linhas contiverem as variáveis, mesmo que estas não mais forem usadas nos programas, serão examinadas cada vez que haja um GOTO ou um GOSUB. Por isso convém colocar no final do programa, como uma subrotina, as variáveis e todos os outros elementos a definir, passando então a primeira linha a ser algo como:

```
-10 GOSUB 9000
```

Assim o programa nunca mais volta a essas instruções depois de executar o primeiro GOSUB. Uma excepção a esta regra é a das instruções DEF FN, as quais devem ser colocadas no início do programa pois são procuradas sempre que o programa necessita delas.

As subrotinas mais usadas devem ser também colocadas no início do programa, pela ordem de frequência do seu uso, e não no final. Só então virá o programa principal, e no fim as subrotinas menos usadas. Um programa “estruturado” terá portanto o seguinte aspecto:

```

10 REM Título
20 GOSUB 9000
100 SUBROTINA A
200 SUBROTINA B
300 SUBROTINA C
1000 PROGRAMA PRINCIPAL
5000 SUBROTINA D
5100 SUBROTINA E
9000 SUBROTINAS DE INICIAÇÃO

```

Cada módulo deve terminar com uma direcção (normalmente um GOTO) para evitar que o programa prossiga na linha seguinte.

A Gravação de Programas em Código e o Registo de Imagens e Dados

Para se gravar em cassette um programa em código de máquina, uma ficha de dados ou ficha de imagem, coloca-se no Datacorder, no Spectrum + 2, ou num gravador-reprodutor, no Spectrum + 3, uma cassette virgem. As cassetes já gravadas são pouco dignas de confiança porque o ruído de fundo aumenta com a sobreposição das gravações. Releiam-se as páginas 18 e 19.

Note também que:

- * Para que um programa em BASIC arranque automaticamente quando for carregado, a última linha deverá ser:

SAVE nome do programa LINE *número da linha em que o programa deve arrançar.*

- * Para gravar um programa em código (*bytes*) deverá fazer-se:

SAVE nome do programa CODE *endereço, comprimento*

- * Para gravar uma imagem do ecrã (*screen\$*):

SAVE nome da imagem SCREEN\$

O que equivale a:

SAVE nome do programa CODE 16384, 6912

- * Para gravar matrizes:

DATA designação dos dados DATA *letra* ()

Por "letra" entende-se a que serve de referência à matriz. Por exemplo: se a matriz for DIM a(n), a letra será a. Se a matriz for de cadeias ter-se-á:

SAVE designação dos dados DATA *letra* \$ ()

Para fazer VERIFY de um programa em código introduz-se:

VERIFY nome do programa CODE *endereço, comprimento*

Se o comprimento do programa for maior que o indicado, surgirá a mensagem de erro:

R Tape loading error

Se se fizer:

VERIFY nome do programa CODE endereço

Os bytes gravados serão comparados com os que se encontrarem na memória a partir do endereço indicado, sem atenção ao comprimento. Se se fizer:

VERIFY nome do programa CODE

Os bytes gravados sob o nome designado serão comparados com os que se encontram na memória a partir do endereço do primeiro byte gravado.

VERIFY nome da imagem SCREEN\$

equivale a

VERIFY nome da imagem CODE 16384, 6912

mas falha a regra.

VERIFY designação dos dados DATA letra ()

ou

VERIFY designação dos dados DATA letra \$ ()

permite verificar matrizes.

Quanto a LOAD o procedimento é idêntico:

LOAD nome do programa CODE endereço, comprimento

carrega na memória os bytes gravados a partir do endereço, apagando tudo quanto af esteja. Se os bytes excederem o comprimento indicado para o programa, surgirá a mensagem de erro no carregamento. Então poder-se-á usar:

LOAD nome do programa CODE endereço

Os bytes serão carregados independentemente do comprimento do programa.

Se o endereço e o comprimento do programa não forem conhecidos far-se-á:

LOAD nome do programa CODE

Os bytes serão carregados a partir do endereço do primeiro byte gravado.

Quanto às matrizes bastará fazer:

LOAD designação dos dados DATA letra ()

ou

LOAD designação dos dados DATA letra \$ ()

Se não houver espaço na memória para as novas matrizes, surgirá a mensagem de erro **4 Out of memory**, mas as que já se encontrarem na memória não serão apagadas.

O comando MERGE, acessível através de EXTENDED, SYMBOL SHIFT e (T), permite carregar um programa em BASIC sem apagar outro que já exista na memória. No entanto quando houver duas linhas com o mesmo número, a do novo programa substituirá a do anterior. Se não houver espaço na memória para os dois programas e para as variáveis, surgirá a mensagem **4 Out of memory**.

O Comando MERGE não é aplicável a bytes ou matrizes.

Para gravar os caracteres definidos pelo utilizador há que seguir procedimentos especiais. O mais fácil é o de gravar os 21 caracteres fazendo:

SAVE designação CODE USR "a", 21 * 8

depois do que se pode fazer:

VERIFY designação CODE

Recorde-se que o endereço do primeiro byte a ser gravado é USR "a" — o do primeiro dos oito bytes do primeiro carácter — e que o número total de bytes é de 21 * 8 — vinte e um caracteres com oito bytes cada.

Para carregar o programa far-se-á simplesmente:

LOAD designação CODE

Ou, para evitar surpresas com alterações de endereços:

LOAD designação CODE USR "a"

Note-se que todas as instruções dadas se referem à gravação em cassette. A cassette é um meio económico de registo, mas deve-se notar que não satisfaz as aplicações educacionais e profissionais.

Para que essas aplicações sejam realmente práticas, é indispensável que o utilizador possa dispor de um grande volume de dados e possa obtê-los e manipulá-los rapidamente. Com as cassetes e os gravadores vulgares, o utilizador tem de fazer avançar ou recuar a fita até encontrar os dados de que necessita — o que pode ser demorado, em particular se o gravador não dispuser de um contavoltas. Terá nova demora para introduzir os dados — e poderá encontrar dificuldades ao fazer LOAD ou MERGE, porque a fita magnética é um meio pouco seguro: por muito bom

que seja o gravador e por muito bem concebido que seja o computador, basta que a fita magnética se distenda um pouco para que o carregamento se torne impossível.

Daí o interesse das disquetes, mas antes de aprender a utilizá-las, conheça as diferenças entre o 48 BASIC, o 128 BASIC e o + 3 BASIC.

Spectrum + 2 e o 128 Basic

Tudo quanto atrás se disse sobre o 48 BASIC é aplicável ao 128 BASIC mas este possui comandos e possibilidades suplementares, nomeadamente quanto ao som, e ao disco de memória.

Os programas escritos em 48 BASIC funcionam de uma maneira geral no modo 128 BASIC. Exceptuam-se:

Os que usem o comando VAL com funções trigonométricas ou condicionais. O 128 BASIC (e o + 3 BASIC) só aceitam esse comando com as operações aritméticas.

Os que usem os sinais < . > e <> com uma variável. Nesse caso deve-se colocar um espaço entre ele e a variável. Quando são usados com números o espaço não é necessário.

Os programas escritos em 128 BASIC podem funcionar no modo 48 BASIC se não usarem o gerador de som de três canais e o disco de memória.

Os programas em código de máquina concebidos para os Spectrum de 48 Kb e 128 Kb funcionam de uma maneira geral no Spectrum + 2 mas o contrário nem sempre é possível. Se os programas do Spectrum + 2 tiverem uma extensão superior a 48 Kb ou usarem o gerador de som ou o disco de memória não poderão correr nos Spectrum 48K. Por outro lado os programas do Spectrum + 3, se transferidos para cassete, só serão aceites pelo Spectrum + 2 se não tiverem comandos do sistema de disquetes e do disco de memória.

As diferenças mais importantes entre o 48 BASIC por um lado e o 128 BASIC (e o + 3 BASIC) por outro, são o facto de os comandos e instruções serem introduzidos letra a letra, e não por teclas específicas e a presença de um editor de ecrã muito mais flexível, assim como a presença do disco de memória.

O editor do ecrã

O Spectrum + 2 dispõe de um editor de ecrã que permite criar, modificar e "correr" programas em 128 BASIC. Para se passar ao editor, seleccione no menu inicial a opção 128 BASIC, levando a barra à coincidência com ela por meio das teclas cursoras e premindo então ENTER.

Note então três coisas:

* Primeiro, um quadrado relampejante, azul e branco, no canto superior esquerdo do ecrã. É o *cursor*. Prima as teclas de algumas letras. Veja que elas surgem no ecrã a seguir ao cursor.

* Segundo, uma barra negra na parte inferior do ecrã — a *barra de rodapé*. Diz-se nela qual é a parte do "software" incluído na ROM que se está a usar. No momento deve dizer 128 BASIC.

* Terceiro, o espaço entre a barra de rodapé e o extremo inferior do ecrã está em branco, mas pode apresentar duas linhas de texto. É aí que o Spectrum + 2 apresenta as suas mensagens de erro, mas tem outros usos que adiante se descreverão.

Prima a tecla de EDIT. Note que:

- * O cursor desaparece.
- * Surge um novo menu — o *menu de edição* com as seguintes opções:

128 BASIC
Renumber
Screen
Print
Exit.

128 BASIC — Cancela o menu de edição e determina o regresso do cursor. É útil quando se prime EDIT por engano.

Renumber — Determina que as linhas dos programas em BASIC sejam renumeradas começando por 10 e continuando por intervalos de 10. Todos os comandos que incluam referências a números de linhas são automaticamente renumerados, exceptuando aqueles em que esses números sejam calculados. Nesses casos, assim como no de não haver qualquer programa introduzido no computador ou de a remuneração conduzir a números de lista superiores a 9999, o Spectrum + 2 emite um som de aviso.

No apêndice ???? encontram-se duas rotinas que permitem remunerar os programas do Spectrum + 2 entre duas linhas quaisquer e com um intervalo qualquer.

Screen — Permite passar o cursor para o espaço abaixo da barra de rodapé e introduzir ali comandos ou instruções de BASIC ou modificá-las, sem alterar o que estiver acima da barra, o que é muito importante quando se trabalha com gráficos. Para voltar a trabalhar na parte superior do ecrã basta seleccionar de novo *Screen*.

Print — Corresponde a LLIST no 48 BASIC. Isto é, envia para a impressora a listagem do programa corrente. No final, o menu desaparece e é substituído pelo cursor. Se a impressora estiver desligada ou de algum modo não estiver acessível, prime-se duas vezes **BREAK** para regressar ao editor.

Exit — Determina o regresso ao menu inicial mas sem perda do programa que estiver na memória. Para voltar ao programa, selecciona-se a opção **128 BASIC**, mas se se escolher **48 BASIC** o programa perde-se. No entanto se se escolher a opção **Calculator**, disponível no menu inicial, o programa não se perderá.

Para uma compreensão mais completa do funcionamento do editor do Spectrum + 2 introduza no computador exactamente a seguinte linha de programa:

10 for f=1 to 255 step 10

Prima ENTER. O computador alterará a linha, escrevendo FOT, TO e STEP com maiúsculas:

10 FOR f=1 TO 255 STEP 10

Se as palavras referidas não forem convertidas em maiúsculas, e se o computador emitir um som de aviso, isso indica que houve um erro na composição da linha. A cor do cursor muda de azul para vermelho e a linha só será aceite pelo computador quando o erro for corrigido. Para isso usam-se as teclas cursoras para levar o cursor até à parte da linha onde se verificou o erro e então ou se insere o carácter ou caracteres em falta ou se apaga o que estiver errado com o auxílio da tecla DELETE e se procede depois à correcção.

Introduza em seguida:

20 plot 0,0 draw f, 175: plot 255,0: draw -f.175

Prima ENTER. Note que é conveniente inserir as instruções com caracteres minúsculos para se localizar mais facilmente qualquer erro.

Acrescente:

30 next f

Prima a tecla cursora com a seta virada para cima. Note que o cursor sobe até à linha 30 e fica aí.

Prima de novo a tecla em causa e depois a que tem a tecla virada para a direita até o cursor se situar sobre a linha 20 e o número 1 na instrução DRAW — f.175.

Prima a tecla cursora com a seta virada para baixo. O Cursor saltará para o fim da linha 30. Isso acontece porque o cursor “foge” aos espaços em branco e procura o texto mais próximo. Prima de novo a tecla com a seta virada para cima. Pelo que atrás se disse, poder-se-ia esperar que o cursor subisse na vertical, mas não. Volta à posição anterior porque se “recorda” dela. Mova várias vezes o cursor e note o comportamento dele. Esta faculdade do cursor denomina-se “tracking” — isto é *rastreio*. Pode confundir um tanto os iniciados, mas torna a edição dos programas muito mais fácil quando o utilizador está familiarizado com ela.

Faça RUN.

Então acontece muita coisa. O ecrã fica limpo e apresenta um desenho, depois do que surge a mensagem:

0 OK, 30 : 1

indicando que o programa foi completado.

Premindo de novo ENTER, a barra de rodapé reaparece, tal como a listagem do programa, o que demora cerca de um segundo. Não tente usar o teclado durante esse intervalo porque mesmo que fosse suficientemente ágil, o computador ignoraria a tentativa.

Prima EDIT e escolha a opção **Screen**. O editor desce abaixo da barra de rodapé e nesse espaço surge a linha 10 do programa. Mova as teclas cursoras acima e abaixo e veja o que acontece.

Faça RUN.

O programa será executado tal como antes. Prima ENTER. Note que, ao contrário do que aconteceu antes, o ecrã não é limpo, e as linhas do programa, na parte inferior, continuam a poder ser movidas por meio de teclas cursoras. Prima EDIT. Contra o que poderia esperar, quando acabar de trabalhar com o menu, o ecrã volta a apresentar o desenho que nele se encontrava.

Para se certificar de que o editor está a trabalhar na parte inferior do ecrã, prima ENTER e altere a linha 10 como se segue:

```
10 FOR f=1 TO 255 STEP 7
```

Escreva em seguida:

```
go to 10
```

Prima ENTER.

O desenho será substituído por outro, diferente. Altere a linha quantas vezes quiser — verá outros tantos desenhos diferentes.

Um aviso: — se a instrução a editar ocupar mais de duas linhas do ecrã o editor pode ficar “confuso”, pois não sabe onde está o fim ou o princípio dela. Então haverá que usar o modo normal de edição, no ecrã superior.

Note que, tal como nos modelos anteriores do Spectrum, quando [CAPS SHIFT] está premida, as teclas [5], [6], [7], [8] também movem o cursor e a tecla [0] actua como a de DELETE, enquanto o [9] permite passar ao modo gráfico.

Premindo-se a tecla [L] no modo 128 BASIC, a listagem surge automaticamente no ecrã. Fazendo:

```
LIST número da primeira linha, número da última linha
```

é possível apresentar uma parte da listagem no ecrã.

O disco de memória

Uma das características particulares do Spectrum + 2 (e do Spectrum + 3) é o disco de memória, também denominado “disco de silício” e *RAM disk*. Na prática trata-se de uma área da memória

reservada para armazenar programas ou dados para serem chamados e processados quando necessário tal como num disco rígido, ou numa disquete, mas de uma maneira praticamente instantânea. Tudo quanto se faz com as cassetes pode-se fazer com o disco de silício mas há que ter presente que se se desligar o computador ou houver uma falta de corrente, o conteúdo do disco de silício perde-se. Contudo o comando NEW não actua sobre ele.

Os comandos usados para o disco de memória são idênticos aos usados para as cassetes, mas são seguidos de um ponto de exclamação — por exemplo: para “gravar” um programa no disco de memória faz-se: SAVE !

Os únicos comandos suplementares, para uso com o disco de memória, são:

CAT !

que fornece uma lista dos programas contidos no disco, e

ERASE ! *nome da ficha*

que apaga um programa ou os dados que deixem de ser necessários.

Um dos usos mais evidentes do disco de silício é o de armazenar partes de programas em BASIC para, através do comando MERGE !, constituir um programa maior. Desse modo é possível constituir programas em BASIC até cerca de 90 K.

Um dos usos mais interessantes do disco de memória é a *animação*. O BASIC permite com muito menos trabalho que o código de máquina, desenhar toda a espécie de imagens, mas a formação destas no ecrã é muito lenta. Essas imagens podem ser armazenadas no disco de memória e chamadas de uma maneira quase instantânea. O programa seguinte permite fazer uma ideia — embora pálida — dessa possibilidade:

```
10 INK 5: PAPER 0: BORDER 0: CLS
20 FOR f=1 TO 10
30 CIRCLE f * 20,150,f
40 SAVE ! "BOLA" + STR$(f) CODE 16384,2048
50 CLS
60 NEXT f
70 FOR f=1 TO 10
80 LOAD ! "BOLA" + STR$(f) CODE
90 NEXT f
100 BEEP 0.01, 0.01
110 FOR f=9 TO 2 STEP -1
```

```

120 LOAD ! "BOLA" + STR$(f) CODE
130 LOAD ! NEXT f
140 BEEP 0.01, 0.01
150 GO TO 70
160 REM use GO TO 160 para apagar as imagens do disco
170 FOR f=10 TO 1 STEP -1
180 ERASE ! "BOLA" + STR$(f)
190 NEXT f

```

As linhas 20 a 60 criam uma série de dez imagens e gravam-nas no disco de memória, as linhas 70 a 130 chamam-nas, a 140 avisa do final da série, a 150 determina a repetição e — fazendo **BREAK** e usando a instrução **GO TO 160** — as linhas até 180 apagam as imagens no disco de memória. *Note que o apagamento se faz no sentido inverso ao da gravação. É mais rápido e simplifica as operações.*

O Spectrum + 3 e o + 3 Basic

Tudo quanto atrás se disse sobre o 48 BASIC e sobre o 128 BASIC é aplicável ao + 3 BASIC, excepto no que diz respeito ao comando do leitor de disquetes, do disco de memória, do gravador-reprodutor de cassetes e das impressoras.

A capacidade das disquetes do Spectrum + 3 é de 173 Kb por face.

A capacidade do disco de memória é de cerca de 58 Kb.

Os programas escritos em BASIC ou em código de máquina para o Spectrum de 48 Kb funcionam de uma maneira geral em Spectrum + 3, mas os programas do Spectrum + 3 só funcionam nos Spectrum de 48 Kb se não usarem o gerador de som de três canais e o disco de memória. E só funcionarão nos Spectrum de 128 kB — inclusive no Spectrum + 2 — se não contiverem instruções especiais para o comando do leitor de disquetes e do disco de memória.

Tal como caso do 128 BASIC, as diferenças mais importantes entre o 48 BASIC por um lado e o + 3 BASIC por outro, são o facto de os comandos e instruções serem introduzidas letra a letra, e não por teclas específicas, a existência do disco de memória e a presença de um editor de ecrã muito mais flexível. Os comandos do editor de ecrã são precisamente iguais aos do 128 BASIC, mas os do disco de memória são diferentes.

O leitor de disquetes

O Spectrum + 3 dispõe de um leitor de disquetes (*driver*) de 3 polegadas e pode ser ligado a um segundo leitor. O leitor integrado é designado por A: — notem-se os dois pontos porque é através deles que o + 3 BASIC reconhece que a letra A se refere ao leitor. Se existir um segundo leitor, é designado por B:.

Mesmo que não se disponha de um segundo leitor, é possível usar o computador como se o leitor B: estivesse presente. Se, em circunstâncias que adiante serão descritas, se ordenar ao computador que funcione com o leitor B:, surgirá no ecrã a seguinte mensagem:

**Please put the disk for B: into
the drive then press any key**

ou seja:

*Por favor introduza a disquete para B:
no leitor e depois prima qualquer tecla*

O computador tratará então o leitor A: como se fosse o B:. Quando o computador precisar de trabalhar com a disquete que estava antes no leitor, pedirá:

Please put the disk for A: into the drive then press any key

ou seja:

Por favor introduza a disquete para A: no leitor e depois prima qualquer tecla

Esta possibilidade é particularmente útil para a cópia de disquetes.

O Spectrum + 3 é acompanhado por uma disquete de demonstração. A forma de usar essa disquete e as que já estiverem gravadas foi descrita na página 16.

Note-se que as disquetes gravadas noutros computadores não são aceites pelo Spectrum + 3, excepto em certas circunstâncias e até certo ponto as dos Amstrad PCW e CPC.

Para usar uma disquete virgem, há que formatá-la, em primeiro lugar — isto é: dividi-la em pistas e sectores, para que o leitor possa identificar rapidamente as gravações que forem nela feitas. A formatação faz-se através do comando FORMAT, que deve ser seguido pela designação (entre aspas) do leitor onde se encontra a disquete a formatar:

FORMAT "a:"

Se se esquecer de colocar a disquete no leitor antes de introduzir o comando, não se preocupe. O computador avisará:

Drive nor ready

Nesse caso basta inserir a disquete e repetir o comando.

Se a disquete já estiver formatada, o computador avisará:

Disk is already formatted. A to abandon, other key to continue.

isto:

Disquete já formatada, Prima A para abandonar, outra tecla para continuar.

Trata-se de uma segurança que permite evitar que uma disquete já gravada seja formatada. Recorde-se que se uma disquete gravada for formatada, tudo quanto nela estiver gravado será apagado, mesmo que a disquete esteja protegida.

Quando se está a trabalhar com as disquetes, é possível surgir a mensagem:

Drive A: not ready — Retry. Ignore or Cancel?

Isso indica que a disquete não foi colocada no leitor, ou não está em boas condições.

Se a disquete não tiver sido colocada no seu lugar por esquecimento, corrige-se o erro e prime-se a tecla [R].

Se a disquete estiver no seu lugar mas mesmo assim surgir a mensagem, é possível que haja uma falha accidental no início da leitura. Prime-se também a tecla [R] para repetir a tentativa de leitura.

Se se estiver a meio de uma leitura e surgir uma mensagem do tipo:

Missing adress mark

Isso indica que a disquete está danificada. Tenta-se premir a tecla [R] várias vezes e se isso falhar, tem-se de facto uma disquete em más condições. Prime-se a tecla [I] para ignorar o problema e, se possível, salvar uma parte dos dados. Nessas circunstâncias, e como adiante se verá, pode-se usar o comando COPY para copiar para uma disquete em bom estado as partes utilizáveis.

Finalmente, se se der conta de que o problema é devido a um erro do programa e não vale a pena insistir, prime-se [C], para cancelar a leitura.

Fichas e Ficheiros

Apesar do uso das disquetes os comandos SAVE e LOAD são idênticos aos usados no 48 BASIC e no 128 BASIC para as cassetes, conforme foram descritos na pág. 13 e 14.

Aos programas, textos, listas de dados, etc., inseridos numa disquete dá-se o nome de "fichas". (É vulgar, mas errado, designá-los por "ficheiros", à semelhança do inglês file e do francês fichier, mas na língua portuguesa a distinção é clara: ficha é o documento onde se contém os dados, enquanto ficheiro é o lugar onde se guardam as fichas).

Quando um computador tem possibilidade de armazenar um grande número de fichas numa disquete ou, principalmente, num disco rígido, as fichas podem ser organizadas em ficheiros. À lista das fichas contidas num ficheiro dá-se o nome de directório. É o que acontece, por exemplo, no sistema operativo MS-DOS, da Microsoft Corp., usado nos PCs.

No Spectrum + 3 não existem ficheiros e portanto não existem directórios, mas o formato das fichas é idêntico ao do sistema operativo normalmente usado nos computadores de 8 bits — o CP/M. da Digital Research Corp.. Portanto as disquetes do + 3 podem ser usadas nos computadores que usem esse formato, para transferir dados, programas em código máquina ou texto. É extremamente improvável que deste modo possam ser transferidos programas em BASIC.

Os nomes das fichas (*filenames*) no Spectrum + 3 são constituídos por quatro "campos" (*fields*) segundo as normas do CP/M:

número do utilizador *letra do leitor: nome da ficha. tipo da ficha*

Dos quatro campos só o nome da ficha tem sempre de ser mencionado. Os outros são auxiliares e logo facultativos.

O "número do utilizador" tem uma função semelhante à dos ficheiros no MS-DOS: destina-se a separar as fichas por tipos de utilização e logo por assuntos. Pode ir de 0 a 15, mas como as disquetes do + 3 não podem conter mais de 64 fichas, é raro haver necessidade de o usar. Mas se se especificar um número de utilizador haverá que juntar a letra do leitor. Por exemplo: se se pretender gravar uma ficha com o nome "TESTE" no número de utilizador 5 no leitor A:, far-se-á:

SAVE "5a:teste"

A gravação com número de utilizador cria um problema: se esse número for esquecido, a localização da ficha torna-se difícil porque o comando CAT que fornece o catálogo ou lista das fichas presentes na disquete só refere de cada vez as contidas numa área.

A "letra de leitor" destina-se a indicar ao computador qual o leitor que deve pôr em acção para gravar ou ler uma ficha. Há três letras utilizáveis: A:, B: ou M:, sendo M: a designação do disco de memória. Se nenhuma letra for especificada, o computador entenderá que o leitor designado é o A:. Para que o computador passe a considerar outro leitor como sendo o prioritário, basta inserir o comando SAVE seguido pela letra desse leitor entre aspas. Por exemplo, se se fizer:

SAVE "m:"

e em seguida se fizer:

SAVE "TESTE"

A ficha "TESTE" não será gravado no leitor A: mas sim armazenada no disco de memória.

A mudança também se verificaria se se tivesse feito LOAD "m:"

O "nome da ficha" não deve conter mais de 8 caracteres. As regras a que obedece são as mesmas que se indicam para o 48K BASIC. **Note-se que para o sistema de disquetes do Spectrum + 3 é indiferente que o nome da ficha seja expresso em maiúsculas ou minúsculas.** Os caracteres reconhecidos são os seguintes:

Letras	abcdefghijklmnopqrstuvwxyz (minúsculas ou maiúsculas)
Algarismos	0123456789
Outros	“ # \$ _ 0 ~

Os caracteres * e ? tem um significado especial para o + 3 BASIC, como adiante se verá, e portanto não podem ser usados nos nomes das fichas.

O "tipo de ficha" é opcional e tem por fim indicar ao utilizador se uma ficha contém um programa de BASIC, ou código ou dados. Ao contrário de outros, o + 3 BASIC não opõe automaticamente à ficha a indicação do respectivo tipo. O utilizador deve, portanto, cuidar disso, juntando ao nome da ficha as designações:

BAS se se tratar de um programa em BASIC

BIN se se tratar de código de máquina.

Não é obrigatório que os tipos das fichas sejam os indicados, mas trata-se da convenção geralmente aceite.

Os caracteres utilizáveis para os tipos das fichas são os mesmos indicados para os respectivos nomes.

As seguintes variantes de designação de fichas são aceites pelo + 3 BASIC:

Z
 teste
 Teste
 m:imagem.bin
 a:pedro
 13a:vendas
 0M:COMPRAS
 teste.bas
 filipe
 copos.vid
 a:a.a

As seguintes são ilegais pelas razões adiante indicadas:

pac man (não pode conter espaços)
 (teste) (não pode conter parenteses)
 ⌂+#!& (caracteres não admitidos)
 bas (falta nome da ficha)
 depois... (só pode haver um ponto)
 7:dubio (se for especificado o número do utilizador, também tem de o ser a letra do leitor:
 7a:dubio)

Se na disquete se gravar um programa com a designação de DISK contendo, por exemplo, as instruções para a listagem das fichas nele gravadas, o + 3 BASIC executará automaticamente

a sua execução se no menu inicial se escolher a opção **Loader** e se premir **ENTER**.
Para se obter a listagem ou "catálogo" das disquetes ou do disco de memória há que usar o comando **CAT**.

A lista é apresentada por ordem alfabética e cada ficha é seguida pela indicação da respectiva extensão em quilobytes arredondada por excesso. No fim da lista é indicado o espaço livre na face da disquete.

Para obter o catálogo das fichas num leitor que não seja o prioritário, há que indicar a respectiva letra. O mesmo para o disco de memória, para o qual se terá, portanto:

CAT "m:"

Quando se pretende seleccionar fichas de um determinado tipo ou com designações semelhantes, ou ainda quando não se tem o conhecimento preciso do nome da ficha, recorre-se aos subtítulos (*wilcards*).

O + 3 BASIC usa dois caracteres como substitutos. * e ?.

- * substitui a designação e (ou) o tipo das fichas.
- ? substitui uma letra da designação e (ou) do tipo das fichas.

Por exemplo:

CAT "*.*BAS"

apresentará uma lista das fichas do tipo **.BAS**, e

CAT "TESTE.*"

apresentará uma lista das fichas com a designação "TESTE".

CAT "*.*"

apresentará uma lista geral, equivalendo portanto a **CAT**.

Fazendo por exemplo:

CAT "ma?"

ter-se-á uma lista de todas as fichas cujo nome contenha três caracteres dos quais os primeiros são "ma".

Fazendo:

CAT "?ma?"

ter-se-á uma lista de todas as fichas cujo nome tenha quatro caracteres dos quais o segundo e o terceiro sejam "ma".

Eis alguns exemplos de aplicações admissíveis do substituto ?:

- ?.*bas** (fichas do tipo **.BAS** com um nome de um caracter)
- T??????.*** (todas as fichas começadas por **T**)
- ca??.*i?** (todas as fichas de quatro caracteres iniciados por "ca" e de todos os tipos cujo caracter central seja um "i".)
- ??????.???** (todas as fichas de todos os tipos, como ***.***)

Se se pretender imprimir o catálogo far-se-á:

CAT #3

Este comando admite também as anteriores especificações de fichas, desde que elas sejam adicionadas com a interposição de uma vírgula. Por exemplo:

CAT #3,"a:*.bas"

Como é óbvio, o comando só será executado se a impressora estiver ligada. Se não se dispuser de impressora e portanto o computador não puder executar a instrução, deve-se premir **BREAK** para a anular.

O comando **CAT** pode também ser expandido, se lhe for adicionado **EXP**. Assim, se se fizer:

CAT "a:" EXP

As fichas serão seguidas pelas abreviaturas dos seus atributos.

Há três espécies de atributos:

PROT — indica que a ficha pode ser lida mas não escrita.

SYS — indica que a ficha é de sistema. É o único tipo de ficha realmente compatível com outros sistemas CP/M.

ASC — indica que a ficha é de arquivo. Não tem outro uso prático além do de assegurar a compatibilidade ao nível da transferência de fichas com outros computadores que disponham de CP/M.

Os atributos podem ser alterados pelo comando **MOVE**. Por exemplo: se se fizer:

MOVE "teste" TO "+p"

a ficha "teste" ficará protegida contra escrita e apagamento. Se em seguida fizer:

ERASE "teste"

o computador recusar-se-á a executar a ordem e responderá:

File is read only

ou seja:

A ficha é só de leitura

Para retirar a protecção faz-se:

MOVE "teste", TO "-p"

Para os outros atributos, o procedimento é semelhante: "+a" para que uma ficha seja considerada de arquivo e "-a" para que deixe de o ser; "+s" para que uma ficha seja considerada de sistema e "-s" para que deixe de o ser.

Para apagar uma ficha usa-se o comando ERASE. Por exemplo:

ERASE Teste

O computador perguntará então:

Erase teste ? (Y/N)

Se tiver havido um erro, o utilizador poderá então rectificá-lo premindo a tecla [N]. Se tudo estiver em ordem, premirá [Y].

O comando ERASE pode ser usado com os substitutos * e ?.

Se se fizer:

ERASE **

Todas as fichas não protegidas serão apagadas, depois de o computador pedir a confirmação do comando.

Se se fizer:

ERASE a: ou **ERASE b:** ou **ERASE m:**

Todas as fichas não protegidas serão apagadas sem que seja pedida confirmação.

Para mudar o nome de uma ficha usa-se o comando MOVE. Por exemplo:

MOVE "TESTE" TO "TESTEI"

Quando uma ficha é gravada com o mesmo nome de outra já existente, a anterior conserva o nome mas é-lhe atribuído o tipo .BAK. Assim, se se verificar que na nova ficha se cometeu

um erro, basta que se use do comando ERASE para apagar a nova ficha e do comando MOVE para repor a designação da ficha antiga.

Para copiar fichas, dentro da mesma disquete, ou de uma disquete para outra, ou ainda de qualquer para o disco rígido, usa-se o comando COPY. Por exemplo:

COPY "a:Teste" TO "m:"

a ficha "Teste" será copiada para o disco de memória m:

O comando COPY pode trabalhar com substitutos, mas estes só podem ser usados em relação às fichas a copiar. Por exemplo:

COPY *.bas TO M:

determinará a cópia de todas as fichas do tipo .BAS para o disco de memória. E:

COPY ** TO B:

determinará a cópia de todas as fichas do leitor prioritário para o leitor B:, mas

COPY *.BAS TO M: *.TXT

determinará a mensagem:

Destination cannot be wild

o que pode ser entendido como:

O destino não pode ser indefenido.

Uma ficha não pode ser copiada dentro da mesma disquete sem que o nome ou o tipo sejam alterados. Se isso for tentado, o computador avisará que a ficha já existe:

File already exists

ou

File already in use

Pode-se alterar o nome de uma ficha copiando-a dentro da mesma disquete mas atribuindo outro nome á cópia e apagando o original.

O comando COPY tem ainda outras aplicações. Por exemplo:

COPY nome e tipo da ficha TO SCREEN\$

apresenta no ecrã o conteúdo das fichas de texto, eliminando os caracteres de controlo. E:

COPY nome e tipo da ficha TO LPRINT

envia o conteúdo da ficha para a impressora.

O comando COPY pode ainda ser usado para adicionar a uma ficha de código de máquina um "header" que permita o seu reconhecimento pelo BASIC. Por exemplo, se a ficha tiver a designação de JOGO.COD, far-se-á:

COPY "jogo.cod" TO SPECTRUM FORMAT

Em sequência será automaticamente formada uma ficha com a designação de JOGO.HED. Como o BASIC não pode saber qual o endereço em que o código deve ser carregado, é preciso indicá-lo, fazendo:

LOAD "jogo.hed" CODE endereço.

As cassetes

O Spectrum + 3 quando usa o + 3 BASIC considera o gravador-reprodutor de cassetes como outro leitor de disquetes, o T:. Assim, estando nesse modo e pretendendo-se por exemplo, carregar um programa de cassette, far-se-á:

LOAD "t:"

Ou, se se quiser gravar o programa em vez de o carregar:

SAVE "t:"

Esses comandos determinam somente a passagem ao modo de cassette. Depois actua-se como no 48 BASIC, fazendo LOAD "" ou LOAD nome da ficha, ou SAVE "" ou SAVE nome da ficha. Além dos comandos LOAD e SAVE, ficam disponíveis os comandos MERGE e VERIFY — que não podem ser usados no modo de disquete. Em contrapartida, os comandos característicos das disquetes — FORMAT, COPY, MOVE e ERASE — não são aplicáveis.

Note-se que, ao contrário do que acontece com as disquetes, o nome das fichas gravadas em cassette pode incluir quaisquer caracteres e espaços em branco.

O comando CAT pode ser usado para as cassetes sob a forma de:

CAT "t:"

Correndo a cassette, obter-se-á uma lista de fichas nela contidas com indicação do respectivo tipo — BASIC, DATA, CODE ou SCREEN\$ — e, no caso das fichas em código, do respectivo endereço e da extensão. Nas fichas em BASIC, se elas forem de arranque automático, o parâmetro usado em LINE é também mostrado.

Por exemplo:

“SIMPLES ” (BASIC)
 “EXECUTAR ” LINE 10 (BASIC)
 “NUMEROS ” DATA n()
 “TEXTO ” DATA t\$()
 “CÓDIGO ” CODE 30000.12345
 “IMAGEM ” CODE 16384.6912

Note-se que a última indicação se refere a uma ficha gravada como:

SAVE "imagem" SCREEN\$

Uma vez que essas fichas são de código e o endereço e a extensão são respectivamente 16384 e 6912.

As fichas de código podem ser transferidas de cassette para disquete usando as indicações dadas por CAT "t:". Primeiro faz-se:

CLEAR endereço — 1

Isto é: no caso da ficha "CÓDIGO", acima indicado, apor-se-ia a CLEAR o número 29999. Depois far-se-ia:

LOAD "" CODE

E depois de terminado o carregamento:

SAVE NOME DA ficha CODE endereço, extensão

Este método não deve ser usado para copiar programas comercializados porque implica uma infracção de "copyright". Há também largas probabilidades de não resultar, por razões como a de o código cobrir algumas variáveis do sistema, o endereço superior variar, o código não ter "header", ou estar protegido de algum modo, e não ser referido por CAT "t:", nem poder ser carregado a partir de BASIC, o comprimento do código ser muito grande, etc..

O Gerador de Som

Como se disse, os Spectrum + 2 e + 3, além do "beep" já descrito — e que serve somente para assegurar a compatibilidade com os programas dos 48K — podem gerar sons musicais, programáveis através do 128 BASIC ou do + 3 BASIC, respectivamente. Os comandos são idênticos nestes dois tipos de BASIC.

Note-se que para extrair do gerador de sons o melhor rendimento é necessário possuir um mínimo de conhecimentos musicais.

Os Spectrum + 2 e + 3 usam a escala de doze notas, que a seguir se indica com as correspondentes da escala clássica:

C,	C#,	D,	D#,	E,	F,	F#,	G,	G#,	A,	A#,	B
Dó	Ré	Mi	Fá	Sol	Lá	Si					

As oitavas são numeradas da maneira seguinte:

-4, -3, -2, -1, 0, 1, 2, 3

A oitava 0 é aquela em que se contem o C internacional.

A frequência das notas é determinada pela fórmula:

$$\text{FREQUÊNCIA} = 440 * (2^{\uparrow(0 + N - 10)/12}))$$

Em que:

0 = número da oitava

N = número da nota

Os métodos vulgares de programação do gerador de som são complexos, pois implicam o cálculo do período da nota, que depende não só da frequência da mesma mas também da dos impulsos transmitidos pelo computador ao gerador, e a sua inserção nos registos deste.

O sistema de programação por cadeias de caracteres, adoptado nos Spectrum + 2 e + 3, é muito mais prático, mas é essencial que nelas as letras maiúsculas sejam sempre representadas por maiúsculas e as minúsculas por minúsculas.

O comando PLAY é muito mais flexível do que BEEP porque pode actuar sobre três vozes em harmonia com toda a espécie de efeitos e com uma muito melhor qualidade de som. É também muito mais fácil de usar.

Estando no 128 BASIC ou no + 3 BASIC, introduza:

```
PLAY "gA"
```

Ouvirá duas notas, a segunda um pouco mais aguda que a primeira. É o que se chama "um tom". Tente em seguida:

```
PLAY "g$a"
```

O "salto" da primeira nota para a segunda será menos brusco. É um "meio tom". Se não tiver notado a diferença, repita os dois exemplos em sequência. Quando se der por satisfeito, faça:

```
PLAY "gD"
```

Note a diferença. É uma "quinta". E faça em seguida:

```
PLAY "gG"
```

As duas notas são agora mais semelhantes. A última é o que se chama uma "oitava", o ponto em que a música começa a repetir-se.

Para tocar A acima de C (ou seja DÓ) médio durante meio segundo basta fazer:

```
PLAY "a"
```

E para tocar a escala de C (DÓ) maior:

```
PLAY "cdefgabC"
```

Note que o último C é maiúsculo. Isso diz ao comando *PLAY* que toque a nota uma oitava acima do c minúsculo. Uma *escala* é a série de notas que abrange uma escala. O exemplo acima constitui a chamada escala de C maior porque é o conjunto de notas entre dois C. Maior porque? Há duas classes principais de escala, chamadas maior e menor, o que é apenas uma maneira de as diferenciar. A escala de C (DÓ) menor é:

```
PLAY "cd$efg$a$bC"
```

Preceder uma nota por \$ baixa-a de um meio tom, isto é, torna-a mais grave, enquanto precedê-la por # eleva-a de meio tom, isto é, torna-a mais aguda. O comando *PLAY* abrange 9 oitavas e para lhe indicar qual deve usar insere-se um O maiúsculo seguido pelo número da oitava.

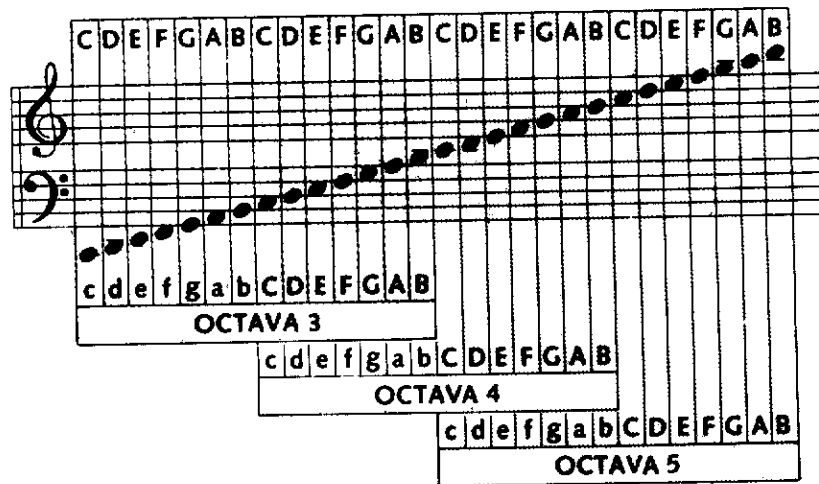
Experimente o seguinte programa:

```
10 LET o$ = "05"
20 LET n$ = "DECcg"
30 LET a$ = o$ + n$
40 PLAY a$
```

Note em primeiro lugar que PLAY executa do mesmo modo uma variável de cadeia ou de constante de cadeia — PLAY a\$, no caso, é o mesmo que PLAY "05DECcg". O comando PLAY pode ter milhares de notas e o melhor é constituir cadeias das notas que tenham de ser repetidas.

Corra o programa. Edite a linha 10 de modo que "05" se torne em "07" e experimente. Editar de novo para "02". Se não se especificar um número de oitava, então o computador assume que se pretende a quinta oitava. (Para facilitar a programação nos Spectrum + 2 e + 3 as oitavas não são numeradas de -4 a 3 mas sim de 0 a 7).

O diagrama seguinte mostra os números e oitavas da escala musical bem temperada:



Note-se que há largas sobreposições: por exemplo, "03D" é o mesmo que "04d". Isso torna mais fácil escrever músicas sem ter de alterar constantemente as oitavas. Algumas das notas nas oitavas mais baixas (0 ou 1) não são exactas, por razões técnicas, e o computador somente pode tentar aproximar-se dos valores reais.

PLAY pode também atribuir muitos comprimentos diferentes para as notas. Edite o programa atrás apresentado de modo que a linha 10 seja:

```
10 LET o$ = "2"
```

Experimente e depois altere o\$ atribuindo-lhe valores entre "1" e "9", ou mesmo um número decimal entre esses limites.

O valor indicado é válido para todas as notas subsequentes até que outro seja atribuído. Os nove comprimentos de nota têm nomes específicos musicais e correspondem os símbolos diferentes quando se escreve música, como se pode ver pela tabela seguinte.

NÚMERO	NOME DA NOTA	SÍMBOLO
1	semicolcheia	♪
2	semicolcheia com mais meio tempo	♪
3	colcheia	♪
4	colcheia com mais meio tempo	♪
5	semínima	♪
6	semínima com mais meio tempo	♪
7	mínima	♪
8	mínima com mais meio tempo	♪
9	semibreve	♩

PLAY pode também trabalhar com *tercinas*, ou seja com três notas tocadas no tempo de duas. Ao contrário dos comprimentos de notas simples, o número de tercinas aplica-se somente às três notas que se seguem de imediato, e depois volta-se ao anterior número de comprimento de nota. Os números das tercinas são as seguintes:

NÚMERO	NOME DA NOTA	SÍMBOLO
10	tercina de semicolcheia	♪♪♪
11	tercina de colcheia	♪♪♪
12	tercina de semínima	♪♪♪

PLAY pode "calar-se" por um período simbolizado por "&". A extensão desse período é igual ao comprimento de nota corrente. Substitua as linhas 10 e 20 no programa da página 108.

```
10 LET o$ = "04"
```

```
20 LET n$ = "DEC&cg"
```

Duas notas tocadas juntas sem interrupção são o que se chama *notas ligadas*. São representadas num comando PLAY por um sublinhado — de modo que uma semínima c e uma mínima c ligadas entre si seriam "5_7c". (O segundo valor é então usado como o comprimento da nota para todas as notas subsequentes, tal como antes).

Há ocasiões em que surgem ambiguidades. Pense que uma peça de música precisa da oitava 6 e um comprimento de nota de 2. Então...

10 LET o\$ = "O62"

parece correcto... mas não é! O computador encontra o O e tenta ler o número que se lhe segue. Quando encontra o número 62. pára com a mensagem de erro **n Out of range**. Nesses casos usa-se uma "nota fingida", denominada N, que serve somente de separador, de modo que a linha 10 deve ser:

10 LET o\$ = "06N2"

O volume pode ser regulado entre 0 (mínimo) e 15 (máximo) usando "V" seguido por um nome. Na prática, só 10 a 15 são em regra úteis porque 1 a 9 são demasiado suaves a menos que o computador seja usado com um amplificador.

É muito fácil activar mais que um canal ao mesmo tempo: basta separar as listas das notas por comas. Experimente o programa seguinte:

```
10 LET a$ = "04cCcCgGgG"
20 LET b$ = "06CaCe$bd$bD"
30 PLAY a$, b$
```

Normalmente não há diferença entre os três canais e qualquer cadeia de notas pode ser enviada para qualquer canal. A velocidade geral da música, o *tempo*, deve estar na cadeia dirigida para o canal A (a primeira cadeia depois de PLAY), de outro modo seria ignorada. Para determinar o tempo em batimentos (semínimas) por minuto, usa-se "T" seguido por um número entre 60 e 240. O valor normal é de 120, ou duas semínimas por segundo. Modifique o programa para:

```
5 LET t$ = "T120"
10 LET a$ = t$ + "04cCcCgGgG"
20 LET b$ = "06CaCe$bd$bD"
30 PLAY a$, b$
```

e corra-o várias vezes, alterando o tempo na linha 5.

Uma prática corrente na música é a repetição de um grupo de notas. Qualquer parte de uma cadeia pode ser repetida incluindo-a entre parênteses, portanto pode-se alterar a linha 10 para:

```
10 LET a$ = t$ + "04(cC) (gG)"
```

PLAY executa-se do mesmo modo que na anterior linha 10. Se se inclui um parêntese final (sem o correspondente parêntese inicial) então a cadeia até esse ponto é repetida indefinidamente. Isso é útil para efeitos de ritmo e linhas de baixo. Tente o que se segue — terá de usar BREAK para parar...

PLAY "O4N2cdefgfed"

e...

PLAY "O4N2cd(efgf)ed"

Se se criar uma linha de baixo com repetição indefinida e depois tocar uma melodia, será bom que a linha de baixo pare quando a melodia parar. Há um meio de o fazer — se PLAY encontra um "H" (*de Half*) em qualquer das cadeias que está a tocar, então pára imediatamente todos os sons. Corra o seguinte programa — mais uma vez terá de usar BREAK para parar...

```
10 LET a$ = "cegbdfaC"
20 LET b$ = "04cC)"
30 PLAY a$, b$
```

Modifique agora a linha 10:

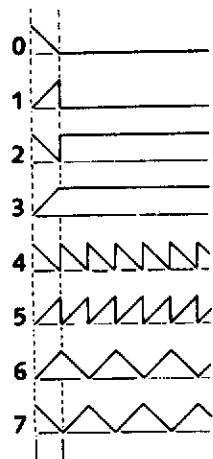
```
10 a$ = "cegbdfaCH"
```

e corra de novo o programa.

Até agora usámos notas que começam e acabam ao mesmo nível de volume. Os Spectrum + 2 e + 3 podem alterar o volume de uma nota enquanto está a ser tocada, portanto pode começar muito alto e acabar em piano, ou subir e descer com um cão a rosnar. Para aplicar esses efeitos use "W" (*de Waveform* — isto é: envolvente) seguido por um número entre 0 e 7 juntamente com "U" para cada canal em que se queira usar o efeito. Os canais que tenham uma regulação de volume ("V") não respondem a "U", a tabela seguinte mostra graficamente como o volume muda para cada regulação...

0	0 — Decai e pára
1	1 — Ataca e pára
2	2 — Decai e sustenta
3	3 — Ataca e sustenta
4	4 — Decai e repete
5	5 — Ataca e repete
6	6 — Ataca decai e repete
7	7 — Decai, ataca e repete

O programa seguinte toca a mesma nota com cada efeito em sucessão, de modo que se possa fazer a comparação com o diagrama seguinte:



- 0 — Decai e pára
- 1 — Ataca e pára
- 2 — Decai e sustenta
- 3 — Ataca e sustenta
- 4 — Decai e repete
- 5 — Ataca e repete
- 6 — Ataca, decai e repete
- 7 — Decai, ataca e repete

10 LET a\$=

''UX1000W0C&W1C&W2C&W3C&W4C&W5C&W6C&W7C''

20 PLAY a\$

O "U" inicia os efeitos e "W" selecciona as envolventes a usar. Há também um "X1000". O "X" determina a duração do efeito (de 0 a 65535). Se não se incluir um "X" os Spectrum + 2 e + 3 escolherão o valor mais alto. As envolventes que param (0 a 3 na tabela anterior) depois da parte inicial, trabalham melhor com regulações de "X" de cerca de 1000, enquanto os efeitos repetitivos (4 - 7) são mais eficazes com valores curtos como 300. Tente variar o valor de X no programa acima para ter uma ideia de como cada um actua.

O comando PLAY não se limita às notas musicais puras. Há também três geradores de "ruído branco" (um ruído semelhante ao de um receptor de FM ou TV não sintonizado) e qualquer dos três canais pode tocar notas, gerar ruído branco ou uma mistura das duas coisas. Para seleccionar uma mistura de ruído e nota, pode-se usar "M" seguido por um número entre 1 e 53. Para se calcular esse número pode-se usar a tabela seguinte:

Número	Canais de tom			Canais de ruído		
	A	B	C	A	B	C
	1	2	4	8	16	32

Escreva os números correspondentes aos efeitos que deseja e depois adicione-os. Se quiser que A seja de ruído, B de tom, e C simultaneamente de tom e ruído, some 8, 2, 4 e 32 e terá 46 (a ordem das cadeias e a ordem das cadeias de caracteres que se seguem ao comando PLAY). Os melhores efeitos são obtidos com o canal A — não hesite em experimentar. Agora já pode escrever sinfonias. No entanto terá dificuldade em saber o que uma certa parte de uma cadeia irá fazer. Para aliviar o problema, a cadeia de música pode conter "comentários", incluídos entre pontos de exclamação. Por exemplo:

1098 LET z\$=z\$+'CDcE3Ge4_6f! fim do verso 75 ! egeA''

O comando PLAY "saltará" simplesmente os comentários inseridos na cadeia. Se tiver um instrumento musical com MIDI — sintetizadores, simuladores de percussão ou sequenciadores — os Spectrum + 2 e + 3 poderão tocá-lo através de PLAY, usando até 8 canais de música. O comando PLAY é exactamente igual ao descrito, excepto o facto de cada cadeia incluir um "Y" seguido por um número entre 1 e 16. O número depois do Y determina o canal para o qual são enviados os dados da música. Podem ser usadas até oito cadeias, as primeiras três soam também através do televisor, mas pode-se fechar o som dele. Pode-se também enviar códigos de programação de MIDI através do comando PLAY, usando "Z" seguido pelo número de código. O volume de som é calculado e transmitido a 8 vezes o valor de V (— portanto "V6" corresponde a 48 de volume).

Por fim uma breve lista dos parâmetros que podem ser usados na cadeia de um comando PLAY, com os valores que podem assumir...

CADEIA FUNÇÃO

- a - g) Especifica a altura da nota dentro da oitava corrente.
- A - G) Especifica que a nota que se segue deve ser abafada.
- \$ Especifica que a nota que se segue deve ser alterada.
- # Especifica o número da oitava a usar (seguido por 0 a 8).
- l-12 Especifica o comprimento das notas.
- & Especifica que haverá uma pausa.
- Especifica que uma nota ligada será tocada.
- N Separa dois números.
- V Especifica o volume a usar (seguido por 0-15).
- W Especifica o efeito de volume a usar (seguido por 0-7).
- U Especifica que o efeito de volume será usado numa cadeia.
- X Especifica a duração do efeito de volume (seg. por 0-65536)
- T Especifica o tempo da música (seguido por 60-240).
- () Especifica que a frase incluída deve ser repetida.
- !! Especifica que o comentário incluído será ignorado.
- H Especifica que o comando PLAY deve parar.
- M Especifica o canal a ser usado (seguido por 1-63).
- Y Especifica o canal MIDI a ser usado (seguido por 1-16).
- Z Especifica o código de programação MIDI (seguido pelo número de código).

As impressoras

Com o Spectrum 48K foi comercializada a ZX Printer, uma pequena impressora electroestática, de custo muito baixo, que servia principalmente para listagens. A seu tempo o papel metalizado que usava subiu de preço a tal ponto que se tornou excessivamente dispendioso. Outras impressoras surgiram, que como a ZX Printer podiam ser montadas directamente sobre o conector de expansão e que usavam papel térmico ou mesmo papel comum, como a Seikosha GP50-S. Mas essas impressoras deixaram de ser fabricadas porque as impressoras profissionais foram sendo aperfeiçoadas, a sua procura aumentou e os seus preços desceram.

A ligação das impressoras profissionais aos Spectrum não pode ser feita directamente sobre o conector de expansão. Exige uma interface — isto é, um dispositivo que transmita os dados entre o computador e a impressora. Há dois tipos de transmissão: série ou paralelo.

O Spectrum + 2 possui uma interface RS232, para comunicações ou para a ligação de uma impressora série.

O Spectrum + 3 possui não só uma interface série RS232 mas também uma interface Centronics, para a ligação de uma impressora paralelo.

Na ligação série, os dados são transmitidos um a um, enquanto na ligação paralelo o são por grupos. Consequentemente, a transmissão série tem de ser feita à razão de um número de caracteres na unidade de tempo, para não se tornar demasiado lenta, mas quanto mais rápida for a transmissão maior é a probabilidade de erro. Daí que a maior parte das impressoras profissionais modernas usem a ligação paralelo, e é por isso que o Spectrum + 3 dispõe de uma interface desse tipo. Mas por outro lado este tipo de ligação só é eficiente quando a distância é curta, como a do computador para a impressora. Para distâncias maiores é essencial a ligação série. Daí que no Spectrum + 3 tenha sido conservada a interface série RS232, com vista à comunicação com outros computadores, se necessário, ou para a utilização de uma impressora desse tipo, se já se dispuser de uma.

Existem interfaces adicionais, não comercializadas pela Amstrad, que permitem usar impressoras paralelo com o Spectrum + 2. Algumas necessitam de carregamento de "software" e da introdução dos códigos da impressora, outras não. Este é o caso do Multiprint, da J.G. Components, que permite também obter cópias de ecrã com várias dimensões e proporções.

Note-se que as impressoras profissionais só podem trabalhar nas interfaces integradas no Spectrum + 2 e + 3 quando no modo 128 BASIC ou no + 3 BASIC, respectivamente. Para as usar no modo 48K BASIC é necessário dispôr de uma segunda interface, montada no conector de expansão — quem já dispuser de uma dessas interfaces, série ou paralelo, poderá usá-la no Spectrum + 2, mas não é aconselhado fazer isso no Spectrum + 3 porque as ligações do conector deste são diferentes...

As pequenas impressoras, como a Seikosha GP50-S só funcionam no modo 48K. No entanto a sua montagem no conector do Spectrum + 3 também é desaconselhada.

O "plotter" Tricom pode trabalhar no Spectrum + 2, tanto no modo 48K, se a sua interface estiver ligada ao conector de expansão, como no modo 128K, se estiver directamente ligada à interface Centronics. No Spectrum + 3 só pode ser ligada à interface Centronics.

Recorda-se que os Spectrum + 2 e + 3 não são acompanhados pelo cabo da impressora. Essa é uma prática comum, pois que o cabo é normalmente adquirido com a impressora e não com o computador. Note-se que o cabo para a ligação em série é idêntico ao que era usado para o Sinclair QL.

Impressoras série

Spectrum + 2

Como acontece com todas as impressoras profissionais, é conveniente proceder a um ensaio antes da aquisição. Para isso pode ser usado o seguinte programa:

```
10 PRINT "Este programa..."
20 LLIST
30 LPRINT "...IMPRIME O CONJUNTO DE CARACTERES:"
40 FOR n=32 TO 255
50 LPRINT CHR$ n;
60 NEXT n
```

Note-se que no menu de edição do 128 BASIC existe a opção Print que equivale a LLIST e que foi ali incluída apenas para facilitar a obtenção de listagens.

Se a impressora não funcionar, a causa normal será uma disparidade no ritmo da transmissão de dados (*baud rate*). O Spectrum + 2, quando a ligação é feita, assume que a "baud rate" é de 9600. Se a da impressora for diferente, haverá em primeiro lugar que consultar o seu manual, para saber esse valor. Depois introduz-se o comando:

```
FORMAT "p"; valor da "baud rate"
```

Um problema com as impressoras série é o de que nem todas dispõem do chamado "hardware handshaking", que assegura que só sejam transmitidos a elas os sinais que têm os valores correctos. Se a impressora não dispuser de "handshaking" haverá que ligar entre si os pernos 4 e 5 da interface RS232 — veja-se o apêndice PERIFÉRICOS. Nesse caso, se a "baud rate" for muito elevada, podem desaparecer caracteres na impressão. O mal, obviamente, será da impressora,

e não do computador. Tudo quanto se pode fazer é tentar baixar a "baud rate", mas o melhor é ensaiar a impressora antes de a comprar...

Só quando se recorre aos comandos LPRINT e LLIST se pode ter a certeza de obter uma impressão correcta, pois que eles anulam os códigos usados para definir as cores de INK, PAPER, etc. — códigos esses que se fossem transmitidos poderiam ter consequências desagradáveis, pois são usados pelas impressoras para outros fins. Pela mesma razão não é possível introduzir através do BASIC numa impressora as sequências de escape e os códigos necessários à execução de efeitos especiais, como o itálico, o sublinhado, etc.. Para tal é necessário dispôr de um programa utilitário adequado, em código de máquina, ou de uma impressora que possa definir directamente esse ou outros efeitos, como por exemplo a largura dos caracteres.

O comando COPY permite, tal como nos Spectrum 48K, copiar o écran. Para ensaiar a impressora, depois de executar o programa acima, introduza o comando:

COPY

A cópia deverá demorar de 18 a 30 segundos a iniciar-se. Se a impressora não copiar perfeitamente o que mostra o ecrã haverá que ensaiar outro modelo. As impressoras adequadas são as compatíveis com os códigos Epson e devem possuir um modo gráfico de imagem de bit de quádrupla densidade. A maior parte das impressoras profissionais existentes no mercado correspondem a essa especificação.

Para parar a impressora basta primir a tecla de BREAK. Quase todas as impressoras modernas dispõem de uma memória para armazenamento temporário (*buffer*) em que o texto é guardado antes de ser impresso. Nesse caso o computador parará logo que for premida a tecla BREAK, mas a impressora continuará a funcionar até o *buffer* ficar vazio.

Se se usarem os comandos LLIST, LPRINT ou COPY sem a impressora estar ligada, o computador parará e haverá que premir BREAK para ele continuar a funcionar.

Experimente:

```
10 FOR n=31 TO 0 STEP -1
20 PRINT AT 31-n,n; CHR$(CODE"0"+n);
30 NEXT n
```

No ecrã surgirá uma série de caracteres em diagonal, de alto a baixo, até o programa parar para perguntar se se pretende fazer Scroll.

Substitua AT 31-n,n na linha 20 por TAB n. O programa terá exactamente o mesmo efeito.

Mude PRINT na linha 20 para LPRINT. Dessa vez o programa só parará quando se fizer BREAK.

Mude TAB n de novo para AT 31-n mantendo LPRINT. Dessa vez obterá uma única linha de símbolos. Isso acontece porque os caracteres não são impressos imediatamente, mas sim guardados no "buffer" até que tenha sido acumulada uma linha ou que o "buffer" seja esvaziado por outra causa. Consequentemente, a impressão só ocorre quando:

- 1 — O "buffer" esteja cheio.
- 2 — Depois de uma declaração LPRINT que não termine numa vírgula ou ponto e vírgula.
- 3 — Quando uma vírgula, um apóstrofe ou um TAB determinem uma nova linha.
- 4 — No fim de um programa, se algo ficou por imprimir.
- 5 — Segundo a impressora, quando ela estiver "Off line".

O número 3 explica porque o programa com TAB funciona assim. O número de linha apostro a AT é ignorado e a posição de LPRINT (como a de PRINT) passa para número de coluna. A instrução AT nunca pode servir para levar a impressora a uma determinada linha.

Spectrum + 3

Introduzir de início o comando:

```
FORMAT LPRINT "R"
```

em que o "R" funciona como uma abreviatura da interface RS232.

Se a impressora não funcionar, a causa normal será uma disparidade no ritmo da transmissão de dados (*baud rate*). Os Spectrum + 2 e + 3, quando a ligação é feita, assumem que a "baud rate" é de 9600. Se a da impressora for diferente, haverá em primeiro lugar que consultar o seu manual, para saber esse valor. Depois introduz-se o comando;

```
FORMAT LINE valor da "baud rate"
```

e repete-se o comando FORMAT LPRINT "S". Se se quiser voltar do modo Centronics (ou seja ao da ligação em paralelo), bastará introduzir o comando;

```
FORMAT LPRINT "C"
```

Tal como para o Spectrum + 2, um problema com as impressoras série é o de que nem todas dispõem do chamado "hardware handshaking", que assegura que só sejam transmitidas a elas os sinais que têm os valores correctos. Se a impressora não dispuser de "handshaking" haverá que ligar entre si os pernos 4 e 5 da interface RS232 — veja-se o apêndice PERIFÉRICOS. Nesse caso, se a "baud rate" for muito elevada, podem desaparecer caracteres na impressão. O mal, obviamente, será da impressora, e não do computador. Tudo quando se pode fazer é tentar baixar a "baud rate", mas o melhor é ensaiar a impressora antes de a comprar..

Em tudo o mais, o uso das impressoras série no Spectrum + 3 é idêntico ao do Spectrum + 2.

Impressoras paralelo

Quanto às impressoras paralelo, não é necessário mais do que ligá-las. O Spectrum + 3 foi concebido para as usar.

Uma característica extremamente importante do Spectrum + 3 é a possibilidade de introduzir através do + 3 BASIC as "sequências de escape", ou sejam os códigos que permitem que a impressão seja feita com caracteres pica ou elite, itálico, negro, condensado, alargado, etc.. Para isso usa-se o comando:

FORMAT LPRINT "U"

e introduzem-se os códigos indicados no manual da impressora, seguidos pelo texto a imprimir e por novos códigos e novos textos conforme os tipos de impressão que sucessivamente forem desejados. A inserção pode ser feita directamente ou num programa.

Convém notar que depois de introduzir o comando **FORMAT LPRINT "U"** os comandos usuais do Spectrum, como o **LLIST**, deixam de ser aplicáveis. Mas a solução é simples: basta fazer

FORMAT LPRINT "E"

e tudo volta ao princípio.

A cópia de ecrã

O comando **COPY** tem três formas no Spectrum + 3:

- 1 — **COPY**, normal.
- 2 — **COPY EXP**, que permite obter cópias com as cores convertidas em tons de cinzento, graças a uma maior ou menor densidade dos pontos de impressão.
- 3 — **COPY EXP INVERSE**, em que as imagens sobre fundo negro são invertidas para poupar a fita da impressora.

No caso de **COPY EXP** ou **COPY EXP INVERSE**, se as proporções da cópia não forem correctas, deve-se antepôr **POKE 23419,15** aos comandos indicados. Se mesmo assim não se obtiver o resultado desejado, o valor 15 pode ser substituído por outro, até 8.

A Memória

Bits, bytes e outras coisas

Os computadores "falam" uma linguagem que para eles é a mais simples possível, pois têm somente um elemento básico: o "bit", que por sua vez só pode ter dois valores: 0 ou 1, pois correspondem a estados que são o inverso um do outro: ligado/desligado, entrada/saída, sim/não, voltagem alta/voltagem baixa, etc. É o "código binário", assim chamado por ser constituído somente por dois algarismos ou "dígitos". Donde a designação de "bit", que é abreviatura de "binary digit".

Acontece também que "bit", na língua inglesa significa "uma dentadinha" - o que influenciou o resto da nomenclatura: um grupo de 4 bits é um "nibble" — um pedacinho arrancado com uma dentadinha — enquanto um grupo de 8 bits é um "byte" — um bom pedaço arrancado com uma boa dentada.

Um byte corresponde a $2^8=256$ combinações, pelo que o seu valor pode ser expresso entre 0 e 255. O kilobyte ou Kb não é igual a 1000 bytes, mas sim a $2^{10}=1024$ bytes.

Todas as informações registadas num computador são-o em bytes, e cada lugar da memória onde os bytes são guardados tem um endereço. O microprocessador dos Spectrum reconhece $256 * 256 = 65536$ endereços numerados de 0 a 65535, e em cada endereço podem ser inseridos dois bytes.

O conteúdo dos endereços 0 a 16383 pode ser conhecido (lido), mas não pode ser alterado. É a memória permanente, ou "só de leitura" (ROM = *Read Only Memory*).

O conteúdo dos endereços 16384 a 65535 pode ser lido e alterado (escrito). É a memória virtual, de acesso aleatório (RAM = *Random Access Memory*).

Nos Spectrum primitivos a RAM dispunha de espaço para $65536-16384=49152$ bytes. Portanto, como cada kilobyte equivale a 1024 bytes, tinham $49152/1024=48$ Kb, donde a designação "48K". (Nos países de língua inglesa usa-se o sufixo "K" em vez da abreviatura "Kb").

Nos Spectrum + 2 e + 3 existem mais endereços do que aqueles que o microprocessador pode reconhecer: nada menos de 131 072 bytes de RAM (donde a designação de "128 K" pois $131\ 072/1024=128$ Kb). O Spectrum + 2 tem 32768 bytes de ROM (ou $32768/1024=32$ Kb), enquanto o Spectrum + 3 tem 65536 Kb de ROM, ou sejam $65536/1024=64$ Kb. No entanto, o microprocessador trabalha a cada momento somente com 16 Kb de ROM e 48 Kb de RAM, tal como uma pessoa lê somente uma página de um livro de cada vez, é o que se chama "paginação" (*paging*). Para se conhecer o conteúdo de um endereço, usa-se a função **PEEK**. Por exemplo: para se conhecer os primeiros 21 bytes no ROM e dos seus endereços, pode-se fazer:

```
10 PRINT "Endereço"; TAB 9; "Byte"
20 FOR a=0 TO 20
30 PRINT a; TAB 9; PEEK a
40 NEXT a
```

Para quem não conhece o código de máquina, esses bytes nada significam, mas o computador entende-os.

para alterar o conteúdo de um endereço — se ele pertencer à RAM — usar-se-á o comando **POKE**, que tem a forma:

POKE endereço, conteúdo

em que o *endereço* e o *conteúdo* são expressões numéricas. Por exemplo, se se inserir:

POKE 31000,57

e se em seguida se fizer:

PRINT PEEK 31000

obter-se-á como resposta 256.

A possibilidade de usar o comando **POKE** confere ao utilizador um grande domínio do computador, mas não se deve recorrer a ela sem um profundo conhecimento do código de máquina, pois se mal aplicada pode conduzir à perda de programas que demoraram longo tempo a preparar. No entanto, em caso algum pode danificar o computador.

UM MANUAL COM OS DADOS NECESSÁRIOS PARA A PROGRAMAÇÃO AVANÇADA EM CÓDIGO DE MÁQUINA PODERÁ SER OBTIDO A UM PREÇO NOMINAL ATRAVÉS DO DISTRIBUIDOR.

O mapa da memória e as variáveis do sistema

Os bytes na memória, do endereço 23296 a 23733 no Spectrum + 2 e 23734 no Spectrum + 3 são reservados para fins específicos do sistema — rotinas para a paginação e que se designam por **variáveis do sistema**. Há largas diferenças entre as variáveis do sistema nos modos 48K BASIC, 128K BASIC e + 3 BASIC. No modo 48K BASIC não existem variáveis ou rotinas abaixo de 23552 — o espaço é ocupado por uma memória temporária (*buffer*) usado para comandar a impressora. É muito usado para pequenas rotinas de código de máquina nos Spectrum 48K e Plus e se uma dessas rotinas for usada no 128K BASIC ou no + 3 BASIC o computador parará (isto é, fará "crash"). Os programas antigos que usem **PEEK**, **POKE** e **USR** devem ser carregados no modo 48 BASIC (ou no + 3 BASIC e transferidos para o 48 BASIC pelo comando **Spectrum**). É também possível que um desses programas active as portas de entrada/saída do + 3. Nesse caso há que introduzir:

OUT 32765,48

A lista das variáveis do sistema e o mapa da memória dos Spectrum + 2 e + 3 está contida no manual de programação avançada em código de máquina.

Vias e Canais, Entradas e Saídas

Vias e canais (Streams — Channels)

O Spectrum + 3 pode “ler” dados do teclado através de INPUT e INKEY\$ e pode “escrever” no ecrã ou numa impressora usando PRINT e LPRINT. No entanto esses comandos não são mais do que abreviaturas concebidas para evitar que o utilizador tenha de recorrer a procedimentos mais complexos. O que na realidade eles fazem é enviar dados para um canal de comunicação (*channel*).

Há três canais disponíveis em BASIC:

- * O ecrã (canal S)
- * O teclado (canal K)
- * A impressora (canal P)

O do ecrã funciona somente em saída, o do teclado é de entrada e saída, e o da impressora é só de saída (se usar a interface paralelo PRINTER) ou de entrada e saída (se usar a interface série RS232).

Para se obter acesso a um canal há que o *abrir*, ligando-o a uma corrente (*channel*). Partindo do BASIC, isso faz-se usando um comando do tipo:

```
OPEN #4. "p"
```

o que significa “ligar a corrente 4 ao canal da impressora”. As vias podem ser comparadas a vias do caminho de ferro que “fazem agulha” entre os vários canais quando eles são indicados pelo respectivo número. Isso permite escrever programas que enviam informações para os vários dispositivos auxiliares (*devices*), sem usar comandos diferentes. É o que se chama um “sistema” de entradas e saídas (*input/output ou I/O*) redirigível (*redirectable*).

Isso pode parecer complicado, e é essa a razão da existência dos comandos PRINT e LPRINT. Mas PRINT e LPRINT são na realidade os mesmos comandos. Quando se trabalha com BASIC há três vias (*streams*) normalmente abertas.

- * A *Stream* #1 está ligada ao dispositivo “teclado” (K, de *Keyboard*) e é usada por INPUT e INKEY\$
- * A *Stream* #2 está ligada ao dispositivo “ecrã” (S de *Screen*) e é usada por PRINT e LIST.
- * A *Stream* #3 está ligado ao dispositivo “impressora” (P de *Printer*) e é usado por LPRINT e LLIST.

Não obstante todos estes comandos podem ser redirigidos para outros dispositivos se lhe for adicionado o símbolo # (*cardinal*) seguido pelo número da *stream*. Por exemplo:

```
PRINT #1; "Parte inferior do ecrã"
```

imprime o texto entre aspas na parte inferior do ecrã, normalmente reservado para as mensagens do computador. Do mesmo modo:

```
PRINT #3 "Quem precisa de LPRINT?"
```

passará o texto entre aspas para a impressora. Mas LPRINT também pode ser usado como PRINT...

```
LPRINT #2; "Quem diria?"
```

passará o texto entre aspas para o ecrã. Isto é LPRINT #2 equivale a PRINT.

A redirigibilidade destes comandos, embora pareça uma simples curiosidade, simplifica muito os programas cujos resultados devem ir para o ecrã ou para a impressora como convier ao utilizador:

```
10 REM Programa de quadrados para imprimir
20 INPUT "Quer imprimir os resultados?"; a$
30 LET stream=2
40 IF a$="y" OR a$="Y" THEN LET stream=3
50 FOR n=0 TO 10
60 PRINT #stream;n,n*n
70 NEXT n
```

O Spectrum + 3 pode usar 16 *streams*. Como três são usados pelo BASIC e um o é internamente, restam 12 que podem ser usados, por exemplo, para dirigir para o ecrã os caracteres recebidos através da interface RS232.

```
10 Programas para ler dados da RS232
20 FORMAT LINE 9600
30 FORMAT LPRINT "r"
40 OPEN #4."p"
50 PRINT INKEY$ #4;
60 GO TO 50
```

Para se ler directamente da memória os dados recebidos pela RS232 substitui-se a linha 50 por:

```
POKE endereço, CODE (INKEY$ #4)
```

Como se disse antes o ecrã e a interface paralelo PRINTER só podem ser usados pelo Spectrum para a saída de dados (*output*). Consequentemente, se se usar algo como PRINT INKEY\$2 o resultado é uma mensagem de erro.

Teóricamente pode-se redirigir o *output* do BASIC do écran para a impressora, usando:

```
10 CLOSE #2
20 OPEN #2, "p"
```

Então, tudo quanto deveria surgir no ecrã através de PRINT passará a ser impresso. Na prática, se isso for feito durante a edição dos programas, pode ter resultados imprevistos.

As vias e os canais são de interesse quase académico no Spectrum + 3 normal, mas alguns periféricos e certas extensões do BASIC usam-nos.

Entradas e saídas (IN e OUT)

O microprocessador, que é o coração de todo e qualquer computador, pode "ler" o conteúdo do ROM e da RAM, usando PEEK e "escrever" na RAM usando POKE. Mas o tipo da memória é-lhe indiferente: para ele, tudo quanto existe são 65536 endereços de memória nos quais pode ler bytes ou para os quais pode enviar bytes (mesmo que nem sempre sejam usados). Pode-se dizer assim, por simples analogia, que os trata como se fossem 65536 portas de entrada e saída: *input/output ports* ou *I/O ports*. É por elas que o microprocessador comunica com o teclado e a impressora, por exemplo, e comanda a memória do gerador de sons. Algumas podem ser usadas pelo BASIC através da função IN e do comando OUT, mas outras devem ser evitadas porque podem determinar uma paragem do processamento (*crash*), resultando na perda de programas e dados.

IN actua como PEEK. Tem a forma de:

IN endereço

e permite "ler" um byte nesse endereço.

OUT actua como POKE. A sua forma é:

OUT endereço, valor

e "escreve" o valor indicado no endereço designado.

O byte escrito ou lido tem 8 bits que são usualmente referidos por:

D7, D6, D4, D3, D2, D1, D0

A letra D significa DATA.

Os endereços mais usados em conjunto com a função IN são os seguintes:

O teclado está dividido em 8 meias filas de 5 teclas cada:

```
IN 65278 lê a meia-fila CAPS SHIFT a V
IN 65022 lê a meia-fila A a G
IN 64510 lê a meia-fila Q a T
IN 63486 lê a meia-fila 1 a 5 (e "joystick 2")
IN 61438 lê a meia-fila 0 a 6 (e "joystick 1")
IN 57342 lê a meia-fila P a Y
IN 49150 lê a meia-fila ENTER a H
IN 32766 lê a meia-fila espaço a B
```

Note-se que os endereços correspondem a $254 + 256 \times (255 - 2 \uparrow n)$, em que o valor de n vai de 0 a 7.

No byte lido, os bits D0 a D4 referem-se às cinco teclas nas respectivas meias-linhas. D0 é a tecla do lado exterior e D4 a mais próxima do meio do teclado. Se a tecla estiver premida, o bit é 0; se não estiver o bit é 1.

O bit D6 está ligado à interface do leitor-gravador e o seu valor, quando não recebe dados, é aleatório.

No "joysticks 1", bit 0 é fogo, bit 1 é acima, bit 2 é abaixo, bit 3 é direita, bit 4 é esquerda. No "joystick 2" os valores são simétricos: bit 0 é esquerdo, bit 1 é direito, bit 2 é abaixo, bit 3 é acima, bit 4 é fogo.

OUT 254 actua sobre o som em D4; o sinal de SAVE em D3; e a cõr de BORDER em D2, D1 e D0.

Os endereços 254, 247 e 239 são reservados. No Spectrum + 2 destinam-se às "ZX microdrives", à rede ZX, à interface RS232 e ao teclado numérico suplementar (não disponível).

O endereço 32765 dá acesso à memória suplementar. Fazer OUT 32765 a partir do BASIC resulta quase sempre na paragem do processamento e à perda do programa e dos dados.

O endereço 49149 dá acesso aos registos do gerador de som. OUT 65533 escreve um registo e IN 65333 faz a leitura. O uso judicioso dos registos permite gerar sons enquanto está a correr um programa em BASIC, mas importa ter em conta que eles comandam também as interfaces RS232, "keypad", e MIDI no Spectrum + 2 e as RS232/MIDI e AUX no Spectrum + 3.

Exclusivamente no Spectrum + 3:

O endereço 4093 é usado para a interface paralelo (a PRINTER). Quando se faz IN 4093, o bit 0 refere o sinal gerado pela impressora. Se o bit é 1, a impressora ou está "off-line", ou desligada, ou não existe.

O endereço 8189 é usado entre outras coisas para gerir a ROM, a transferência RAM/ROM, o comando do motor do leitor/gravador de disquetes e a porta STROBE da interface paralelo. Não é conveniente usar o comando OUT 8189 sem um conhecimento profundo da sua actuação.

O endereço 12285 pode ser usado para ler o registo principal controlador da unidade de disquetes mas isso não terá muita utilidade sem um profundo conhecimento da maneira como ele funciona.

O endereço 16381 pode ser usado para ler o registo de dados do controlador da unidade de disquetes. É também possível usá-lo para escrever, mas isso é desaconselhável sem um profundo conhecimento das consequências, pois pode perturbar as operações de SAVE e LOAD, tornando-as falíveis.

O programa seguinte mostra como o teclado funciona:

```
10 FOR n=0 TO 7: REM Numero de meia-fila.
20 LET a=254+256*(255-2 ↑ n)
30 PRINT AT 20,0; IN a: GOTO 30
```

Depois de premir as teclas de cada meia-fila (a começar pela de CAPS SHIFT a V). Quando acabar uma meia-fila prima [BREAK] e insira:

```
NEXT n
```

O barramento (*Bus*) do conector EXPANSION I/O na traseira do Spectrum + 3 dá acesso a todos os endereços, dados e comandos, o que permite fazer quase tudo o que o microprocessador Z80, só por si, pode fazer — embora na prática o funcionamento interno do computador o possa impedir. Na secção LIGAÇÃO A PERIFÉRICOS encontra-se um esquema do conector.

Mensagens

Quando o computador ao executar um programa em BASIC pára por qualquer razão natural ou por haver um erro, na parte inferior do écran surge uma mensagem constituída por um número ou uma letra (para referência à tabela seguinte), uma breve mensagem explicando o que aconteceu, o número da linha e o número da instrução dentro da linha onde o programa parou. (um comando é referido como linha 0). Dentro de uma linha, a instrução 1 é a primeira, a instrução 2 é a que se segue ao primeiro dois pontos ou a THEN, e assim por diante.

O comportamento de CONTINUE está muito ligado às mensagens. Normalmente, quando se interrompe um programa, CONTINUE arranca da linha e da instrução especificadas na última mensagem, mas há excepções com os relatórios 0, 9 e D.

A tabela seguinte contém todas as mensagens e as circunstâncias em que podem ocorrer. Antes de a ler convém recordar que no BASIC se dá o nome de:

- * **Comandos** — (*Commands*) — quando a acção por elas expressa é executada imediatamente. Por exemplo: RUN;
 - * **Instruções** — (*Statements*) — quando a acção por eles expressa só é executada no momento em que o programa o determina. Por exemplo: DRAW
 - * **Funções** — (*Functions*) — quando produzem um valor de qualquer espécie. Por exemplo: RND
 - * **Operadores lógicos** — (*logical operators*) — quando são usados para exprimir a lógica numa instrução ou comando. Por exemplo: AND, OR, NOT.
- e que:
- * **Argumento** — (*Argument*) é um valor usado por uma função para obter um certo resultado.
 - * **Atributos** - (*Attributes*) são os códigos dos caracteres ou pontos de imagem (pixels).
 - * **Constante** — (*Constant*) — Um número ou um grupo de uma ou mais letras ou de outros caracteres que se mantém inalterado durante a execução de um programa.
 - * **Matriz** — (*Array*) — Um grupo de dados (*data*) relacionados, inseridos juntamente numa secção da memória.
 - * **Variável** — (*Variable*) — Uma ou mais unidades de memória em que se introduzem valores para o computador usar durante a execução de um programa e que são identificadas por um nome ou uma letra. Os Spectrum distinguem entre variáveis numéricas (*numeric variables*) e variáveis de cadeia (*string variables*).

SIGNIFICADO DO CÓDIGO	SITUAÇÃO
<p>0 OK Execução completa ou salto para uma linha mais alta do que qualquer existente. Este relatório não muda a linha e a instrução de que CONTINUE arrancará.</p>	Qualquer
<p>1 NEXT without FOR A variável de controlo não existe (não foi definida por uma instrução FOR), mas há uma variável normal com o mesmo nome.</p>	NEXT
<p>2 Variable not found Uma variável simples foi usada antes de lhe ter sido atribuído um valor por uma declaração LET, READ ou INPUT, ou carregada da cassete ou disquete, ou colocada numa declaração FOR. Para uma variável subscripta acontecerá se a variável for usada antes de dimensionada numa instrução DIM, ou carregada de cassete ou disquete.</p>	Qualquer
<p>3 Subscript wrong Um subscripto está além da dimensão da matriz ou o número de subscriptos está errado. Se o subscripto for negativo ou maior que 65535, então resultará o erro B.</p>	Variáveis subscriptas Sub-cadeias
<p>4 Out of memory Não há espaço bastante na memória do computador para o que se pretende. Se o computador persistir em não sair dessa situação, pode-se ter de "limpar" a linha de comando usando [DELETE] e depois apagar uma ou duas linhas do programa (repondo-as depois) para obter espaço de manobra.</p>	LET, INPUT, FOR, DIM GO SUB, LOAD, MERGE. Por vezes durante a avaliação de uma expressão.
<p>5 Out of screen Uma instrução de INPUT tentou gerar mais de 23 linhas na metade inferior do ecrã. Também ocorre com PRINT AT 22,XX</p>	INPUT, PRINT AT
<p>6 Number too big Os cálculos conduziram a um número de cerca de 10^{38}</p>	Qualquer cálculo
<p>7 RETURN without GO SUB O número de RETURNS é maior que o de GO SUBs.</p>	RETURN

<p>8 End of file</p>	Operação com fichas. (microdrives, etc.)
<p>9 STOP statement Depois disto, CONTINUE não repetirá o STOP, mas prosseguirá com a instrução seguinte.</p>	STOP
<p>A Invalid argument O argumento de uma função é inadequado por qualquer razão).</p>	SQN, LN, ASN, ACS, USR (c/argumento de cadeia.
<p>B Interger out of range Quando um número inteiro (<i>integer</i>) é necessário, o argumento de vírgula flutuante é arredondado até ao número inteiro mais próximo. Se este está fora dos limites adequados, isso resulta no erro B. Para o acesso às matrizes ver também o erro 3.</p>	RUN, RANDOMIZE, POKE, DIM, GOTO, GOSUB, LIST, LLIST, PAUSE, PLOT, CHR\$, PEEK CHR\$, PEEK, USR (com argumentos numéricos). No acesso às matrizes.
<p>C Nonsense in BASIC O texto do argumento (de uma cadeia) não forma uma expressão válida. O aumento de uma função ou comando está completamente errado.</p>	VAL, VAL\$
<p>D BREAK — CONT repeats Premiu-se [BREAK] durante a operação de um periférico. Normalmente, o uso de CONTINUE depois desta mensagem repete a instrução. Comparar com a mensagem L, ou a barra dos espaços.</p>	LOAD, SAVE, VERIFY, MERGE. Também usada quando o computador pergunta 'scroll ?' e se prime N, [BREAK]
<p>E Out of DATA Tentou-se ler (READ) para além do fim da lista de dados (DATA).</p>	READ
<p>F Invalid file name SAVE tentado com o nome em vazio ou mais de 10 caracteres.</p>	SAVE
<p>G No room for line Não há espaço livre na memória para acomodar uma nova linha no programa.</p>	Ao introduzir uma linha no programa

H STOP in INPUT Em INPUT introduziram-se dados começados por STOP. Ao contrário da mensagem 9, depois da mensagem H, CONTINUE comportar-se-á normalmente, repetindo a instrução de INPUT.	INPUT
I FOR without NEXT Há um ciclo FOR a ser executado zero vezes (como em FOR n=1 TO 0) e a correspondente instrução NEXT não pode ser encontrada.	FOR
J Invalid I/O device	Nas microdrives e outros periféricos.
K Invalid colour O número especificado não tem um valor adequado.	INK, PAPER, BORDER FLASH, BRIGHT, INVERSE, OVER; também depois de um dos correspondentes caracteres de comando.
L BREAK into program Premiu-se [BREAK] e isso foi detectado entre duas instruções. A linha e o número da instrução na mensagem referem-se à instrução antes de [BREAK] ter sido premido, mas CONTINUE parte da instrução seguinte (considerando os saltos previstos), de modo que não serão repetidas quaisquer instruções.	Qualquer
M RAMTOP no good O número especificado para o RAMTOP é demasiado grande ou demasiado pequeno.	CLEAR; possivelmente em RUN
N Statement lost Salto para uma instrução que já não existe.	RETURN, NEXT, CONTINUE
O Invalid Stream	Microdrives, etc.
P FN without DEF Função definida pelo utilizador usada sem o correspondente DEF no programa.	FN

Q Parameter error Número errado de argumentos, ou um argumento de tipo errado (cadeia em vez de número ou vice-versa).	FN
R Tape loading error Encontrou-se uma ficha numa cassete mas por qualquer razão não pode ser lida ou verificada.	VERIFY, LOAD ou MERGE
a MERGE error MERGE ! não é executada por qualquer razão — ou o tamanho da ficha ou o tipo desta.	MERGE !
b Wrong file type Uma ficha de tipo inadequado foi chamada durante a operação do disco de memória, por exemplo uma ficha do tipo CODE em LOAD “nome da ficha”.	MERGE !, LOAD !
c CODE error O tamanho da ficha vai além do topo da memória.	LOAD ficha CODE
d Too many brackets Demasiados parênteses envolvendo uma frase repetida num dos argumentos.	PLAY
e File already exists O nome indicado para a ficha já existe.	SAVE !
f Invalide filename O nome indicado é vazio ou tem mais de 10 caracteres.	ERASE !
h File does not exist Não há ficha no disco de memória que tenha o nome indicado.	LOAD ! ERASE !
i Invalid device O nome do dispositivo que se segue ao comando FORMAT não existe ou corresponde a um dispositivo físico.	FORMAT
j Invalid baud rate A “baud rate” para a RS232 foi posto a zero.	FORMAT
k Invalid note name O comando PLAY encontrou uma nota ou outro comando que não reconheceu, ou um comando em minúsculas.	PLAY

- l Number too big** **PLAY**
Um parâmetro para um comando com uma ordem de grandeza demasiado grande.
- m Note out of range** **PLAY**
Uma nota para além da gama do microprocessador de som.
- n Out of range** **PLAY**
Um parâmetro para um comando que é demasiado pequeno ou demasiado grande. Se o erro for muito grande ter-se-á a mensagem l.
- o Too many tied notes** **PLAY**
Tentou-se ligar demasiadas notas.

Mensagens exclusivas do Spectrum + 3

- Bad filename** **CAT, COPY**
O nome da ficha usado em qualquer dos comandos não se conforma com as especificações definidas nas páginas 92/93 deste manual. **ERASE, FORMAT, LOAD, MERGE, MOVE, SAVE**
- Bad parameters** **Improvável**
Um dos valores passados ao + 3 DOS pelo BASIC está fora dos limites. É improvável que este erro surja.
- RIC CRC data error** **CAT, COPY**
A verificação cíclica de redundância (*cyclic redundancy check* ou *checksum byte*) é incorrecta. É um erro raro produzido quando uma disquete está corrompida. **ERASE, LOAD, MERGE, MOVE,**
- Code length error** **LOAD... CODE**
Tentou-se carregar da disquete uma ficha CODE com um comprimento maior que o indicado no comando LOAD.
- Destination cannot be wild** **COPY... TO**
Tentou-se dar uma máscara (*wild card*) à ficha de destino num comando COPY quando a fonte também usa caracteres de máscara. Neste caso o destino só pode ser a letra de uma unidade de disquetes.

- Destination must be drive** **COPY... TO**
O nome da ficha fonte num comando COPY contém caracteres de máscara, mas o destino é um único nome de ficha. Neste caso, o destino só pode ser a letra de uma unidade de disquetes.
- Directory full** **COPY, SAVE**
Tentou-se criar uma 65.^a ficha na disquete. (O directório comporta normalmente 64 fichas).
- Disk full** **COPY, SAVE**
A gravação das fichas num disco fez uso do último byte de espaço livre. O comando CAT pode ser usado antes para verificar se há suficiente espaço livre antes de se tentar gravar ou copiar uma ficha. Ao copiar, as fichas parcialmente copiadas devem ser apagadas. Ao gravar, é possível que uma parte da ficha tenha ficado na disquete. Essa parte deve ser apagada porque qualquer tentativa para a usar falhará.
- RIC Disk has been changed** **CAT, COPY**
Ao executar um comando, o + 3 DOS notou que a disquete inserida não é a mesma que estava presente no início da execução do comando. Se um programa de código de máquina abriu fichas na disquete e depois esta foi mudada, e um comando do + 3 BASIC tentar o acesso à disquete, então será apresentada esta mensagem. **ERASE, LOAD, SAVE**
- Disk is not bootable** **LOAD """"**
Foi tentado carregar o programa de arranque (*boot-strap*) de uma disquete que não tem o sector de arranque (*boot*).
- RIC Disk is write protected** **COPY, ERASE, FORMAT, MOVE, SAVE**
Foi feita uma tentativa para escrever numa disquete cujo furo de protecção contra escrita estava aberto. A protecção contra escrita pode ser desactivada deslizando a patilha que tapa o furo, antes de se escrever na disquete.
- Drive B: is not present** **FORMAT**
Foi feita uma tentativa para usar o comando FORMAT na unidade de disquete externa (*drive B*) quando ela não estava ligada.

Drive in use Foi feita uma tentativa para usar uma unidade de disquetes que tinha fichas abertas. É muito improvável que este erro surja.	Improvável
Drive not found Um nome de ficha usado num comando de disquete contém uma letra designando uma unidade não presente. Por exemplo: ERASE "C: manuel".	CAT, COPY, ERASE, LOAD, MERGE, MOVE, SAVE,
RIC Drive not ready Foi tentado um comando de disquete quando a unidade não estava pronta. Em regra isso acontece por não haver disquete inserida. Normalmente basta inserir a disquete e escrever R.	CAT, COPY, ERASE, FORMAT, LOAD, MERGE, MOVE, SAVE
End of File found Foi feita uma tentativa para ler um byte para além da posição de fim da ficha (<i>end-of-file</i>). É improvável que esta mensagem seja vista.	Improvável
File already exists O nome da ficha de destino num comando MOVE (que está a ser usado para mudar o nome de uma ficha) já existe.	MOVE... TO
File already in use Se um programa em código de máquina abriu a fichas 1... 3, então um comando do + 3 BASIC poderá falhar quando tentar abrir uma ficha que já está aberta. É improvável que esse erro surja.	Improvável, COPY, LOAD, MERGE, SAVE
File is read only Tentou-se actualizar, apagar ou gravar uma ficha que tem um atributo de protecção colocado com o comando MOVE <i>nome da ficha</i> TO "+P". Usar o comando MOVE <i>nome da ficha</i> TO "-P" para retirar a protecção.	COPY, ERASE, MOVE, SAVE
File not found O nome da ficha dado a um dos comandos de leitura refere-se a uma ficha que não existe.	COPY, ERASE, LOAD, MERGE, MOVE.
File not open Um comando de disquete tentou operar numa ficha que não estava aberta. É muito improvável que este erro ocorra.	Improvável

File too big Uma tentativa foi feita para escrever uma ficha de comprimento superior a 8 Kb. É muito improvável que este erro ocorra.	Improvável
Invalid Attribute O caracter de atributo que se segue a + ou - num comando MOVE não é P, S ou A (ou há mais de um caracter depois de + ou de -).	MOVE... TO
Invalid drive A letra de uma unidade de disquete diferente de A: ou B: foi especificada num comando FORMAT.	FORMAT
RIC Missing Adress Mark Um sector da disquete não contém a informação habitualmente usada pelo sistema para identificar o que está na disquete. Isso quase invariavelmente significa que a disquete não estava formatada, mas o erro pode ocorrer quando se tenta ler uma disquete corrompida ou que usa qualquer protecção especial.	CAT, COPY, ERASE, LOAD, MERGE, MOVE, SAVE
Missing extent As fichas são constituídas por blocos de 16 Kb, cada um dos quais constitui o que se chama um <i>extent</i> , termo que significa "extensão", "comprimento". Este erro pode ocorrer enquanto se lê uma ficha de uma disquete se esta for mudada depois de o sistema ter lido o seu nome no directório mas antes de ter começado a ler o seu conteúdo. É muito improvável que este erro ocorra.	Improvável, COPY, LOAD, MERGE
RIC No data O sector de identificação da disquete não pode ser encontrado. Este erro pode ocorrer ao tentar-se copiar uma disquete protegida.	CAT, COPY, ERASE, LOAD, MERGE, MOVE
No rename between drives Foi feita uma tentativa para usar o comando MOVE especificando nomes de fichas fonte e destino que estão em unidades diferentes.	MOVE... TO

RIC Seek fail

A unidade de disquete é incapaz de localizar a pista que foi indicada. Se este erro persistir, é possível que o computador tenha de ser reparado.

CAT, COPY,
ERASE, FORMAT, LOAD,
MERGE, MOVE, SAVE

Uncached

Erro interno de ocorrência muito improvável.

Improvável

RIC Unknown disk error

Erro não familiar. É muito improvável que tal apareça.

Improvável, CAT,
COPY, ERASE, FORMAT,
LOAD, MERGE, SAVE

RIC Unrecognized disk format

Ao tentar ler/escrever uma disquete, o + 3 DOS foi incapaz de reconhecer o seu formato — isto é: leu a especificação da disquete mas encontrou informações para ele sem significado. Este erro pode ocorrer quando se tenta obter acesso.

CAT, COPY,
ERASE, LOAD, MERGE,
MOVE, SAVE

RIC Unsuitable media

A disquete inserida tem um formato inadequado. Este erro pode ocorrer, por exemplo, quando se pretende escrever numa disquete de 80 pistas colocada na unidade de 40 pistas do + 3.

CAT, COPY,
ERASE, FORMAT, LOAD,
MERGE, MOVE, SAVE

Palavras Reservadas

As palavras seguintes são reservadas para o BASIC e portanto não podem ser usadas para outros fins, excepto quando entre aspas:

Funções

ABS, ACS, AND, ASN, ATN, ATTR, BIN, CHR\$, CODE, COS, EXP, FN, IN, INKEY\$, INT, LEN, LN, NOT, OR, PEEK, PI, POINT, RND, SCREEN\$, SGN, SIN, SQR, STR\$, TAN, USR, VAL, VAL\$, +, -, *, /, ↑, =, <, >, <=, >=, <>, =.

Instruções

BEEP, BORDER, BRIGHT, CAT!, CIRCLE, CLEAR, CLEAR __, CLOSE #, CLS, CONTINUE, COPY, DATA __, __, __, __, __, __, DEF FN, DIM, DIM __\$, DRAW, ERASE, ERASE!, FLASH, FOR __ TO __, FOR __ TO __ STEP __, FORMAT __; __, GOSUB, GOTO, IF __ THEN __, INK, INPUT, INVERSE __, LET __=__ , LIST, LIST __, LLIST, LLIST __, LOAD, LOAD!, LPRINT, MERGE __, MERGE! __, MOVE, NEW, NEXT, OPEN, OUT, OVER, PAPER, PAUSE, PLAY, PLOT, POKE, PRINT, RANDOMIZE; READ, REM, RESTORE, RESTORE __, RETURN, RUN, RUN __, SAVE __, SAVE! __, SPECTRUM, STOP, VERIFY.

Instruções exclusivas do Spectrum + 3

CAT, CAT __ EXP, COPY EXP, COPY EXP (INVERSE), COPY __ TO, COPY __ TO SCREEN\$, COPY __ TO LPRINT, COPY __ TO SPECTRUM FORMAT, FORMAT LINE, FORMAT LPRINT.

Para a utilização destas palavras reservadas vejam-se as páginas:

48K BASIC 00 a 00
128K BASIC 00 a 00
+ 3 BASIC 00 a 00

Como Usar a Calculadora

Os Spectrum + 2 e + 3 podem ser usados como máquinas de calcular. Para isso recorre-se ao menu inicial e escolhe-se a opção 'Calculator'. (Ver pág. 11).

A calculadora pode ser usada logo que se liga o Spectrum, mas se se estiver a trabalhar num programa em 128K BASIC ou + 3 BASIC, pode-se passar à calculadora escolhendo a opção 'Exit' do menu de edição, que conduz ao menu inicial e dá assim acesso a 'Calculator'. Note-se que o programa em BASIC que estiver em curso quando se decide usar a calculadora não é perdido: quando se sai de "calculator" continua a partir do passo em que se encontrava.

Quando se entra na opção 'Calculator' surge na parte inferior do ecrã uma barra com essa designação. Introduza, por exemplo:

$$6 + 4$$

Logo que se prima [ENTER] surge a resposta **10**. Note-se que não é necessário premir a tecla [=] como nas calculadoras vulgares.

O cursor fica colocado à direita da resposta, o que significa que se trata de um *total corrente*. Assim, para continuar o cálculo não é necessário repetir o resultado. Se, por exemplo, se quiser dividir este por 5, bastará inserir

$$/5$$

e ao premir-se [ENTER] a resposta será 2. Do mesmo modo, se a seguir se fizer:

$$* \text{PI}$$

O resultado será **6.2831853**. Note-se que não se usou o valor de PI, mas sim a palavra reservada para a função. O mesmo pode ser feito com as outras funções matemáticas. Por exemplo:

$$* \text{ATN } 60$$

dará o resultado **9.7648943**.

É possível "editar" o conteúdo do ecrã. Mova o cursor (premiendo a tecla com a seta apontada para a esquerda) até ao princípio da linha e depois escreva INT:

A linha passará a ser:

$$\text{INT } 9.7648943$$

Assim que seja premido [ENTER] tem-se a resposta 9. Isso demonstra que o computador não tem de efectuar um cálculo para apresentar o valor de uma expressão. Para outro exemplo, prima [ENTER] e depois escreva;

$$1\text{E}6$$

No ecrã surgirá o valor dessa expressão. Note que quando se escreveu '1E6' e se premiu [ENTER], indicou-se ao computador que se ia iniciar um novo cálculo.

Uma característica extremamente útil da calculadora dos Spectrum + 2 e + 3 é a de atribuir valores a variáveis e depois usá-los em cálculos subsequentes. Tal como no BASIC isso é conseguido através da instrução LET. Faça:

$$\text{LET } x = 10$$

Depois prima [ENTER] *duas vezes* para que o computador aceite a atribuição do valor à variável. Para verificar o uso da variável x faça:

$$x + 90$$

e depois:

$$+ x * x$$

Se se estiver a usar a calculadora enquanto está a ser processado um programa de BASIC, deve-se ter o cuidado de não escolher para ela variáveis que possam entrar em conflito com as usadas pelo programa.

As palavras reservadas do BASIC não podem ser usadas como nomes de variáveis.

Premindo a tecla [EDIT], surge no ecrã um menu com duas opções:

Calculator
Exit

Seleccionando 'Calculator' volta-se ao modo de calculadora. Seleccionando 'Exit' volta-se ao menu de abertura. Se então se quiser voltar ao 128 BASIC (no Spectrum + 2) ou ao + 3 BASIC (no Spectrum + 3), selecciona-se essa opção.

Ligação a Periféricos

SPECTRUM + 2

Joystick(s)

Recomenda-se o uso de "joysticks" Sinclair SJS1. Os de outros tipos (como os Atari) não funcionam se inseridos directamente nas fichas porque estas estão ligadas de um modo diferente. Para os usar é necessário dispôr de um adaptador. De um modo geral, os "joysticks" com interface, usados nos anteriores modelos do Spectrum, trabalham no + 2 se a interface for usada.

No Spectrum + 2 há duas fichas para "joysticks". De uma maneira geral, os jogos usam a indicada por JOYSTICK ONE.

Se um programa apresentar várias opções de tipos de "joysticks", deve-se usar a "Interface Two" (ou "Sinclair") pois o circuito dos "joysticks" no Spectrum + 2 foi concebido para funcionar exactamente como o da "Interface Two".

Os "joysticks" podem ser ligados (ou desligados) com o computador ligado. São os únicos periféricos com os quais isso pode ser feito sem perigo. (Note-se que o mesmo acontece se o "joystick" for de outro tipo que não o Sinclair e estiver ligado a uma interface, mas esta só deverá ser montada ou retirada com o computador desligado).

PINO	FUNÇÃO
1	não usado
2	Terra
3	não usado
4	fogo
5	acima
6	direita
7	esquerda
8	terra
9	abaixo

Conectores dos
JOYSTICK 1 e 2

Monitor

O + 2 pode trabalhar com um monitor monocromático ou de cor (RGB) em vez de um televisor ou conjuntamente com este. O cabo depende do tipo do monitor e por isso não é fornecido com o + 2. Consulte o fornecedor do monitor.

Conector de RGB.



PINO	SINAL	NÍVEL
1	PAL composto	1.2 V-pk/75 ohms
2	0 volts	
3	bright	TTL
4	sync composto	TTL
5	sync vertical	TTL
6	verde	TTL
7	vermelho	TTL
8	azul	TTL

Note que nem todos os monitores aceitam o sinal de BRIGHT. Os que não o aceitam, só mostram 8 cores, em vez de 16.

Se o monitor tiver entrada de som, esta deve ser ligada à saída na traseira do + 2. Se o monitor não dispuser de som, há que usar um amplificador externo. Veja a secção seguinte.

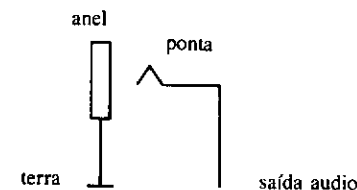
Amplificador

Normalmente, o + 2 reproduz o som através do televisor ao qual estiver ligado. Se for usado um monitor sem circuito de som ou se se desejar gravar ou amplificar o som, deve-se usar a saída **SOUND** na traseira do computador. É uma ficha fêmea do tipo "jack" com uma saída de 200 mV pk-pk a uma impedância de cerca de 5 ohms. Quando se usar um amplificador deve-se recordar que o som dos sinais de "save" e "load" são também enviados para a saída, pelo que convém baixar o volume de som do amplificador durante essas operações.

Note-se também que o nível do som produzido por BEEP é igual aos dos três canais em PLAY, actuando em conjunto. Consequentemente, na prática BEEP é um som muito mais forte que PLAY, do que podem resultar problemas quanto à regulação do volume de som.

Não há inconveniente em ligar ou desligar um amplificador, um gravador, etc., à ficha SOUND enquanto o computador estiver a funcionar.

Conector SOUND




Impressora (e outros dispositivos com ligação série)

Para a ligação a impressoras, em geral, veja o capítulo IMPRESSORAS. O + 2 pode ser ligado directamente à maior parte das impressoras série que seguem a norma RS232. Os utilizadores não experimentados não devem fazer qualquer espécie de experiência com ligações de interfaces. Para o cabo de ligação consulte o distribuidor Sinclair e siga à letra as indicações do fabricante da impressora.

A impressora deve ser ligada ao conector RS232/MIDI na traseira do + 2.


Para ligar qualquer outro dispositivo ao Spectrum + 3 através do conector RS232 consulte o distribuidor. As ligações são as seguintes:

PINO	FUNÇÃO	Conector RS232
1	GND	
2	TXD	
3	RXD	
4	DTR	
5	CTS	
6	+ 12V	

MIDI

Muito embora a saída do MIDI (*Musical Instrument Digital Interface*) seja a RS232, necessita de um cabo diferente, o qual deve ser ligado ao conector MIDI IN no instrumento a comandar (sintetizador, máquina de percussão, etc.). A MIDI do + 2 não recebe dados — só actua como *fonte*. Não é necessário configurar a MIDI antes de a usar: basta ligá-la através de PLAY com o parâmetro Y.

O uso da interface MIDI não perturba a regulação da "baud rate" da RS232.

PINO	FUNÇÃO	Conector MIDI
1	RETURN	
2	não usado	
3	não usado	
4	não usado	
5	DATA OUT	
6	não usado	

Keypad

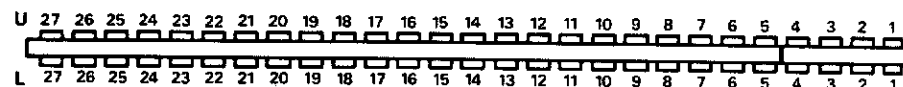
O conector **Keypad** destinava-se a um teclado numérico suplementar que esteve disponível na versão espanhola do Spectrum 128K mas não foi produzida para o mercado britânico nem para o + 2.

Interface One e "microdrives"

O Spectrum + 2 é compatível com a "Interface One" e as 'microdrives'. A ligação faz-se como para os anteriores modelos do Spectrum, através do barramento do conector *EXPANSION I/O* na traseira do computador.

Outros dispositivos de expansão

O Spectrum + 2 pode ser ligado a muitos periféricos através do conector de expansão *EXPANSION I/O* na traseira do computador, mas muito embora este seja muito semelhante ao dos modelos anteriores, não é possível garantir que um dispositivo que funciona correctamente num Spectrum 48K funcione também no + 2. Consequentemente, aconselha-se que não se adquiram periféricos sem verificar se eles funcionam correctamente com o Spectrum + 2.



ATENÇÃO — É MUITO PERIGOSO COLOCAR QUALQUER DISPOSITIVO NO CONECTOR I/O OU RETIRÁ-LO ESTANDO O COMPUTADOR LIGADO. É PROVÁVEL QUE O COMPUTADOR E (OU) O DISPOSITIVO SE AVARIEM

Conector EXPANSION I/O

PINO	FILA SUPER. (U)	FILA INFER. (L)
1	A15	A14
2	A13	A12
3	D7	-5V
4	não usado	-9V
5	DO	0V
6	D1	0V
7	D2	CK
8	D6	AO
9	D5	A1
10	D3	A2
11	D4	A3
12	<u>INT</u>	<u>IORQGE</u>
13	<u>NMI</u>	OV
14	<u>HALT</u>	não usado
15	<u>MREQ</u>	não usado
16	<u>IORQ</u>	não usado
17	<u>RD</u>	não usado
18	<u>WR</u>	<u>BUSRQ</u>
19	-5V	<u>RESET</u>
20	<u>WAIT</u>	A7
21	+12V	A6
22	-12V	A5
23	<u>MI</u>	A4
24	<u>RFSH</u>	<u>ROMCS</u>
25	A8	<u>BUSACK</u>
26	A10	A9
27	not used	A11

SPECTRUM + 3

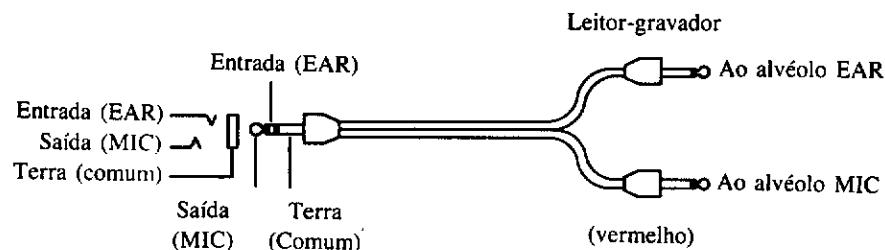
O que se disse quanto aos periféricos para o Spectrum + 2 aplica-se ao Spectrum + 3 excepto quanto aos adiante indicados.

Leitor-gravador de cassetes

Os programas podem ser carregados de cassetes ou gravados nelas. Os comandos que instruem o computador para trabalhar com cassetes ou disquete foram descritos anteriormente neste manual.

Para ligar o leitor-gravador de cassetes no Spectrum + 3 é necessário um cabo especial, diferente dos usados nos modelos anteriores. Esse cabo não é fornecido com o computador, mas pode ser obtido através do distribuidor ou fabricado como se indica a seguir:

Conector TAPE/SOUND no + 3



Uma das fichas-macho deve ser do tipo "jack", com o espigão em três secções, como as usadas nos sistemas estereofônicos, e destina-se a ser inserida no conector marcado **TAPE/SOUND** na traseira do computador.

Os espigões das outras duas fichas devem ser divididos somente em duas partes, para serem inseridas nos alvéolos do leitor-gravador de cassetes marcados MIC e EAR. (Em muitos leitores-gravadores, o alvéolo de MIC tem a cor vermelha. Noutros, o alvéolo de MIC pode ser designado por COMPUTER IN, ou INPUT, e o de EAR pode sê-lo por COMPUTER OUT ou OUTPUT).

Importa recordar que o sucesso na transferência de programas para as cassetes depende muito da regulação dos níveis de som e tonalidade do leitor-gravador. Se verificar que não consegue gravar ou carregar quaisquer programas, tente trocar as fichas de EAR e MIC.

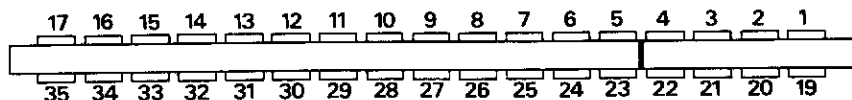
O modo de operar com cassetes foi descrito em pormenor anteriormente neste manual.

Impressora

No capítulo IMPRESSORAS descreveu-se em pormenor como ligar as impressoras ao Spectrum + 3. Recorde-se que neste computador não é possível usá-las no modo 48K, mas na maior parte dos casos é possível transferir os programas em 48K BASIC ou as fichas para o modo + 3 e imprimi-las então.

As impressoras série necessitam de um cabo idêntico ao usado no Sinclair QL. As impressoras paralelo necessitam de um cabo especial que pode ser obtido através do distribuidor. A impressora Amstrad DMP3160 é fornecida com o cabo adequado.

As impressoras série são ligadas ao conector RS232/MIDI. As impressoras paralelo são ligadas ao conector PRINTER.



(vista da rectaguarda)

Conector PRINTER

PINO	FUNÇÃO	PINO	FUNÇÃO
1	STROBE	19	GND
2	D0	20	GND
3	D1	21	GND
4	D2	22	GND
5	D3	23	GND
6	D4	24	GND
7	D5	25	GND
8	D6	26	GND
9	D7	27	não usado
10	não usado	28	GND
11	BUSY	29	não usado
12	não usado	30	não usado
13	não usado	31	não usado
14	GND	32	não usado
15	não usado	33	GND
16	GND	34	não usado
17	não usado	35	não usado
18	não existe	36	não existe

Embora existam apenas 34 contactos no conector PRINTER, a sua numeração vai de 1 a 17 e de 19 a 35 (o 18 e o 36 não existem) para equivalência com o conector Centronics na impressora.

Unidade adicional de disquetes

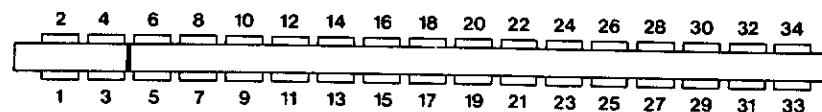
A unidade suplementar de disquetes AMSTRAD FD-1 pode ser adicionada ao Spectrum + 3.

A versatilidade do + 3 BASIC permite fazer facilmente todas as operações com fichas numa só unidade de disquetes. No entanto uma segunda unidade tem a vantagem de tornar mais rápidas e mais seguras essas operações.

Para ligar a unidade FD-1 ao Spectrum + 3 é necessário o cabo AMSOFT DL+. Ligue a ponta do cabo com o conector maior ao conector marcado com DISK na traseira do computador. A outra ponta deve ser ligada ao conector na traseira da unidade FD-1.

ATENÇÃO! — Antes de ligar ou desligar a unidade adicional, certifique-se de que nenhuma disquete está inserida em qualquer das unidades e de que o sistema está desligado. Se as ligações forem alteradas enquanto o sistema está a funcionar, é provável que o sistema entre em "crash", levando à perda de programas e dados. Grave sempre os programas antes de tocar nas ligações!

Quando a FD-1 estiver ligada ao + 3, ligue primeiro a FD-1 (usando o interruptor de cursor, na traseira da unidade) e depois ligue o + 3 (inserindo o cabo da fonte de alimentação. Os indicadores vermelho e verde na frente da unidade devem acender-se. Então o sistema de duas unidades estará pronto a funcionar.



(vista da traseira)

Conector de DISK B:

PINO	FUNÇÃO	PINO	FUNÇÃO
1	READY	20	GND
2	GND	21	não usado
3	SIDE SELECT	22	GND
4	GND	23	DRIVE SELECT
5	READ DATA	24	GND
6	GND		

7	WRITE PROTECT	25	não usado
8	GND	26	GND
9	TRACK 0	27	INDEX
10	GND	28	GND
11	WRITE GATE	29	não usado
12	GND		
13	WRITE DATA	30	GND
14	GND	31	não usado
15	STEP	32	GND
16	GND		
17	DIRECTION SELECT	33	não usado
18	GND	34	GND
19	MOTOR ON		

Monitor

O Spectrum + 3 pode ser ligado a um monitor monocromático ou de cor, ou aos televisores construídos segundo a norma francesa PERITEL TV, em alternativa (ou juntamente) com um televisor vulgar. Se o cabo não for fornecido com o monitor, consulte o distribuidor do + 3.

O monitor (ou o televisor PERITEL) deve ser ligado ao conector RGB/PERITEL na traseira do + 3.

PINO	SINAL
1	+ 12 V
2	GND
3	audio out
4	sync composto
5	+ 12V
6	verde
7	vermelho
8	azul



conector RGB/PERITEL

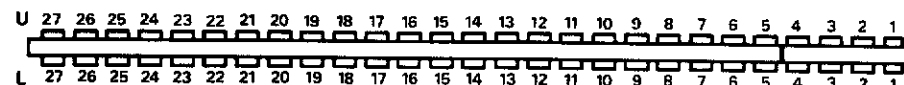
Se o monitor tiver capacidade de som, a respectiva entrada de audio deve ser ligada ao pino 3 do conector **RGB/PERITEL** ou à saída de som **SOUND** na traseira do computador. Se o monitor não possuir circuito de som há que usar um amplificador. Ver atrás as instruções para o Spectrum + 2.

Interface auxiliar

A interface auxiliar **AUX** tem duas linhas de entrada (*input*) e duas linhas de saída (*output*), comandadas pelos "chips" 1488 e 1489, por sua vez ligados às linhas de entrada e saída (*I/O*) do processador AY-3-8912. Para o seu comando é necessário dispôr da folha de dados do fabricante e de um conhecimento bastante do código de máquina. A interface pode também ser usada como uma segunda porta **RS232/MIDI**, ou para comandar robôs ou outros dispositivos externos. **UM MANUAL COM OS DADOS NECESSÁRIOS PARA A PROGRAMAÇÃO AVANÇADA EM CÓDIGO DE MÁQUINA E INCLUINDO OS DADOS EXISTENTES SOBRE A INTERFACE AUXILIAR PODERÁ SER OBTIDO A UM PREÇO NOMINAL ATRAVÉS DO DISTRIBUIDOR.**

Outros dispositivos de expansão

O Spectrum + 3 pode ser ligado a muitos periféricos através do conector de expansão **EXPANSION I/O** na traseira do computador, mas não é possível garantir que um dispositivo que funciona correctamente num Spectrum 48K ou mesmo num + 2 funcione também no + 3. Consequentemente, aconselha-se que não se adquiram periféricos sem verificar se eles funcionam correctamente com o Spectrum + 3.



ATENÇÃO! — É MUITO PERIGOSO COLOCAR QUALQUER DISPOSITIVO NO CONECTOR I/O OU RETIRÁ-LO ESTANDO O COMPUTADOR LIGADO. É PROVÁVEL QUE O COMPUTADOR E (OU) O DISPOSITIVO SE AVARIEM.

Conector EXPANSION I/O

PINO	FILA SUPER (U)	FILA INFER (L)
1	A15	A14
2	A15	A12
3	D7	+5V
4	ROMIOE	não usado
5	D0	GND
6	D1	GND
7	D2	CK
8	D6	A0
9	D5	A1
10	D3	A2
11	D4	A3
12	INT	não usado
13	NMI	GND
14	HALT	ROM2OE
15	MREQ	DISKRD
16	IORQ	DISKWR
17	RD	MOTORON
18	WR	BUSRQ
19	não usado	RESET
20	WAIT	AT
21	+12V	A6
22	-12V	A6
23	MI	A4
24	RFSH	não usado
25	A8	BUSACK
26	A10	A9
27	RESET	A11

