PACOTE GRÁFICO:

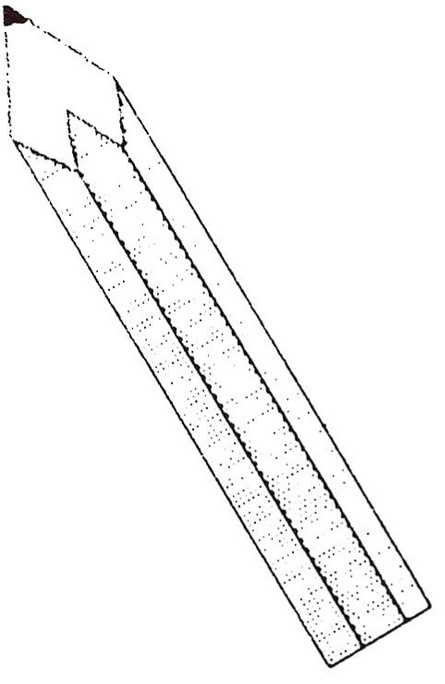
Basic 64

TC 2068

PACOTE GRAFICO

BASIC 64

EXTENSÃO AO BASIC SINCLAIR



VERSÃO PARA O TC 2068

NOTA: As rotinas BASIC-64 e o conteúdo deste manual podem ser utilizados em programas de utilizador.

Programas comerciais que as utilizem deverão mencionar apenas, que são usadas as rotinas BASIC-64.

PACOTE GRAFICO : BASIC_64

CAP 0 - Introdução

O programa BASIC_64, versão 1.1, possibilita ao utilizador o uso das capacidades gráficas e de impressão de texto especiais, suportadas pela arquitectura do microcomputador TC2068 (em modo 2068).

Foram destacados canais especiais, para impressão em 64 colunas (canal #4) ou impressão com número definivel de colunas (canal #5). Está implementado um novo conjunto de comandos BASIC, em adição ao já existente, permitindo a utilização destes modos gráficos em programas escritos em BASIC. Por outro lado, está igualmente previsto o uso de rotinas em linguagem máquina, que desemvolvem estas capacidades.

Refira-se que esta implementação utiliza o mesmo tipo de procedimentos, por parte do utilizador, em relação ao BASIC standard.

Cap 1 - Inicialização do Sistema

Introduza a cassete que contém o programa BASIC_64, no seu gravador e execute o seguinte comando :

LOAD ""

Após ter sido carregado um 'loader' surge a seguinte mensagem:

TEM O SISTEMA FDD LIGADO AO COMPUTADOR ? (S/N)

Se a resposta è não, então è carregado o código **BASEST1A** , em caso contrário (sim), o código a carregar è o **BASEST1** , o que se justifica, pois com o sistema FDD ligado a rotina máquina è ligeiramente diferente, do que no caso com o FDD desligado (interfaçe TIMEX desligado). Seguidamente è carregado o segundo bloco de código após o que surge a mensagem :

0 OK, 0:2

Pode agora verificar que existe uma linha 0 que inclui uma chamada directa à rotina e uma mensagem de COPYRIGHT. A chamada à rotina vai promover a recolocação do BASIC, inicializar as novas variáveis introduzidas pelo BASIC_64, abrir os novos canais de impressão (#4 e #5) e preparar o interpretador BASIC, de modo a que este aceite a sintaxe dos novos comandos criados.

Refira-se que o utilizador deve executar o comando **RANDOMIZE USER 32280** , após a execução dum **CLEAR n** , dum **RUN n** ou de um **GOSUB n** para ter acesso aos comandos estendidos.
Por outro lado aconselha-se que a edição de programas em BASIC, seja feita em 32 colunas, para o que è necessário executar o seguinte comando :

OUT 255,0

Após a edição, pode retornar aos modos estendidos com os comandos **INK* n** ou **PAPER* n**

1.1 - Salvar e chamar programas em BASIC_64

Existem duas formas de salvar programas em BASIC_64, envolvendo uma delas a alteração do programa em BASIC, fornecido na cassete.

Num primeiro passo comum aos dois processos, o programa è gravado na cassete pela forma usual. Para o chamar novamente pode-se primeiro chamar a rotina BASIC_64, e depois chama-se o programa em causa com um simples **LOAD "nome"** . Se se pretende chamar num só passo (i.e. fazendo apenas um **LOAD ""**), há que proceder da seguinte forma: primeiro faz-se **NEW** ,depois a partir da cassete BASIC_64 executa-se **MERGE ""** após o que se fizer **LIST** verificará a existência da linha 15 que possiu um comentário bastante elucidativo. Coloque então nessa linha a instrução conveniente e grave o programa com **SAVE "BASIC_64"LINE 10** .

Cap 2 - BASIC Estendido

2.1 Modos de Resolução

No que respeita às capacidades gráficas, o utilizador possui dois modos de resolução permitidos pelo sistema :

0 - 192*256 pixels, (176*256 pixels na zona do écran dedicada à impressão).
1 - 192*512 pixels, (176*512, idem).

Estes modos, ambos utilizando dois DISPLAY FILES, são relativos aos novos comandos gráficos, que passamos a introduzir e que serão explanados na próxima secção :

- **PLOT* x,y**
- **LINE x,y**
- **LINE x,y,z**
- **CIRCLE* x,y,r**
- **SCREEN$ n**

Convèm referir que os modos 0 e 1 podem ser utilizados simultaneamente no mesmo programa e são independentes dos comandos de impressão de texto e de expressões. Assim, em qualquer deles è possivel a utilização dos canais #4 e/ou #5 da maneira que se descreve .

#4 : Este canal è especialmente dedicado à impressão em 64 colunas. Utiliza o set de caracteres da ROM do microcomputador, apontado pela variável CHARS. Cada caracter está definido numa matriz de pontos de 8X8. A impressão è invocada utilizando os comandos, a introduzir, PRINT #4 ... e LIST #4 ... .

#5 : Este canal permite uma impressão relativamente flexível, já que o modo de impressão è programável por parte do utilizador. Este possui dois graus de liberdade :

- Número de colunas no écran (variável do sistema MAXCL).
- Número de pixels, na horizontal, da matriz de pontos de definição dos caracteres (variável do sistema LARG).

Existe um novo set de caracteres, definidos numa matriz de 8X6 pontos, apontados pela variável do sistema CHRST. Trata-se dum set especialmente vocacionado para a impressão em 85/80 colunas, embora possa igualmente ser utilizado em 64 colunas. Assim, o canal #5 está definido, por defeito, da seguinte maneira :

- Impressão do novo set de carcteres em 85 colunas e com uma matriz de 8X6, utilizando os comandos PRINT #5 ... e LIST #5 ... .

2.2 Novos Comandos BASIC

2.2.1 Comandos de Impressão de Texto

- **PRINT #n ...** O parâmetro n especifica o canal utilizado (#4 ou #5). O campo ... indica uma seqência de parâmetros do comando PRINT, intercalados por separadores (, ; e "). Os parâmetros podem ser : expressão numèrica, expressão literal e ainda caracteres de controle (AT m,n , TAB n).

- **LIST #n;m** O parâmetro n especifica o canal em uso (#4 ou #5). Este comando provoca uma listagem do programa a partir da primeira linha cujo número é pelo menos m. Se m não è especificado então a listagem e feita a partir da primeira linha do programa.

2.2.2 Comandos Grâficos

- **CLS*** Apaga os dois DISPLAY FILES.

- **INK* n** O utilizador apenas pode definir, para o conjunto dos dois DISPLAY FILES, um INK e um PAPER com cores complementares. Este comamdo coloca INK=n e PAPER=7-n.

- **Paper* n** Comando semelhante ao anterior. Coloca PAPER=n, INK=7-n.

- **SCREEN$ n** Estabelece o modo de resolução grâfica. Se n=1, temos o modo de 192X512. Se n=0, o modo entao è de 192X256.

- **PLOT* x,y** Imprime Um ponto de INK, sendo em conta OVER e INVERSE, no pixel (x,y) e actualiza a posiçào PLOT.

- **LINE x,y,z** Desenha uma linha, desde a posição actual de PLOT, mudando x horizontalmente e y verticalmente e percorrendo um arco de circunferencia de z rad.

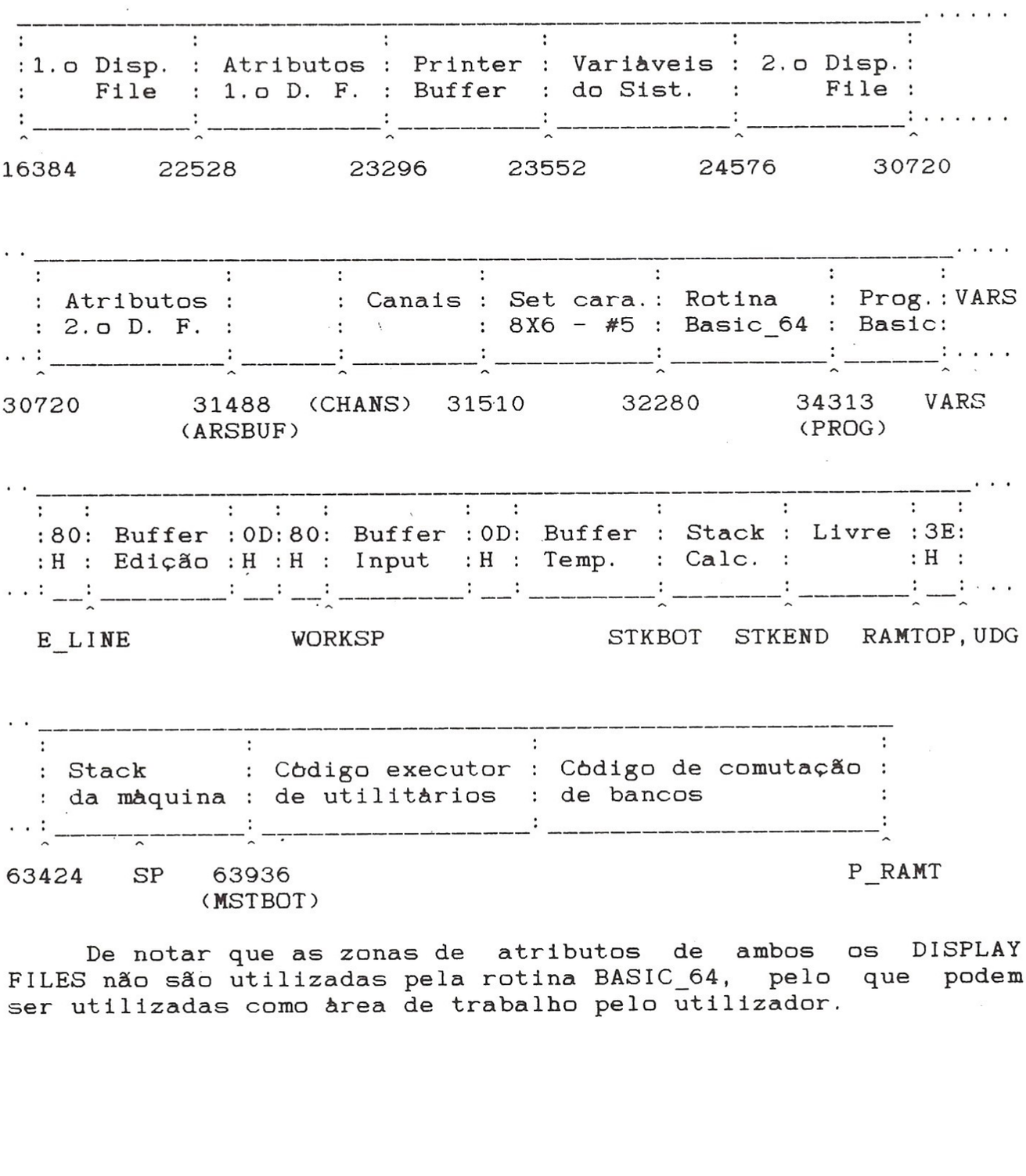
- **LINE x,y** **LINE x,y,0.**

- **CIRCLE* x,y,r** Desenha uma circunferência de centro (x,y) e raio r, se o modo grâfico è 0 (SCREEN$ 0). Se o modo è 1 (SCREEN$ 1), então desenha uma oval.

2.3 Organização de Memòria

O novo mapa de memòria è caracterizado pela existência de dois DISPLAY FILES e pela recolocação da àrea reservada ao BASIC bem como das àreas de 'STACK e CODIGO RESIDENTE EM RAM. A àrea BASIC passa a situar-se a partir do endereço 34313 apontado pela variàvel do sistema PROG). Logo a seguir à area reservada aos CANAIS (CHANS), situa-se o novo SET de caracteres (comprimento = 768 bytes), seguindo-se a rotina BASIC_64 (comprimento = 2030 bytes), que anteçede a àrea do BASIC. As zonas de STACK e CODIGO RESIDENTE EM RAM ficam colocados no segmento de 8K bytes superior, do mapa de memòria.

Esquemàticamente teremos :



De notar que as zonas de atributos de ambos os DISPLAY FILES não são utilizadas pela rotina BASIC_64, pelo que podem ser utilizadas como àrea de trabalho pelo utilizador.

2.4 Novas Variàveis do Sistema

Em adição às variàveis do sistema do BASIC do microcomputador, a rotina utiliza as seguintes variàveis :

N.o BYTES	Endereço	Nome	Conteúdo
2	31491	XCOOR	Coordenada x do ultimo ponto colocado
1	31493	YCOOR	Coordenada y do ultimo ponto colocado
1	31494	FLAG	Flag do sistema
2	31495	COUNT	Contador
2	31497	WORK	Buffer
1	31494	VDMOD	Modo de resolução: 0 se res.=176X256, 1 se res.=176X512
1	31500	COL	Coluna do ùltimo caracter imprimido
1	31501	LIN	21 menos a linha do ùltimo caracter imprimido
2	31502	XCR	Utilizado pelo canal #5
1	31504	YCR	Idem
1	31505	CURST	Canal què està a ser utilizado
1	31506	NCOL	No. màximo de colunas em utilização
1	31507	LARG	Largura, em pixels, dos caracteres impressos usando o canal #5
1	31508	MAXCL	No. màximo de colunas da impressão usando o canal #5
1	31509	CHRST	256 menos do que o endereço de set. de caracteres utilizado pelo canal #5. Por defeito è igual a 23755-256=23499.

Cap 3 - Utilização da linguagem máquina

3.1 Impressão de um caracter em 64, 85, ou 128 colunas

A impressão de um caracter em 64 ou 85 colunas é perfeitamente directa, envolvendo apenas previamente a abertura do canal adequado seguida de um RST 10h, tendo o acumulador, como é normal, o código do caracter a ser impresso. Este caracter pode ser, como será visto adiante, um caracter de controlo.

3.1.1 Impressão em 64 colunas

O canal #4 tem de ser aberto. Esta operação é feita através da rotina do ROM situada apartir do endereço 1230h. Esta rotina tem como parâmetro o número do canal a abrir, que é colocado no acumulador. Assim para imprimir a caracter "!", código ASCII = 21h, teríamos:

```
LD A, 4         ;é o canal #4 que vái ser aberto
CALL 1230h      ;abertura do canal
LD A, 21h
RST 10h         ;impressão do caracter "!",
RET             ;na linha e coluna correntes.
```

3.1.2 Impressão em 85 colunas

E perfeitamente igual à anterior, simplesmente agora utiliza-se a canal #5. Por exemplo, para a impressão do caracter "!" teríamos:

```
LD A,5          ;abertura
CALL 1230h      ;do canal #5
LD A,21h
RST 10h         ;impressão do caracter.
RET
```

3.1.3 Impressão em 128 ou outro número de colunas

Para realizar esta impressão é necessário primeiro definir um novo set de caracteres, que no caso das 128 colunas terá de ser definido numa matriz de 8x4 pixels. Em qualquer caso cada caracter fica sempre definido pelos normais 8 bytes.

A definição de cada caracter faz-se de forma inteiramente análoga à definição dos UDGs, só que no caso presente apenas as 4 primeiras colunas, a contar da esquerda, podem ser usadas para gerar o set de caracteres (128 colunas). Este número 4 foi encontrado tendo em atenção que a resolução horizontal é agora de 512 pixels, e assim numa matriz de 128 colunas, a largura de cada coluna é de 512 / 128 = 4 pixels.

Após colocado o set de caracteres em memória, por exemplo apartir do endereço 50000, as variáveis do sistema LARG e MAXCl (note-se que esta impressão refere-se apenas ao canal #5) teriam de ser alteradas em conformidade; LARG tomaria o valor 4 e MAXCL o valor 128.

3.2 Utilização dos caracteres de controlo

A utilização dos caracteres de controlo é directa: abertura do canal adequado seguido de um RST 10h. Depois da "impressão" do caracter de controlo são enviados os parametros também através do RST 10h. Por exemplo, para fazer o equivalente a 'PRINT #5; AT 10,50; "!";', teríamos:

```
LD A,5          ;esta abertura do canal e só
CALL 1230h      ;necessária fazer uma vez
LD A,16h
RST 10h
LD A,10         ;linha
RST 10h
LD A,50         ;coluna
RST 10
LD A,21h
RST 10
RET
```

A utilização do TAB control seria feita de forma inteiramente análoga. Note-se que os caracteres de controlo referentes aos atributos (INK, PAPER, BRIGHT) não têm efeito.

3.3 Impressão de um 'string' de caracteres

A impressão de um string de caracteres não e mais do que uma sequência de RST 10h, em que vão sendo enviados sucessivamente os códigos dos caracteres que compoêm o 'string'. Existe uma rotina na ROM que realiza este trabalho, situa-se a partir do endereço 21DBh. Esta rotina tem como parametros os pares de registos DE e BC que têm respectivamente, o primeiro endereço a partir do qual está armazenado o 'string' e o seu comprimento. Por exemplo, a impressão de um 'string' armazenado apartir do endereço 50000 e de comprimento 100 bytes, seria feita da seguinte forma:

```
LD DE,50000
LD BC,100
CALL 21DBh
RET
```

NOta: o canal adequado teria de ser aberto.

3.4 Impressão de uma constante numérica

Utiliza-se para este efeito a rotina PRINT_FP situada a partir do endereço 31A1h. Esta rotina imprime o número que se encontra no topo do stack do calculador. Assim se se desejar imprimir um numero de 2 Bytes em 64 colunas ,por exemplo 28054 seria:

```
LD A,4
CALL 1230h
LD BC, 28054
CALL 30E9h        ;por BC no stack do calculador
CALL 31A1h        ;PRINT_FP.
RET
```

Adenda - Utilização do sistema FDD

Se o seu sistema informàtico inclui um **FlOPPY DISK DRIVE SYSTEM (FDD)** da **TIMEX** ,então torna-se desejàvel a alteração do programa 'LOADER' (BASIC_64) fornecido na cassete, com vista a uma eficiente inicialização do sistema. Assim apòs ter percorrido os procedimentos descritos no **CAPITULO 1** , com o sistema **FDD** ligado, deverà executar os seguintes comandos :

SAVE * "BASEST1"CODE 32280,2030 , seguido de :

SAVE * "SETMM'CODE 31510,768

Seguidamente, deverà executar (a partir da cassete) **MERGE "BASIC_64"** . Depois de fazer **LIST** , deverà apagar as linhas 2 a 12, inclusivè. Serà então introduzida a seguinte linha de programa :

12 LOAD * "BASEST1"CODE

... sendo a linha 14 alterada para :

14 RANDOMIZE USER 31488: OUT 255,0: LOAD * "SETMM"CODE 31510

Nesta altura o programa de 'LOADER' jà està convenientemente escrito e restarà apenas guardà-lo em disco. Deverà executar :

SAVE * "START"LINE 12

... e o programa BASIC_64 inicializa-se por pressão no botão de reset do interface **TIMEX** .