SHARP

All and more about Sharp PC-1500 at http://www.PC-1500.info

MANUAL DE INSTRUÇÕES

PC-1500 RP

WWW.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

WWW.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

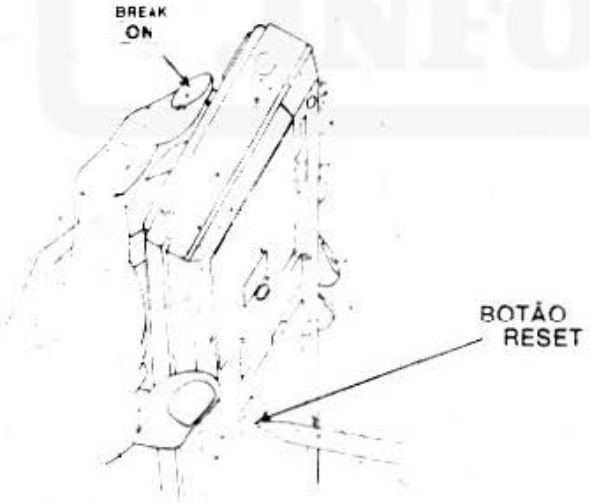
NOTAS OPERACIONAIS

Como o visor de cristal líquido é feito de vidro, ele deve ser tratado com alguns cuidados:

Para assegurar o bom funcionamento do seu computador de bolso Sharp, recomendamos:

1. Conservá-lo em ambientes livres de extremas flutuações de temperatura, umidade e poeira.
   Durante o calor, veículos deixados à luz do sol direta estão sujeitos à elevação de temperatura em seu interior. Exposição prolongada à alta temperatura pode causar danos no seu computador de bolso.
2. Para a limpeza, use somente um pano macio e seco. Evite o uso de solventes ou pano úmido.
3. Para evitar descarga de pilhas, remova as pilhas quando o computador de bolso não for usado, por um longo período de tempo.
4. Caso o computador de bolso necessitar de serviço técnico, enviá-lo somente ao serviço autorizado Sharp.
5. Guarde este manual para futuras consultas.

Esta unidade, caso submetida a um ruído externo forte ou impacto durante a operação, pode ficar com todas as suas teclas, inclusive a tecla [BREAK ON], inoperantes.
Caso isto ocorra, pressione o botão ALL RESET localizado no painel traseiro da unidade por aproximadamente 8 segundos, com a tecla [BREAK ON] pressionada.



Pressione o botão ALL RESET com qualquer objeto pontiagudo, tal como uma caneta esferográfica. Não use jamais, objetos que possam quebrar facilmente como lapiseira ou ponta de agulha.

Então, verifique se [NEW0?: CHECK] é indicado no visor, e pressione as teclas [CL] [N] [E] [W] [0] [ENTER]. (0 : zero, tecla numérica).
Caso o visor não mostre NEW0?: CHECK, efetue a operação acima novamente. Com esta operação, o programa, dados e todos os conteúdos reservados são zerados, logo não pressione o botão ALL RESET exceto quando a falha acima ocorrer.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

ESPECIFICAÇÕES DO PC-1500

Modelo:	Computador de bolso PC-1500
Número de dígitos para cálculo:	10 dígitos (mantissa) 2 dígitos (expoente).
Sistema de cálculo:	De acordo com a fórmula matemática (com função de avaliação de prioridade).
Linguagem de programação:	Basic
CAPACIDADE:	CPU: CMOS 8 bits Sistema ROM: 16K bytes Capacidade de memória da RAM: 3.5K bytes. - Área de sistema: 0.9K bytes Buffer de entrada: 80 bytes Stack: 196 bytes Outros: 600 bytes - Área do usuário: 2.6K bytes Área de memória fixa: 624 bytes (A ~ Z, A$ ~ Z$) Área de dados de programa básico: 1850 bytes Área de reserva: 188 bytes
Cálculos:	Quatro cálculos aritméticos, potenciação, funções trigonométricas e trigonométricas inversas, funções logarítmicas e exponenciais, conversão angular, extração de raiz quadrada, função de sinal, absolutos, inteiros e cálculos lógicos.
Função de revisão:	Mudança de cursor (▶◀) Inserção (INS) Cancelamento (DEL) Linha superior e inferior (↓ , ↑)
Proteção de memória:	Bateria de reserva (programa, dados e memórias de reserva são protegidos).
Visor:	Cristal líquido 26 caracteres Gráficos de 7 x 156 pontos
Teclas:	65 teclas incluindo Alfabéticas Numéricas Funções definíveis pelo usuário, Pré-programadas
Fonte de energia:	6.0V, DC: 4 pilhas (tipo UM-3,AA ou R6)
Consumo de força:	6.0V, DC: 0.13W
Tempo de operação:	Aprox. 50 horas com pilhas (tipo UM-3, AA ou R6)
Temperatura de operação:	0°C ~ 40°C
Dimensões:	195 x 86 x 25.5 mm
Peso:	Aprox. 375 g (com as pilhas)
Acessórios:	Capa protetora, quatro pilhas, dois gabaritos, etiqueta de nome, manual de instruções, manual de programas e módulo de memória para expansão, com RAM de 8 Kbytes, tipo encaixável. Interface para Impressora/Gravador de fita cassete.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

ÍNDICE

Introdução 1
Notas Operacionais 2
Especificações do PC-1500 3
Índice 4

0. **Programação Instantânea** 8
A. Exemplo 1 8
B. Exemplo 2 9

I. **Algumas Informações** 11
A. Teclas LIGA e DESLIGA (ON e OFF) 12
B. Teclas Alfabéticas 12
C. Teclas Numéricas e Teclas de Operação Aritmética 12
D. Tecla de Segunda Função (SHIFT) 12
E. Letras minúsculas e a Tecla SMALL 12
F. O Visor 13
G. O Cursor e o "Pronto" 13
H. Limpeza 13
I. Entrada (ENTER) 13
J. Mensagens de Erro 13
K. Indicação de Carga da Pilha 14
L. Tecla RCL (de chamada) 14

II. **Mergulhando** 15
A. Modo (MODE) 15
B. Cálculos Simples 15
C. Cálculos em Série 16
D. Cálculos com Números Negativos 16
E. Cálculos Compostos 17
F. Uso de Parênteses 17
G. Comparações Lógicas e Desigualdade 18
H. Teclas de Edição e Funções 19
H.1. Tecla Seta para esquerda DEL (cancelamento) 19
H.2. Tecla Seta para direita(INS (inserção) 20
H.3. Função de chamada 21
I. Variáveis 21
J. Pausa 24
SUMÁRIO 24

III. **A Arte Misteriosa da Programação** 26
A. O que é um programa? 26
B. Como programar? 26
C. Comandos vs. Declarações 27
D. Números de Linha 27
E. Teclas de Revisão da Linha de Programa 27
F. Uma Visão Minuciosa em alguns Velhos Amigos 29
F.1. O Comando New 29

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F.2. A Declaração LET 29
F.3. A Declaração PRINT 29
G. A Declaração PAUSE 34
H. Entrada (INPUT) 35
I. Sugestões Úteis para agilizar o Trabalho 39
I.1. Abreviações 39
I.2. Declarações Múltiplas e os Dois Pontos 41
J. Correção de Erro no Modo de Programa (PRO) 42
K. O Comando LIST 43
L. Quanto mais, melhor 43
L.1. A Declaração END 43
L.2. RUN com número de linha especificada 44
M. Declarações de controle 44
N. IF e THEN 44
O. GOTO 46
P. FOR e NEXT 51
Q. WAIT 54
R. READ, DATA e RESTORE 55
S. REM (REMARK) 57
T. GOSUB... RETURN 58
SUMÁRIO 59

IV. Cálculos Avançados 60
A. Notação Científica 60
B. Capacidade de Cálculos 62
C. Raiz, Potência e Pi 62
D. Modos Angulares 64
E. Funções Trigonométricas
SEN, COS, TAN, ASN, ACS, ATN 64
F. Funções Logarítmicas
LN, LOG, EXP 65
G. Conversão Angular
DEG, DMS 66
H. Funções Diversas
ABS, INT, SGN 66

V. Programação Avançada 67
A. Matrizes e a Declaração DIM 67
B. Algo Mais Sobre Cordões de Caracteres 70
B.1. Dimensionando os Cordões 70
B.2. Concatenação 70
B.3. Comparação dos Cordões 72
C. Funções 72
C.1. ASC 72
C.2. CHR$ 72
C.3. INKEY$ 73
C.4. LEN 75
C.5. LEFT$ 75
C.6. MID$ 76
C.7. RIGHT$ 77
C.8. RND 77
C.9. RANDOM 78
C.10. STR$ 78

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.11. STATUS 79
C.12. TIME 79
C.13. VAL 80
D. PRINT USING 80
E. Transferência de Controle Computado 83
ON GOTO, ON GOSUB, ON ERROR GOTO 84
F. Programação do Visor 84
F.1. BEEP 84
F.2. CURSOR 85
F.3. CLS 88
F.4. GCURSOR 88
F.5. GPRINT 91
F.6. POINT 94
G. DEBUGGING
TRON, TROFF, teclas de Letra 96
H. Números Hexadecimais e Funções Booleans 98
H.1. Números Hexadecimais 98
H.2. Função AND 98
H.3. Função OR 99
H.4. Função NOT 99
I. Parando a execução do Programa
STOP, CONT 100
J. Controle do Modo
LOCK, UNLOCK 100

VI. Expandindo o PC-1500 101
A. A Impressora/Interface Cassete 101
A.1. Conexão do Computador à Interface 101
A.2. Fonte de Alimentação 103
A.3. Conectando um gravador de Fita Cassete na Interface 103
A.4. Colocando o Papel 105
A.5. Substituindo as Canetas 106
B. Usando um Gravador de Fita Cassete 108
B.1. Operação do Gravador 108
B.2. O Comando CSAVE 108
B.3. O Comando CLOAD 109
B.4. Comandos PRINT # e INPUT # 110
B.5. Uso do Comando MERGE 112
B.6. A Declaração CHAIN 113
B.7. Usando Dois Gravadores de Fita 114
C. Usando a Impressora 116
C.1. Especificações da Impressora 116
C.2. O Comando TEST 116
C.3. Impressão de Cálculos 117
C.4. Modos da Impressora 118
C.5. Listagem de Programas 118
C.6. Controle Programável da Impressora 120
CSIZE 120
ROTATE 120
COLOR 121
LF 121

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

LPRINT 122
LCURSOR 123
TAB 124
SORGN 124
GLCURSOR 124
LINE 125
RUNE 127

VII. Modo RESERVE 129
A. Definição e Seleção das Teclas de Reserva 129
B. Identificação das Teclas de Reserva 130
C. Cancelamento de Programas de Reserva 131

VIII. Iniciando a Execução do Programa 132
A. A Tecla DEF 132
A.1. Execução de Programas Definíveis 132
A.2. Palavras-Chave pré-atribuídas 132
A.3. A Declaração AREAD 133
B. Iniciando o Programa Automaticamente
ARUN 134
C. Comparação entre Métodos de Iniciação 134
C.1. Área de Memória Fixa 135
C.2. Tabela Comparativa do Programa 135

IX. Apêndice 137
A. Abreviações 138
B. Substituição das Pilhas para o PC-1500 142
C. Código de Caracteres ASCII para o PC-1500 146
E. Lista de Códigos de Erros do PC-1500 147
O. Ordem das Evoluções de Expressões 152
X. Comparação de Comandos 155
Z. Tabela de Referências dos Comandos 157

Etiqueta de Nome

Escreva o seu nome na etiqueta, e cole-a na parte traseira do Computador.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PROGRAMAÇÃO INSTANTÂNEA

(Não é necessário água)

Esta secção é dedicada exclusivamente para um grupo de pessoas selecionadas, cuja curiosidade excede a sua paciência (e talvez o seu senso comum). Para aqueles de vocês que precisem fazer alguma coisa com esta maravilha da eletrônica moderna, nós apresentamos um simples exercício de programação. (Advertência: Os tímidos ou medrosos devem seguir para capítulo 1; Preparando-se para uma introdução mais minuciosa e pausada sobre o PC-1500 da Sharp).

Antes de você prosseguir, mais uma precaução é necessária. É importante seguir todos os passos listados na ordem dada. Ao contrário da opinião popular, os computadores não são "super-cérebros" e não tem a qualidade da habilidade humana de "imaginar" o que você deseja. O PC-1500 simplesmente espera as suas instruções e as efetua. Você está pronto? Ótimo, vamos começar.

EXEMPLO 1:

Primeiro, procure a tecla marcada "ON" no canto direito superior do teclado.
Pressionando esta tecla, você fará com que o espírito eletrônico adormecido acorde (não espere um bufo de fumaça!)
A parte do visor do computador deverá ser semelhante à ilustração abaixo:

Pressione a tecla MODE (do lado direito) até a abreviação PRO aparecer no topo do visor. (Se você pressionar a tecla mais de uma vez, pressione-a novamente até o resultado desejado ser obtido). O PC-1500 da Sharp está pronto para aceitar a série de instruções que constituem um programa de computador.
Digite a seguinte seqüência de teclas:

[1] [Ø] [A] [=] [1] [ENTER]

Note que quando você pressionar a tecla ENTER, o computador modificará o que você digitou.
O visor deve agora aparecer assim:

NOTA: Nós usamos Ø para o número zero, assim você pode distinguir entre a letra e o número. O visor e a impressora fazem desta maneira - nós faremos o mesmo a cada passo do manual.

Continue pressionando as teclas seguintes. Não fique espantado quando cada linha desaparecer assim que você digitar o seu sucessor.

[2] [Ø] [P] [A] [U] [S] [E] [A] [ENTER]
[3] [Ø] [A] [=] [A] [+] [2] [ENTER]
[4] [Ø] [G] [O] [T] [O] [2] [Ø] [ENTER]

Neste ponto, o seu primeiro programa está completo. Agora você deve falar ao computador para "executar" as instruções que ele contém. Este processo é conhecido como "Execução do Programa" e é efetuado no modo RUN. Pressione a tecla [ENTER] novamente e as letras PRO são substituídas pelas letras RUN no topo do visor. Um último passo: digite as letras [R] [U] [N] e pressione [ENTER].

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Parabéns! Seu primeiro programa BASIC está rodando.
Suas instruções estão sendo seguidas, e o computador está ocupado, listando todos os números ímpares positivos na ordem.

"Então", você fala para você mesmo: "Eu sou um gênio. Mas quando ele vai parar?" Bem,... infelizmente, sem a sua intervenção ou falha de bateria, este programa em particular nunca vai parar. Para vermos porque, vamos rever o nosso programa:

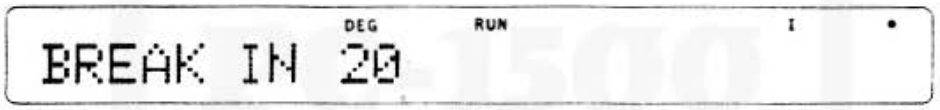
```
10 A= 1
20 PAUSE A
30 A = A + 2
40 GOTO 20
```

O efeito da linha de número 40 é fazer com que o computador efetue novamente todas as linhas após a de número 20. Isto inclui a linha 40, a qual evidentemente, fala ao computador para tornar a reefetuar as linhas 20, 30 e 40... e assim por diante. Esta repetição sem fim é conhecida no Jargão de computação como "Looping".

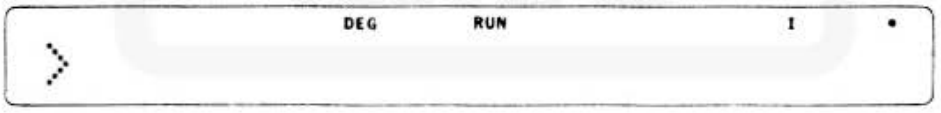
O Nosso programa está fixo em um "Loop Infinito" e somente você tem a capacidade de parar este trágico desgaste de pilhas. Para fazer isto, pressione a tecla ON. Como o PC-1500 já está ligado, você está realmente selecionando a função BREAK.

Não se espante. Apesar do que o nome parece indicar, esta não é uma tecla de autodestruição. Somente para lembrá-lo, você não pode de maneira alguma, danificar o computador simplesmente pressionando as teclas, então, sinta-se livre para experimentar.

Se você tiver pressionado a tecla BREAK, uma mensagem similar à seguinte será visível no visor:



Ela informa qual a instrução que estava sendo "Executada", quando você interrompeu o computador. Pressione a tecla BREAK mais uma vez, e o computador esperará a sua próxima instrução:



Para aqueles de vocês que acabam de lembrar-se que deixaram a torneira aberta em casa, esta é uma boa oportunidade para parar. (Antes de deixá-lo, por favor, pressione a tecla OFF para conservar as baterias). Outros de vocês, já estão entregues à programação e desejam continuar a sua instrução com o nosso segundo exemplo.
(Nós não nos responsabilizamos se você estiver atrasado para o jantar).

EXEMPLO 2

Para começar o nosso segundo programa, é necessário introduzir o modo de programa pressionando a tecla MODE até as letras PRO (abreviação para PROGRAMA) substituir as letras RUN no topo do visor. O PC-1500 nos deixará submetê-lo a um novo programa ou modificar um velho programa. Como o nosso novo programa não aproveitará as instruções do nosso velho programa, nós devemos limpar aquelas instruções da memória do computador. Para fazer isto, digite a palavra NEW e pressione ENTER. Após uma pausa, o caracter > (chamado Pronto) retornará.

Digite as seguintes teclas para introduzir a primeira linha do programa:

1 0 I N P U T SHIFT " T A M A N H O SPACE D A
SPACE L I S T A SHIFT / SHIFT " SHIFT + A ENTER.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Note que pressionando a tecla [SHIFT] seguida pela tecla que tenha um outro caracter inscrito acima da tecla, introduzirá o caracter de cima. A tecla [SHIFT] permite dois caracteres compartilharem a mesma tecla, e é as vezes chamada de tecla de "Segunda Função". Então, na primeira linha do nosso programa (linha 10 acima), uma digitação da tecla [SHIFT] seguida pela tecla [+] faz com que [:] (ponto e vírgula) seja introduzida. A linha toda é armazenada no computador como:

```
                DEG        PRO        ¤    •
10: INPUT "TAMANHO DA LISTA
```

Pressione a tecla [▶]

```
                DEG        PRO        ¤    •
10: "TAMANHO DA LISTA?";A_
```

Neste manual nós ilustraremos a seleção do caracter de segunda função com a tecla [SHIFT] e o caracter desejado. Por exemplo, a linha numerada de 10 acima será mostrada como:

[1] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [T] [A] [M] [A] [N] [H] [O] [SPACE] [D] [A]

[SPACE] [L] [I] [S] [T] [A] [SHIFT] [?] [SHIFT] ["] [SHIFT] [;] [A] [ENTER]

Complete a entrada do seu segundo programa com a seguinte digitação de teclas:

[2] [Ø] [I] [F] [A] [SHIFT] [<] [=] [Ø] [G] [O] [T] [O] [9] [9] [ENTER]

[3] [Ø] [F] [O] [R] [I] [=] [1] [T] [O] [A] [ENTER]

[4] [Ø] [P] [A] [U] [S] [E] [I] [SHIFT] [,] [I] [*] [I] [ENTER]

[5] [Ø] [N] [E] [X] [T] [I] [ENTER]

[9] [9] [B] [E] [E] [P] [2] [SHIFT] [:] [E] [N] [D] [ENTER]

Nosso segundo programa é agora armazenado na memória do PC-1500. Você se lembra o que deve ser feito agora? Se você disser "Rodar o Programa", você está bem no caminho da competência em programação.

Retorne para o modo RUN (Sugestão: use a tecla MODE) e digite a palavra RUN. Pressione [ENTER] para iniciar a "Execução" do nosso segundo programa.

O computador está te perguntando como indicado abaixo? (Caso não esteja, retorne para o modo PRO e reverifique o seu programa).

```
                DEG        RUN        ¤    •
TAMANHO DA LISTA?
```

Parabens! Nosso programa está solicitando que o usuário (você) dê a informação necessária para ele efetuar a tarefa que nós, como programadores, o instruímos. Chame a primeira linha do nosso programa BASIC (que você gentilmente digitou para nós);

```
                DEG        RUN        ¤    •
10: INPUT "TAMANHO DA LISTA
```

Pressione a tecla [▶]

```
                DEG        RUN        ¤    •
10: "TAMANHO DA LISTA?";A_
```

As instruções contidas nesta linha estão sendo seguidas pelo computador. O resultado é que o computador está esperando por você para "introduzir" (digitar) alguma informação.

Este programa imprimirá uma lista de números e seus quadrados (o número multiplicado por ele mesmo). Primeiro ele pergunta ao usuário, quantos números e quadrados imprimir (tamanho da lista). O usuário (você) responde digitando um número e pressionando [ENTER]

Digite [8] e pressione [ENTER].

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Observe cuidadosamente como os pares de números aparecem brevemente no visor. O segundo número do par (no lado direito) será o quadrado do primeiro. Oito pares de números serão visualizados, porque isto é o que você solicitou quando você respondeu para a questão do programa.

Quando o "PRONTO" retornar, rode novamente o programa (digite RUN, e pressione ENTER e dê uma resposta diferente para a questão "TAMANHO DA LISTA?". Rode o programa várias vezes experimentando com diferentes tamanhos da lista. Você está experimentando uma das importantes vantagens dos computadores programáveis; eles podem efetuar uma tarefa maçante repetidamente, variando-a um pouco cada vez, em função da resposta à questão.

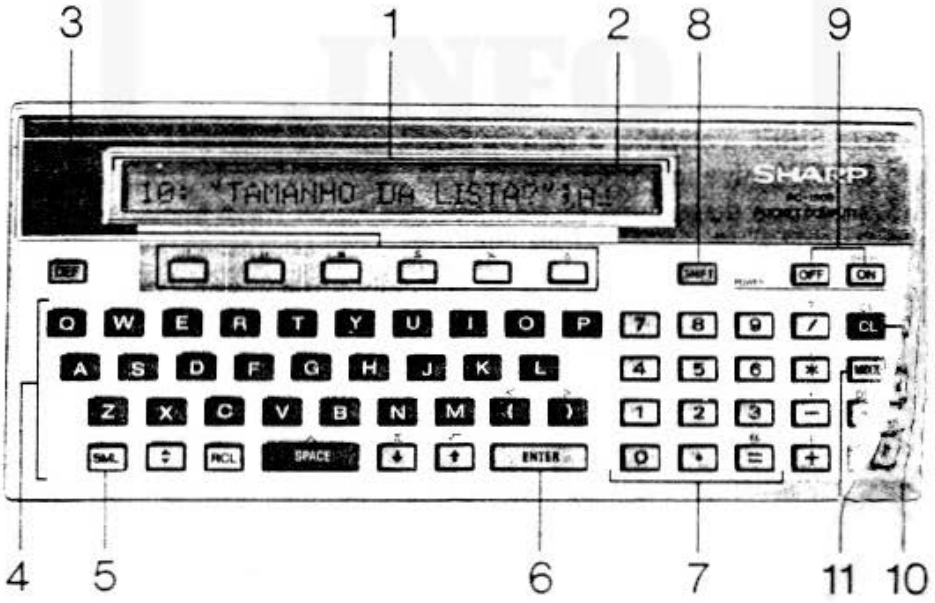
Rode o programa mais uma vez, mas desta vez, introduza um zero para o tamanho da lista. O que aconteceu? Sim, o programa termina sem ter produzido uma lista. Embora isto possa parecer estranho, o PC-1500 está simplesmente seguindo as nossas instruções.

Isto ilustra o porquê o computador é uma ferramenta poderosa. Ele pode ser programado para seguir diferentes conjuntos de instruções e processar as várias informações que são fornecidas. Por causa das instruções da linha 20, se a entrada do usuário for zero, (ou menos) o computador omite a computação da lista e vai para o fim do programa. De fato, ele toma uma decisão baseada no pedido do usuário. Como um programador, você controla quais as decisões que são possíveis e quando elas são tomadas. Então, todo o poder do computador está disponível para você solucionar o seu problema específico da maneira que você achar que é melhor.

I. ALGUMAS INFORMAÇÕES

Após desembalar o seu Computador de Bolso Sharp (daqui em diante, chama-lo-emos de Sharp) e admirar seu belo amigo, você pode imaginar a maravilha que você está iniciando.

Vamos examinar o Sharp.



1 – Visor
2 – Indicador de carga da pilha
3 – Teclas reserváveis
4 – Teclas Alfabéticas
5 – Teclas p/ alfabeto minúsculo
6 – Tecla de entrada
7 – Teclas numéricas e Teclas de memória
8 – Teclas de segunda função (SHIFT)
9 – Teclas liga (ON) e desliga (OFF)
10 – Tecla de limpeza
11 – Tecla de modo

Nós descreveremos o visor brevemente. Primeiro, mesmo antes de ligar o Sharp, note as várias características importantes do teclado:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

A) TECLAS LIGA E DESLIGA (ON e OFF)

Obviamente, estas teclas servem para ligar e desligar o computador.

Para observar as suas pilhas, o Sharp desliga automaticamente caso não seja digitado nada num período de cerca de sete minutos, a não ser que um programa esteja sendo executado.

Observe a inscrição acima da tecla [ON] , a palavra BREAK.

A tecla [ON] pode ser usada como BREAK, ou interromper a execução de um programa. Esta função é detalhada mais tarde neste manual.

B) TECLAS ALFABÉTICAS

As teclas alfabéticas permitem que o usuário do computador (você) dê instruções e introduza dados. Além disso, estas teclas podem ser usadas para designar "Áreas de Armazenamento" dentro da memória do computador nas quais você será capaz de guardar ou recuperar dados. Este uso será discutido na secção das variáveis. Letras minúsculas são obtidas através do uso das teclas [SHIFT] e [SML] (SMALL) descrito abaixo.

C) TECLAS NUMÉRICAS E TECLAS DE OPERAÇÃO ARITMÉTICA

Com estas, você introduz os números para os cálculos. As teclas [+] [−] [*] [/] fazem o Sharp somar, subtrair, multiplicar e dividir respectivamente. A tecla [E] permite a entrada dos números na "Notação Científica". O uso desta notação e outras funções sofisticadas são descritas no capítulo de Cálculos Avançados.

D) TECLA DE SEGUNDA FUNÇÃO (SHIFT)

Esta tecla libera as funções inscritas acima de muitas teclas não-alfabéticas. Por exemplo, para digitar dois pontos, pressione a tecla [SHIFT] e então a tecla [*] (asterísco). Quando a tecla SHIFT é seguida por uma tecla alfabética, a letra minúscula é visualizada. (NOTA: No modo SMALL, a SHIFT seguida da tecla alfabética produzirá uma letra maiúscula).

Quando a tecla SHIFT é ativada, a palavra SHIFT aparece no canto esquerdo superior do visor. O modo SHIFT é ativado somente para uma digitação de tecla por vez.

As seis teclas no topo do teclado, diretamente abaixo do visor, são chamadas de TECLAS RESERVÁVEIS. Usando a tecla SHIFT de maneira que nós descrevemos mais tarde, você poderá atribuir comandos frequentemente usados ou outras operações a estas teclas.
NOTA: Se você pressionar a tecla [SHIFT] por engano, pressione-a novamente para cancelá-la.

E) LETRAS MINÚSCULAS E A TECLA SMALL

A tecla [SML] permite você especificar os caracteres minúsculos para todas as teclas alfabéticas. Se você não especificar o modo SMALL, o Sharp selecionará os caracteres maiúsculos cada vez que você pressionar uma tecla alfabética. (Nós chamamos isto de modo "Default", significando o modo no qual a máquina opera, a menos que você determine o contrário). Você pode digitar letras minúsculas individuais pressionando a tecla [SHIFT] antes da letra.

A tecla [SML] pode ser usada para ativar o modo SMALL.

No modo SMALL, as letras minúsculas são obtidas através do pressionamento de uma tecla alfabética, e as letras maiúsculas individuais são obtidas através do pressionamento da tecla [SHIFT] primeiro. Quando o computador estiver no modo SMALL, a palavra SMALL aparecerá na parte superior do visor. Tendo você, posicionado o computador no modo SMALL uma vez, ele permanecerá neste modo até você pressionar a tecla [SML] novamente.
NOTA: Nós recomendamos que você evite o uso de caracteres minúsculos no momento. Isto porque, o Sharp reconhece somente as instruções em caracteres maiúsculos.

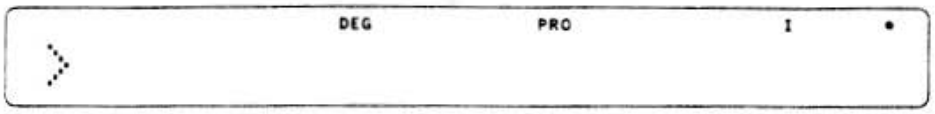
Quando você aprender a programar, você utilizará os caracteres minúsculos facilmente.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F) O VISOR

Pressione ON . A parte do "Vidro" do computador é chamado de "visor". Ele se parece com o que temos abaixo:



No visor, você deve ver um > (chamado "Pronto"), várias palavras ou abreviações, e um ponto (indicando que a pilha está carregada). Não fique preocupado se as abreviações específicas que aparecerem no seu visor não forem aquelas da nossa ilustração. Estes símbolos mudam conforme você operar o Sharp.

G) O CURSOR E O "PRONTO"

No lado esquerdo do visor, encontra-se o símbolo "PRONTO" (>); ele informa que você pode falar com o Sharp. Quando o "pronto" aparece, ele significa que o Sharp não tem planos imediatos e espera a sua ordem.

Digite uma letra de sua escolha. Ela substituirá > no lado esquerdo do visor, enquanto que no lado direito da sua letra aparecerá um – (símbolo de sublinhar). Isto é um cursor. Conforme você pressiona cada tecla, o cursor avança de um lado para o outro do visor, indicando onde aparecerá o próximo símbolo que você digitar. **Digite o seu nome e note o movimento do cursor.**

Se você digitar mais que 25 caracteres (limite que pode ser visualizado em uma vêz), a linha inteira desloca-se para a esquerda. (Tente). Os caracteres colocados para fora do visor não são perdidos: eles permanecem no computador como parte da linha digitada, até um máximo de 80 caracteres. Nós veremos como "chamar" e como mudar estes caracteres na secção adiante.

H) LIMPEZA

(A tecla CL vermelha no canto direito superior)

Pressione este tecla, e você limpa o conteúdo do seu visor.

Use-a para apagar os caracteres que você acabou de digitar.

Note que o pronto retornou, indicando que o computador está novamente esperando pelos seus comandos.

A tecla de limpeza é usada também para cancelar um comando incorreto. (Veja a secção abaixo de Mensagens de Erro).

I) ENTRADA (ENTER)

Conforme você digita no computador, as letras ou números aparecem no visor. O Sharp não tomará Ação, enquanto você não informar que terminou de digitar (afinal, ele não pode ler a sua mente). Isto é feito através do pressionamento da tecla ENTER após suas outras digitações. Neste ponto, o computador varrerá os caracteres que você digitou na forma correta. Alguns erros causarão a rejeição da sua entrada nesta fase.

LEMBRE: Pressione a tecla ENTER cada vez que você desejar introduzir uma instrução ou item de dados na máquina.

J) MENSAGENS DE ERRO

Pressione as teclas seguintes:

1 + 1 =

Agora, pressione ENTER

O resultado deve ser visualizado. Três, certo? Não?

Você obteve "ERROR 1" como a sua resposta?

Seu computador está com defeito? Nunca! Existe um erro no seu coman

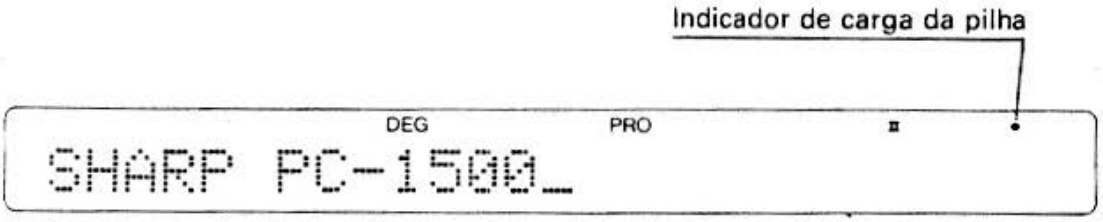
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

um código de erro o qual indica que você efetuou o cálculo incorretamente. (Para matar a sua curiosidade, uma lista completa de mensagens de erro está incluída como um apêndice). Nós nos responsabilizaremos por este primeiro erro, e na secção posterior, nós mostraremos como usar as outras teclas para corrigir um comando errado. No momento, você pode usar a tecla CL para apagar a mensagem de erro.

K) INDICAÇAO DE CARGA DA PILHA

Quando este ponto desaparecer, está na hora de rejuvenecer o Sharp, substituindo as suas pilhas. Veja o apêndice para as instruções.



L) TECLA RCL (DE CHAMADA)

Esta tecla se usa para ler informações ou sentenças previamente armazenadas no modo RESERVE para efetuar a leitura de um registro do modo reserve, efetuar os seguintes passos:

- Apertar a tecla ▲▼ para selecionar o grupo I, II ou III
- Pressionar a tecla [RCL] e a tecla reservável previamente assinalada

Em casos de dúvidas, seguir instrucões do parágrafo B na página 130.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

II. MERGULHANDO

Realmente, este capítulo não está com o nome adequado. Aprender a usar o PC-1500 não é tão chocante como mergulhar numa piscina. Entretanto, é necessário que você dirija-se ao computador sem receio. Como nós já falamos antes, **VOCÊ NÃO CONSEGUIRÁ DANIFICAR O COMPUTADOR SIMPLESMENTE PRESSIONANDO AS TECLAS.**

Neste capítulo, nós exploraremos as características fundamentais do computador nas quais estão baseados os programas e os cálculos avançados. Tome o tempo necessário para trabalhar através de exemplos em cada secção. Um bom entendimento desta base permitirá que você explore todo o potencial desta máquina.

Embora não seja recomendável, se você sentir que está suficientemente avançado, você pode pular para o sumário no fim deste capítulo.

A. MODO (MODE)

Vamos começar com uma tecla que nós temos ignorado até agora. No lado direito do teclado, encontramos uma tecla muito importante chamada MODE. Pressione-a repetidamente. Note que cada vez que você pressionar esta tecla, muda a abreviação do lado direito superior, mudando o modo da máquina. Para cada novo modo, o funcionamento da máquina é diferente.

O Sharp deve ser colocado no modo próprio para operação de acordo com o planejado. E, quando você tentar operar o Sharp de modo impróprio, o computador avisar-te-á rapidamente do seu erro.

Através do pressionamento repetitivo da tecla MODE, você alterará os dois mais importantes, dos três modos do computador: RUN e PROGRAMA. O terceiro modo, RESERVE, é ativado através do pressionamento da tecla SHIFT antes da tecla MODE. Nos capítulos posteriores deste manual descreveremos como cada um desses modos contribuem para o funcionamento suave do computador. Por enquanto, lembre-se que para usar o Sharp como uma calculadora, você deve estar no modo RUN. Mais tarde, no modo PROGRAMA, você escreverá e alterará os programas. Com o modo RESERVE, você pode atribuir os comandos freqüentemente usados a uma tecla. Isto é explicado com mais detalhes no capítulo 7.

Note que a primeira vez que você ligar o computador após inserir as pilhas, ele se posicionará no modo PRO. Da outra vez que o computador for ligado, será posicionado no modo o qual foi operado pela última vez antes de ser desligado.

B. CÁLCULOS SIMPLES

Com o Sharp posicionado no modo RUN, vamos testar as computações matemáticas fundamentais. Antes de cada cálculo, é necessário pressionar [CL]. Isto limpa o visor de qualquer dado anterior, o qual pode interferir com um cálculo novo.

*Ache as respostas para os simples problemas seguintes.

Entrada	Visor
[5] [+] [2] [ENTER]	7
[5] [−] [2] [ENTER]	3
[5] [/] [2] [ENTER]	2.5
[5] [*] [2] [ENTER]	10

NOTA: Não digite o sinal de igual. Lembre-se do capítulo I, que é a tecla ENTER que informa o Sharp que você terminou de digitar, e deseja ter seu comando ou cálculo efetuado.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C. CÁLCULOS EM SÉRIE

Você pode utilizar uma resposta de um cálculo no cálculo seguinte, seguindo diretamente para o segundo cálculo. (Não pressione CL entre os cálculos). Então, se você está efetuando o balanço de um livro caixa, você operará o Sharp desta maneira!

Entrada	Visor
161.16 – 47.5Ø [ENTER]	113.66
– 12.33	113.66–12.33_
[ENTER]	101.33

Como você observará, o resultado do primeiro cálculo pula para a esquerda do visor, quando você começar o segundo cálculo.

NOTA: Não digite sinais de cifrão ou vírgula quando introduzir números nos cálculos. Estes símbolos possuem significado especial na linguagem BASIC.

Outras operações podem ser efetuadas similarmente. Tente estas:

Entrada	Visor
5 + 3 [ENTER]	8
8 + 3 – 1 [ENTER]	10
1Ø * 3 – 1 [ENTER]	29
29/3 – 1 [ENTER]	8.666666667

D. CÁLCULOS COM NÚMEROS NEGATIVOS

Imagine você ter presenteado o seu professor de Ciências, Sr. Pedro, com duas maçãs. Você ficou com cinco maçãs, e quer saber, "Quantas maçãs teria agora se eu não tivesse sido tão generoso com o Sr. Pedro?" Para encontrar a resposta, você pode imaginar a subtração de uma quantidade negativa; isto é, você subtraria a subtração. Digite no computador: 5– –2 [ENTER]. Sete seria o seu resultado hipotético. Tente estes cálculos similares com números de sinais negativos.

```
5*-2
5+-2
5--2
-5-2.3
-5+-2
-5--2
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Lembre-se: Pressione [CL] entre os cálculos para limpar os resultados anteriores.

E. CÁLCULOS COMPOSTOS

Você pode enfileirar juntos, uma sequência de cálculos antes de perguntar uma resposta para o Sharp. Por exemplo, você e dois amigos desejam dividir 05 maçãs duas vezes ao dia por uma semana. Quantas maçãs deverá comprar até o fim da semana? 5 maçãs [/] 3 amigos [*] 2 por dia [*] 7 dias:

Digitação de Teclas	Visor
[5] [/] [3] [*] [2] [*] [7] [ENTER]	23. 33333333

(Compre 24 maçãs e um papagaio, para quem você pode dar os 2/3 da maçã que sobrou). Rode os cálculos seguintes, (mas desta vez, invente a sua própria estória).

Entrada	Visor
5 * 2 – 3.675 [ENTER]	6. 325
5 / 3 * 6.2 + 7 – 47 [ENTER]	–29. 66666667

F. USO DE PARÊNTESES

Um problema que aparece conforme nós indagamos os cálculos compostos é a prioridade. Por exemplo, a expressão 5-3/4 pode ser lida de duas maneiras: (5 menos 3) dividido por 4, onde a resposta é .5, ou 5 menos (3 dividido por 4), onde a resposta é 4.25.

Localizado na primeira carreira de teclas, estão os parênteses o qual você pode usar para classificar tal ambiguidade. Execute os seguintes cálculos:

Entrada	Visor
5 – 3 / 4 [ENTER]	4. 25
5 – (3 / 4) [ENTER]	4. 25
(5 – 3) / 4 [ENTER]	0. 5

O Sharp está disposto (tem prioridades na ausência de indicação) para efetuar alguns cálculos antes de outros. (para uma completa listagem da ordem que o Sharp calcula, veja o apêndice 0). Divisão e multiplicação serão efetuadas antes de subtração e adição, a não ser que seja indicada uma ordem diferente. Então, o Sharp está estruturado para interpretar a primeira equação acima como sendo igual à segunda ao invés da terceira. Para certificar-se que a resposta que o Sharp te dá é o que você necessita, use parênteses para indicar a ordem correta que ele deve efetuar.

O Sharp pode interpretar várias leis de parênteses como neste problema:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Entrada	Visor
((6 – 4) / 2) * (((3 – 1) / 4) * 6) ENTER	3

As equações dentro de parênteses serão sempre calculadas em primeiro lugar.

LEMBRE: Na dúvida, use parênteses para esclarecer a ordem da sua operação aritmética.

G. COMPARAÇÕES LÓGICAS E DESIGUALDADE

O Sharp permitirá que você compare dois valores ou equações, e indicará o resultado da comparação. Esta capacidade é básica para programas com decisões. A maneira pela qual isto é feito será reconhecível pelos estudantes de "Nova Matemática" como uma desigualdade (não se desespere se você não foi educado com a Nova Matemática; o autor não foi também).

Uma desigualdade pode ser pensada como uma comparação que pode ser verdadeira ou falsa. Por exemplo, a instrução "seis dividido por três é igual à dois" é uma comparação verdadeira. Por outro lado, a instrução "seis dividido por três é maior que cinco" é uma comparação falsa.

Computadores e matemáticos usam os seguintes símbolos para os possíveis tipos de comparações:

<	menor que
>	maior que
=	igual à
<=	menor que ou igual à
>=	maior que ou igual à
<>	não é igual à

Então, nós podemos expor novamente as desigualdades acima simbolicamente, como: 6/43 = 2 e 6/3 > 5 respectivamente.

Dada uma desigualdade, o Sharp determinará se a comparação é verdadeira ou falsa. De acordo com a prática de designio do computador, o Sharp indicará uma instrução verdadeira com um 1 e uma instrução falsa com um Ø. Por exemplo, se você digitar:

6/3 = 2

O Sharp responderá com um 1 (para verdadeira). Digitando a sequência:

6/3 > 5

Concluirá uma resposta Ø (para falsa)

Tente os testes seguintes de avaliação do Sharp (certifique de que você está no modo RUN):

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de Teclas	Resultado
4 SHIFT > 5 ENTER	Ø
5 SHIFT > 4 ENTER	1
2 * 2 SHIFT < 2 * 3 ENTER	1
1 8 SHIFT < SHIFT > 1 8	Ø
2 = 2 ENTER	1
5 SHIFT < = 5 ENTER	1
2 5 * 2 SHIFT > = 1 Ø Ø / 3 + 4 ENTER	1

Como você pode ter observado, as equações a serem comparadas podem ser tão complexas quanto o necessário (o único limite é a restrição de 80 símbolos por linha).

Aqui está um problema muito simples para ilustrar um uso prático de desigualdades:

Pedro entrou numa loja de ferragens. Lá, ele encontrou cimento que vem em sacos de 4,8 e 12 kg. O dinheiro que ele tem, dá para comprar dois sacos de 12 kg ou três sacos de 4 kg e um saco de 8kg. Ele gostaria de saber em qual caso ele obterá mais cimento. Ele pergunta para o Sharp, se

$$2 * 12 > (3 * 4) + 8$$

O Sharp responde 1; Pedro compra dois sacos de 12kg.

H. TECLAS DE EDIÇÃO E FUNÇÕES

Os mais humanos (exceto os gênios como nós) tem uma tendência de cometer erros. Reconhecendo as facilidades humanas, os projetistas do PC-1500 incorporaram várias características que facilitam as mudanças e correções.

H.1. Seta para esquerda/tecla DELete DEL ◄ (cancelamento)

Vários de vocês devem ter percebido a tecla de seta para esquerda no lado direito inferior do teclado.

Esta tecla atua como a tecla de retrocesso nas mais modernas máquinas de escrever; ela permite que você mova os caracteres digitados anteriormente.

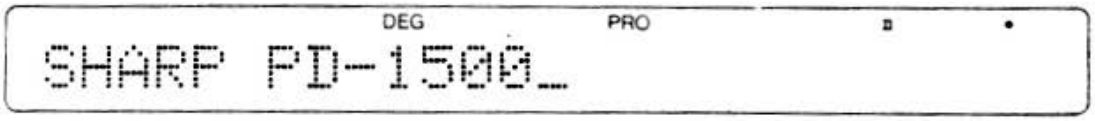
No modo RUN, iniciando com um visor limpo (isto é, com o símbolo Pronto) **digite os seguintes caracteres**:

S H A R P SPACE P D - 1 5 0 0

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

O visor deve aparecer conforme segue:



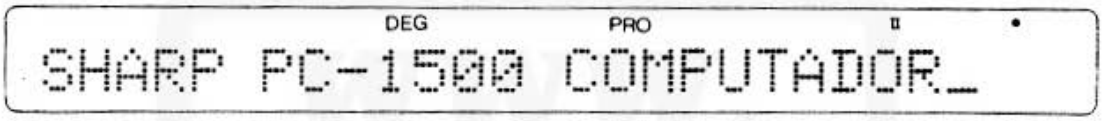
Pressione a tecla Seta para esquerda uma vez, e note a maneira que o cursor se move. Esta forma de "grade pisca-pisca" do cursor permite que você veja o caracter na posição corrente do cursor. **Pressione a tecla Seta para esquerda repetidamente (ou mantenha-a pressionada) até o cursor posicionar sobre o D.** (caso você passe o D acidentalmente, mova o cursor para frente novamente com a tecla Seta para direita).

Digite um C. O C substitui o D e o cursor se move para a frente. Isto não deve ser surpresa, se você lembrar que o cursor indica onde o próximo caracter será colocado.

Continue digitando:

[SPACE] [C] [O] [M] [P] [U] [T] [A] [D] [O] [R]

No visor teremos:

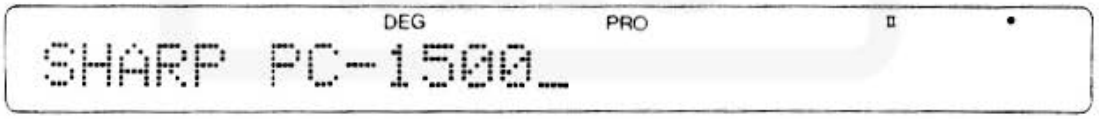


Conforme o visor mostra, os caracteres que foram digitados por cima desaparecem.

Em adição ao retrocesso, a tecla de Seta para esquerda tem um segundo uso, a função de cancelamento, cuja a abreviação está inscrita sobre a tecla (DEL). para cancelar um caracter, posicione o cursor sobre o caracter, e pressione [SHIFT] [DEL] .

Vamos tentá-lo: mova o cursor para trás até o 1.° C, e pressione a sequência de [SHIFT] [DEL] dez vezes.

Agora, o visor mostra:



H.2. Seta para direita/Tecla INSert (Inserção) [INS ►]

Conforme nós já vimos, a tecla de Seta para direita move o cursor para a frente sem apagar os caracteres. Como a Seta para esquerda, o cursor moverá repetidamente se a tecla for mantida pressionada.

Mova o cursor para o fim da linha.

A segunda função da tecla de Seta para direita nos dá a capacidade de inserir os caracteres dentro de uma linha já digitada. Esta característica é útil para aqueles que tendem a esquecer pequenas coisas (como letras e palavras).

Mova o cursor para trás até o P em PC.

Pressione a sequência de SHIFT INS seis vezes e veja os caracteres presentes deslocarem para a direita. Os novos caracteres parecidos com um bloco, podem ser imaginados como "guardadores de lugar", e podem ser ocupados com nova informação.

Digite: [S] [U] [P] [E] [R] [SPACE]

Pronto! Você acaba de inserir uma palavra.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

H.3. FUNÇÃO DE CHAMADA

Até agora, nós temos discutido os meios de corrigir as instruções, que não tenham sido introduzidas ainda (isto é, antes de ter pressionado ENTER após as suas digitações). Uma vez que você pressionou ENTER, o Sharp tentará efetuar o seu cálculo imediatamente. Se o computador for bem sucedido, o resultado substituirá a sua equação no visor. Porém, a equação não está perdida, e pode ser chamada (visualizada novamente) através da tecla Seta para a esquerda ou Seta para a direita.

Limpe o visor e digite a equação de sua escolha.

Pressione ENTER para computar o resultado. **Chame a sua equação.** Note que, para ver o resultado novamente, é necessário computar novamente a equação (usando a tecla [ENTER]).

Felizmente, a função de chamada trabalhará também se o Sharp encontrar um erro enquanto estiver tentando avaliar (compreender) a sua entrada. Isto permite que você chame e corrija a equação errada usando qualquer uma das características de edição que você já aprendeu.

Para testar isto, **introduza a seguinte expressão digitada incorretamente:**

```
45 * 63 / * 2  ENTER
             ↑
```

Quando a mensagem de erro aparecer (ERROR 1) **pressione qualquer uma das teclas de Seta para chamar a expressão.**

Você deve ver o cursor de grade pisca-pisca sobre o segundo símbolo de multiplicação (asterisco). Este é o meio do Sharp indicar o local do erro (e por boa razão). Daqui para frente, você pode prosseguir fazendo qualquer correção que considerar apropriada.

I. VARIÁVEIS

A capacidade de trabalhar de maneira abstrata através de variáveis é uma das características mais poderosas do Sharp. Variáveis podem ser consideradas como um grupo de pequenos blocos, onde cada um pode ser ocupado com um dado tais como, um número ou um nome.

Você pode lembrar das variáveis da álgebra no colegial.

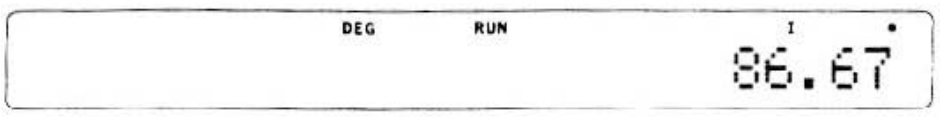
Você aprendeu (pelo menos, você foi ensinado) que se 5A = 30, então A deve ser 6, mas se 5A = 35, então A é igual à 7. A neste caso é uma variável que retém um número chamado de valor da variável. A capacidade de usar uma letra (tal como A) na equação em vez de um número específico é muito útil. Vamos ver porque, com o exemplo seguinte:

Sr. João quer adquirir um jogo de tacos de golfe prateados. O jogo consiste de 5 tacos à Cr$ 12,00 cada, uma sacola à Cr$ 21,99 e três bolas à Cr$ 1,56 cada. A loja A está oferecendo este jogo com 10% de desconto, mas eles tem um encargo de polimento e de entrega de Cr$ 8,00. A loja B está oferecendo o mesmo jogo com um desconto de 5% e livre de encargos.

Para computar qual é o melhor negócio, o Sr. João decide pedir auxílio do seu PC-1500 da Sharp. Ele calcula o custo básico do jogo e acumula o seu resultado na variável G:

```
G = (5 * 12) + 21.99 + (3 * 1.56)  ENTER
```

O resultado é visualizado e pode ser chamado da memória através da introdução do nome da variável; [G] [ENTER]

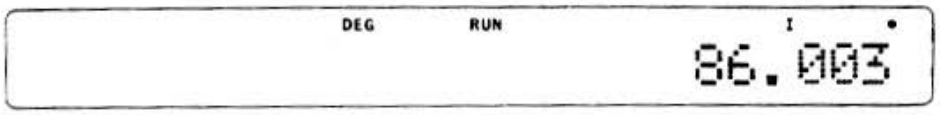


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Agora, ele calcula o preço de aquisição real na loja A:

G – (G * .10) + 8 [ENTER]



E o preço de aquisição na loja B:

G – (G * .05) [ENTER]

Claramente, a loja B tem a melhor oferta.

Existem várias lições para aprendermos com este exemplo.
Observe que os primeiros cálculos do Sr. João tiveram a seguinte forma:

nome da variável = expressão

Uma instrução com esta forma é conhecida como uma Declaração de Atribuição. É importante não confundirmos uma Declaração de Atribuição com uma desigualdade. Diferente da Declaração de Atribuição, as desigualdades não são usadas separadamente, e sim como partes de outras instruções de programação.

A declaração de atribuição instrui o Sharp para armazenar o resultado, obtido através do cálculo da expressão, dentro da localização de memória associada ao nome da variável. Consequentemente, usar o nome da variável (no nosso exemplo, G) é o mesmo que usar o resultado.

Note também que uma variável pode ser usada quantas vezes for necessária, no mesmo cálculo.

Variáveis podem ser usadas de outras maneiras também. O valor de uma variável pode ser atribuido para a outra, como na declaração:

H = G

que copiará o nosso resultado anterior em H. G não é alterado; o mesmo resultado agora está armazenado em duas variáveis diferentes.

Variáveis que retém números podem ser incrementadas ou decrementadas em uma declaração, como neste exemplo:

G = G + 5

Esta instrução fará com que o Sharp chame o valor de G, some 5 a este valor, e armazene de volta o novo valor em G.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Esta capacidade é útil para todos os tipos de cálculos:

O custo de um produto é armazenado na variável X. Assumindo uma taxa de venda de 6.5%, qual é o preço de venda?

```
X = X * 1.065
```

Evidentemente, nós poderíamos ter escrito este exemplo comoP = X * 1.065a fim de deixar X inalterado. Se nós tivessemos armazenado o índice de taxa na variável T, nós poderíamos ter falado X = X * T ou P = X * T.

Até agora, nós usamos letras singulares como os nomes das variáveis. Isto nos fornece 26 variáveis (A à Z). Na realidade, o Sharp permite o uso de acima de 950 nomes de variáveis para números. Um grupo adicional acima de 950 variáveis pode ser usado para armazenar até 16 caracteres cada. Finalmente (como se isso não fosse suficiente), usando técnicas mais avançadas, os usuários podem criar variáveis que retém tantos números ou caracteres quanto desejar, sendo limitado somente pela quantidade de memória disponível no computador.

O esquema de nome para as variáveis é simples e fácil para aprender. Nomes de variáveis numéricos (usadas para reter números) podem ser escolhidas usando as seguintes regras:

- o nome pode ser uma letra; A à Z
- o nome pode ser uma letra, seguida por um dígito (0 à 9) ou por uma outra letra.

Então, os seguintes são nomes válidos para variáveis numéricas:

S, Q1, TX, MM, Z9, RO, E

NOTA: Devido aos conflitos com as abreviações, que tem outros significados na linguagem BASIC, o Sharp não permite o uso desses nomes em variáveis: LF, IF, LN, PI, TO.

Nomes de variáveis de caracteres (usados para reter caracteres) seguem as mesmas regras acima exceto que os nomes terminam com um $ (sinal de cifrão). O $ alerta o Sharp, que a variável retém informações de caracter. Os seguintes são exemplos de nomes de variáveis de caracter válidos:

TS, P2$, T7$, AA$, YR$, X$, ZH$, B5$

NOTA: Devido aos conflitos com palavras que fazem parte da linguagem BASIC, o Sharp não permite o uso destes nomes em variáveis: LF$, IF$, LN$, PI$, TO$.

É importante entender-se que uma variável A e uma variável A$ são duas variáveis diferentes, a primeira pode somente manter um número, e a segunda pode somente manter caracteres. Conforme veremos mais tarde, o Basic do Sharp inclui instruções para converter caracteres em números e números em caracteres.

Para armazenar os caracteres nas variáveis de caracter, nós usamos uma variação da nossa amiga, a Declaração de Atribuição:

```
nome da variável de caracter = "caracteres"
```

Como exemplo, digite o seguinte:

```
D$ = "João Carlos"
```

Agora, chame o conteúdo da D$ digitando [D] [$] [ENTER]

```
João Carlos
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Note que o espaço entre O e C foi armazenado e que "(aspas) não foram. As aspas servem como "delimitadores", um guia para indicar que outros caracteres serão armazenados. Qualquer caracter (incluindo espaços), exceto as aspas, podem ser armazenados. Esta sequência de caracteres fechados pelas aspas é chamada de "cordão de caracteres." Cada variável de caracter pode reter até 16 caracteres. Alguns cuidados devem ser tomados a fim de não exceder este limite ou a informação será perdida. Isto é demonstrado na seguinte atribuição.

F$ = "João come jabuticaba"

Agora, chame à informação: [F] [$] [ENTER]. Que diferença!

Um último lembrete sobre as variáveis: elas tem a memória de um elefante. A informação permanece armazenada numa variável até que:

1) Outra declaração de atribuição seja executada para a mesma variável.
2) Um comando NEW ou CLEAR seja dado.
3) Um programa seja rodado com o uso do comando RUN.
4) As pilhas do computador sejam trocadas.

O valor de uma variável fica retido mesmo quando o computador é desligado! Teste isto desligando-o, e então ligando-o.

No modo RUN, solicite o valor de G (Digite [G] [ENTER]).

Ficou impressionado? Quando você for começar a programar no próximo capítulo, você achará que as variáveis são indispensáveis.

J. PAUSA . . .

Congratulações! Se você persistiu até este ponto, agora você sabe os fundamentos do PC-1500 da SHARP, o suficiente para efetuar as largas variedades de cálculos. O PC-1500 é tão versátil, que cada leitor encontrará muitas utilidades para o seu próprio campo de interesse. Não importa qual é a sua aplicação. Você desejará aprender programação a fim de explorar totalmente, a potência desta maravilhosa máquina.

Neste ponto, você tem uma escolha. Para aqueles cujo o coração está acelerado, cuja a respiração está mais rápida no pensamento da programação (e para qualquer um que ainda não esteja com sono), nós pedimos que você volte e leia o capítulo 0, integrando os seus novos conhecimentos com a informação de lá. Então, prossiga para o capítulo 3.

Outros leitores, que tenham a imediata necessidade de usar tais características como, notação científica e funções trigonométricas, poderão seguir diretamente para o capítulo de Cálculos Avançados.

SUMÁRIO

1) **Modos** – O PC-1500 da SHARP opera em um dos três diferentes modos: RUN, PRO e RESERVE: Cada mudança no modo causa uma leve diferença no funcionamento interno, semelhante à mudança de marcha no carro. O modo RUN é para cálculos, e é o modo no qual você deve "rodar" (executar) programas. No modo PRO, todas as escritas e edições (adições e correções) de programas são efetuadas.
2) **Cálculos** – O PC-1500 da SHARP efetua cálculos aritméticos comuns no modo RUN. A tecla CL (limpeza) deve ser pressionada antes de cada cálculo para limpar os resultados do cálculo anterior. **NÃO** pressionando a tecla CL entre os cálculos, nos permite uma série de cálculos utilizando-se os resultados dos cálculos anteriores.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

3) **Caracteres Especiais** –O $ (sinal de cifrão) e a , (vírgula) tem significados especiais na linguagem BASIC e não podem ser usados como parte de um número em um cálculo.
4) **Números Negativos** – São caracterizados quando temos um – (sinal de menos) precedendo o número, como por exemplo: –5 + 2 = > –3
5) **Cálculos Compostos** – Seguem leis comuns de álgebra.
Parênteses são usados para indicar o significado correto de uma expressão dada. Informações completas sobre a ordem de avaliação das expressões são dadas no apêndice 0.
6) **Desigualdades** – Desigualdades usando os símbolos <, >, < =, > =, e < > (menor que, maior que, menor ou igual, maior ou igual, e diferente) posicionará Ø para FALSO e 1 para verdadeiro.
7) **Tecla de Seta para Esquerda** – A tecla de seta para esquerda ◄ volta o cursor. Mantendo-a pressionada, causará repetição automática.
Pressionando SHIFT seguido pela tecla de Seta para Esquerda invoca a função de Cancelamento, que elimina o caracter onde o cursor está posicionado.
8) **Tecla de Seta para Direita** – A tecla de seta para direita move o cursor para frente. Mantendo-a pressionada, causará repetição automática. A sequência de SHIFT, Seta para Direita, invocará a função de Inserção; inserindo um "Retentor de Lugar" na posição de cursor corrente. Este caracter pode ser então, substituido pela informação a ser inserida.
9) **Função de Chamada** – Após ENTER ser pressionada, e o resultado de um cálculo ser visualizado, a equação original pode ser chamada através do pressionamento tanto da tecla de seta para esquerda como da tecla de seta para direita. Neste ponto, mudanças podem ser feitas e introduzidas novamente na equação modificada. A função de chamada funcionará também para qualquer expressão não programada que produza um erro. Neste caso, o cursor será posicionado no ponto de detecção de erro.
10) **O Uso de Variáveis** – no modo RUN aumenta grandemente a capacidade de computação do PC-1500 e simplifica as expressões complexas.
11) **Declarações de Atribuição** – usando as formas:
   - nome da variável = expressão
   - nome da variável de caracter = "caracteres" permitem o armazenamento de um número ou um cordão de caracteres de até 16 caracteres, respectivamente. Qualquer caracter digitável pode ser usado dentro do cordão de caracteres, exceto as aspas.
12) **Nomes da Variável** – para as variáveis numéricas são:
   1. Uma letra (A à Z)
   2. Uma letra seguida por um dígito (0 à 9) ou outra letra.

   Os nomes da variável de caracter seguem a mesma regra, com a adição de $ (sinal de cifrão), como um sufixo para todos os nomes.
13) **Exceções** – As seguintes são as exceções para o esquema de nome e não podem ser usadas como nomes de variáveis: IF, LF, LN, PI, TO, IF$, LF$, LN$, PI$, TO$.
14) **Vida da Informação** – A informação dentro das variáveis são retidas até:
   1. Ser dado um comando de CLEAR (limpeza) ou NEW (novo).
   2. Um programa ser rodado através do uso de comando RUN.
   3. Outra declaração de atribuição ser executada para a mesma variável.
   4. As pilhas do computador serem trocadas.

   O fato de desligar o computador não afeta os valores armazenados nas variáveis.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

III. A ARTE MISTERIOSA DA PROGRAMAÇÃO

A arte da programação tem sido encoberta pelo véu do mistério desnecessariamente por tanto tempo, que a maioria das pessoas associam-na com a magia ou o gênio matemático. O fato é que não é necessário talento especial para arrancar fora os coelhos do chapéu. Nem é necessário que você seja perito para resolver as equações diferenciais parciais. Suas maiores vantagens serão a sua paciência, sua capacidade de raciocínio lógico, sua atenção para detalhes, e sua ânsia de aprender. A boa vontade para aceitar desafios é também útil.

Programação é uma arte, e como tal, necessita de uma pequena habilidade, um pouco de aprendizagem e bastante prática. Não é nossa intenção fazer de você, com este manual, um programador experimentado. Nós familiarizar-te-emos com as operações básicas e conceitos de programação. Para ser um programador competente é necessário mais, da mesma forma que um bom motorista necessita mais do que o conhecimento de como dirigir e mudar as marchas.

A. O QUE É UM PROGRAMA?

Você deve estar surpreso em descobrir que um programa é somente uma lista de instruções que o computador segue uma de cada vez. Estas instruções devem ser dadas ao computador na linguagem que ele "entende". O PC-1500 da Sharp "fala" um dialeto de BASIC, uma linguagem de programação largamente usada e muito popular. Como outras linguagens, o BASIC tem um vocabulário especial e regras de gramática que são combinadas para formar as declarações. Se você fala com o Sharp "com erros de gramática" ou em vocabulário não familiar, o computador avisa o seu erro. Mas não é difícil instruir o Sharp corretamente. A linguagem BASIC foi desenvolvida originalmente para ensinar os princípios de programação e muitas das suas declarações contém palavras em Inglês e outros símbolos familiares.

B. COMO PROGRAMAR?

Usando o Sharp para programar, você seguirá uma certa rotina. As instruções de um programa são introduzidas no modo PRO. Estas instruções são conhecidas como "declarações" na linguagem BASIC. Para começar a execução destas declarações, é necessário posicionar no modo RUN, e então instruir o Sharp para processar, digitando o comando RUN. Vamos tentar introduzir e rodar um programa:

Posicione no modo PRO, e digite o "comando" BASIC (mais detalhes no item C):

[N] [E] [W] [ENTER]

Este comando "limpará" toda a memória da máquina. Digite a seguinte linha:

```
Listagem do Programa:
10 PRINT "PARABENS!"
```

Digitação de teclas:

[1] [0] [P] [R] [I] [N] [T] [SHIFT] ["] [P] [A] [R] [A] [B] [É] [N] [S]
[SHIFT] [!] [SHIFT] ["] [ENTER]

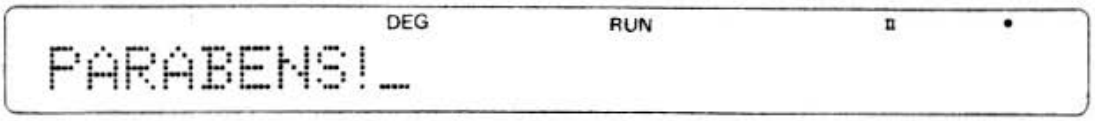
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Nosso programa de uma linha está completo. Mude para o modo RUN e digite:

[R] [U] [N] [ENTER]

Este comando instrui o Sharp a começar o processamento das instruções (ou, neste caso, da instrução) no nosso programa.
Seguindo as ordens, o Sharp mostra no visor:



(Pressione ENTER para informar o Sharp que já pode limpar o visor).

Para fazer qualquer adição, mudanças ou cancelamentos no nosso programa, nós precisamos voltar para o modo PRO.
Caso o nosso programa esteja com um erro, e não foi completado com êxito, será necessário que se volte para o modo PRO para corrigir a instrução incorreta. Então, a operação de programas é efetuada no modo PRO, enquanto a execução e teste de programas são efetuados no modo RUN.

C. COMANDOS VS. DECLARAÇÕES

Você deve ter notado no exemplo anterior que nós nos comunicamos com o Sharp usando dois métodos diferentes. As instruções "NEW" e "RUN" são executadas logo após ao pressionamento do "enter". Este tipo de instrução é conhecida como um "comando". A instrução PRINT, por sua vez, foi um tanto diferente. Ela foi introduzida no modo PRO, precedida por um número (10), e não foi executada imediatamente. Este tipo de instrução é conhecida como uma "declaração".

De certo modo, os comandos informam ao Sharp o que fazer com as declarações. Por exemplo, o comando NEW apagará todas as declarações guardadas. É importante lembrar que os comandos não podem ser usados dentro de um programa, ao passo que as declarações são quase sempre grupadas para formar um programa.

D. NÚMEROS DE LINHA

Um programa BASIC consiste de uma série de linhas numeradas, cada uma contendo uma ou mais declarações.

Os "números da linha" são usados pelo Sharp para manter a seqüênca correta durante a execução. Declarações podem ser introduzidas em qualquer ordem, mas são processadas pelo computador seqüencialmente pelo número da linha (sujeito a modificação, conforme veremos mais tarde).

Como demonstração, vamos acrescentar uma declaração ao nosso programa de uma linha. Posicione no modo PRO e digite:

[5] [P] [R] [I] [N] [T] [SHIFT] ["] [Q] [U] [E] [SPACE]
[SHIFT] ["] [SHIFT] [;] [ENTER]

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Agora, rode o programa revisado. O que acontece
Pressione [ENTER] depois de aparecer QUE.

O Sharp colocou na ordem e executou as linhas 5 e 10 na ordem correta. Mas, para "cutucar" o Sharp na linha 5, para ver o resultado da linha 10, você precisou pressionar ENTER.

Embora o número da linha possa ser qualquer número de 1 a 65.279, nós o aconselhamos a numerar as suas linhas com acréscimo de 10 (isto é, 10, 20, 30... etc).

Isto permite que você insira até 9 declarações entre as suas declarações correntes (11 à 19, por exemplo). Alguns programadores recomendam um intervalo maior regular; numerando de 20 em 20. Entretanto você pode imaginar que com o projeto e escrita de programas cuidadosos, você raramente necessitará inserir declarações. Não conte com isto! É muito mais fácil numerar de 10 em 10 agora, do que renumerar muitas declarações mais tarde.

Lembre que, duas linhas não podem ter o mesmo número. Se esta condição ocorrer, a primeira linha introduzida será perdida. Esta característica pode ser utilizada para cancelar uma linha que não se deseja mais, simplesmente pressionando o número da linha a ser cancelada e então, pressionando [ENTER] Deste modo, uma linha vazia, cancelará efetivamente uma linha existente com o mesmo número.

Caso sejam usados números de linha duplicados involuntariamente, teremos sempre um desarranjo no programa. Para demonstrar isto, introduza a linha seguinte (em modo PRO evidentemente):

Digitações de teclas:

[1] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [H] [O] [R] [R] [I] [V] [E] [L]
[SHIFT] ["] [ENTER]

Agora, rode o programa da mesma maneira que você fez antes. É horrível perder uma linha de programa, não é mesmo? E devido à perda da linha podemos ser levados a alguns erros muito graves. Você deve exercitar-se e tomar cuidado durante a escrita de seus programas.

E. TECLAS DE REVISÃO DA LINHA DE PROGRAMA

Você pode perguntar para você mesmo: "Como eu posso lembrar qual linha que eu introduzi?" Medo não encoraja o programador! Esta necessidade de revisar foi antecipada e providenciada pelas teclas [↑] (seta para cima) e [↓] (seta para baixo). Você pode considerá-las como teclas de Revisão de linhas de programa. Pressionando a tecla apropriada (no modo PRO), você poderá mover as linhas do programa para frente ou para trás. Posicione no modo PRO e use a tecla de seta para baixo para revisar as linhas do nosso programa. Agora, use a tecla de seta para cima para mover de volta para o início do programa. Note que caso qualquer tecla de seta seja mantida pressionada, linhas sucessivas serão visualizadas automaticamente.

(Infelizmente, esta característica não é notada facilmente em um programa de duas linhas).

Uma vez que você alcançou uma dada linha usando as teclas de revisão de linha de programa, você pode então, prosseguir a edição daquela linha usando as teclas de Seta para direita ou esquerda. Note que a operação dessas teclas no modo PRO é idêntica à sua operação no modo RUN (veja o capítulo 2). As funções de Cancelamento e Inserção são também acessíveis para usar na edição de declaração, e são invocadas da mesma forma que antes.

NOTA: Após efetuar qualquer mudança numa linha de programa, você deve pressionar ENTER a fim de confirmar estas mudanças.
Não use as teclas de Setas para cima ou para baixo antes de pressionar [ENTER] . Caso contrário a edição que você efetuou será perdida.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F. UMA VISÃO MINUCIOSA EM ALGUNS VELHOS AMIGOS

Agora que nós sabemos como introduzir, rodar e editar programas, nós ampliaremos o nosso vocabulário de declarações e comandos úteis. Vamos começar examinando alguns velhos amigos: o comando NEW, a declaração LET, e a declaração PRINT.

F.1. O COMANDO NEW

Conforme vimos nos nossos exemplos de programação anteriores, o comando NEW cancela todas as linhas de programa existentes na memória. Nós usaremos o comando NEW (no modo PRO) antes de cada programa para assegurar que as únicas instruções na memória do Sharp são as instruções do nosso programa atual.

Embora seja possível (e muitas vezes desejável) ter vários programas na memória simultaneamente, nós adiaremos o uso desta característica para evitar confusão.

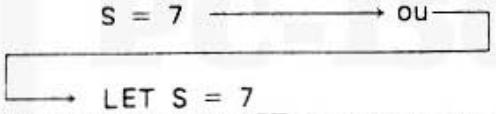
No modo PRO, emita o comando NEW. Que efeito as teclas de Seta para Cima e para Baixo tem agora?

F.2. A DECLARAÇÃO LET

Não fique assustado se você não reconhecer a declaração LET como um "velho amigo"; você já foi apresentado a ela. Realmente, a declaração LET é, somente uma declaração de Atribuição disfarçada. (se você não reconhecer a declaração de atribuição também, por favor releia a parte das variáveis no capítulo 2 imediatamente!).

No início da linguagem BASIC, todas as declarações começavam com uma "palavra-chave" (como PRINT, INPUT,etc...) a qual indicava o que a instrução fazia. LET foi a palavra-chave que identificava o fato de que um valor era para ser armazenado dentro de uma variável. Hoje, é geralmente estabelecido de que a palavra LET realmente não é necessária.

Como resultado, a palavra-chave LET é opcional no BASIC do PC-1500. Então, uma declaração que armazena o número 7 numa variável S pode ser escrita também da seguinte forma.



Uma exceção: Uma posição onde LET deve sempre ser usada, é uma atribuição que ocorre como parte de uma declaração IF. Apesar de nós não termos discutido ainda a declaração IF,é possível uma explicação.

Usando a declaração IF, você pode escrever uma instrução como a seguinte:

IF expressão THEN declaração

Caso a declaração que segue o THEN seja uma declaração de atribuição, a palavra-chave LET deve ser usada. Isto resultaria numa declaração com a seguinte forma:

IF expressão THEN LET nome da variável = expressão

NOTA: Neste caso, a omissão da declaração LET resultaria em ERROR 19 (ou outra mensagem de erro).

F.3. A DECLARAÇÃO PRINT

Na maioria dos programas que você escreverá, as instruções seguirão uma forma básica. Haverá as instruções para leitura de dados que ainda não foram processadas, instruções para processar os dados, e instruções para imprimir ou visualizar os resultados. Esta forma foi referenciada como o ciclo de Entrada, Processamento e Saída. Logo não é surpresa, que a declaração PRINT tenha muitas variações diferentes. O formato geral da declaração PRINT é a palavra PRINT, seguida por um item ou uma lista de itens a serem impressos. Estas incluem, cordões de caracteres, expressões, ou nomes de variáveis, cujos valores serão impressos. Cada item numa lista é separado por uma vírgula ou um ponto e vírgula.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Introduza o seguinte programa para demonstrar a impressão de alguns itens: (lembre-se de executar o comando NEW antes de começar).

```
Listagem de Programa:
10 Z$ = " = NADA"
20 ZZ = 0
30 PRINT "NO PERU  SUSPEITA-SE QUE"
40 PRINT ZZ
50 PRINT Z$
```

Digitação de teclas:

[1] [Ø] [Z] [SHIFT] [$] [=] [SHIFT] ["] [=] [N] [A] [D] [A] [SHIFT]
["] [ENTER]

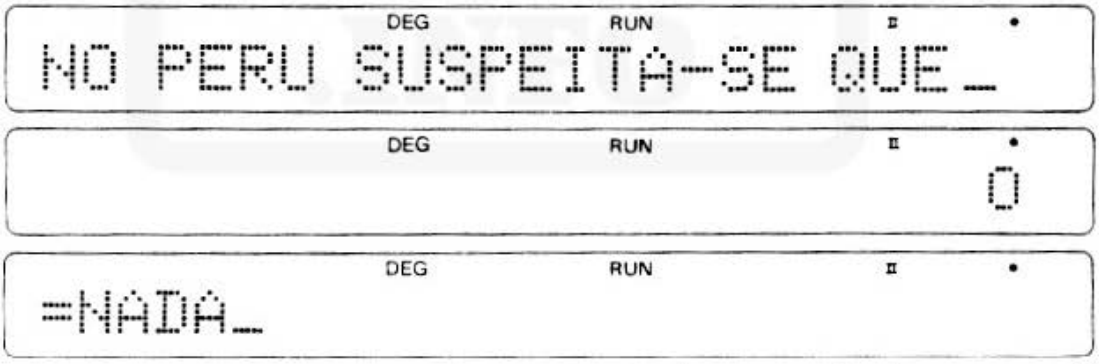
[2] [Ø] [Z] [Z] [=] [0] [ENTER]

[3] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [N] [O] [SPACE] [P] [E] [R] [U]
[SPACE] [S] [U] [S] [P] [E] [I] [T] [A] [-] [S] [E] [SPACE] [Q] [U] [E] [SHIFT] ["] [ENTER]

[4] [Ø] [P] [R] [I] [N] [T] [Z] [Z] [ENTER]

[5] [Ø] [P] [R] [I] [N] [T] [Z] [SHIFT] [$] [ENTER]

Quando rodar, este programa deve produzir três linhas de saída conforme segue (pressione [ENTER] após ler cada linha.):



A primeira das suas instruções PRINT (linha 30) visualiza um cordão de caracteres. Note que o símbolo de aspas não é impresso como parte da saída. Eles são necessários para "delimitar", ou marcar o início e o fim da seqüência de caracteres que você deseja imprimir. Esta seqüência pode incluir qualquer caracter exceto o próprio caracter de aspas.

Os itens na segunda e terceira instruções PRINT (linhas 40 e 50) devem ser reconhecíveis para todos os leitores como variáveis. Quando um nome de variável é usado dentro de uma lista de impressão, o valor da variável é impresso. Neste caso, nós sabemos quais seriam os valores de ZZ e Z$ desde que eles sejam especificados pelas linhas 10 e 20. Imprimindo uma variável "vazia" resultará em um zero ou um espaço dependendo da variável ser numérica ou caracter.

Leitores espertos observarão que os cordões de caracteres e os valores de variáveis de caracter são impressos começando do lado esquerdo do visor. Isto é conhecido como "alinhamento à esquerda". Por outro lado, os números e os valores de variáveis numéricas usam o "alinhamento à direita".

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

É possível também imprimir o resultado de uma expressão contida na declaração PRINT. O seguinte programa de uma linha ilustra isto:

Listagem do Programa:

```
10 PRINT (1982-1956) * 365.25
```

Digitação de teclas:

[1] [Ø] [P] [R] [I] [N] [T] [(] [1] [9] [8] [2] [−] [1] [9] [5] [6] [)]
[*] [3] [6] [5] [.] [2] [5] [ENTER]

Como você pode esperar, o resultado, um número, é alinhado à direita.

Para maior eficiência, é melhor evitar computação de expressões dentro da instrução PRINT a não ser que a instrução (e a sua expressão associada) seja executada somente uma vez no programa.

Devido a maioria dos programas calcularem vários resultados de uma vez, a impressão simultânea de vários itens é uma prática comum.

Talvez a mais simples das instruções PRINT de múltiplos itens é aquela que divide o visor em duas secções. Cada secção contém um ou dois itens especificados na lista de impressão. Os itens são separados na lista por uma vírgula. Explore este formato usando o seguinte programa:

Listagem do Programa:

```
10 A = 2 * PI
20 PRINT "2 vezes PI = ", A
```

Digitação de teclas:

[1] [Ø] [A] [=] [2] [*] [P] [I] [ENTER]
[2] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [2] [SPACE]
[V] [E] [Z] [E] [S] [SPACE] [P] [I] [=]
[SHIFT] ["] [SHIFT] [,] [A] [ENTER]

Execute o programa. Como nas instruções PRINT de um único item, números são alinhados à direita e caracteres são alinhados à esquerda. Neste caso, o alinhamento ocorre dentro de cada uma das duas secções do visor.

Usando as suas habilidades de edição, altere a linha 20 para ler:

```
20 PRINT A, " = 2 vezes PI"
```

A saída desta versão modificada deve ajudar você a identificar as duas secções do visor.

Do not sell this PDF !!!

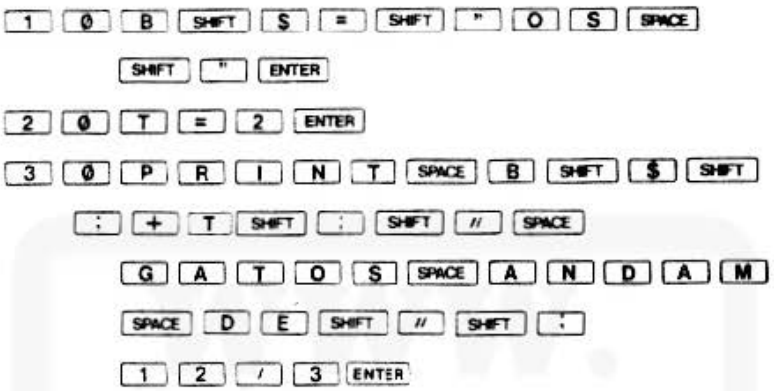
All and more about Sharp PC-1500 at http://www.PC-1500.info

Podem ser visualizados mais que dois itens na mesma linha através de uma variação da instrução PRINT, o qual utiliza o ponto e vírgula. Construa e verifique o programa seguinte:

Listagem do Programa:

```
10 B$=" OS "
20 T=2
30 PRINT B$;+T;"GATOS ANDAM DE"; 12/3
```

Digitação de Teclas:



Execute esta composição eletrônica que deve produzir a seguinte saída:

OS 2 GATOS ANDAM DE 4

Assim ilustramos, a ação da instrução PRINT quando os itens prontos para a impressão são separados por ponto e vírgula.
Neste formato, os itens são visualizados adjacentemente com uma separação mínima entre eles.

Esta capacidade é muito útil para criar saídas "naturais".

NOTA: Caso o comprimento da informação a ser visualizada exceda o espaço disponível no visor (25 caracteres), os itens do fim da linha de impressão não serão mostrados.

Outro local onde usamos ponto e vírgula é bem no fim da instrução PRINT. Nesta situação, o ponto e vírgula indica que a saída, deve ser mantida no visor, e que qualquer nova saída (da próxima instrução PRINT) deve juntar-se à velha saída na mesma linha. Este processo é mais fácil de se programar do que descrever. Vamos então experimentar com o programa a seguir. (não esqueça o comando NEW):

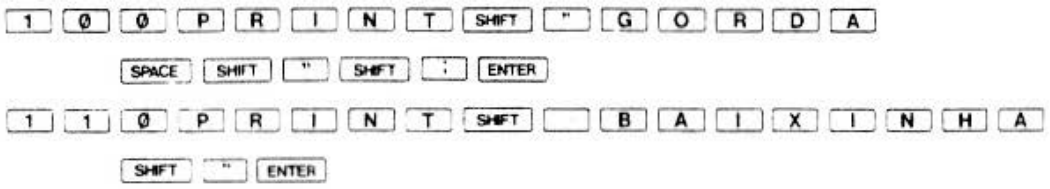
Listagem de Programa:

```
100 PRINT "GORDA";
110 PRINT "BAIXINHA"
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de Teclas:

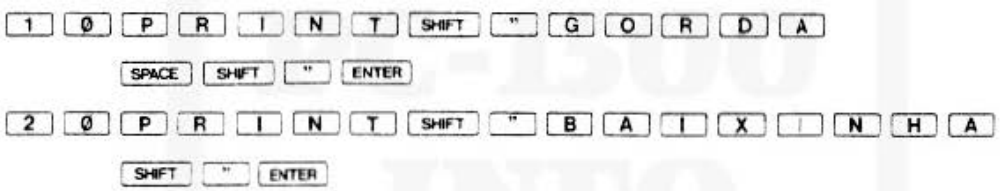


Agora, rode-o. Como de costume, você deve pressionar ENTER após a primeira instrução PRINT ter mostrado "GORDA". Por causa do ponto e vírgula, "GORDA" é retido e "BAIXINHA" compartilha o visor com ele. Ao contrário deste, com a execução do seguinte programa, que não utiliza o ponto e vírgula, a diferença se tornará mais clara.

Listagem do Programa:

```
10 PRINT "GORDA"
20 PRINT "BAIXINHA"
```

Digitação de Teclas:



O próximo exemplo de programa ilustra que o ponto e vírgula pode juntar tantas linhas de saídas assim como encaixar no visor. Impressão de informações no visor em excesso é um erro que o Sharp não avisará. Cabe a você, como um programador, assegurar que isto não acontecerá. Tente o programa educacional seguinte:

Listagem do Programa:

```
20 PRINT "DO";
40 PRINT "RE";
60 PRINT "MI";
80 PRINT "FA";
100 PRINT "SOL";
120 PRINT "LA";
140 PRINT "SI";
160 PRINT "DO"
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de teclas:

2 Ø P R I N T SHIFT " D O SPACE
SHIFT " SHIFT : ENTER

4 Ø P R I N T SHIFT " R E SPACE
SHIFT " SHIFT : ENTER

6 Ø P R I N T SHIFT " M I SPACE
SHIFT " SHIFT : ENTER

8 Ø P R I N T SHIFT " F A SPACE
SHIFT " SHIFT : ENTER

1 Ø Ø P R I N T SHIFT " S O L SPACE
SHIFT " SHIFT : ENTER

1 2 Ø P R I N T SHIFT " L A SPACE
SHIFT " SHIFT : ENTER

1 4 Ø P R I N T SHIFT " S I SPACE
SHIFT " SHIFT : ENTER

1 6 Ø P R I N T SHIFT " D O SHIFT "
ENTER

Agora, rode o programa, pressionando ENTER repetidamente cantando alegremente cada nota na escala!

G. A DECLARAÇÃO PAUSE

A declaração PAUSE é uma forma semi-automática de declaração PRINT. Ela visualiza os vários itens da sua lista associada, por um periodo de tempo breve e fixo. O usuário está então liberado da carga de estimular o Sharp a continuar pressionando ENTER. Imagine a PAUSE como uma declaração PRINT seguido por uma contagem regressiva.

Quando a contagem regressiva acabar, o programa continua.

Os formatos da declaração PAUSE são idênticos àqueles da declaração PRINT. Todas as várias técnicas que nós já discutimos para a declaração PRINT são aplicáveis para a declaração PAUSE embora a saída resultante varie um pouco, evidentemente. Para ilustrar o uso da declaração PAUSE, vamos escrever novamente o nosso programa da escala musical:

Listagem do Programa:

```
10 PAUSE "DO";
20 PAUSE "RE";
30 PAUSE "MI";
40 PAUSE "FA";
50 PAUSE "SOL";
60 PAUSE "LA";
70 PAUSE "SI";
80 PAUSE "DO";
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de Teclas:

1 Ø P A U S E SHIFT " D O SPACE
SHIFT " SHIFT : ENTER

2 Ø P A U S E SHIFT " R E SPACE
SHIFT " SHIFT : ENTER

3 Ø P A U S E SHIFT " M I SPACE
SHIFT " SHIFT : ENTER

4 Ø P A U S E SHIFT " F A SPACE
SHIFT " SHIFT : ENTER

5 Ø P A U S E SHIFT " S O L SPACE
SHIFT " SHIFT : ENTER

6 Ø P A U S E SHIFT " L A SPACE
SHIFT " SHIFT : ENTER

7 Ø P A U S E SHIFT " S I SPACE
SHIFT " SHIFT : ENTER

8 Ø P A U S E SHIFT " D O
SHIFT " ENTER

Note o que acontece após a última "nota" ser impressa; o programa termina e o visor retorna para o "pronto". Isto acontece porque não existe outra declaração seguindo a última PAUSA.

Para "congelar" o visor após a última nota a ser impressa, nós podemos mudar a linha 80 para:

80 PRINT "DO"

Digitação de Teclas:

8 Ø P R I N T SHIFT " D O
SHIFT " ENTER

Após tudo isto, não há razão para nós não podermos misturar as declarações PRINT e PAUSE no nosso programa do começo ao fim. Tente isto você mesmo.

H. ENTRADA

Através da utilização de várias formas da declaração PRINT, sozinha ou combinada, a informação pode ser apresentada agradavelmente ao usuário do computador. A maioria dos itens que serão impressos, entretanto, são o resultado do processamento de algum dado inicial. Este dado inicial é fornecido ao programa pelo usuário do computador. A instrução que controla este processo é a instrução INPUT (entrada).

Na sua forma mais simples. INPUT nome-da-variável, a instrução INPUT simplesmente imprime um ? (ponto de interrogação) e então espera que o usuário digite a informação requisitada. O que é "requisitado" depende da variável que aparece como parte da instrução INPUT. Por exemplo, se a variável é numérica, o usuário deve introduzir um número, que será armazenado na variável.

Como vocês podem notar, a não ser que o usuário do programa seja também um programador (e provavelmente ainda não é), ele não saberá qual tipo de dado introduzir quando ele visualizar o ponto de interrogação. Cabe ao programador manter o usuário informado em todos os momentos.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Como exemplo de um programa falho neste aspecto, introduza o seguinte:

Listagem do Programa:

```
10 A = 0
20 INPUT A
30 PRINT A*PI
```

Digitação de Teclas:

[1] [Ø] [A] [=] [Ø] [ENTER]

[2] [Ø] [I] [N] [P] [U] [T] [A] [ENTER]

[3] [Ø] [P] [R] [I] [N] [T] [A] [*] [P] [I] [ENTER]

Agora, imagine o usuário quando ele roda o programa... A primeira coisa que aparece no visor é o ponto de interrogação. O usuário instruído compreenderá que algum dado é requisitado, mas ele não terá idéia de qual dado deve digitar. Suponhamos que ele tome uma decisão extraordinária, e por sorte, introduza um número. Subitamente, um longo e complicado número aparece. O que ele significa? Ele pressiona ENTER para continuar, mas o programa termina. Do seu ponto de vista, toda a experiência tem sido uma perda de tempo. Por quê? Por causa da programação pobre.

Uma solução para este problema é usar declarações PRINT ou PAUSE para ajudar o usuário a localizar-se. Com esta idéia nós podemos escrever novamente os nossos programas como segue:

Listagem de Programa:

```
10 A = 0
20 PAUSE "INTRODUZA QUALQUER NÚMERO"
30 INPUT A
40 AP = A*PI
50 PRINT A; "VEZES PI = ";AP
```

Digitação de Teclas:

[1] [Ø] [A] [=] [Ø] [ENTER]

[2] [Ø] [P] [A] [U] [S] [E] [SHIFT] ["] [I] [N] [T] [R] [O] [D] [U] [Z] [A] [SPACE]
[Q] [U] [A] [L] [Q] [U] [E] [R] [SPACE] [N] [U] [M] [E] [R] [O] [SHIFT] ["] [ENTER]

[3] [Ø] [I] [N] [P] [U] [T] [A] [ENTER]

[4] [Ø] [A] [P] [=] [A] [*] [P] [I] [ENTER]

[5] [Ø] [P] [R] [I] [N] [T] [A] [SHIFT] [;] [SHIFT] ["] [SPACE]
[V] [E] [Z] [E] [S] [SPACE] [P] [I] [SPACE] [=]
[SPACE] [SHIFT] ["] [SHIFT] [;] [A] [P] [ENTER]

Esta versão é muito mais útil porque ela "prepara" (ou provoca) o usuário para a digitação e identifica as saídas. Esta necessidade (de preparar o usuário para a digitação) é tão

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

importante que formas mais avançadas da declaração INPUT foram criadas. A primeira delas com um ponto e vírgula:

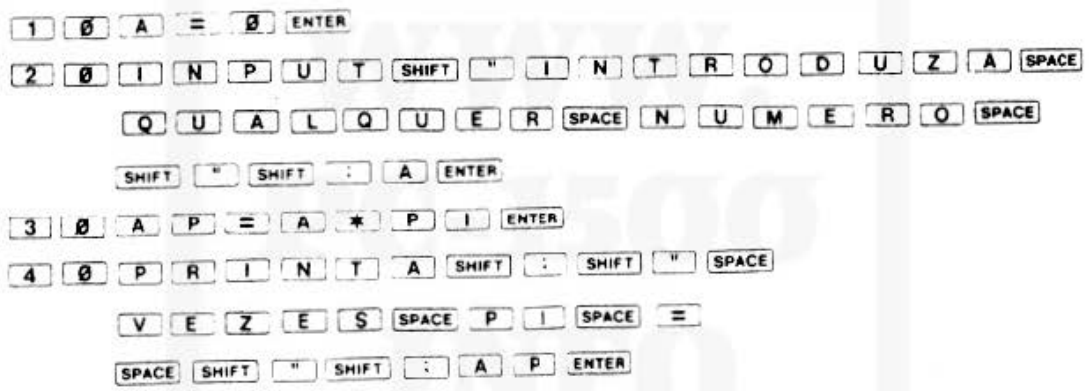
INPUT "caracteres" ; nome da variável

Leitores perspicazes conhecerão nosso velho amigo, o cordão de caracteres. Este cordão de caracteres será impresso no visor como um "pronto". O cursor, substituirá o ponto de interrogação, na mesma linha, e o Sharp esperará pela entrada do usuário. Esta forma de declaração INPUT permite-nos a escrever novamente o nosso exemplo anterior como:

Listagem de Programa:

```
10 A = 0
20 INPUT "INTRODUZA QUALQUER NÚMERO" ; A
30 AP = A*PI
40 PRINT A; "VEZES PI = "; AP
```

Digitação de Teclas:



Assim que você rodar este exemplo, note como o uso da declaração INPUT difere da declaração PAUSE.

A segunda forma de entrada é bastante semelhante a primeira, exceto que ela usa uma vírgula ao invés de um ponto e vírgula:

INPUT "caracteres", nome da variável

Quando esta forma de declaração é executada, o cordão de caracteres associado é visualizado e o computador espera novamente pela entrada. Neste momento, entretanto, assim que o usuário começa a digitar, a mensagem é apagada e o dado do usuário aparece no seu lugar. Isto permite a entrada de dado após uma longa mensagem sem perder o final do visor.

Mude o ponto e vírgula para uma vírgula na linha 20 do nosso programa e reexecute o programa.

Evidentemente, a entrada de dados não é limitada a números, como os nossos exemplos vistos até agora. Para introduzir caracteres, nós simplesmente especificamos uma variável de caracter na declaração INPUT como neste bri hante programa dedutivo:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Listagem de Programa:

```
10 INPUT "SEU ÚLTIMO NOME";L$
20 INPUT "SEU PRIMEIRO NOME";F$
30 I$ = LEFT$ (F$,1)+"."+ LEFT$ (L$,1) + ". "
40 PAUSE "VAMOS LÁ, "
50 PRINT "SUAS INICIAIS";I$
```

[1] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [S] [E] [U] [SPACE]
[U] [L] [T] [I] [M] [O] [SPACE] [N] [O] [M] [E] [SPACE]
[SHIFT] ["] [SHIFT] [;]
[L] [SHIFT] [$] [ENTER]

[2] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [S] [E] [U] [SPACE]
[P] [R] [I] [M] [E] [I] [R] [O] [SPACE] [N] [O] [M] [E] [SPACE]
[SHIFT] ["]
[SHIFT] [;] [F] [SHIFT] [$] [ENTER]

[3] [Ø] [I] [SHIFT] [$] [=] [L] [E] [F] [T] [SHIFT] [$] [(] [F]
[SHIFT] [$] [SHIFT] [,] [1]
[)] [+] [SHIFT] ["] [-] [SHIFT] ["] [+]
[L] [E] [F] [T] [SHIFT] [$] [(] [L] [SHIFT] [$]
[SHIFT] [,] [1] [)] [+] [SHIFT] ["] [-]
[SHIFT] ["] [ENTER]

[4] [Ø] [P] [A] [U] [S] [E] [SHIFT] ["] [V] [A] [M] [O] [S] [SPACE] [L] [Á] [SHIFT] [,]
[SHIFT] ["] [ENTER]

[5] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [S] [U] [A] [S] [SPACE]
[I] [N] [I] [C] [I] [A] [I] [S] [SPACE]
[SHIFT] ["] [SHIFT] [;] [I] [SHIFT] [$] [ENTER]

NOTA: Não se assuste com a linha 30; ela usa técnicas avançadas que você aprenderá mais tarde.

A instrução INPUT pode ser usada também para colher e armazenar vários valores. Para programar um processo desta maneira, simplesmente liste as variáveis que receberão dados na instrução INPUT. Cada variável na lista é separada por uma vírgula.

Como exemplo de uma entrada de item múltiplo, construa este programa estatístico:

Listagem de Programa:

```
10 W = 0 : X = 0 : Y = 0 : Z = 0
20 INPUT "IDADE DE 4 PESSOAS",W,X,Y,Z
30 S = W+X+Y+Z : A = S/4
40 PAUSE "TOTAL DE ANOS = "; S
50 PRINT "IDADE MÉDIA";A
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de teclas:

```
1 Ø W = Ø SHIFT : X = Ø SHIFT : Y = Ø
SHIFT : Z = Ø ENTER
2 Ø I N P U T SHIFT "
I D A D E SPACE D E SPACE 4 SPACE
P E S S O A S SHIFT " SHIFT ,
W SHIFT , X SHIFT , Y SHIFT ,
Z ENTER
3 Ø S = W + X + Y + Z SHIFT : A = S / 4
ENTER
4 Ø P A U S E SHIFT " T O T A L SPACE
D E SPACE A N O S SPACE = SPACE SHIFT "
SHIFT ; S ENTER
5 Ø P R I N T SHIFT " I D A D E SPACE
M E D I A SPACE = SPACE SHIFT "
SHIFT ; A ENTER
```

Quando rodar este programa um tanto indiscreto, solicitará que você introduza 4 números. O primeiro número que você digitar substituirá a mensagem, porque nós usamos uma vírgula diretamente após o texto na declaração INPUT. Quando o primeiro número for digitado completamente você deve pressionar ENTER. O ponto de interrogação precederá todos os números sucessivos, onde cada número deve ser seguido por pressionamento da tecla ENTER.

Se nós tivéssemos usado um ponto e vírgula na declaração INPUT, o primeiro número introduzido não teria substituído a mensagem, mas teria compartilhado a linha junto com ela. Todas as entradas sucessivas teriam tido o mesmo efeito do que com a vírgula. Tente você mesmo alterando a linha 20 para:

```
20 INPUT "IDADE DE 4 PESSOAS";W,X,Y,Z
```

Digitação de teclas:

```
2 Ø I N P U T SHIFT "
I D A D E SPACE D E SPACE 4 SPACE
P E S S O A S SHIFT " SHIFT ,
W SHIFT , X SHIFT , Y SHIFT ,
Z ENTER
```

I. SUGESTÕES ÚTEIS PARA AGILIZAR O TRABALHO

I.1. Abreviações

Aqueles cujo os dedos de ouro não voam velozmente sobre as teclas do computador, já notaram que nossos programas estão cada vez maiores. Os projetistas hábeis do Sharp anteciparam as suas dificuldades. Eles permitiram ao Sharp, reconhecer as abreviações das declarações e comandos freqüentemente.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

A forma geral de uma abreviação é uma ou mais letras designadas, seguidas por um ponto. O ponto é crucial para evitar confusão com nomes variáveis. Como um exemplo, introduza o programa seguinte, cujas declarações são abreviadas:

```
LISTAGEM DO PROGRAMA
15 PA. "OLA, HUMANO."
25 I. "QUAL E O SEU NOME?"; N$
35 P. "PRAZER EM CONHECE-LO!"; N$
```

Digitação de teclas:

[1] [5] [P] [A] [.] [SHIFT] ["] [O] [L] [A] [SHIFT] [,]
[H] [U] [M] [A] [N] [O] [.] [SHIFT] ["] [ENTER]
[2] [5] [I] [.] [SHIFT] ["] [Q] [U] [A] [L] [SPACE] [E] [SPACE]
[O] [SPACE] [S] [E] [U] [SPACE] [N] [O] [M] [E] [SHIFT] [?] [SHIFT] ["]
[SHIFT] [;] [N] [SHIFT] [$] [ENTER]
[3] [5] [P] [.] [SHIFT] ["] [P] [R] [A] [Z] [E] [R] [SPACE] [E] [M] [SPACE]
[C] [O] [N] [H] [E] [C] [E] [R] [SHIFT] [,] [SPACE]
[SHIFT] ["] [SHIFT] [;]
[N] [SHIFT] [$] [ENTER]

Conforme você completar a entrada de cada linha (pressionando a tecla ENTER), notará que o Sharp expande automaticamente a abreviação naquela linha. Este fato é imensamente valioso para você mais tarde, verificar o programa. Um benefício oculto ádicional é que o espaço usado para armazenar as abreviações das declarações não é maior (nem menor) que o espaço usado para armazenar a declaração inteira. Então, a facilidade de abreviação é rigorosamente uma conveniência para você, o programador.

Para aumentar a facilidade de leitura,nós não usaremos estas abreviações na nossa listagem do programa. As seguintes, são as abreviações permitidas para comandos e declarações já apresentadas:

PRINT	P. PR. PRI.
PAUSE	PA. PAU.
IMPUT	I. IN. INP.
RUN	R.

Uma lista completa de abreviações está inclusa no fim deste manual como um Apêndice.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

I.2 Declarações Múltiplas e os Dois Pontos.

Como mencionamos na secção de números de linha, várias declarações podem compartilhar a mesma linha. O Sharp executará as declarações na sequência da esquerda para a direita. Para que o Sharp seja hábil em distinguir o fim de uma declaração e o começo da próxima, deve ser usado um caracter de "sinalização". Este caracter é os dois pontos (:). Pela analogia, a função de dois pontos é similar à função de pontos que separam as sentenças nesta página; ele distinçue quando parar a leitura.

O uso de dois pontos já foi ilustrado várias vezes nos últimos programas. Sua forma geral é a seguinte:

Número da linha declaração 1: declaração 2: declaração 3 (etc)

A questão de quando usar dois pontos é um problema de estilo de programação. Uso indiscriminado de dois pontos para escrever um programa muito compacto produz um programa muito difícil de ler, de corrigir e de expandir. Devido às vantagens primárias do PC-1500 serem pessoais e de ação interativa, é altamente desejável que você esteja habilitado a modificar e extender programas para satisfazer as suas próprias necessidades. Logo, nós recomendamos que você pratique e domine o uso de dois pontos.

Caso você use dois pontos, será bastante vantajoso somente para grupos de declarações que são inter-relacionadas conceitualmente. Isto é, colocar somente aquelas poucas declarações que efetuam uma tarefa única, dentre muitas tarefas possíveis que compreendem o programa, em uma linha. Por exemplo, considere as declarações do programa seguinte: lê-se três números, encontra-se a sua média, computa-se a diferença de cada número da média, e imprime-se a soma dessas diferenças:

Listagem de Programa:

```
10 N1 = 0 : N2 = 0 : N3 = 0
20 INPUT "INTRODUZA 3 NÚMEROS", N1, N2, N3
30 A = (N1 + N2 + N3)/3
40 D1 = N1 - A : D2 = N2 - A : D3 = N3 - A
50 SD = D1 + D2 + D3
60 PRINT "SOMA DAS DIFERENÇAS="; SD
```

Digitação de teclas:

[1] [Ø] [N] [1] [=] [Ø] [SHIFT] [:] [N] [2] [=] [Ø] [SHIFT] [:] [N] [3]
[=] [Ø] [ENTER]

[2] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [I] [N] [T] [R] [O] [D] [U] [Z] [A] [SPACE]
[3] [SPACE] [N] [U] [M] [E] [R] [O] [S] [SHIFT] ["] [SHIFT] [;]
[N] [1] [SHIFT] [;] [N] [2] [SHIFT] [;] [N] [3] [ENTER]

[3] [Ø] [A] [=] [(] [N] [1] [+] [N] [2] [+] [N] [3] [)] [/] [3] [ENTER]

[4] [Ø] [D] [1] [=] [N] [1] [−] [A] [SHIFT] [:] [D] [2] [=] [N] [2] [−] [A]
[SHIFT] [:] [D] [3] [=] [N] [3] [−] [A] [ENTER]

[5] [Ø] [S] [D] [=] [D] [1] [+] [D] [2] [+] [D] [3] [ENTER]

[6] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [S] [O] [M] [A] [SPACE]
[D] [A] [S] [SPACE] [D] [I] [F] [E] [R] [E] [N] [Ç] [A] [S]
[=] [SPACE]
[SHIFT] ["] [SHIFT] [;] [S] [D] [ENTER]

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Note como cada linha efetua um passo completo e necessário no programa. Linha 10 apaga qualquer valor anterior que pode estar contido nas variáveis N1, N2 e N3 (isto é uma precaução no caso do usuário falhar na introdução dos três números). Linha 20 colhe os dados do usuário. Linha 30 calcula a média. Linnha 40 computa as três diferenças. Linha 50 soma as diferenças e a linha 60 imprime o resultado.

Não é casualmente que as instruções de cada linha do programa corresponde à descrição anterior do programa. Este é um dos princípios de boa programação que você deve tentar seguir. Ao contrário do uso das abreviações, o uso dos dois pontos afeta a quantidade de memória usada para armazenar o programa.

Esta é a principal justificação para usar os dois pontos para posicionar as várias declarações em uma linha.

Cada número de linha reserva vários "passos" (uma unidade de armazenamento) do programa e portanto, quanto menor for o número de linhas do programa, menor será o tamanho da memória usada pelo programa.

A afirmação final sobre o uso de dois pontos é que cada programador deve avaliar a legibilidade e mutabilidade do programa contra as necessidades de armazenamento da sua aplicação.

J. CORREÇÃO DE ERRO NO MODO DE PROGRAMA (PRO)

Embora as abreviações e dois pontos ajudem-nos a introduzir programas facilmente, eles não podem prevenir-nos de cometer erros. Mesmo programadores profissionais falham em descobrir os erros quando revisam seus próprios programas. O que isto significa é que mais cedo ou mais tarde você encontrará um erro durante a execução do seu programa (se você já não encontrou). A maioria destes erros são facilmente corrigidos se você simplesmente aceitá-los como um quebra-cabeça para ser solucionado e seguir cuidadosamente a pista do problema.

Várias características do PC-1500 ajudarão você nisto. Ao descobrir uma declaração incorreta, o Sharp vai parar e indicar o problema com uma breve mensagem tal como:

```
                                    RUN
ERROR  1  IN  20
```

Vamos criar um programa com um erro deliberado para ilustrar esta função. Introduza:

Listagem do Programa:

```
25 PAUSE "PC-1500"
50 PRINT "É UMA MÁQUINA ÓTIMA"
```

Digitação de teclas:

[2] [5] [P] [A] [U] [S] [E] [SHIFT] ["] [A] [SPACE]
[P] [C] [-] [1] [5] [0] [0] [SHIFT] ["] [ENTER]
[5] [0] [P] [R] [I] [N] [T] [SHIFT] ["] [É] [SPACE] [U] [M] [A] [SPACE] [M] [Á] [Q] [U] [I] [N] [A]
[SPACE] [Ó] [T] [I] [M] [A] [SHIFT] ["] [ENTER]

Agora, rode o programa. Quando a mensagem de erro aparecer, pressione a tecla-[↑] (seta para cima). Enquanto você estiver pressionando esta tecla, o visor mostrará a linha que tornou o Sharp confuso. A grade pisca-pisca pode dar uma sugestão quanto a natureza do problema.

Para corrigir a má declaração, pressione [CL] para abandonar o programa e chaveie no modo PRO. Pressione a Tecla de Seta para cima, mas não mantenha-a pressionada, e o visor mostrará novamente a linha errada:

Do not sell this PDF !!!

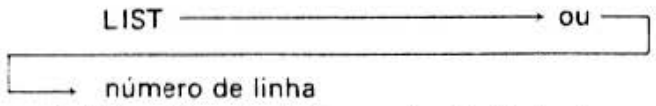
All and more about Sharp PC-1500 at http://www.PC-1500.info

```
50 PRIMT "E UMA MÁQUINA ÓTIMA"
```

Você pode seguir agora a editar a linha usando as teclas familiares INS, DEL e as setas. Quando tiver terminado a edição, você deve pressionar ENTER para instruir o Sharp a armazenar a linha corrigida.

K. O COMANDO LIST

Outra maneira de visualizar uma linha particular de um programa é usar o comando LIST cuja a forma é:



Caso o número de linha não seja dado, a primeira linha do programa será visualizada.

Caso um número de linha seja especificado, a linha do programa com aquele número é visualizada. Caso o programa não tenha o número dado, a próxima linha do programa cujo número é imediatamente maior, é visualizada. Por exemplo, adotando o programa seguinte:

```
15 PRINT "SHARP"
30 PRINT "COMPUTADOR"
45 PRINT "PC-1500"
```

O comando LIST 40 visualizará a linha 45. O comando LIST visualizará a linha 15, e o comando LIST 30 visualizará a linha 30.

NOTA: Se você especificar um número de linha que é maior do que qualquer número de linha de programa existente, ocorrerá "ERROR 11".

L. QUANTO MAIS, MELHOR

Como nós já mencionamos anteriormente, é possível manter na memória mais de um programa ao mesmo tempo. O truque para fazer isto é dar a cada programa, sua própria numeração de linhas. Por exemplo, um programa poderia ser numerado de 300 à 500. Você deve, evidentemente, tomar cuidado para não misturar as declarações dos vários programas acidentalmente (pela nuneração incorreta) ou os resultados imprevisíveis poderão surgir.

L.1. A declaração END

Outro problema que ocorre durante o armazenamento de vários programas simultaneamente é que, cada programa é somente um grupo de declarações numeradas e as declarações são executadas em ordem ascendente. Logo, como o Sharp saberá quando ele termina as declarações da execução de um dado programa? A resposta é que ele não saberá a não ser que você o diga.A declaração usada para assinalar ao computador que foi alcançado o final de programa é a Declaração END.

Até agora nós não usamos, nem precisamos, da Declaração END. O Sharp tem executado simplesmente todas as linhas do nosso programa em ordem ascendente até ele terminar as declarações.

Quando ele acaba, o computador decide que o programa foi encerrado e retorna para esperar pelo nosso próximo comando. Agora, entretanto, nós devemos informar ao Shap que ele deve parar a execução das instruções antes de continuar no próximo programa. Para ilustrar o uso de programas múltiplos, introduza as linhas seguintes:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
10 PAUSE "BAIXINHA"
20 PAUSE "TEVE UM ESPLENDIDO OUTONO"
30 PRINT "MAS NAO UMA BOA PRIMAVERA"
40 END
200 PAUSE "O VELHO CÁLICE"
210 PRINT "FOI ENCHIDO COM VINHO"
220 END
```

L.2. RUN com número de linha especificada

Bem, agora que você tem dois programas na memória, como você fará para rodar cada um deles separadamente?

Se você é ousado o suficiente para ter experimentado o comando RUN normal, você terá descoberto que ele é perfeitamente adequado para iniciar o primeiro programa. (Se você não é ousado, tente isto agora). Para começar o segundo programa, nós necessitamos usar uma variação do comando RUN que informe ao Sharp em qual linnha iniciar. Isto é o comando RUN com número de linha especificada. Para iniciar o segundo programa digite:

[R] [U] [N] [2] [Ø] [Ø]

e pronto!

Como todos os comandos, o comando RUN com número de linha especificada pode causar problemas. Ele instrui o Sharp onde iniciar as instruções, e como o Sharp é um criado tão prestativo, programas podem ser iniciados no meio.

Como você pode supor, isto não deve ser feito propositalmente. Se isto for feito, produzirá alguns resultados excepcionais. Tente-o no nosso primeiro programa emitindo o comando:

```
RUN 30
```

Este é um erro suave comparado ao que aconteceria num programa mais complexo.

M. DECLARAÇÕES DE CONTROLE

Cs nossos programas até aqui tem sido uma sequência consecutiva de instruções efetuadas uma vez no computador. Aqueles com experiência em instruções, sabem que esta não é a melhor maneira de efetuar uma tarefa. Muitas vezes, você permite ao seu ouvinte, escolher um dos vários planos. Às vezes, você "condensa" as suas instruções incluindo um comando como, "Agora repita os últimos 03 passos até você terminar".

Estas capacidades foram incorporadas nas declarações no BASIC e chamadas de declarações de controle. Estas determinam quando, e se, quantas vezes outras declarações serão executadas. Declarações de controle permitem que você construa programas muito poderosos e versáteis. Nas próximas secções discutiremos as declarações de controle do BASIC; a IF . . . THEN, a GOTO, e a GOSUB.

N. IF e THEN

A capacidade de decisão dentro de um programa BASIC é fornecida pela declaração IF. Um programa equipado com uma declaração IF, pode avaliar a sua entrada e tomar decisões.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Listagem do Programa:

```
10 PAUSE "VOCÊ ESTA COM SONO?"
20 INPUT "DIGITE SIM OU NÃO"; SX$
30 IF SX$ = "SIM" THEN PRINT "OH, DESCULPE
O INCOMODO"
40 END
```

[1] [0] [P] [A] [U] [S] [E] [SHIFT] ["] [V] [O] [C] [E] [SPACE]
[E] [S] [T] [A] [SPACE] [C] [O] [M] [SPACE] [S] [O] [N] [O] [SHIFT] [?]
[SHIFT] ["] [ENTER]

[2] [0] [I] [N] [P] [U] [T] [SHIFT] ["] [D] [I] [G] [I] [T] [E] [SPACE]
[S] [I] [M] [SPACE] [O] [U] [SPACE] [N] [A] [O] [SPACE]
[SHIFT] ["] [SHIFT] [:] [S] [X] [SHIFT] [$] [ENTER]

[3] [0] [I] [F] [S] [X] [SHIFT] [$] [=] [SHIFT] ["] [S] [I] [M] [SHIFT] ["]
[T] [H] [E] [N] [P] [R] [I] [N] [T] [SHIFT] ["] [O] [H]
[SHIFT] [,] [SPACE] [D] [E] [S] [C] [U] [L] [P] [E] [SPACE] [O]
[SPACE] [I] [N] [C] [O] [M] [O] [D] [O] [SPACE]
[SHIFT] ["] [ENTER]

[4] [0] [E] [N] [D] [ENTER]

Este soporífico programa somente produzirá saída se você responder afirmativamente. A forma geral da declaração IF, que a linha 30 ilustra, é:

IF condição THEN declaração

Durante a execução, o teste encaixado na condição IF é efetuado. Se a declaração é efetuada ou não, depende do resultado do teste. O teste é geralmente uma desigualdade e é chamado de "condição". Lembre que as desigualdades são comparações que são também Verdadeiras ou Falsas. Se a desigualdade for verdadeira, então a declaração é executada. Se a desigualdade for falsa, então a declaração é omitida.

No nosso programa de exemplo, o teste é se a variável SX$ é igual à variável alfanumérica de caracter "SIM". Se ela for, e somente se ela for, então a instrução PRINT é executada. Se SX$ não for igual à "SIM", então a instrução PRINT é ignorada. Em qualquer caso, seja a declaração PRINT executada ou ignorada, o Sharp seguirá, normalmente, para a próxima linha. (No nosso programa de exemplo a próxima linha é a linha 40).

Note que nós poderíamos ter invertido o nosso teste alterando várias linhas da seguinte maneira:

```
30 IF SX$ = "NÃO" THEN END
40 PRINT "DESCULPE O INCOMODO"
```

Entretanto, este programa não é perfeitamente igual ao original. Este permite que o nosso computador fale com alguém que digitou incorretamente ou quem não responder com um "NÃO". Este programa realmente tem dois pontos finais; a declaração END na linha 30 e o END implícito depois da linha 40.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Vários fins em um mesmo programa não resultam da prática de boa programação. Nossa inversão destas declarações demonstram entretanto, que a distribuição correta de declarações e testes são necessários para um programa operar corretamente. Na próxima secção, nós obervaremos uma terceira maneira de expressar o nosso programa que involve a declaração GOTO e soluciona os problemas da segunda versão.

Embora a condição de uma declaração IF seja geralmente uma desigualdade, ela não o é necessariamente. No PC-1500 BASIC qualquer expressão que resulta em um número positivo, diferente de zero, é considerada como Verdadeira. Isto explica porque as desigualdades funcionam como condições: lembre que uma desigualdade que é verdadeira retorna um 1 e as desigualdades que são falsas retornam um 0. Este método de representação Verdadeira e Falsa não é padronizada em todos os computadores e pode produzir programas obscuros. Use esta facilidade sensatamente.

Mais um lembrete: se a declaração que segue o THEN for uma declaração de atribuição, a palavra-chave LET DEVE ser usada. Caso contrário uma condição de ERRO ocorrerá. Isto foi discutido na secção da declaração LET.

IMPORTANTE – veja a página 29.

0. GOTO

Você deve ter notado que, na última secção, nossas opções foram limitadas após o teste feito com a declaração IF. Nós só podemos utilizar uma declaração se o seu resultado do teste for verdadeiro. Por conveniência, nós gostaríamos de executar várias declarações. A declaração GOTO nos permite fazer isto.

A declaração GOTO modifica o "fluxo" da execução de declarações. Ela diz ao Sharp "go to" (vá para) uma linha diferente da próxima e comece a execução das declarações sequencialmente a partir daí. O efeito deste "salto" é que algumas declarações podem ser omitidas inteiramente. Por exemplo, vejamos o programa seguinte:

Listagem de Programa:

```
10 PAUSE "SHARP";
20 GOTO 50
30 PRINT X * 3 / 4 + 2
40 PRINT "COMPUTADOR DE BOLSO";
50 PRINT "PC-1500"
60 END
```

Digitação de Teclas:

[1] [Ø] [P] [A] [U] [S] [E] [SHIFT] ["] [S] [H] [A] [R] [P]
[SPACE] [SHIFT] [*] [SHIFT] [;] [ENTER]

[2] [Ø] [G] [O] [T] [O] [5] [Ø] [ENTER]

[3] [Ø] [P] [R] [I] [N] [T] [X] [*] [3] [/] [4] [+] [2] [ENTER]

[4] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [C] [O] [M] [P] [U] [T] [A] [D] [O] [R]
[SPACE] [D] [E] [SPACE] [B] [O] [L] [S] [O]
[SHIFT] [*] [SHIFT] [;] [ENTER]

[5] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [P] [C] [-] [1] [5] [O] [O]
[SHIFT] ["] [ENTER]

[6] [Ø] [E] [N] [D] [ENTER]

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Evidentemente, este programa seria executado normalmente na sequência de números de linha ascendente. O efeito do GOTO na linha 20, é fazer com que o Sharp siga imediatamente para a linha 50 e comece na sequência ascendente a partir daí. As linhas 30 e 40 nunca serão executadas.

A forma geral da declaração GOTO é:

GOTO expressão

onde:

Expressão deve ser um número que é um número de linha do programa válido (isto é, entre 1 e 65279).

NOTA: Especificando um número de linha que não existe, resultará em um ERROR 11.

Caso nós desejemos ter várias instruções executadas como resultado de um certo teste, nós usaremos a declaração GOTO juntamente com IF.

```
10   IF teste THEN GOTO 100
20   | Estas declarações são executadas |
 :   | somente se o teste for falso     |
 :
90   GOTO 200
100  | Estas declarações são executadas |
 :   | somente se o teste for verdadeiro.|
 :
200  | Estas declarações são sempre     |
 :   | executadas                       |
 :
900  END
```

A lógica deste programa é aplicável para muitas situações. Qualquer número de declarações pode ser inserido em cada uma das secções que são formadas pelas declarações GOTO.

Como um exemplo desta estrutura, na prática, o próximo segmento de um programa de livro caixa determina se um dado valor de entrada é um depósito (número positivo) ou uma retirada (número negativo). Um balanço inicial é introduzido pelo usuário:

Listagem de Programa:

```
10 INPUT "BALANÇO INICIAL", B
20 INPUT "VALOR DA TRANSAÇÃO?",TA
25 IF TA = 0 THEN GOTO 90
30 B = B + TA
40 IF TA < 0 THEN GOTO 70
50 PRINT "DEPOSITO DE $";TA
60 GOTO 80
70 PRINT TA; "RETIRADA DE DINHEIRO"
80 PRINT "BALANÇO FINAL = ";B
90 END 85 GOTO 20
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação de Teclas:

```
1 Ø I N P U T SHIFT " B A L A N Ç O
    SPACE I N I C I A L
    SHIFT ? SHIFT " SHIFT ; B ENTER
2 Ø I N P U T SHIFT " V A L O R
    SPACE D A SPACE T R A N S A Ç Ã O SHIFT ?
    SHIFT " SHIFT ; T A ENTER
2 5 I F T A = Ø T H E N G O T O
    9 Ø ENTER
3 Ø B = B + T A ENTER
4 Ø I F T A SHIFT < Ø T H E N
    G O T O 7 Ø ENTER
5 Ø P R I N T SHIFT " D E P Ó S I T O
    SPACE D E SPACE SHIFT $
    SHIFT " SHIFT : T A
6 Ø G O T O 8 Ø ENTER
7 Ø P R I N T T A SHIFT ; SHIFT "
    SPACE R E T I R A D A SPACE D E
    SPACE D I N H E I R O SHIFT " ENTER
8 Ø P R I N T SHIFT " B A L A N Ç O
    SPACE F I N A L = SPACE
    SHIFT " SHIFT ; B ENTER
8 5 G O T O 2 Ø ENTER
9 Ø E N D ENTER
```

Observe que este programa termina somente se o valor de transação for zero. A segunda declaração IF é que ilustra a estrutura que nós mencionamos antes. Note que além das duas ações separadas associadas com a declaração IF (linhas 50 e 70), existem algumas declarações (linhas 80 e 90) executadas independentemente do resultado do teste do IF.

Nós podemos escrever agora, a terceira versão de um programa a partir da seção anterior.

Listagem de Programa:

```
10 PAUSE "VOCÊ ESTÁ COM SONO?"
20 INPUT "DIGITE SIM OU NÃO";SX$
30 IF SX$ < > "SIM" THEN GOTO 99
40 PRINT "OH, DESCULPE O INCÔMODO"
99 END
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Digitação das Teclas:

```
1 Ø P A U S E SHIFT " V O C E SPACE
  E S T A SPACE C O M SPACE S O N O
  SHIFT ? SHIFT " ENTER
2 Ø I N P U T SHIFT " D I G I T E SPACE
  S I M SPACE O U SPACE N A O SPACE
  SHIFT " SHIFT : S X SHIFT $ ENTER
3 Ø I F S X SHIFT $ SHIFT < SHIFT >
  SHIFT " S I M SHIFT " T H E N
  G O T O 9 9 ENTER
4 Ø P R I N T SHIFT " O H SHIFT 1
  SPACE D E S C U L P E SPACE O SPACE
  I N C O M O D O SHIFT "
  ENTER
9 9 E N D ENTER
```

Esta versão inverte o teste condicional através da especificação de algumas ações a serem efetuadas se a entrada do usuário NÃO for igual a "SIM". Então ele elimina os problemas da segunda versão, e pode ser expandido na forma maior da declaração IF descrita acima. Reorganizar as declarações neste estilo é muitas vezes uma manobra útil durante a programação. Tome algum tempo para experimentar você mesmo e você será recompensado com programas melhores.

Outro uso muito comum da declaração GOTO é causar a execução repetida de uma sequência de declarações. Este processo é conhecido como "looping". Um simples exemplo de looping é ilustrado por este programa.

Listagem de Programa:

```
10 WAIT 30
20 PRINT "CHUG CHUG CHUG CHUG"
30 PRINT "        CHUG CHUG CHUG"
40 GOTO 20
```

Digitação de Teclas:

```
1 Ø W A I T 3 Ø ENTER
2 Ø P R I N T SHIFT " C H U G SPACE
  C H U G SPACE C H U G SPACE C H
  U G SHIFT " ENTER
3 Ø P R I N T SHIFT " SPACE SPACE SPACE
  SPACE SPACE C H U G SPACE C H U G
  SPACE C H U G SPACE C H U G SHIFT "
  ENTER
4 Ø G O T O 2 Ø ENTER
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Infelizmente, este programa manterá somente "chugdando" incessantemente. (Você pode parar o programa usando a tecla BREAK).

Nós devemos fornecer uma maneira para tais programas chegarem ao fim. Nós podemos fazer isto usando um "contador" acompanhando a declaração IF. Um contador é uma variável em que nós mantemos o registro de quantas vezes nós fizemos alguma coisa (isto é, nós contamos com ele). Com esta técnica, a declaração IF, testará se nós efetuamos as nossas declarações PRINT (após 10 vezes, isto satura).

Usando o nosso contador e a declaração IF, nós podemos escrever:

Listagem de Programa

```
10 WAIT 30
15 C = 1
20 PRINT "CHUG CHUG CHUG CHUG"
30 PRINT "CHUG CHUG CHUG CHUG"
40 C = C + 1
50 IF C < = 10 THEN 20
60 END
```

Digitação de Teclas:

[1] [Ø] [W] [A] [I] [T] [3] [Ø] [ENTER]

[1] [5] [C] [=] [1] [ENTER]

[2] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [C] [H] [U] [G] [SPACE]
[C] [H] [U] [G] [SPACE] [C] [H] [U] [G] [SPACE] [C] [H]
[U] [G] [SHIFT] ["] [ENTER]

[3] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [SPACE] [SPACE] [SPACE]
[SPACE] [SPACE] [C] [H] [U] [G] [SPACE] [C] [H] [U] [G]
[SPACE] [C] [H] [U] [G] [SPACE] [C] [H] [U] [G] [SHIFT] ["]
[ENTER]

[4] [Ø] [C] [=] [C] [+] [1] [ENTER]

[5] [Ø] [I] [F] [C] [SHIFT] [<] [=] [1] [Ø] [T] [H] [E] [N]
[2] [Ø] [ENTER]

[6] [Ø] [E] [N] [D] [ENTER]

Siga a operação do contador à medida que cada "loop" é executado. Note que na linha 15 nós devemos atribuir ao contador um valor inicial. Na linha 40 nós acrescentamos este valor para refletir mais uma execução das declarações 20 e 30.

Existem muitas maneiras de usar contadores e "loops" dentro dos programas. Infelizmente, nós não temos espaço para descrevê-los aqui.

A instrução GOTO é um comando e também uma declaração. Como um comando, o seu uso é naturalmente diferente do seu uso como uma declaração do programa. Emitido como um comando, no modo RUN, GOTO começa a execução de programa de maneira similar ao

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

do comando RUN. As diferenças estão em certas preparações internas que são feitas antes da execução do programa. (Para uma comparação de métodos usados para começar um programa, veja a secção entitulada de Execução de Programa Inicial no capítulo V).

Para começar a execução do programa com o comando GOTO digite:

GOTO número da linha

onde:

número da linha é o número da primeira linha do programa a ser executada.

NOTA: Especificando um número de linha que não existe, resultará em um "ERROR 11".

P. FOR e NEXT

Como nós vimos na secção anterior, a habilidade de repetir uma série de instruções é muito útil. De fato, esta característica é muito usada, e o BASIC incorpora várias declarações para automatizar o processo. Esta é a declaração FOR e sua companheira, a declaração NEXT. Juntas, as declarações FOR e NEXT incluem uma série de instruções que são repetidas um número de vezez. A declaração FOR tem uma variável contadora associada e uma condição de teste embutida.

Ele permite também a especificação do valor inicial e o valor de acréscimo da variável contadora. A forma para todas estas informações é:

FOR variável contadora = valor inicial TO
valor final
STEP valor de acréscimo

onde:

variável contadora é o nome da variável usada para efetuar a contagem do loop.

valor inicial é o valor armazenado na variável contadora antes de entrar no "loop". A capacidade permitida para este valor é − 32768 até 32767.

valor final é o número que é usado no teste. Se a variável contadora contém um valor maior do que o valor final, o "looping" termina. A capacidade permitida para este número é − 32768 até 32767.

STEP valor de acréscimo é uma cláusula opcional. O valor de acréscimo indica quanto aumentar ou diminuir a variável contadora cada vez que se completa um "loop". Este deve ser um inteiro dentro do limite entre − 32768 e 32767. Se a cláusula inteira for omitida, então o valor de acréscimo é assumido como sendo 1.

Isto é muito trabalhoso, então vamos observar o procedimento de um programa de exemplo simples. O primeiro programa é similar à versão do programa de CHUG CHUG o qual usou um contador. Ao invés de "chugar" nós imprimimos o valor da variável contadora C:

```
15 FOR C = 1 TO 10
30 PAUSE C
50 NEXT C
```

(Para fazer este programa "Chug" como antes, simplesmente insira as declarações 10, 20 e 30 daquele programa). Note que esta versão é mais ordenada e mais resumida; usando menos ceclarações para efetuar as mesmas funções de contagem e "looping" do que na versão anterior.

Se existe ainda alguma confusão sobre o que as declarações FOR e NEXT fazem, então nós apresentamos uma comparação de um "loop" de FOR . . . NEXT com as declarações equivalentes:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

10 FOR I = 1 TO 8	10 I = 1
	12 IF I > 8 THEN 100
20 [algumas declarações : a serem repetidas]	20 [algumas declarações a serem repetidas]
90 NEXT I	90 I = I + 1
100 END	92 GOTO 12 100 END

Nosso segundo exemplo de programa ilustra como podemos programar um "loop geral", cujas repetições são controladas pelo valor de uma variável. Nós começamos perguntando ao usuário, quantos números ele planeja introduzir. Nós armazenamos este número em uma variável N e seguimos para o "loop" através das declarações subseqüentes. O "locp" é executado N vezes, a não ser que N seja menor ou igual a 0 (zero), neste caso, nós terminaremos o processamento.

Listagem de Programa

```
10 N = 0 : V = 0 : T = 0 : A = 0
20 WAIT 0
30 INPUT "QUANTOS VALORES?"; N
40 IF N = 0 THEN GOTO 999
50 FOR I = 1 TO N
60 CLS : CURSOR 0
70 INPUT V
80 T = T + V
90 NEXT I
100 WAIT : CLS : CURSOR 0
110 A = T/N
120 PAUSE "TOTAL ="; T
130 PRINT "MÉDIA ="; A
999 END
```

Digitação de Teclas:

[1] [Ø] [N] [=] [Ø] [SHIFT] [:] [V] [=] [Ø] [SHIFT] [:] [T] [=] [Ø]
[SHIFT] [:] [A] [=] [Ø] [ENTER]

[2] [Ø] [W] [A] [I] [T] [Ø] [ENTER]

[3] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [Q] [U] [A] [N] [T] [O] [S]
[SPACE] [V] [A] [L] [O] [R] [E] [S] [SHIFT] [?] [SPACE]
[SHIFT] ["] [SHIFT] [:] [N] [ENTER]

[4] [Ø] [I] [F] [N] [=] [Ø] [T] [H] [E] [N] [G] [O] [T] [O]
[9] [9] [9] [ENTER]

[5] [Ø] [F] [O] [R] [I] [=] [1] [T] [O] [N] [ENTER]

[6] [Ø] [C] [L] [S] [SHIFT] [:] [C] [U] [R] [S] [O] [R] [Ø] [ENTER]

[7] [Ø] [I] [N] [P] [U] [T] [V] [ENTER]

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

8 Ø T = T + V ENTER

9 Ø N E X T I ENTER

1 Ø Ø W A I T SHIFT : C L S SHIFT :

C U R S O R Ø ENTER

1 1 Ø A = T / N ENTER

1 2 Ø P A U S E SHIFT " T O T A L

SPACE = SPACE SHIFT " SHIFT ;

T ENTER

1 3 Ø P R I N T SHIFT " M E D I A

SPACE = SPACE SHIFT " SHIFT ;

A ENTER

9 9 9 E N D ENTER

Com o "loop" FOR....NEXT não é necessário aumentar sempre a variável contadora, e nem inicializá-la com valor 1. Utilizando a cláusula STEP, o programador pode especificar o tamanho do aumento (ou diminuição). Nosso próximo exemplo demonstra isto:

```
Listagem de Programa
10 WAIT 30
20 FOR HS = 2 TO 6 STEP 2
30 PRINT HS; "!";
40 NEXT HS
50 WAIT 80: CLS: CURSOR 0
60 PRINT " DO PC-1500"
70 PRINT "EU SOU FREGUES!"
80 END
```

Digitação de Teclas:

1 Ø W A I T 3 Ø ENTER

2 Ø F O R H S = 2 T O 6 S T E P

2 ENTER

3 Ø P R I N T H S SHIFT : SHIFT "

SHIFT ! SPACE SHIFT " SHIFT ;

ENTER

4 Ø N E X T H S ENTER

5 Ø W A I T 6 Ø SHIFT : C L S SHIFT :

C U R S O R Ø ENTER

6 Ø P R I N T SHIFT " D O SPACE P C - 1 5 Ø Ø

SHIFT " ENTER

7 Ø P R I N T SHIFT " E U SPACE

S O U SPACE F R E G U E S SHIFT ! SHIFT " ENTER

8 Ø E N D ENTER

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Complementando, a cláusula STEP permite também ao SHARP efetuar a contagem regressiva. Isto é feito através da inversão dos valores iniciais e finais e especificando um acréscimo negativo. O programa seguinte, (dedicado para os consumidores de toda a parte), ilustra isto:

O Empório de Carpetes está oferecendo tapetes redondos à preços extraordinariamente baixos, de Cr$ 99,00 por metro quadrado. A variação dos raios dos tapetes vão desde 40 metros (para o hall do hotel) até 1 metro (para banheiro). Entre cada tapete e o próximo tamanho menor existe uma diferença de três metros no raio. Sr. João decide fixar preço para cada tapete usando um programa no seu computador Sharp. Ele escreve o seguinte programa.

```
10 FOR R = 40 TO 1 STEP -3
20 P = .99 * (PI * R ^ 2)
30 PRINT R; "M (RAIO) = $";P
40 NEXT R
50 END
```

e descobre que os tapetes variam de preço de Cr$ 497.628,27 (para tapete do hall) à Cr$ 311,01 (para banheiro).

Q. WAIT

A declaração WAIT permite o programador mudar a operação da declaração de impressão. Informação visualizada por uma declaração PRINT permanecerá no visor por um período de tempo especificado por uma declaração WAIT.
O formato para a declaração WAIT é:

WAIT argumento

O argumento é opcional. Se nenhum argumento for especificado, o período de tempo corresponde a "infinito"; isto é, a informação permanecerá no visor até o usuário pressionar ENTER. Este é o modo de operação usado na maioria dos nossos programas até este ponto.

Se um argumento for dado, todas as declarações PRINT subseqüentes "segurarão" suas informações no visor por um período de tempo proporcional ao número especificado como argumento. Este tipo de impressão é similar à declaração PAUSE, exceto que o período de tempo da declaração PAUSE é fixo. Note que a declaração WAIT não tem efeito sobre a operação da declaração PAUSE.

O número (ou expressão que resulta em um número) dado como um argumento deve estar dentro do limite de 0 à 65535. WAIT 0 faz a informação ser visualizada tão rapidamente que ela é virtualmente impossível de se ler. WAIT 65535 fará com que cada declaração PRINT visualize suas informações cerca de 17 minutos! Mais praticamente, WAIT 64 dá um período de aproximadamente um segundo, e WAIT 3840 cerca de um minuto. Para reposicionar a operação da declaração PRINT, de maneira que ele espere até o usuário pressionar ENTER, use a instrução WAIT sem argumento.

Programa de Demonstração:

Este programa ilustra o efeito da declaração WAIT sobre declarações PRINT subseqüentes. Aqui nós variamos o tempo WAIT de 0 à 52 através de aumentos de 2 enquanto imprimimos em um loop. O BEEP é exclusivamente para ajudar você na compreensão do intervalo de tempo, devido a algumas das ações acontecerem muito rápidas para serem vistas.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Listagem de Programa:

```
10 FOR W = 0 TO 52 STEP 2
20 WAIT W
30 BEEP 1,5
40 PRINT ".";
50 NEXT W
60 END
```

Digitação de Teclas

1 Ø F O R W = Ø T O 5 2 S T E P
2 ENTER
2 Ø W A I T W ENTER
3 Ø B E E P 1 SHIFT , 5 ENTER
4 Ø P R I N T SHIFT " . SHIFT "
SHIFT : ENTER
5 Ø N E X T W ENTER
6 Ø E N D ENTER

R. READ, DATA e RESTORE

Nem todos os dados dentro de um programa devem ser introduzidos pelo usuário do programa. Muitas vezes, dados úteis são relativamente estáticos, tal como tabelas de taxa dentro de uma aplicação financeira ou constantes de pressão numa aplicação de engenharia.

Estes tipos de informação podem ser colocados dentro de um programa, e utilizados quando necessário, através do uso de declarações DATA, READ e RESTORE. Estas declarações atuam de modo a especificar os dados usados pelo programa, para transferir os dados para variáveis, e repetir o processo conforme a necessidade.

A declaração DATA consiste da palavra chave DATA seguido por uma lista de itens de dados. Estes incluem números, em notação real ou científica, e cordões de caracteres.

Os itens nas listas são separados por vírgulas. Declarações de DATA podem aparecer em qualquer parte dentro de um programa, mas muitos programadores preferem agrupá-los no início do programa. Isto os permite encontrá-los com maior facilidade ao ler o programa.

Uma declaração DATA típica terá o seguinte formato:

```
10 DATA "MOBY DICK", 20000, "BRANCO", "M", 112
```

A declaração READ consiste em uma palavra chave READ seguida de uma lista de nomes de variáveis. Estes podem ser nomes de variáveis numéricas ou caracteres. Os nomes de variáveis dentro de uma lista são separados por vírgulas. A declaração READ faz com que um item, ou itens de dados sejam "lidos" da declaração DATA e armazenados nas variáveis associadas. Uma declaração READ que corresponde à nossa declaração DATA anterior é:

```
120 READ N$, WT, C$, SX$, L
```

O Sharp exige que toda vez que uma declaração READ seja executada, exista um item de dados correspondente dentro de uma declaração DATA. Então, o programa seguinte produzirá um erro na linha 30 porque todos os itens de dados foram "consumidos" pela declaração READ na linha 20:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
10 DATA 1, 2, 3
20 READ A, B, C
30 READ D
```

Para corrigir esta situação, nós podemos acrescentar um item de dado na linha 10:

```
10 DATA 1, 2, 3, 65
```

ou nós podemos usar uma declaração DATA, em qualquer parte do programa:

```
10 DATA 1, 2, 3
20 READ A, B, C
30 READ D
40 DATA 65
```

Isto ilustra que o Sharp considera todas as declarações DATA no programa como uma lista singular de itens de dados. Como o computador encontra cada nome da variável dentro de uma declaração READ, ele fornece o próximo item de dados da lista para aquela variável. Se o Sharp não pode cumprir uma solicitação para um item de dado, ele pára o programa e assinala um erro. Itens extras, que não são usados quando o programa termina de maneira normal, são ignorados.

Se o tipo (caracter ou numérico) do próximo item não bater com o tipo da variável a ser preenchido, ocorre um erro.

Bons programadores agrupam itens de dados em declarações DATA separados, onde cada um corresponde à sua declaração READ dentro do programa. Isto é ilustrado no programa seguinte que lê 3 itens de dados 4 vezes:

```
10 DATA 1, "A", 1
20 DATA 2, "B", 3
30 DATA 5, "C", 8
40 DATA 13, "D", 21
50 FOR I = 1 TO 3
60 READ A, A$, Z
70 T = T - A*Z
80 NEXT I
```

As linhas de 10 a 40 poderiam ter sido escritas assim:

```
10 DATA 1, "A", 1, 2, "B", 3, 5, "C", 8, 13, "D", 21
```

ou mesmo assim:

```
10 DATA 1
20 DATA "A"
30 DATA 1
40 DATA 2
 :
(etc)
```

Na realidade, ambas as formas alternativas escondem o fato que 3 itens de dados são lidos por vez pela declaração READ. As formas alternativas fazem também com que fique

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

muito difícil de verificar os tipos dos itens de dados, devido a configuração de: número, caracter, número, não ser perceptível imediatamente.

Às vezes é desejável que seja usado novamente algum ou todos os valores nas declarações DATA. A declaração RESTORE permite-nos fazer isto. RESTORE faz com que o Sharp use itens de dados novamente começando da primeira declaração DATA do programa. Então, qualquer declaração READ efetuada após RESTORE receberá itens de dados que já foram usados anteriomente.

A declaração RESTORE pode também especificar uma repassagem por ciclo mais seletivo de itens de dados. A declaração:

RESTORE número da linha

fará com que itens de dados seiam redistribuídos, começando da declaração DATA na linha especificada. Por exemplo:

```
10 DATA .8, 4
15 DATA 1, 3, 6, 8E2
100 READ X, Y
110 READ M, N, O, P
120 RESTORE 15
130 READ A, B
```

Quando este programa terminar, os valores das variáveis A e B serão 1 e 3, respectivamente.

Acrescentando aos números de linha, a declaração DATA pode ser "rotulada" com um simples caracter. A declaração RESTORE pode então ser usada para reemitir itens de dados começando na declaração DATA com o rótulo dado. Um exemplo de uma declaração DATA rotulada é:

```
20 "A": DATA 1, -12.2, 8
```

O trecho do programa a seguir, faz com que a declaração DATA rotulada de "X" seja usada novamente.

```
10 DATA 4.2, 3, 1
20 "X": DATA -2, 0, 3,5
100 READ Q, Y, Z
110 READ MA, MB, MC, MD
120 RESTORE "X"
130 READ N, Z
```

No final do programa, N contém − 2 e Z contém 0.

S. REM (REMARK)

A declaração REM fornece capacidade para inserir comentários entre as declarações de um programa. Embora, estes comentários sejam ignorados pelo Sharp, eles são extremamente importantes porque ajudam outras pessoas a ler o seu programa. O seu programa será mais facilmente lido e entendido e outros programadores poderão usá-lo e, talvez, aperfeiçoá-lo.

Para reduzir a carga de digitação, nós temos omitido comentários neste manual. Nós recomendamos que você não siga o nosso exemplo. Comentário é, provavelmente, o mais importante quando você estiver imprimindo, guardando ou distribuindo os seus programas. Não confie na sua capacidade de lembrar qual é o significado de cada variável em um programa. Use um comentário! Se você não utilizar, alguns meses depois você terá esquecido.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comentários, seguem a palavra chave REM e podem ser inseridos nas suas próprias linhas ou no fim de outra declaração ou série de declarações. Comentários não podem aparecer antes ou no meio de declarações executáveis. A razão para isto é que quando o Sharp vê a palavra chave REM, ele ignora todos os demais caracteres da linha. Portanto, se REM for utilizado no começo da linha, declarações válidas de programa seriam ignoradas.

T. GOSUB...RETURN

Assim que você começar a criar programas, perceberá que utiliza certas funções repetidamente dentro de um único programa. Por exemplo, tais funções podem incluir cálculos da área de um círculo ou receber e verificar um número dado pelo usuário. Tal repetição causa duplicação de declarações que efetuam a mesma função. Para evitar esta desperdiciosa duplicação, o programador pode usar a declaração GOSUB.

A declaração GOSUB permite que um grupo de declarações utilizadas em vários locais do programa, sejam colocadas a parte. Este grupo de declarações é chamado de "sub-rotina". Em cada local do programa onde o grupo de declarações ocorreria, uma instrução GOSUB é inserida.

A declaração GOSUB instrui o Sharp a iniciar a execução do grupo de declarações que fora colocado de lado. Este processo é conhecido como "chamando uma sub-rotina". Devido a declaração GOSUB causar uma mudança no curso seqüencial normal da execução, ela é similar a declaração GOTO. A diferença , entretanto, é que antes do Sharp efetuar as declarações de sub-rotina ele "lembra" onde ela está. Quando o computador terminar de efetuar a sub-rotina, ele retorna para o ponto de onde ele saiu. Isto é conhecido como "retorno" de uma sub-rotina.

Mas como o Sharp percebe o fim das declarações que formam a sub-rotina? A resposta é que você deve informá-lo com a declaração RETURN. A forma da declaração GOSUB é:

GOSUB número da linha

onde o número da linha é o número da primeira linha da sub-rotina. A forma da declaração RETURN é simplesmente:

RETURN

Como um exemplo de sub-rotina, considere um programa para calcular e comparar a área de dois retângulos, dados o comprimento e a largura.

Listagem de Programa:

```
10 REM LER COMPRIMENTO E
20 REM LARGURA DE DOIS RETÂNGULOS
30 FOR I = 1 TO 2
40 PAUSE "RETÂNGULO";I; ":"
50 INPUT "INTRODUZA COMP. e LARG", L, W
60 GOSUB 200
70 IF I = 1 LET A1 = A
80 IF I = 2 LET A2 = A
90 NEXT I
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
100 REM IMPRIMA E COMPARE AS
110 REM AREAS DOS RETÂNGULOS
120 PRINT "AREA DO RETÂNGULO 1"; A1
130 PRINT "AREA DO RETÂNGULO 2"; A2
140 IF A1 > A2 THEN 170
150 PRINT "AREA DO 2 E > AREA 1"
160 GOSUB 180
170 PRINT "AREA DO 1 E > AREA 2"
180 END
200 REM SUBROTINA P/ CALCULAR A AREA
210 REM DO RETÂNGULO DANDO COMPRIMENTO
220 REM DOS LADOS EM L E W
230 A = L*W
240 RETURN
```

Note onde a sub-rotina está localizada. Todas as sub-rotinas devem ser colocadas **após** a declaração final END do programa principal. Isto evita seu uso acidental quando em processo de execução normal. Uma declaração RETURN deve concluir cada e todas as sub-rotinas. Em uma sub-rotina pode-se incluir qualquer declaração legal e pode-se efetuar qualquer tipo de processamento desejado. Bons programadores de BASIC constroem os seus programas como um conjunto de peças, ou "módulos". Geralmente uma sub-rotina é usada para codificar cada módulo. O programa principal é então usado para controlar a ordem em que as sub-rotinas são executadas.

SUMÁRIO

1) Linhas de programa devem ser numeradas em intervalos de pelo menos dez. Isto permitirá que uma linha adicional seja inserida entre duas linhas já existentes, simplesmente escolhendo um número de linha que está entre os números já existentes. Por exemplo, podemos inserir entre as linhas 40 e 50 uma nova linha com número de 41 a 49.

2) Para cancelar uma linha existente, simplesmente digite o número da linha e pressione ENTER.

   NOTA: Devido a possibilidade de várias declarações de programa serem agrupadas numa mesma linha, maiores cuidados devem ser tomados ao cancelar uma linha.

3) As teclas de Seta para cima e Seta para baixo, devem ser usadas para rolar (mover para cima e para baixo) o programa uma linha por vez.
   Mantendo a tecla de Seta pressionada, causará repetição automática do movimento.

4) O comando LIST deve ser usado para seguir diretamente para uma linha dada. O comanco LIST visualizará a primeira linha do primeiro programa na memória. LIST N (onde N é um número), visualizará a linha N ou, se não existir nenhuma linha numerada com N, a primeira linha com número imediatamente superior a este será visualizada.

5) Com a linha sendo visualizada, as teclas de seta para esquerda e direita podem ser usadas para posicionar o cursor em qualquer parte da linha. As funções INS (inserção) e DEL (cancelamento) podem então ser usadas para efetuar as mudanças.

   NOTA: Se mudanças forem feitas em uma linha, será necessário pressionar ENTER antes de visualizar a próxima linha ou estas mudanças não serão consideradas. (não serão efetuadas).

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

6) Se um erro for encontrado em um programa durante a sua execução, a tecla de seta para cima "lembrará" a linha na qual o erro ocorreu. Para chamar a linha, posicione no modo PRO e pressione a tecla de Seta para cima.

IV. CALCULOS AVANÇADOS

A. NOTAÇÃO CIENTÍFICA

Para introduzir um número em notação científica (Ax10^B), introduza a mantissa, pressione a letra E e introduza o expoente.

Exemplo 1: Para digitar 6.7 x 10^8:

Digitação de Teclas	Visor
[6] [.] [7]	6. 7_
[E]	6. 7E_
[8]	6. 7E8_

Exemplo 2: Para digitar −9.12 x 10^−34:

Digitação de Teclas	Visor
[−] [9] [.] [1] [2]	−9. 12_
[E]	−9. 12E_
[−] [3] [4]	−9. 12E−34_

Somente os 10 primeiros dígitos da mantissa são significativos (veja o exemplo). Para um número menor que 1, mas maior que −1, o dado é exato para um máximo de 10 dígitos.

Exemplo 3: Digite 1234567898765:

Digitação de Teclas	Visor
[1] [2] [3] [4] [5] [6] [7] [8]	
[9] [8] [7] [6] [5]	1234567898765_
[ENTER]	1. 234567898E 12

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Exemplo 4: Digite 9.87654321234:

Digitação de Teclas	Visor
[9] [.] [8] [7] [6] [5] [4]	
[3] [2] [1] [2] [3] [4]	9. 87654321234_
[ENTER]	9. 876543212

Exemplo 5: Digite 0.0000000002345678:

Digitação de Teclas	Visor
[.] [0] [0] [0] [0] [0]	
[0] [0] [0] [0] [2] [3]	
[4] [5] [6] [7] [8]	.000000002345678_
[ENTER]	2. 345678E-10

Exemplo 6: Digite 0.00001234567 x10 ^ 24:

Digitação de Teclas	Visor
[.] [0] [0] [0] [0] [1] [2]	
[3] [4] [5] [6] [7] [E] [2] [4]	.00001234567E24_
[ENTER]	1. 234567E 19

Note que para expoentes, somente os últimos dois dígitos digitados são efetivos.
Exemplo 7: Digite 3 x 10 ^ 123:

Digitação de Teclas	Visor
[3] [E] [1] [2] [3]	3E123_
[ENTER]	3E 23

Exemplo 8: Digite 4 x 10 ^ – 3210:

Digitação de Teclas	Visor
[4] [E] [–] [3] [2] [1] [0]	4E-3210_
[ENTER]	4E-10

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

B. CAPACIDADE DE CÁLCULOS

A maioria das máquinas possuem um limite de números com os quais elas podem operar. No PC-1500 este limite é qualquer número entre 9.999999999 x 10^ – 99. Quando um número exceder este limite, ele se tornará muito grande para o computador manejar, ocorrendo uma condição de "sobrecapacidade" (designado de erro 37). Existe também uma condição de "subcapacidade"; o ponto no qual um número se torna muito pequeno. Uma condição de subcapacidade não será designada; nem ocorrerá mensagem de erro ou parada. Qualquer número dentro da faixa de -1×10^{-99} a 1×10^{-99} será considerado como zero. Isto é ilustrado pela seguinte tabela:

		-1×10^{-99}	1×10^{-99}		
$-9.999999999 \times 10^{99}$		0		$9.999999999 \times 10^{99}$	
ERRO	FAIXA COMPUTACIONAL	CONSIDERADO COMO ZERO	FAIXA COMPUTACIONAL		ERRO

Exemplo 1: Se você tentar solucionar a equação $(5.67 \times 10^{55}) * (8.90 \times 10^{65})$, você causará uma sobrecapacidade:

Digitação de Teclas	Visor
5 . 6 7 E 5 5 ×	
8 . 9 0 E 6 5	(5. 67E55) * (8. 90E65)_
ENTER	ERROR 37

Erro 37 indica sobrecapacidade de cálculo:

C. RAIZ, POTÊNCIA, Pi

Raiz

Exemplo 1: Para encontrar a raiz quadrada de 73:

Digitação de Teclas	Visor
SHIFT √ 7 3	√73_
ENTER	8. 544003745

Exemplo 2: Para encontrar $\sqrt[4]{256}$:

Digitação de Teclas	Visor
SHIFT √ SHIFT √ 2 5 6	√√256_
ENTER	4

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Exemplo 3: Para encontrar $\sqrt{3\,\hat{}\,2+4\,\hat{}\,2}$:

Digitação de Teclas	Visor
SHIFT √ (3 * 3 +	
4 * 4)	√(3*3+4*4)_
ENTER	5

Esta equação pode também ser computada da seguinte maneira:

Exemplo 4

Digitação de Teclas	visor
SHIFT √ (3 SHIFT ^ 2 + 4	√(3^2+4^2)_
SHIFT ^ 2)	
ENTER	5

Potência:

A função de potenciação ou exponenciação permite elevar um número a uma potência.

Exemplo 1: Calcule 4 ^ 3 (= 4 x 4 x 4):

Digitação de Teclas	Visor
4 SHIFT ^ 3	4^3_
ENTER	64

Exemplo 2: Calcule 3 ^ 3.2 x 4 ^ −2.4:

Digitação de Teclas	Visor
3 SHIFT ^ 3 . 2 *	
4 SHIFT ^ − 2 . 4	3^3.2*4^-2.4_
ENTER	1.207380162

Exemplo 3: Calcule 4 ^ (3 ^ 2):

Digitação de Teclas	Visor
4 SHIFT ^ 3 SHIFT	
^ 2	4^3^2
ENTER	262144

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PI [π]

O valor de Pi(3.141592654) é armazenado nos símbolos PI e [π] como uma constante fixa. Um ou outro símbolo poderá ser utilizado em cálculos onde o valor de Pi for necessário.

Exemplo: Encontrar a área de um tapete, com diâmetro de 5 pés, em modo RUN:

Digitação de Teclas	Visor
[P] [I] [*] [(] [5] [/] [2] [)]	
[SHIFT] [^] [2]	PI* (5/2) ^2_
[ENTER]	19. 63495408

D. MODOS ANGULARES

O PC-1500 permite que as funções angulares sejam calculadas em qualquer um dos 3 modos angulares, como segue:

Para posicionar o PC-1500 em modo GRAU, digite:

DEG. [ENTER] (DEG aparecerá no topo do visor).

Para posicionar o PC-1500 em modo RADIANO, digite:

RAD. [ENTER] (RAD aparecerá no topo do visor)

Para posicionar o PC-1500 em modo GRADO, digite:

GRA. [ENTER] (GRAD aparecerá no topo do visor).

E. FUNÇÕES TRIGONOMÉTRICAS

As seis funções trigonométricas fornecidas no PC-1500 são SIN, COS, TAN, ASN, ACS e ATN. Cada função poderá ser calculada tanto no modo DEG como no modo GRAD ou RAD. A execução é conforme segue:

DEG. [ENTER]		Posiciona modo para grau.
SIN 30 [ENTER]	0. 5	SIN 30 em grau
[CL]		Limpa o visor
GRA. [ENTER]		Posiciona modo para grado.
SIN 30 [ENTER]	4. 539904997E-01	SIN 30 em grado.
[CL]		Limpa o visor.
RAD. [ENTER]		Posiciona modo para radiano.
SIN 30 [ENTER]	-9. 880316241E-01	SIN 30 em radiano.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Nos exemplos acima, o seno de 30 é calculado em cada modo, produzindo três diferentes (mas equivalentes) respostas. Funções trigonométricas inversas também podem ser calculadas:

RAD. ENTER	→	Posiciona o modo
ASN −0.5 ENTER	−5. 235987756E−01	Arco seno de −.5
DEG. ENTER	→	Posiciona o modo
ASN −0.5 ENTER	−30	Arco seno de −.5
ACS (−.5 + .1) ENTER	113. 5781785	Arco cosseno
ATN 2.3 ENTER	66. 50143432	Arco tangente

F. FUNÇÕES LOGARÍTMICAS

LN, LOG

A função LN calculará o logarítmo natural (base e) enquanto a função LOG calculará o logarítmo comum (base 10). Estas são executadas no modo RUN conforme segue:

LN 7.4 ENTER	2. 00148
LOG 7.4 ENTER	8. 692317197E−01
LN 25 ENTER	3. 218875825
LOG 100 ENTER	2

EXP

A funçaõ inversa do LOG é um número elevado a uma potência de 10. Por exemplo:

LOG 100 ENTER	2
10 SHIFT ^ 2 ENTER	100

Devido ao log natural (LN) não ser baseado em uma potência de 10, mas em uma potência de e, é necessário a função inversa. Esta função chama-se EXP.

Exemplo:

LN 7.4 ENTER	2. 00148
EXP 2.00148 ENTER	7. 39999998

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

G. CONVERSÃO ANGULAR

O PC-1500 efetua conversões de ângulos da forma DMS (Grau, Minutos, Segundos) para DEG (Grau Decimal). Quando convertemos Grau Decimal para Grau, Minutos, Segundos equivalentes, a resposta é compreendida de uma parte inteira representando graus e uma parte fraccional na qual o primeiro e o segundo dígitos representam os segundos. O quinto até o último digito são graus decimais. Para converter um ângulo dado em grau, minutos segundos para a forma de grau decimal, deve ser introduzido na ordem inteiro, decimal.

Exemplo 1: Converter 16.1932 graus decimais na forma DMS:

DMS 16.1932 ENTER	16. 113552

Exemplo 2: Converter 32.2513 DMS para a forma de grau decimal:

DEG 32.2513 ENTER	32. 42027778

H. FUNÇÕES DIVERSAS

ABS

A função ABS obtém o valor absoluto de um valor numérico ou variável.

Exemplo 1:

ABS (25 – 86) ENTER	61

Normalmente, 25 – 86 = –61. A função ABS usa a diferença real dos números para obter 61.

INT

A função INT arredonda um valor numérico para o maior número inteiro, não maior que o valor numérico dele mesmo.

(25/3) + 7 ENTER	15. 33333333
INT (25/3) + 7 ENTER	15
(31.62 + 21.18) ENTER	52. 8
INT (31.62 + 21.18) ENTER	52

No exemplo 2, a resposta não é arredondada de 52,8 para 53 porque isto faria com que a resposta ficasse maior do que o valor original.

NOTA: Não esqueça a ordem de avaliação. Devem ser usados os parênteses se você deseja o INT do resultado da expressão. Por exemplo, vamos alterar o exemplo anterior, deixando de usar os parênteses:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Exemplo 3:

INT 31.62 + 21.18 ENTER	52. 18

O computador toma o INT do primeiro número 31.62 (resutado: 31) e acrescenta isto para 21.18 para obter 52.18.

SGN

Para qualquer número X, a função SGN retorna para o valor indicado, se o número for negativo, zero ou positivo. Os valores são os seguintes:

1 se X > 0
0 se X = 0
−1 se X < 0

Exemplos: 5 − 10 ENTER	−5
SGN (5 − 10) ENTER	−1
12 − 4 ENTER	8
SGN (12 − 4) ENTER	1
15 − 15 ENTER	0
SGN (15 − 15) ENTER	0

V. PROGRAMAÇÃO AVANÇADA

A. Matrizes e a Declaração DIM

Até agora, a maioria dos exemplos dos nossos programas tem usado um pequeno número de variáveis. Quando você começar a utilizar todo o potencial de processamento do PC-1500, descobrirá que as variáveis que mantém um único número podem ter desvantagens. Talvez, por exemplo, você está considerando um programa que lê 50 números e os classifica. Você pode rapidamente concluir que, embora o PC-1500 tenha mais potencial de variáveis do que o suficiente, deve existir uma maneira mais fácil. Existe, e ele é chamado de uma "variável de matriz".

Uma matriz é simplesmente um grupo de áreas de armazenamento consecutivo, ou "locações", com um único nome. Cada área de armazenamento pode manter um único número ou manter um cordão de caracteres. Todas as áreas de armazenamento dentro de uma matriz devem reter o mesmo tipo de dados.

O número de locações dentro de uma única matriz pode ser de até 256 e é determinado pela sua especificação.

Portanto, se você define uma matriz numérica com 50 locações, isto permite armazenar até 50 números em associação com um único nome. Se você define uma matriz de cordões (chamada de "matriz de caracter") você pode também especificar o tamanho dos cordões,

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

até no máximo 80 caracteres por cordão. A forma de criar os cordões de caracteres de vários tamanhos está descrita na secção B.1.

Para definir uma matriz, é usada a declaração DIM (abreviação de dimensão). As matrizes devem sempre ser "declaradas" (definidas) antes de serem usadas. (não como as variáveis de valor único que nós temos usado até aqui). A forma para a declaração DIMensão numérica é:

DIM nome-variável-numérica (tamanho)

onde:

nome-variável-numérica é um nome variável que obedece às regras normais dos nomes variáveis numéricos discutidos anteriormente.

tamanho é o número de locações de armazenamento e deve ser um número de 0 à 255. Observe que, quando especifica um número para o "tamanho", você obtém mais uma localização além daquela que você especificou.

Exemplos de declaração DIM numérica são:

```
DIM X (5)
DIM AA (24)
DIM Q5 (0)
```

A primeira declaração cria uma matriz X com 6 locações de armazenamento. A segunda declaração cria uma matriz AA com 25 locações. A terceira declaração cria uma matriz com uma locação e é realmente bastante simples (pelo menos para os números), e é o mesmo que declarar uma variável numérica de valor único.

É importante saber que uma variável matricial X e uma variável X estão separadas e são distintas para o PC-1500. A primeira variável X denota uma série de locações de armazenamento numérico, e a segunda, uma única e diferente locação.

Agora que você sabe como criar matrizes, você pode estar querendo saber como é que nós nos referimos a cada locação do armazenamento. Como o grupo inteiro tem um só nome, a forma com a qual nós nos referimos a uma única locação (chamada de "elemento") é seguindo o nome do grupo, com um número entre parênteses. Este número é chamado de um "subscrito"

Então, por exemplo, para armazenar o número 8 no 5.º elemento da nossa matriz X (declarada anteriormente), nós escreveríamos:

```
X (4) = 8
```

Se o uso do 4 estiver embaraçoso, lembre-se que a numeração dos elementos começa do zero e continua até o número de tamanho especificado na declaração DIM. A utilidade real da matriz consiste na habilidade de utilizar uma expressão ou um nome de variável como um subscrito. Por exemplo, para criar uma tabela contendo os quadrados dos números de 0 a 9, nós podemos escrever a seguinte declaração:

```
10 DIM SQ (9)
20 FOR I = 0 T09
30 SQ (I) = I * I
40 NEXT I
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Neste exemplo, a variável I é usada para selecionar qual locação de armazenamento manterá o resultado e é usada também para computar o resultado.

Para declarar uma matriz de caracteres, se utiliza uma forma um pouco diferente da declaração DIM:

DIM nome da variável de caracter (tamanho) * comprimento
onde:

nome da variável de caracter é um nome de variável que obedece as regras para caracteres variáveis normais conforme discutimos anteriormente.

Tamanho é o número de locações de armazenamento e deve estar entre 0 e 255. Note que, quando você especifica um número, você obtém mais uma locação além daquela que você especificou.

* comprimento é opcional. Caso for usado, ele especifica o comprimento de cada um dos cordões que compreendem a matriz. Comprimento é um número de 1 à 80. Caso esta cláusula não for usada, os cordões terão o comprimento de "default" (omissão) de 16 caracteres.

Exemplos de declarações de matriz de caracteres são:

```
DIM X$ (4)
DIM NM$ (10) * 10
DIM IN$ (1) * 80
DIM R$ (0) * 26
```

O primeiro exemplo cria uma matriz de cinco cordões, cada uma capaz de armazenar 16 caracteres. A segunda declaração DIM declara uma matriz NM com 11 cordões de 10 caracteres cada. A definição explícita de cordões menores do que "default" ajuda a conservar o espaço de memória. O terceiro exemplo declara uma matriz de dois elementos de cordões de 80 caracteres e o último exemplo declara um único cordão de 26 caracteres (veja a secção B.1.)

Ao lado da matriz simples que já temos estudado, o PC-1500 permite matrizes "bidimensionais". Por analogia, uma matriz de uma dimensão é uma lista de dados colocados em uma coluna única. Uma matriz bidimensional é uma tabela de dados com linhas e colunas.

A matriz bidimensional é definida pela declaração:

DIM nome variável numérica (linhas, colunas)
ou
DIM nome variável caracter (linhas, colunas) * comprimento.

onde:

linhas especificam o número de linhas na matriz. Este deve ser um número entre 0 e 255. Observe que quando você especifica o número de linhas, você obtém mais uma linha além da especificada.

colunas especificam o número de colunas na matriz. Este deve ser um número entre 0 e 255. Observe que quando você especifica o número de colunas, você obtém mais uma além do especificado.

O diagrama abaixo ilustra as locações de armazenamento que resulta da declaração DIM T (2,3) e os subscritos (agora compostos de dois números) que pertencem a cada locação de armazenamento.

	Coluna 1	Coluna 2	Coluna 3	Coluna 4
linha 0	T (0, 0)	T (0, 1)	T (0, 2)	T (0, 3)
linha 1	T (1, 0)	T (1, 1)	T (1, 2)	T (1, 3)
linha 2	T (2, 0)	T (2, 1)	T (2, 2)	T (2, 3)

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

NOTA: As matrizes bidimensionais podem consumir espaço de armazenamento rapidamente. Por exemplo, matriz com 25 linhas e 35 colunas usa 875 localizações de armazenamento!

As matrizes são ferramentas de programação muito poderosas. Para um tratamento mais completo de matrizes, nós recomendamos que sejam feitas leituras suplementares.

- A tabela seguinte mostra o número de bytes usado para definir cada variável, e o número usado por cada declaração de programa.

VARIÁVEL	NOME DE VARIÁVEL	DADOS	
Variável numérica	7 bytes	8 bytes	
Variável alfanumérica	7 bytes	Arranjo variável	Número de bytes especificado*
		2 caracteres por variável	16 bytes

- Por exemplo, se é especificado DIMZS (2,3) * 10, são reservados 12 variáveis, cada um capaz de enfileirar 10 caracteres.
  Para isto são necessários 7 bytes (nome variável) + 10 bytes (número de caracteres) × 12 = 127 bytes.

Elemento	Número de linha	Declaração & Função	Outros, ENTER
Número de bytes usado	3 bytes	2 bytes	1 byte

B. ALGO MAIS SOBRE CORDÕES DE CARACTERES

B.1. Dimensionando os cordões de caracteres.

Os cordões de caracteres são limitados, através da ausência de indicação, para 16 caracteres de comprimento.

Pelo dimensionamento de um cordão de caracteres é possível criar um cordão com o comprimento de até 80 caracteres. São possíveis reduções no comprimento do cordão para conservar espaço de memória.

O comprimento de um cordão é especificado na declaração DIM conforme segue:

DIM nome-variável (limitado) * comprimento

onde:

nome da variável é o nome da matriz de cordão de caracteres.

limite é o máximo "subscrito" da matriz.

comprimento é o tamanho de cada cordão dentro da matriz.

Se for necessário somente um cordão, uma matriz com um elemento (subscrito como elemento zero) poderá ser especificado para conservar o espaço da memória. A seguir damos um exemplo de declaração de um cordão de caracteres de 26:

```
DIM A$ (0) * 26
```

B.2. Concatenação

Diversos cordões de caracteres (ou caracteres dentro das variáveis de caracteres) podem ser unidas para formar um cordão único. Esta "adição" de cordões de caracteres é chamada de "concatenação". A forma para a concatenação é:

cordão de caracteres / variável de caracteres	+	cordão de caracteres / variável de caracteres

Exemplo 1:

```
10 S$ = "SUPER"
20 T$ = S$ + "HOMEM"
30 PRINT T$
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Saída:

SUPERHOMEM

Na linha 20, os conteúdos da variável S$ ("SUPER") são "acrescentados" ao cordão"HOMEM". Note que não foi inserido espaço durante a concatenação. Vários cordões podem ser concatenados na mesma expressão, como no exemplo seguinte:

Exemplo 2:

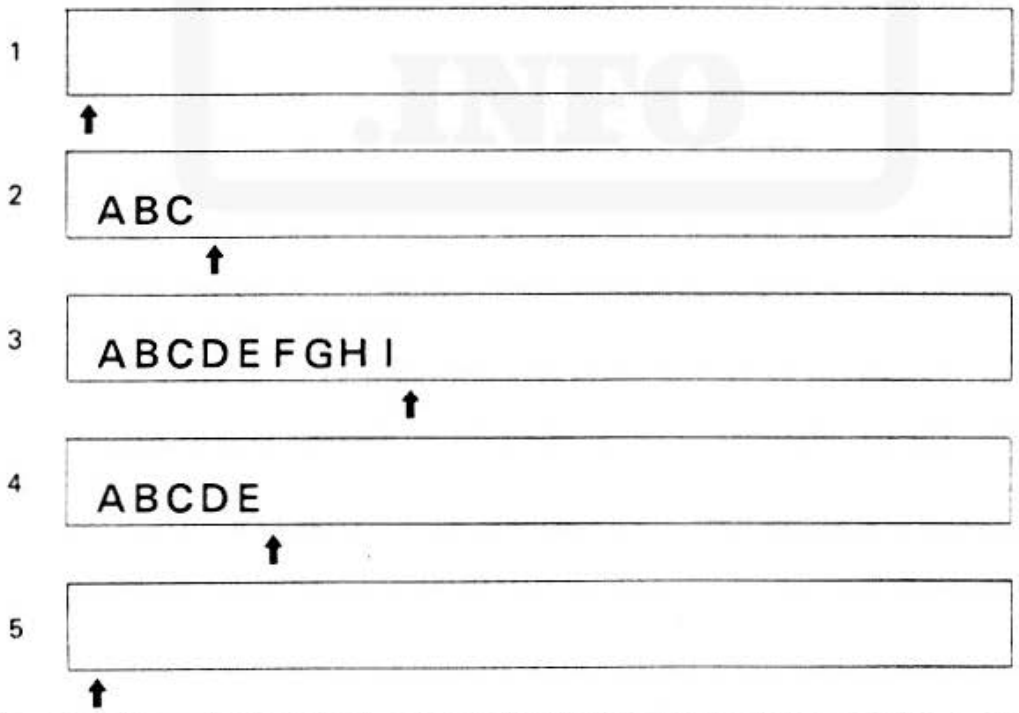
```
10 A$ = "TO"
20 B$ = "VERTI"
30 C$ = "MEN"
40 D$ = "DI"
50 S$ = "BOM" + D$ + B$
60 PRINT S$ + C$ + A$
```

Saída:

BOM DIVERTIMENTO

Quando as operações de concatenação são efetuadas, a área de armazenamento de caracter interna é usada temporariamente para construir um novo cordão alfanumérico. Esta área de armazenamento tem uma capacidade de 80 caracteres. Se o novo cordão alfanumérico exceder este comprimento, ocorrerá um "ERROR 15". Uma ilustração desta área durante uma operação de concatenação é mostrada no exemplo abaixo:

Exemplo 3: X = LEN ("ABC" + LEFT$ ("DEFGHI", 2)) ENTER



NOTA: o símbolo ⬆ representa um apontador interno de caracter que indica a quantidade de caracteres armazenados.

1. No início da execução, a área de armazenamento está limpa e o apontador de caracter está posicionado na posição inicial.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

2. "ABC" é introduzido na área, tomando as três primeiras posições.
3. "DEFGHI" é acrescentado à área de armazenamento de caracter, seguindo "ABC".
4. A função LEFT$, após ter colocado "DEFGHI" para verificar a quantidade total de caracteres, retira "FGHI" da área.
5. A verificação do comprimento do cordão alfanumérico é efetuada, colocada na variável"X" e a área de armazenamento é novamente limpa.

B.3. Comparação dos cordões de caracteres

Cordões de caracteres podem ser comparados, para se determinar qual deles é "maior" ou "menor" do que o outro.
Estas determinações são baseadas na Seqüência de Intercalação (dada no apêndice C) que é a ordem de todos os caracteres reconhecidos pelo computador.

Se os cordões alfanuméricos contém quantidades de caracteres diferentes, o cordão alfanumérico menor é "completado" (preenchido) com caracteres nulos (ASC II 0). Os operadores que são legítimos para a comparação de cordões de caracteres são:

verdadeiro se os dois cordões de caracteres são iguais em comprimento e contém os mesmos caracteres na mesma ordem.

verdadeiro se os dois cordões de caracteres diferem em comprimento, caracteres ou ordem de caracteres.

verdadeiro se o primeiro caracter do primeiro cordão de caracteres, que diferir do caracter do segundo cordão (na mesma posição) for maior do que este.

verdadeiro se o primeiro caracter do primeiro cordão de caracteres, que diferir do caracter do segundo cordão (na mesma posição) for menor do que este.

O formato para comparação de cordão de caracteres é:

cordão de caracteres / variável de caracteres	OP	cordão de caracteres / variável de caracteres

onde OP é um dos operadores de comparação listado acima.

Exemplos:

"Mary" > "Mari"	é verdadeiro
"Mary" = "Mary "	é falso
"abc" <> "ABC"	é verdadeiro
"DATA 1" < "DATA 2"	é verdadeiro
"?" < "#"	é falso

NOTA: A forma A$< = B$, A$> = B$ não pode ser usada para a comparação de cordões de caracteres.
Comparação é possível, entretanto na forma de (A$<B$) ou (A$ = B$) e (A$>B$) ou (A$ = B$).

C. FUNÇÕES

C.1. ASC

Existem duas funções usadas na conversão de caracteres para o código ASC II e vice-versa. A função ASC converte um caracter para seu código decimal ASC II. A função reversa CHR$ converte o código decimal ASC II em um cordão de caracteres.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

ASC | "caracter"
| nome de variável de caracter

O argumento nesta função é qualquer cordão de caracteres ou um nome da variável do cordão de caracteres.
O valor retornado para esta função é o código correspondente em ASC II para o primeiro caracter do cordão especificado.

Exemplo 1:

```
10 LET X$ = "PAULO"
20 LET A = ASC X$
30 PRINT A
```

Saida:

```
RUN
 80
```

No exemplo acima, à X$ é atribuído o valor do cordão de caracteres "PAULO".
A função ASC toma o primeiro caracter (P e converte-o para o seu código ASC II (80).

Exemplo 2:

```
10 PRINT ASC "K"
```

Saida:

```
RUN
 75
```

ASC "K" retorna o código ASC II para "K" que é 75.

C.2. CHR $

CHR$ é o complemento da função ASC. A função CHR$ toma um código decimal ASCII, de 0 à 127, e retorna ao cordão de caracteres equivalente. (NOTA: Alguns códigos representam caracteres especiais que não imprimem).

CHR$ | código decimal ASCII
| variável numérica

Exemplo 1:

```
10  PRINT (CHR$)67) + "OP"
```

Saida

```
RUN
COP
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Exemplo 2:

```
10 Z = 65
20 PRINT CHR$ Z
```

Saída:

```
RUN
A
```

O primeiro exemplo mostra o uso de um código decimal ASC II como um argumento. O resultado é concatenado para o cordão "OP" produzindo "COP". O segundo exemplo atribui um valor numérico para a variável Z. O nome da variável é então usado como o argumento, resultando em caracter "A".

Nosso próximo programa de exemplo converte os caracteres maiúsculos do cordão de texto para o minúsculo usando ambas as funções ASC e STR$:

Exemplo 3:

```
 5 WAIT 0
10 INPUT "ESCREVA MENSAGEM", M$
20 FOR I = 1 TO LEN (M$)
30 T$ = MID$ (M$,I,1)
40 L = ASC (T$)
45 IF (L<65) OR (L>90) THEN 60
50  T$ = CHR$ (L + 32)
60 PRINT T$;
70 NEXT I
80 WAIT : PRINT
```

C.3. INKEY$

Esta função toma qualquer caracter do teclado e armazena-o na variável especificada. Não há necessidade de pressionar ENTER porque o caracter irá automaticamente ser aceito.

```
variável = INKEY$
```

Durante a execução desta declaração o caracter "PRONTO" não é visualizado a não ser que uma declaração PRINT for usada previamente. O caracter de entrada não é devolvido para o visor e o visor permanece sem ser afetado.

Exemplo:

```
10 WAIT 0
20 A$ = INKEY$
30 IF A$ = "" THEN PRINT "NO KEY": GOTO 20
40 PRINT A$
50 GOTO 20
```

Esta função aceitará somente um caracter. Caso for digitado mais de um, será lido somente o primeiro caracter; o resto será ignorado.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.4. LEN

Durante a manipulação de caracteres é desejável conhecer o número de caracteres em um cordão. Isto pode ser feito através do uso da função LEN. Esta função faz retornar ao visor o número de caracteres da expressão especificada ou da variável de caracter.

LEN { "cordão de caracteres" / nome de variável de caracter }

Exemplo 1:

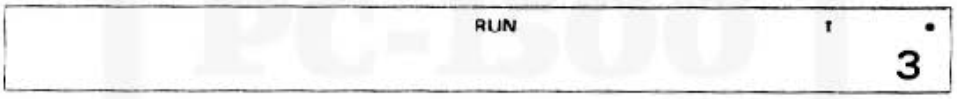
```
10 A$ = "CARLA"
20 C = LEN A$
30 PRINT C
```

Saída:

Exemplo 2:

```
10 C = LEN "CAO"
20 PRINT C
```

Saída:

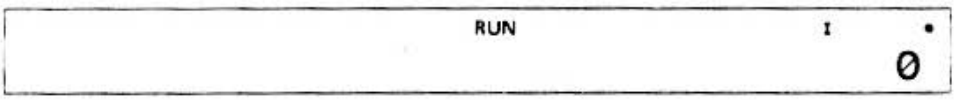


Se LEN for usada em um cordão vazio (isto é, não é colocado nada entre aspas), retornará o zero no visor.

Exemplo 3:

```
10 A = LEN "    "
20 PRINT A
```

Saída:



C.5. LEFT$

Existem três funções usadas para selecionar ou extrair secções especificadas de um cordão de caracteres. LEFT$ extrai caracter da esquerda, RIGHT$ da direita e MID$ do meio.

LEFT$ { ("cordão de caracteres" , número) / (nome da variável de caracter, número) }

Do not sell this PDF !!!

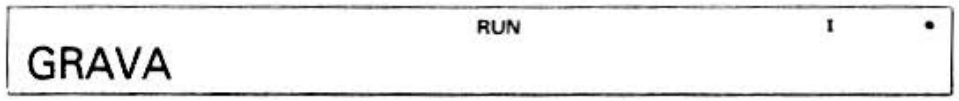
All and more about Sharp PC-1500 at http://www.PC-1500.info

O argumento "número" especifica quantos caracteres devem ser extraídos começando do lado esquerdo.

Exemplo 1:

```
10 A$ = LEFT$ ("GRAVATA", 5)
20 PRINT A$
```

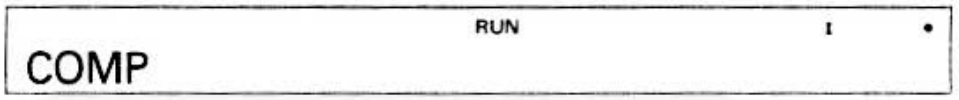
Saída:



Exemplo 2:

```
10 B$ = "COMPUTADOR"
20 A$ = LEFT$ (B$, 4)
30 PRINT A$
```

Saída:



Em ambos os exemplos, começando pelo lado esquerdo do cordão de caracteres, caracteres são extraídos e armazenados em A$. Imprimindo-se A$ resulta em "GRAVA" e "COMP" respectivamente.

C.6. MID$

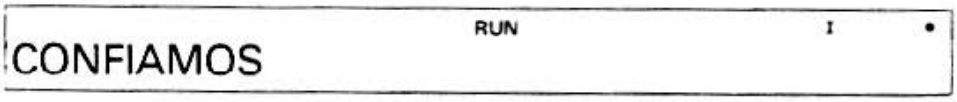
Para extrair a parte do meio do cordão de caracteres, usa-se a função MID$.

```
MID$ ("cordão de caracteres", expressão, expressão)
     (variável do cordão de caracteres, expressão, expressão)
```

Exemplo 1:

```
10 A$ = "NÓS CONFIAMOS EM VOCÊ"
20 B$ = MID$ (A$, 5, 9)
30 PRINT B$
```

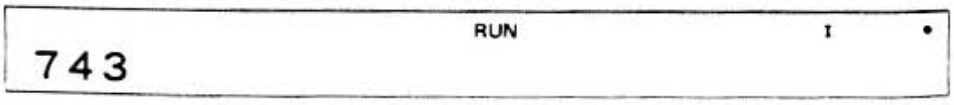
Saída:



Exemplo 2:

```
10 T$ = MID$ ("(415)743 - 1602", 6, 3)
20 PRINT T$
```

Saída:



Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

O primeiro argumento para esta função é um cordão de caracteres ou uma variável do cordão de caracteres. O segundo argumento é um número representando o primeiro caracter a ser extraído. O terceiro argumento é o número total de caracteres, incluindo o primeiro a ser extraído.

No primeiro exemplo, "NÓS CONFIAMOS EM VOCÊ" é armazenado em A$. MID$ extrai 9 caracteres começando do quinto e os coloca em B$. Quando imprime-se B$, ele verifica seu conteúdo que é "CONFIAMOS". No segundo exemplo, um cordão contendo um número de telefone é o primeiro argumento. O sexto caracter ("7") é localizaco e, junto com os dois caracteres seguintes, é armazenado na variável T$. Quando impresso, o resultado é "743".

C.7. RIGHT$

A função RIGHT$ como LEFT$. A única diferença é que começa do lado oposto (direito) ou seja, no fim do cordão. Os argumentos são os mesmos que LEFT$:

RIGHT$ { ("cordão de caracteres", número)
RIGHT$ { (variável de caracter, número)

O argumento "número" nesta função especifica quantos caracteres devem ser extraídos do cordão de caracteres começando do lado direito.

Exemplo 1:

```
10 X$ = "LEIA COM ATENÇÃO"
20 Y$ = RIGHT$ (X$, 7)
30 PRINT Y$
```

Neste programa a função RIGHT$ toma sete caracteres do fim do lado direito do cordão e os armazena na variável Y$. O conteúdo de Y$ agora é "ATENÇÃO".

C.8. RND

Talvez, você deseje "arranjar" o seu programa com números aleatórios. A função RND permite que o computador gere números que vão de 1 a um número qualquer especificado. (Nota: a geração de números sempre começa de 1).

Exemplo 1:

```
10 A = RND 5
```

Neste exemplo, A poderia ter qualquer um dos valores de 1 à 5, inclusive. Caso você deseje números aleatórios na faixa de valores que iniciem com um outro número diferente de 1, (por exemplo de 40 à 50) você terá que simular isto através da geração de números aleatórios de 1 à 10 e somar uma constante.
(39 no nosso exemplo):

Exemplo 2:

```
10 FOR I = 1 TO 5
20 B = 40 + RND 10
40 PRINT B;
50 NEXT 1
```

Saída:

```
RUN
42  44  47  48  42
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.9. RANDOM

Números aleatórios são gerados através de uma fórmula matemática e são acessíveis pelo uso da função RND. Todas as vezes que o computador é ligado, uma série de números aleatórios são gerados pelo computador. Esta lista se mantém inalterada a não ser que seja usada a função RANDOM. Isto significa que um programa usará a mesma série de números "aleatórios" cada vez que o computador é ligado. Para evitar isto, a função RANDOM reposiciona a "inicialização" usada pela fórmula para gerar seus números aleatórios.

Exemplo 1:

```
10 FOR I = 1 TO 5
15 A = RND 3
20 PRINT A;
30 NEXT I
```

Saída: 12231

Exemplo 2:

```
10 FOR I = 1 TO 5
15 A = RND 3
20 PRINT A;
30 NEXT I
```

Saída: 12231

Para obter números aleatórios verdadeiros, neste caso, a função RANDOM deve aparecer antes da declaração RND. Esta função propaga uma nova inicialização da geração de números aleatórios, o que faz com que os números sejam diferentes. Com isto, um programa executado diversas vezes, nas mesmas condições, irá produzir saídas variadas.

Exemplo 1:

```
10 RANDOM
15 FOR I= 1 TO 5
20 A = RND 3
30 PRINT A;
40 NEXT I
```

Saída: 31223

Exemplo 2:

```
10 RANDOM
15 FOR I = 1 TO 5
20 A = RND 3
30 PRINT A;
40 NEXT I
```

Saída: 23132

C.10. STR$

STR$ funciona de maneira oposta a VAL. Ele converterá uma variável numérica interna para sua representação em cordão de caracteres ou expressão.

Exemplo:

```
10 INPUT "INTRODUZA UM NÚMERO"; I
20 S$ = STR$ (I)
30 PAUSE "ESTE NÚMERO É"
40 PRINT "O CORDÃO"; S$
```

O programa acima aceita um número, forma a representação de caracter daquele número e armazena-o na variável de caracter S$. Devido a representação numérica interna não ser visualizada, ela é automaticamente convertida para o cordão de caracteres pela declaração PRINT.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.11. STATUS

Para visualizar quantas memórias de programa você tem disponível ou quantas memórias um programa utiliza, é usada a função STATUS.

STATUS Ø ENTER

Irá visualizar quantos "passos" de programa estão ainda disponíveis.

STATUS 1 ENTER

Irá visualizar o número de "passos" já utilizados.

STATUS 2 ENTER

Irá visualizar o endereço de memória onde está localizado o final do programa. Observe que este endereço é uma "locação" maior que o endereço real do final do programa.

STATUS 3 ENTER

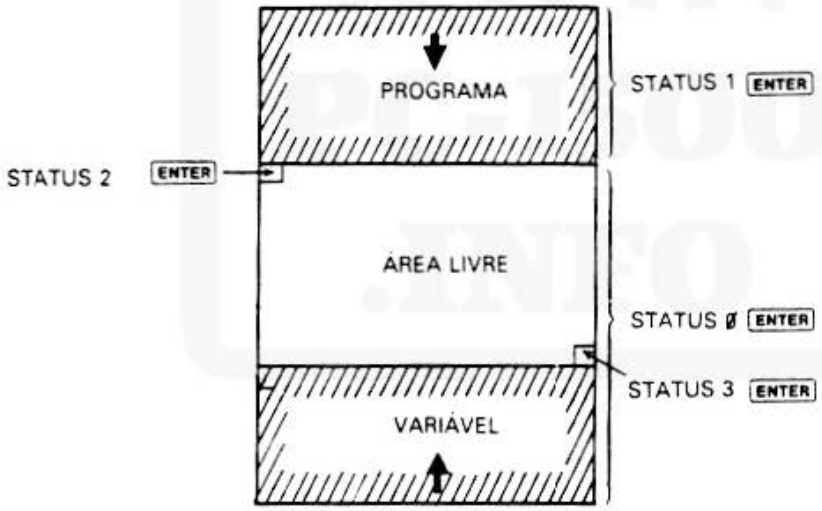
Visualizará o endereço de memória onde as variáveis estão guardadas. Observe que este endereço é uma "locação" menor que o endereço real onde as variáveis estão guardadas.

STATUS 4 ENTER ~ STATUS 100 ENTER

Visualizará o número da linha do programa que estava sendo executada quando o programa foi interrompido.

MEM ENTER

Este comando é equivalente à instrução STATUS Ø



Área livre é obtida por "STATUS 3 – STATUS 2 + 1".

C.12. TIME

Para visualizar ou posicionar o mês, dia e a hora, é usada a função TIME da seguinte maneira:

Posicionamento: TIME = MMDDHH.MMSS ENTER

VISOR: TIME ENTER

onde: MM representa dois dígitos para o mês, DD representa dois dígitos para o dia, e HH dois dígitos para a hora. A parte fraccional define os minutos (MM) e segundos (SS).

Quando estiver visualizando TIME, a saída será no formato acima. O resultado da função pode ser manuseado da mesma maneira que um número qualquer e poderá ser usado também em expressões.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

O programa abaixo simula um relógio. Precavenha-se de posicionar a hora antes de rodar o programa.

Listagem de Programa

```
10 WAIT 0
20 A$ = STR$ TIME
30 IF TIME > 99999 THEN 50
40 A$ = "0" + A$
50 M$ = LEFT$ (A$, 2)
60 D$ = MID$ (A$, 3, 2)
70 H$ = MID$ (A$, 5, 2)
80 DS$ = M$ + "/" + D$ +
         "/82"
90 DS = VAL (M$ + D$ + "00")
100 PRINT DS$;
110 T = TIME - DS
120 IF T >= 1 THEN 140
130 T = T + 12
140 IF T > 23. 5959 GOTO 20
150 CURSOR 18: PRINT USING
                 "# # #.# # # #"; T
160 GOTO 110
```

C.13. VAL

VAL e STR$ são funções complementares que convertem cordões de caracteres para uma variável numérica e vice-versa. A função VAL converte um cordão contendo a representação de caracter de um número dentro de um número, que é então armazenado em uma variável. NOTA: Quando é usado qualquer outro dígito além de 0 à 9, (ponto decimal), + (sinal positivo), − (sinal negativo), ou E (notação científica) na expressão, a conversão terminará com o caracter ilegal.

Exemplo 1: 10 ZS = VAL "−37" Resulta: ZS contém o número −37

Exemplo 2: 10 A = VAL "237.6" Resulta: A contém o número 237.6

D. PRINT USING

A declaração USING permite a um programador controlar rigidamente o formato da informação no visor. Isto permite visualizações padronizadas e evita perdas de informação.

Quando aparece a cláusula USING, sozinha ou com a declaração PRINT ou PAUSE, ela define o formato para todas as declarações subseqüentes PRINT ou PAUSE até que a próxima cláusula USING seja encontrada no programa.

Várias cláusulas USING podem aparecer dentro de uma declaração PRINT ou PAUSE única. Neste caso, cada uma define o formato a ser utilizado para imprimir as variáveis listadas, até que a próxima cláusula USING seja encontrada.

Um formato é especificado através de um cordão de caracteres especial chamado "cordão de edição". Os caracteres dentro do cordão de edição definem as áreas do visor disponíveis para informação e restringe o tipo de informação que pode ser impresso nestas áreas. Este esquema é o mesmo esquema geral empregado pelas outras linguagens, tais como COBOL e PL/I.

Um cordão de edição pode ser armazenado na variável do cordão. O nome da variável colocaria então, o cordão de edição com a cláusula USING. Isto permite múltiplos formatos que são selecionados pelo controle de programa.

Os caracteres que podem ser empregados dentro dos cordões de edição são resumidos abaixo:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

TABELA DE CARACTERES DE EDIÇÃO

Caracter

- # Especifica um campo numérico. Números são justificados à direita dentro deste campo. Se a largura do campo não for suficiente para suportar o número, ocorrerá "ERROR 36".
- * Especifica "Completar com asteriscos" as posições especificadas de um campo de numeração que não contenham dados.
- . Faz com que o ponto decimal seja visualizado dentro de um campo numérico.
- , Usado no começo de um campo numérico para especificar a inserção de vírgula, a cada três dígitos.
- ^ Usado dentro de um campo numérico para fazer com que o número seja visualizado em notação científica
- + Usado no campo numérico para forçar a impressão do sinal dos dados.
- & Especifica um campo de caracter. Caracteres são justificados à esquerda dentro deste campo. Se a largura do campo não for suficiente para suportar o cordão de dados, esta é truncada.

NOTA: A largura de um campo numérico deve sempre ser um campo maior que a largura dos dados, para permitir a atribuição do sinal dos dados, independentemente de utilizar ou não o caracter de edição +.

NOTA: O uso da vírgula requer que você insira um # extra, para cada vírgula no cordão de edição.

Exemplos:

```
X = PI   Y = 1234   A$ = "ABCDEF"
PRINT USING "###"; X
```

```
                                3
```

```
PRINT USING "+####.###"; X
```

```
                           +3.141
```

```
PRINT USING "###.##^"; X
```

```
                         3.14E 00
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
PRINT USING "###.^"; X
```

```
                    3.E 00
```

```
PRINT USING "*######"; Y
```

```
                   **1234
```

```
PRINT USING "****##"; Y
```

```
                     1234
```

```
PRINT USING "&&&&&&&##########"; A$; Y
```

```
ABCDEF       1234
```

```
PRINT USING "&&&"; A$
```

```
ABC
```

```
10 U$ = "*######. ##"
20 USING U$
30 PRINT Y; "$"
```

```
**1234. 00$
```

```
PRINT X; "$"
```

```
*****3. 14$
```

```
PRINT USING; A$; X
```

```
ABCDEF 3. 141592654
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
PRINT USING "###, ###, ###"; 246813
```

```
                246, 813
```

NOTA: Use o seguinte número de símbolos # (inclusive *) para a designação da variável (inteiro):
Quando utilizar pontuação a cada 3 dígitos (,): 14 (incluindo sinal)
Quando não utilizar pontuação a cada 3 dígitos (,): 11 (incluindo sinal)
Este computador tem 10 dígitos significativos para números.
Quando o formato exceder 10 inteiros é designado pela declaração USING, e a visualização e impressão quando exceder 10 inteiros sob a declaração PRINT e declaração LPRINT (para a impressora), pode ser incorreta.

Exemplo:

modo RUN		Visor
USING "########## ####" ENTER	→	>
PRINT 888888883888 ENTER	→	888888888800
(LPRINT 888888883888 ENTER	→	88888888880)
12 dígitos		

E. TRANSFERÊNCIA DE CONTROLE COMPUTADO

Em adição às declarações de controle básicas descritas no capítulo III, o PC-1500 fornece outras duas declarações de controle de grande utilidade. Estas são ON GOSUB e ON GOTO. Como você pode supor pelos nomes, estas declarações atuam como as declarações GOSUB e GOTO vistas anteriormente. A diferença, entretanto, está na habilidade de transferir o controle (isto é, executar declarações em locais diferentes) automaticamente. Isto é, as funções GOTO ou GOSUB irão para uma das várias declarações (ou subrotinas) dependendo do valor da variável numérica. Esta dependência da variável para "guiar" é que dá à declaração ON o "apelido" de "declaração de controle computado".

ON expressão { GOTO linha # 1, linha # 2, linha # 3, (etc)
ON expressão { GOSUB linha # 1, linha # 2, linha # 3

A expressão que segue a palavra chave ON deve evoluir para um inteiro positivo, maior que zero e menor do que o número de linhas listados após a palavra chave GOTO ou GOSUB. Durante a execução, quando o computador encontrar uma declaração ON, ela transfere o fluxo ou execução para o número da linha que corresponde ao valor da expressão.
Uma declaração ON típica pode ser:

```
ON TX GOSUB 100, 200, 250, 300
```

Neste caso, a variável TX DEVE conter um número no limite de 1 à 4 porque existem somente quatro números de linhas listados.
Qualquer outro número em TX resultará na execução da linha seguinte. No caso do número resultante em TX ter uma parte decimal, ela será desprezada e somente a parte inteira será levada em conta.

As declarações ON são muito úteis para automatizar uma série de escolhas. Por exemplo, considere o fragmento de programa abaixo, que permite ao usuário selecionar um dos vários preços de uma tabela. Sem a declaração ON, este pode ser escrito:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
10 PAUSE "ESCOLHA A TABELA:"
20 PAUSE "(1) SOLTEIRO"
30 PAUSE "(2) CASADO"
40 INPUT "(3) NEGÓCIOS?"; TT
50 IF (TT < 1) OU (TT > 3) GOTO 10
60 REM USE A TABELA APROPRIADA
70 IF TT = 1 THEN GOTO 220
80 IF TT = 2 THEN GOTO 300
90 IF TT = 3 THEN GOTO 450
(etc)
```

Usando a declaração ON, nós podemos agrupar as linhas 70, 80 e 90 em uma única declaração:

```
70 ON TT GOTO 220, 300, 450
```

ON ERROR GOTO

O uso de uma outra forma de transferência controlada, permite um programa detectar um erro. Após a detecção, o programa pode executar declarações que tentarão se recuperar do erro. Tais declarações podem informar e instruir ao usuário ou guardar dados valiosos.
As declarações ON ERROR GOTO instrui o Sharp onde ir após detectar a ocorrência de um erro. A forma desta declaração é:

ON ERROR GOTO número da linha

onde o número da linha é o número de uma linha do programa, que contém as instruções a serem seguidas na ocorrência de um erro.

F. PROGRAMAÇÃO DO VISOR

O visor incorporado ao PC-1500 é um notável e flexível dispositivo de saída. Para permitir que os programadores explorem toda a potência do visor, foram adicionadas várias declarações novas na linguagem BASIC utilizada pelo PC-1500. Estas extensões são descritas nesta secção.

O visor em si utiliza a tecnologia de cristal líquido para visualizar até 26 caracteres.
Cada caracter dentro do conjunto de caracteres do computador ocupa uma matriz de pontos de 5x7. Utilizando o comando GPRINT, é possível desenvolver e visualizar seus próprios caracteres.
Para uso gráfico, o campo inteiro do visor pode ser utilizado como uma matriz de pontos de 7x156. Pontos individuais dentro de cada uma das 156 colunas podem ser ativados para criar gráficos, figuras ou símbolos especiais. O comando POINT permite "sensibilização" de qualquer coluna para descobrir quais os pontos que estão ativados no presente momento.

Um alto-falante e um gerador de tons permitem ao programador adicionar sons na interação homem-máquina. Os tons podem ser criados em qualquer uma das 256 frequências (de 230 Hz à cerca de 7K Hz). A repetição automática de um tom e o controle da duração destes também são possíveis sob controle do programa.

F.1. BEEP

A declaração BEEP permite ao programador criar sons para jogos, sinalizar erros, e outras aplicações interativas. O formato da declaração BEEP que cria sons é:

BEEP expressão 1, expressão 2, expressão 3

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

onde:

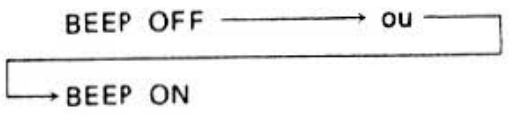
expressão 1 é o único parâmetro requisitado e especifica quantas vezes o som de Beep é repetido. A faixa permitida é de 0 à 65535.

expressão 2 é opcional e especifica a frequência do (s) tom(s).
Este é um número de 0 à 255.

expressão 3 é também opcional e especifica a duração de cada tom.
Esta duração é especificada como um número na faixa de 0 à 65279.

A declaração BEEP pode também ser usada para desligar e ligar o alto-falante interno do PC-1500. Assim, o ruído originado pela operação de cassete, Save ou Load, pode ser eliminado.

O formato da declaração BEEP que controla o alto-falante interno é:



NOTA: Sempre que o PC-1500 é ligado, o alto-falante é reativado.

PROGRAMA DE DEMONSTRAÇÃO

```
10 D=60
20 DATA 14
30 DATA 245, 1, 245, 1, 160, 1, 160, 1
40 DATA 143, 1, 143, 1, 160, 2
50 DATA 180, 1, 180, 1, 195, 1, 195, 1
60 DATA 220, 1, 220, 1, 245, 2
100 READ X
110 FOR I=1 TO X
120 READ N, S
130 BEEP 1, N, (D*S)
140 NEXT I
150 END
```

F.2. CURSOR

A declaração CURSOR posiciona o cursor em uma das 26 posições de caracteres do visor. A forma desta declaração é:

Cursor expressão de posição

onde:

expressão de posição é um número na faixa de 0~25, que especifica para onde o cursor se moverá.

A utilização normal do comando CURSOR é para posicionar o cursor preparando a impressão de alguma informação. Usado desta maneira, ele permite que o programador defina a separação entre os dados, como nas seguintes declarações:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Listagem do Programa

```
10: WAIT 20
20: X=8
30: PRINT "A"
40: CURSOR 4
50: PRINT "B"
60: CURSOR X
70: PRINT "C"
80: CURSOR ((X^2) / 4) – 4
90: PRINT "D"
100: END
```

Digitação de Teclas:

[1] [Ø] [W] [A] [I] [T] [2] [Ø] [ENTER]

[2] [Ø] [X] [=] [8] [ENTER]

[3] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [A] [SHIFT] ["] [ENTER]

[4] [Ø] [C] [U] [R] [S] [O] [R] [4] [ENTER]

[5] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [B] [SHIFT] ["] [ENTER]

[6] [Ø] [C] [U] [R] [S] [O] [R] [X] [ENTER]

[7] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [C] [SHIFT] ["] [ENTER]

[8] [Ø] [C] [U] [R] [S] [O] [R] [(] [(] [X] [SHIFT] [^] [2] [)]
[/] [4] [)] [–] [4] [ENTER]

[9] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [D] [SHIFT] ["] [ENTER]

[1] [Ø] [Ø] [E] [N] [D] [ENTER]

Este programa fará com que as letras A, B, C e D apareçam no visor, nas posições 0, 4, 8 e 12, respectivamente.

```
                                RUN                    •
A    B    C    D
```

NOTA: Especificando uma posição de cursor maior que 25 ou menor que 0 resultará em "ERROR 19".

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Programa de Demonstração

```
10 WAT 0
20 DIM A$ (0) * 13
30 INPUT "DIGITE SEU NOME", A$ (0)
40 C=0 : CLS
50 FOR I=1 TO (LEN A$ (0)-1)
60 CURSOR C
70 PRINT MID$ (A$ (0), I, 1)
80 C=C + 2
90 NEXT I
100 WAIT
110 CURSOR C
120 PRINT RIGHT$ (A$ (0), 1)
130 END
```

Digitação de Teclas

[1] [Ø] [W] [A] [I] [T] [Ø] [ENTER]

[2] [Ø] [D] [I] [M] [A] [SHIFT] [$] [(] [Ø] [)] [*] [1] [3] [ENTER]

[3] [Ø] [I] [N] [P] [U] [T] [SHIFT] ["] [D] [I] [G] [I] [T] [E] [SPACE]
[S] [E] [U] [SPACE] [N] [O] [M] [E] [SHIFT] ["]
[SHIFT] [,] [A] [SHIFT] [$] [(] [Ø] [)] [ENTER]

[4] [Ø] [C] [=] [Ø] [SHIFT] [:] [C] [L] [S] [ENTER]

[5] [Ø] [F] [O] [R] [I] [=] [1] [T] [O] [(] [L] [E] [N] [A] [SHIFT] [$] [(] [Ø] [)]
[-] [1] [)] [ENTER]

[6] [Ø] [C] [U] [R] [S] [O] [R] [C] [ENTER]

[7] [Ø] [P] [R] [I] [N] [T] [M] [I] [D] [$] [(] [A] [$] [(] [Ø] [)]
[SHIFT] [,] [I] [SHIFT] [,] [1] [)] [ENTER]

[8] [Ø] [C] [=] [C] [+] [2] [ENTER]

[9] [Ø] [N] [E] [X] [T] [I] [ENTER]

[1] [Ø] [Ø] [W] [A] [I] [T] [ENTER]

[1] [1] [Ø] [C] [U] [R] [S] [O] [R] [C] [ENTER]

[1] [2] [Ø] [P] [R] [I] [N] [T] [R] [I] [G] [H] [T] [SHIFT] [$]
[(] [A] [SHIFT] [$] [(] [Ø] [)] [SHIFT] [,] [1] [)]
[ENTER]

[1] [3] [Ø] [E] [N] [D] [ENTER]

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F.3. CLS (LIMPEZA DO VISOR)

A declaração CLS apaga o visor desligando todos os pontos. Ela é usada antes de outras declarações para apagar qualquer dado residual do visor. O cursor é reposicionado para a esquerda da tela. O formato é simplesmente CLS.

Programa de demonstração

```
10 GPRINT "7F7F7F7F7F"
20 CLS
30 PRINT "NOVA INFO"
```

Digitação de Teclas

[1] [Ø] [G] [P] [R] [I] [N] [T] [SHIFT] ["] [7] [F] [7] [F] [7] [F]
[7] [F] [7] [F] [SHIFT] ["] [ENTER]
[2] [Ø] [C] [L] [S] [ENTER]
[3] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [N] [O] [V] [A] [SPACE]
[I] [N] [F] [O] [SHIFT] ["] [ENTER]

F.4. GCURSOR

A declaração GCURSOR especifica uma das 156 colunas de pontos, utilizável no visor como a coluna inicial para qualquer visualização de informação. O formato da instrução GCURSOR é:

GCURSOR expressão de posição

onde:

expressão de posição é um número na faixa de 0 à 155. Este número especifica uma das 156 colunas de 7 pontos do visor.

NOTA: Caso uma expressão de posição resultar em um número que seja menor que 0 ou maior que 155, ocorrerá "ERROR 19".

A declaração GCURSOR é geralmente usada em conjunto com a declaração GPRINT, a fim de criar representações gráficas (isto será ilustrado mais detalhadamente na próxima secção).

Outros tipos de instruções podem seguir a declaração GCURSOR devido esta não escrever qualquer informação. GCURSOR simplesmente indica em qual coluna a informação subsequente será escrita.
Isto é ilustrado pelas linhas seguintes:

Listagem do Programa

```
10 GCURSOR 50
20 PRINT "A"
30 GCURSOR 80
40 PRINT 26/3
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

[1] [Ø] [G] [C] [U] [R] [S] [O] [R] [5] [Ø] [ENTER]

[2] [Ø] [P] [R] [I] [N] [T] [SHIFT] ["] [A] [SHIFT] ["] [ENTER]

[3] [Ø] [G] [C] [U] [R] [S] [O] [R] [8] [Ø] [ENTER]

[4] [Ø] [P] [R] [I] [N] [T] [2] [6] [/] [3] [ENTER]

que produz uma saída aparentemente normal nas posições 50 e 80:

```
                         RUN                  •
              A            8. 666666667
```

O que acontece se nós alterarmos a linha 30 do programa, para que a impressão inicie na posição 93? Tente isto substituindo da seguinte forma:

```
30 GCURSOR 93
```

Surpresa! A segunda saída "truncada" (cortada) porque ultrapassou o fim do visor.

Como um exemplo final da declaração GCURSOR, nós apresentamos um programa avançado que cria um efeito não usual, através da sobreposição de caracteres.

Listagem do Programa

```
10 WAIT 0
20 DIM A$ (0)*10
30 A$ (0) = "":C = 0
40 INPUT  "INTRODUZA MENSAGEM" (<10 CARACT)", A$ (0)
50 CLS
60 FOR I = 1 TO (LEN A$ (0))
70 GCURSOR C
80 FOR J = 1 TO 3
90 GCURSOR C
100 PRINT MID$ (A$ (0), I, 1)
105 C = C+3
110 NEXT J
120 C = C+5
130 NEXT I
140 WAIT
150 GCURSOR 155
160 GPRINT "00"
170 END
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
1 Ø W A I T Ø ENTER
2 Ø D I M A SHIFT $ ( Ø ) * 1 Ø ENTER
3 Ø A SHIFT $ ( Ø ) = SHIFT " SHIFT "
  SHIFT : C = Ø ENTER
4 Ø I N P U T SHIFT " I N T R O D U Z A SPACE
  M E N S A G E M SPACE ( SHIFT < 1 Ø
  SPACE C A R A C T ) SHIFT " SHIFT ,
  A SHIFT $ ( Ø ) ENTER
5 Ø C L S ENTER
6 Ø F O R I = 1 T O ( L E N A SHIFT $
  ( Ø ) ) ENTER
7 Ø G C U R S O R C ENTER
8 Ø F O R J = 1 T O 3 ENTER
9 Ø G C U R S O R C ENTER
1 Ø Ø P R I N T M I D SHIFT $ ( A
  SHIFT $ ( Ø ) SHIFT , I SHIFT ,
  1 ) ENTER
1 Ø 5 C = C + 3 ENTER
1 1 Ø N E X T J ENTER
1 2 Ø C = C + 5 ENTER
1 3 Ø N E X T I ENTER
1 4 Ø W A I T ENTER
1 5 Ø G C U R S O R 1 5 5 ENTER
1 6 Ø G P R I N T SHIFT " Ø Ø SHIFT " ENTER
1 7 Ø E N D ENTER
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F.5. GPRINT

A declaração GPRINT permite controle programável diretamente sobre os pontos do visor. Devido a declaração GPRINT posicionar e retirar pontos de qualquer um dos 7 pontos de uma coluna, ela é normalmente usada em conjunto com a instrução GCURSOR. A declaração GCURSOR seleciona a coluna apropriada para a modificação e a declaração GPRINT manipula os pontos dentro daquela coluna. A declaração GPRINT é também capaz de imprimir várias colunas de informações contíguas em uma declaração única.
Para entendermos o formato da declaração GPRINT, é necessário que se entenda como os pontos dentro de uma coluna são controlados. A estrutura de pontos energizados dentro de uma coluna podem ser especificadas também como um número decimal ou como um cordão de caracteres hexadecimal. Se o sistema decimal é utilizado, então cada linha pode ser visualizada como sendo numerada, de cima para baixo, através de potência de dois.
Isto é ilustrado abaixo:

```
 1 - - - - - - - -
 2 - - - - - - - -
 4 - - - - - - - -
 8 - - - - - - - -
16 - - - - - - - -
32 - - - - - - - -
64 - - - - - - - -
```

Com 7 pontos por coluna, cada um podendo estar ligado ou desligado, existem 128 estruturas de pontos possíveis. Então para especificar uma estrutura particular, utiliza-se o formato:

GPRINT estrutura-expressão; estrutura-expressão 2... (etc)...

onde:

estrutura-expressão é um número na faixa de 0 à 127 que especifica a estrutura de pontos energizados. Várias expressões de estrutura podem ser especificados opcionalmente e devem ser separado por um ponto e vírgula ou uma vírgula. Se uma vírgula for usada uma coluna vazia será deixada entre cada coluna impressa.

Vamos ilustrar a utilidade da instrução GPRINT criando um novo caracter; uma Seta para cima. Primeiro, desenhamos os nossos caracteres sobre uma grade representando as linhas e as colunas.

```
 1 - - - - * - - - -
 2 - - * - * - * - -
 4 * - - - * - - - *
 8 - - - - * - - - -
16 - - - - * - - - -
32 - - - - * - - - -
64 - - - - * - - - -
    1  2  3  4  5
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Devido o nosso caracter possuir largura de 5 colunas, necessitaremos cinco números separados na lista de expressão de estrutura da declaração GPRINT. Os números representando as colunas 1 e 5 devem especificar cada um, um ponto único na linha representada por 4. Similarmente, os números representando as colunas 2 e 4 devem especificar cada um, um ponto único na linha 2. A declaração final é:

GPRINT 4; 2 ; 127 ; 2 ; 4

A especificação da terceira coluna (127) é o único número cuja a derivação pode não ser imediatamente óbvia. O número 127 é a soma de um ponto na primeira linha (1), um ponto na segunda linha (2), um ponto na terceira linha (4), e assim por diante. Então, 127 é 1 + 2 + 4 + 8 + 16 + 32 + 64 e especifica todos os 7 pontos na coluna.
Qualquer estrutura pode ser criada especificando uma linha ou uma soma de várias linhas.

Se o esquema de endereçamento hexadecimal for usado, as 7 linhas do visor são conceitualmente divididas em um grupo inferior à 3 linhas e um grupo superior à 4 linhas. Cada grupo é numerado, a partir da sua linha superior, por potência de dois conforme ilustrado abaixo:

```
1 - - - - - - - - -
2 - - - - - - - - -
4 - - - - - - - - -
8 - - - - - - - - -
1 - - - - - - - - -
2 - - - - - - - - -
4 - - - - - - - - -
```

Desta maneira, é possível representar todas as estruturas de um grupo por um dígito hexadecimal único. Devido ao grupo inferior ter somente 3 linhas, o limite de dígitos permitido para este grupo será de 0 à 7. Dos dois dígitos hexadecimais solicitados para especificar uma coluna inteira, o primeiro dígito representará o grupo inferior e o segundo dígito representará o grupo superior.

A forma da GPRINT hexadecimal é:

GPRINT "cordão hexadecimal"

onde:

cordão hexadecimal é um cordão constituído de dígitos hexadecimais, onde cada par especifica a estrutura de pontos de uma coluna única.

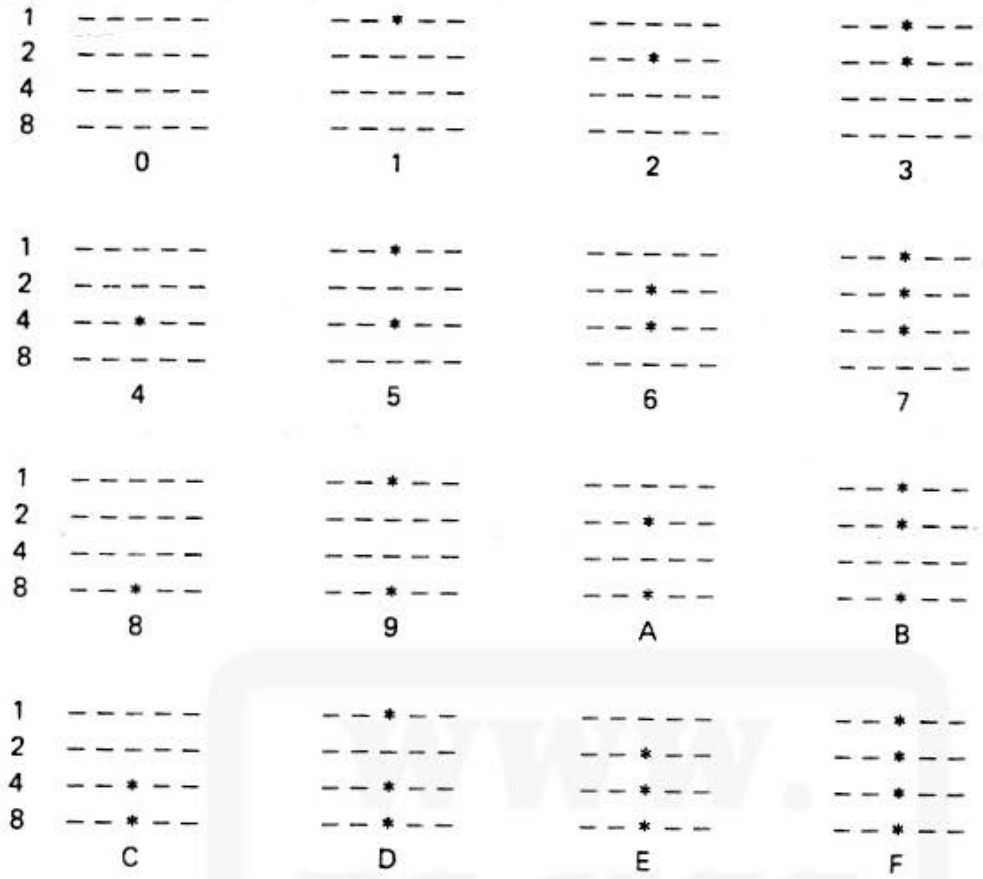
Usando este formato para criar o caracter de Seta para cima do exemplo anterior, nos dará a declaração:

GPRINT "04027F0204"

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

TABELA DE CARACTERES HEXADECIMAL



PROGRAMA DE DEMONSTRAÇÃO

Este programa imprime todos os pontos possíveis, em ordem, de 0 à 127:

Listagem do Programa:

```
10 WAIT 0: CLS
20 FOR I = 0 TO 127
30 GCURSOR I
40 GPRINT I
50 NEXT I
60 WAIT
70 GCURSOR 155: GPRINT "00"
80 END
```

Digitação de teclas

1 Ø W A I T Ø SHIFT : C L S ENTER

2 Ø F O R I = Ø T O 1 2 7 ENTER

3 Ø G C U R S O R I ENTER

4 Ø G P R I N T I ENTER

5 Ø N E X T I ENTER

6 Ø W A I T ENTER

7 Ø G C U R S O R 1 5 5 SHIFT :
G P R I N T SHIFT " Ø Ø SHIFT " ENTER

8 Ø E N D ENTER

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

F.6. POINT

A função POINT devolve um número que representa a estrutura de pontos ativados dentro da coluna dada. A função POINT permite a "sensibilização" de qualquer coluna no visor, sob o controle do programa.

O formato da função POINT é:

POINT expressão de posição

onde:

expressão de posição permite um número na faixa de 0 à 155 e representa a coluna a ser investigada.

O valor devolvido pela função POINT é um número na faixa de 0 à 127. A interpretação deste número é uma soma de potências de 2 conforme explicamos na secção de GPRINT.

Conforme a ilustração, supõe-se que no visor existe uma letra maiúscula I nas colunas de 40 à 44:

```
 1   - - * - * - * - -
 2   - - - - * - - - -
 4   - - - - * - - - -
 8   - - - - * - - - -
16   - - - - * - - - -
32   - - - - * - - - -
64   - - * - * - * - -
       4   4   4   4   4
       0   1   2   3   4
```

A expressão:

POINT 40 devolveria 0,
POINT 41 devolveria 65,
POINT 42 devolveria 127

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PROGRAMA DE DEMOSTRAÇÃO

O programa listado abaixo, preenche o visor com o caracter armazenado em A$ e cria uma estrutura não usual, invertendo a sensibilização dos pontos deste caracter. O programa utiliza algumas das declarações discutidas neste capítulo.

Listagem do Programa

```
10 A$ = "X"
20 WAIT 0
30 Y = 5: X = 155/Y: C = 0
40 FOR I = 1 TO X
50 GCURSOR C
60 PRINT A$
70 C = C + Y
80 NEXT I
90 FOR I = 0 TO 155
100 GCURSOR I
110 A = 127 - POINT I
120 GPRINT A
130 NEXT I
140 GOTO 90
```

Digitação das teclas

[1] [Ø] [A] [SHIFT] [$] [=] [SHIFT] ["] [X] [SHIFT] ["] [ENTER]

[2] [Ø] [W] [A] [I] [T] [Ø] [ENTER]

[3] [Ø] [Y] [=] [5] [SHIFT] [:] [X] [=] [1] [5] [5] [/] [Y]
[SHIFT] [:] [C] [=] [Ø] [ENTER]

[4] [Ø] [F] [O] [R] [I] [=] [1] [T] [O] [X] [ENTER]

[5] [Ø] [G] [C] [U] [R] [S] [O] [R] [C] [ENTER]

[6] [Ø] [P] [R] [I] [N] [T] [A] [SHIFT] [$] [ENTER]

[7] [Ø] [C] [=] [C] [+] [Y] [ENTER]

[8] [Ø] [N] [E] [X] [T] [I] [ENTER]

[9] [Ø] [F] [O] [R] [I] [=] [Ø] [T] [O] [1] [5] [5] [ENTER]

[1] [Ø] [Ø] [G] [C] [U] [R] [S] [O] [R] [I] [ENTER]

[1] [1] [Ø] [A] [=] [1] [2] [7] [-] [P] [O] [I] [N] [T] [I] [ENTER]

[1] [2] [Ø] [G] [P] [R] [I] [N] [T] [A] [ENTER]

[1] [3] [Ø] [N] [E] [X] [T] [I] [ENTER]

[1] [4] [Ø] [G] [O] [T] [O] [9] [Ø] [ENTER]

NOTA: Se a última linha for incluída, o programa entrará em um "loop" infinito, ao ser executado. (isto pode ser interrompido com a tecla BREAK). O caracter visualizado pode ser substituído, posicionando um novo caracter na atribuição da linha 10.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

G. DEBUGGING

Apesar de nós nos cercarmos de todos os cuidados, é possível que criemos um programa que não seja aquilo que esperavamos. A fim de evitar este problema, os projetistas da Sharp criaram um método especial de execução de programas conhecido como o modo "Trace". No modo Trace, o PC-1500 visualizará o número de cada linha do programa e parará após a execução desta linha. Isto permite que você siga (ou trace) a seqüência das instruções conforme eles são efetuadas realmente. Quando o programa pára após a execução de uma linha, é possível inspecionar ou alterar os valores das variáveis.

A forma da instrução para iniciar o modo Trace é simplesmente TRON. A instrução TRON pode ser emitida como um comando (no modo RUN) ou pode ser fixada, como uma declaração, dentro de um programa. Usado como um comando, TRON informa ao Sharp que foi solicitado o método Trace. Os programas a serem traçados são então iniciados de maneira normal, como um comando GOTO ou RUN.

Se TRON for usado como uma declaração, ele iniciará o modo TRACE somente quando a linha que a contém for executada. Se, por alguma razão, esta linha nunca for alcançada, o modo TRACE permanecerá inativo.

Uma vez iniciado, o modo TRACE de operação permanecerá ativo até ser cancelado pela instrução TROFF. A instrução TROFF pode ser também emitida tanto como comando ou como declaração. O modo Trace pode também ser cancelado pela seqüência de teclas:

[SHIFT] [CL]

Como um exemplo de uso do modo Trace, introduza o programa abaixo para calcular o comprimento da hipotenusa de um triângulo, dado o comprimento dos lados:

```
LISTAGEM DO PROGRAMA:

10 INPUT A, B
20 A = A*A: B = B*B
30 H = √(A + B)
40 PRINT "HIPOTENUSA = "; H
```

No modo RUN, emita o comando TRON, seguido pelo comando RUN. Observe que o comando INPUT opera de maneira usual visualizando um ponto de interrogação para cada valor de entrada solicitado. Logo que você introduzir os dois valores, o número da linha da declaração INPUT aparece:

```
                    RUN                    I    •
10:
```

Pressionando a tecla [↑] (seta para cima) e mantendo-a pressionada, você pode revisar a linha inteira:

```
                    RUN                    I    •
10: INPUT  A, B
```

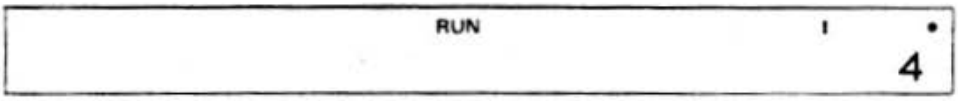
Para continuar o programa, pressione a tecla [↓] (seta para baixo) uma vez. Isto faz com que a próxima linha seja executada e seu número de linha seja visualizado.

Novamente, você pode revisar a linha com a tecla [↑]. Você pode verificar também os conteúdos de qualquer variável digitando o seu nome e pressionando [ENTER]:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Exemplo: [A] [ENTER]
(quando A = 4 é introduzido antes da operação [A] [ENTER])



É necessário pressionar a tecla [↓] uma vez para cada linha, até que todas as linhas do programa sejam executadas. Caso você não deseje continuar a execução linha por linha, pressione a tecla ENTER para suspender a execução do programa. Se você mudar de idéia novamente, os programas suspensos podem ser continuados com o comando CONT.

Um exemplo usando o nosso programa de hipotenusa seria o seguinte:

Digitação de Teclas	Visor
	>
[T] [R] [O] [N]	TRON_
[ENTER]	>
[R] [U] [N]	RUN_
[ENTER]	?
[3]	3_
[ENTER]	?
[4]	4_
[ENTER]	10:
[↑]	10: INPUT A, B_
[↓]	20:
[↑]	20: A=A*A: B=B*B_
[A]	A_
[ENTER]	9
[B]	B_
[ENTER]	16
[↓]	30:
[H]	H_
[ENTER]	5
[↓]	HIPOTENUSA= 5
[↑]	40: PRINT "HIPOTENUSA= ": H_
[↓]	40:
[↓]	>

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

H. NÚMEROS HEXADECIMAIS E FUNÇÕES BOOLEANAS

H.1. Números Hexadecimais

O PC-1500 possui a capacidade de usar um número hexadecimal (base 16) dentro de qualquer expressão, na qual um número decimal deve ser usado. Números hexadecimais são diferenciados de números decimais precedendo-se com um &. Abaixo mostramos exemplos de números hexadecimais válidos.

&16 &F &7ECA &08 &99A –&5B

Números hexadecimais podem ser usados em cálculos:

10 + &A [ENTER]

```
                    RUN                    I    •
                                          20
```

Ou dentro de programas:

Programa:

```
35 PRINT &F, 54, &3E
40 DATA 67, &7F, &2B, 12. 305
```

H.2. Função AND

A função AND fornece um AND booleano entre a representação binária de dois valores. Os valores devem estar na faixa de –32768 à 32767. Números que excedem esta margem farão com que ocorra um 'ERROR 19''.

Exemplo:	Resultado
10 AND &F	10
1 AND 0	0
–1 AND 1	1
55 AND 64	0
16 AND 63	16

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

H.3. FUNÇÃO OR

A função OR efetua um OR booleano entre as representações binária de dois valores. Os valores podem estar na faixa de −32768 à32767.Números que excedam esta margem causarão um 'ERROR 19".

Exemplo:	Resultado
10 OR &F	15
1 OR 0	1
−1 OR 1	−1
55 OR 64	119
16 OR 63	63

H.4. FUNÇÃO NOT

A função NOT efetua o NOT booleano, ou complemento da representação binária de um valor único. O valor deve estar na faixa de −32768 à 32767. Se o valor exceder esta margem causará um 'ERROR 19".

Exemplos:	Resultados
NOT Ø	−1
NOT &F	−16
NOT 55	−56
NOT 1	−2
NOT −2	1

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

I. PARANDO A EXECUÇÃO DO PROGRAMA

STOP, CONT

A declaração STOP faz com que o computador suspenda a execução de um programa. Quando o programa pára, os valores de todas as variáveis ficam retidos e o programador pode inspecionar ou mudá-los.

O programa pode então ter continuidade, do ponto onde ele havia sido interrompido, com o comando CONT.

Quando a declaração STOP for encontrada pelo computador será visualizada uma mensagem semelhante à seguinte:

```
                    RUN                 I    •
BREAK IN 60
```

onde 60 é o número da linha que contém a declaração STOP.

Se você desejar revisar esta linha, mantenha pressionada a tecla [↑].

HQ [ENTER]

```
                                        •
                                  56. 23
```

AS [ENTER]

```
        "DEDUÇÕES"
```

Quando você estiver pronto para a execução, simplesmente retorne para o "PRONTO" (>) e digite CONT ENTER.

J. CONTROLE DO MODO

LOCK, UNLOCK

A instrução LOCK pode ser usada para controlar o modo (RUN, PRO ou RESERVE) nas quais o computador opera. Incluída dentro de um programa ela previne prejuízos ao programa pela mudança acidental de modo. A instrução LOCK invalida a tecla MODE, "travando" o computador no modo em que está sendo utilizado.

Para habilitar novamente a tecla MODE, é usada a instrução UNLOCK. UNLOCK reabilita o funcionamento normal, permitindo mudanças de modo.

Ambas as instruções podem ser usadas como um comando ou como uma declaração. As formas são simplesmente:

LOCK
UNLOCK

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

VI. EXPANDINDO O PC-1500

A. A IMPRESSORA/INTERFACE CASSETE

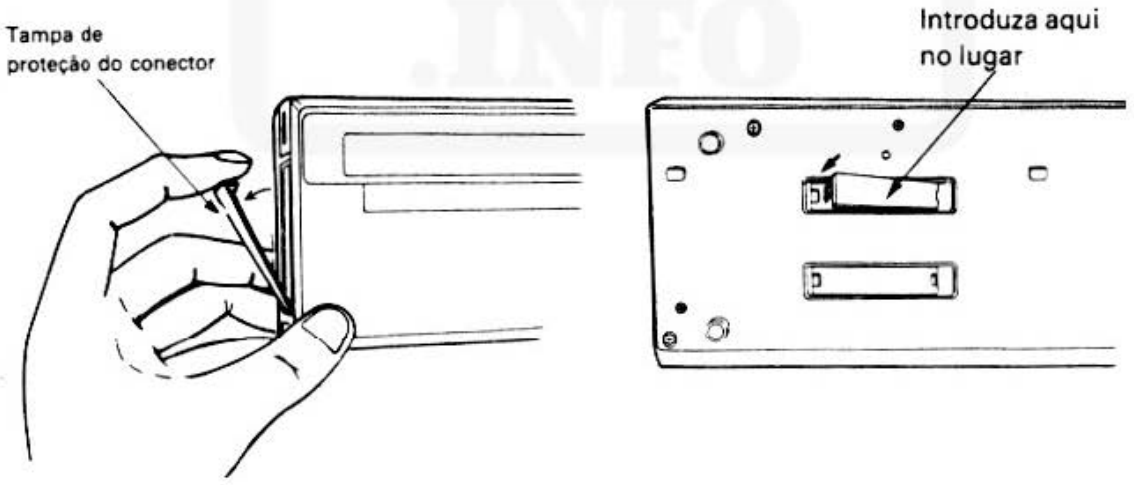
A impressora/interface cassete é um acoplamento para o computador de bolso PC-1500 da Sharp. Esta unidade pode ser conectada a um ou dois gravadores de fita cassete. Os gravadores de fita podem ser utilizados para armazenar programas e dados em fitas comuns de áudio. Programas podem ser "recarregados" no PC-1500, para utilizações posteriores, poupando o trabalho de digitá-los novamente.

1. CONEXÃO DO COMPUTADOR À INTERFACE

Conectar a impressora/interface cassete e o computador (PC-1500) da seguinte maneira:

(1) Desligue o Computador.
Importante! É essencial que o computador esteja desligado. Caso esteja ligado, ele poderá ser "interrompido" (todas as teclas inoperantes). Caso isto ocorra, pressione a chave ALL RESET na parte traseira do computador, pressionando ao mesmo tempo a tecla [ON].

(2) Remova a tampa de proteção do conector do lado esquerdo do computador e introduza-a, na parte traseira da impressora (vide figura).



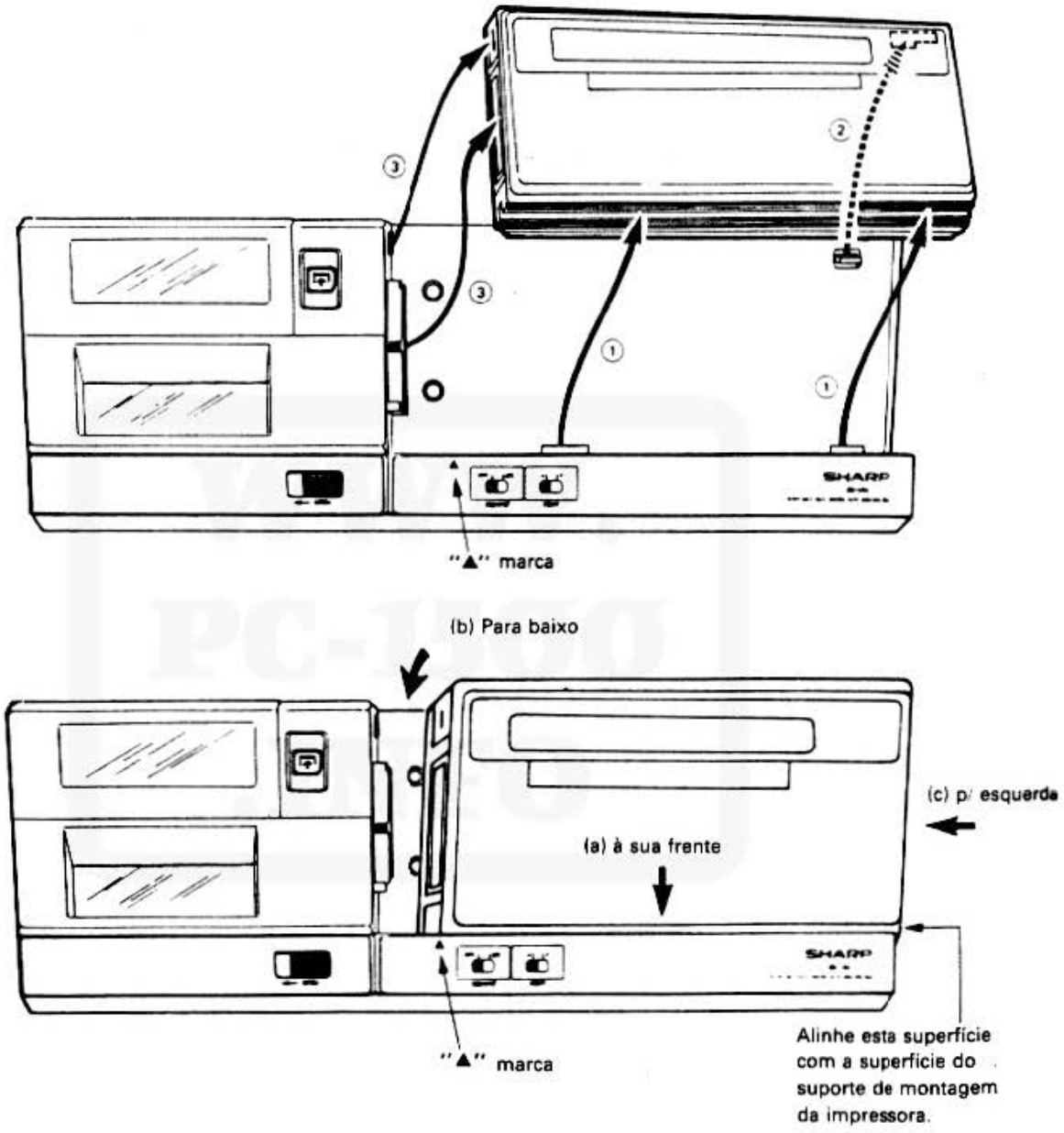
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

(3) Coloque a extremidade inferior do Computador na posição de encaixe de maneira que os guias da impressora coincidam com as fendas-guias do computador.

(4) Coloque o Computador horizontalmente.

(5 Deslize suavemente o Computador para a esquerda de maneira que os conectores da Impressora encaixem no Computador (vide figura).



Não force o computador na impressora. Se não encaixar facilmente, mova o computador cuidadosamente para a esquerda e direita para ajustar corretamente as superfícies de acoplamento.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

2. FONTE DE ALIMENTAÇÃO

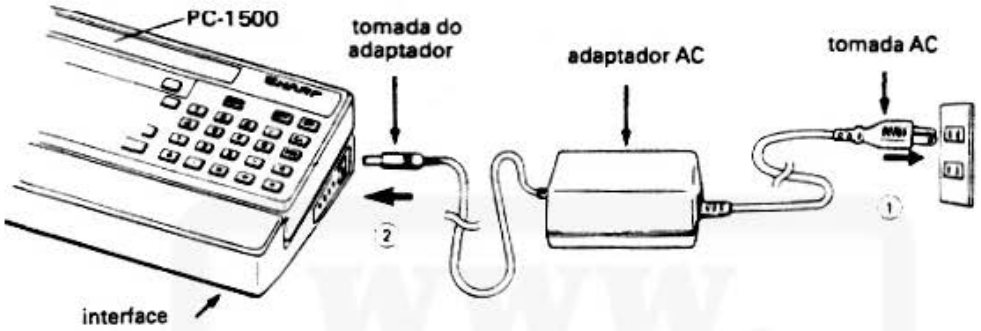
A unidade da interface utiliza como fonte de alimentação, uma bateria recarregável de Ni-Cad. Portanto, é necessário recarregar as baterias após desembalá-la, ou quando a seguinte mensagem for visualizada. (Neste caso, a impressora é travada. Para destravá-la, pressione as teclas [OFF] e [ON] do computador nesta ordem após carregar a bateria).

(1) E R R O R 8Ø ou E R R O R 7 8

NOTA: Quando a impressora estiver em modo de substituição da caneta, o visor pode mostrar ERROR 78.

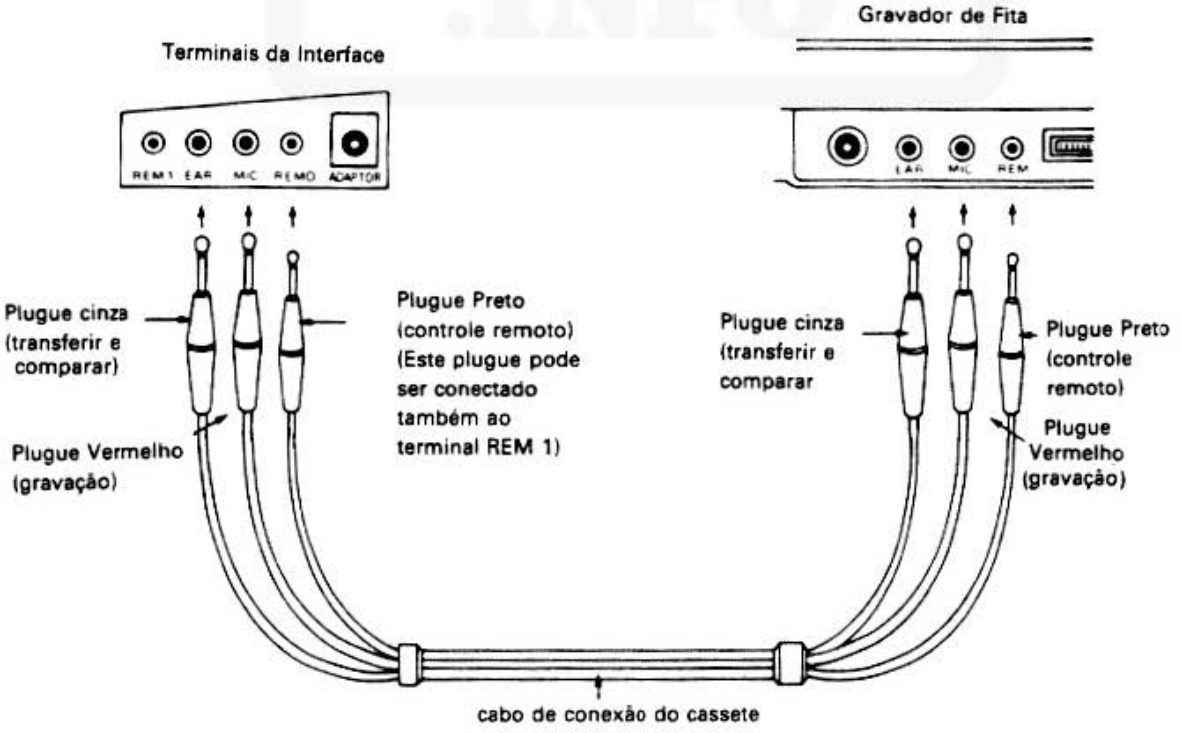
(2) : C H E C K 6 ou N E W Ø ? : C H E C K 6

("Como conectar o adaptador AC")



3. CONECTANDO UM GRAVADOR DE FITA CASSETE NA INTERFACE

Inicialmente, conecte a interface ao computador. Após isso, conecte um gravador de fita cassete à unidade de interface conforme mostramos na ilustração.



Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

A tabela abaixo é uma descrição das especificações mínimas necessárias para acoplar um gravador de fita cassete à impressora/Interface Cassete.

ITEM	REQUISITOS
1. Tipo de gravador	Qualquer gravador de fita cassete, microcassete, ou gravador de fita magnética pode ser utilizado, desde que possua os requisitos descritos abaixo.
2. Tomada de entrada	O gravador deve ter uma mini-tomada chamada "MIC". Nunca use a entrada "AUX".
3. Impedância de entrada	A tomada de entrada deve ser de impedância baixa (200 ~ 1,000 0hm).
4. Nível mín. de entrada	Inferior à 3mV ou -50dB.
5. Tomada de saída	Deve ser uma mini-tomada chamada "EXT. (alto falante EXTerno)", "MONITOR", "EAR" ou equivalente.
6. Impedância de saída	Deve ser inferior à 10 OHM.
7. Nível de saída	Deve ser superior à 1V (saída máxima prática superior à 100mW).
8. Distorção	Deve estar na faixa de 15% no limite de 2KHZ à 4KHZ.
9. Wow e Flutter	0,3% no máximo (W.R.M.S)
10. Outros	O motor do gravador não deve sofrer variações de velocidade.

* No caso do mini-plugue fornecido com o PC-1500RP não ser compatível com as tomadas de entrada/saída do seu gravador de fita, você poderá encontrar plugues especiais de conversão de linha nas lojas do ramo.

NOTA:

- Alguns dos gravadores de fita podem rejeitar a conexão, devido possuirem especificações diferentes. Ou se os mesmos possuirem distorção, ruído excessivo, ou deterioração da potência devido muitos anos de uso.

- Instruções preventivas para o uso do gravador de fita.
  (1) Para qualquer transferência ou comparação, utilize o mesmo gravador de fita que foi usado para a gravação. Se o gravador for diferente daquele usado para a gravação, a transferência ou a comparação talvez não sejam possíveis.
  (2) Se o cabeçote do gravador de fita estiver sujo, aumenta a distorção ou diminui o nível de gravação. Portanto, mantenha-o sempre limpo.
  (3) Utilize fitas livres de resposta de freqüência extremamente baixas, arranhões e dobras.

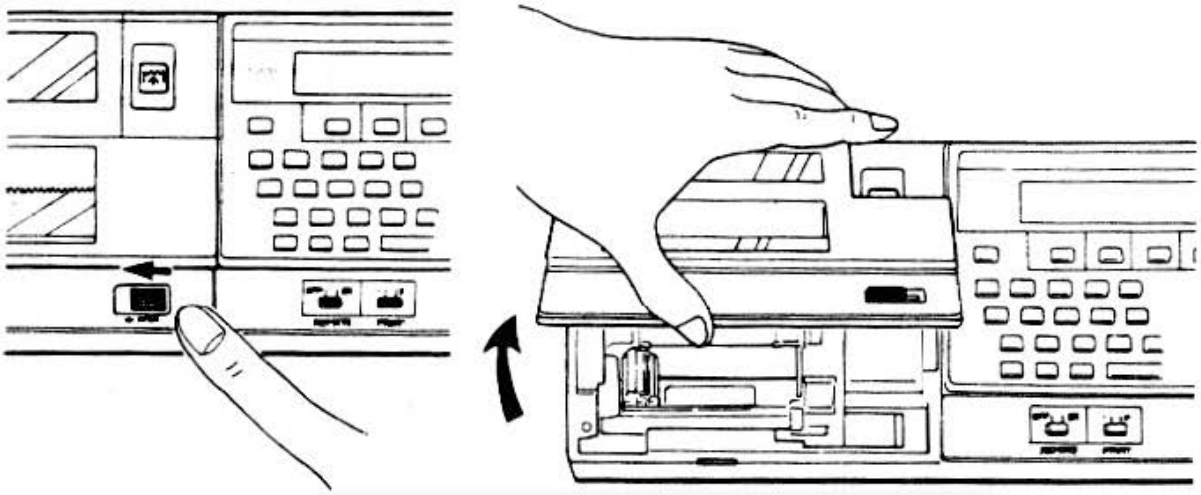
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

4. COLOCANDO O PAPEL

Para maiores detalhes, consulte o manual de instruções da impressora/ interface cassete.

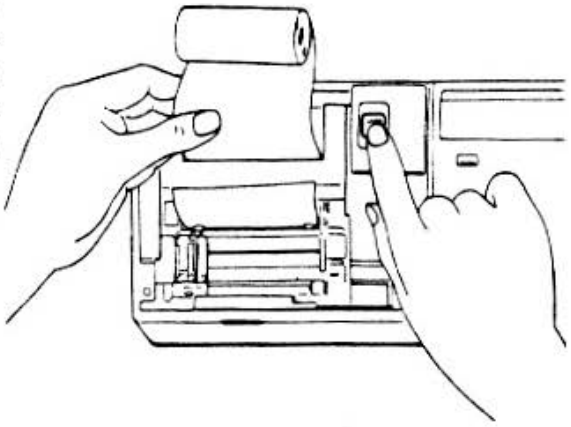
(1) Para remover a tampa da impressora, deslize a alavanca de trava da tampa, na direção da seta.



(2) Corte a ponta do papel retilineamente e insira-o corretamente na entrada apropriada. (Qualquer curva ou dobra próxima ao começo do papel dificultará a inserção do mesmo).

entrada do papel

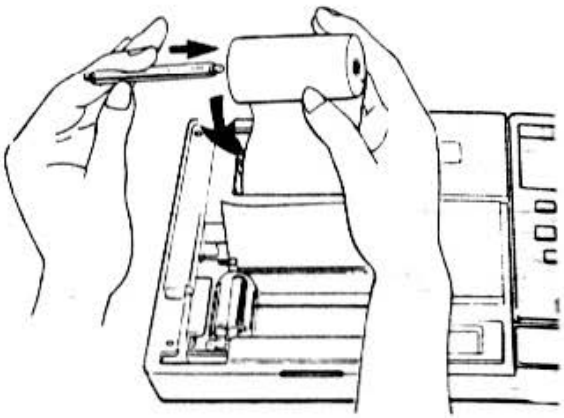
(3) Pressione a tecla [ON] do computador para ligá-lo, e pressione a tecla [↑] para a alimentação do papel. Neste momento, alimente-o de tal maneira que a ponta do mesmo saia para fora da impressora de 3 à 5 cm.



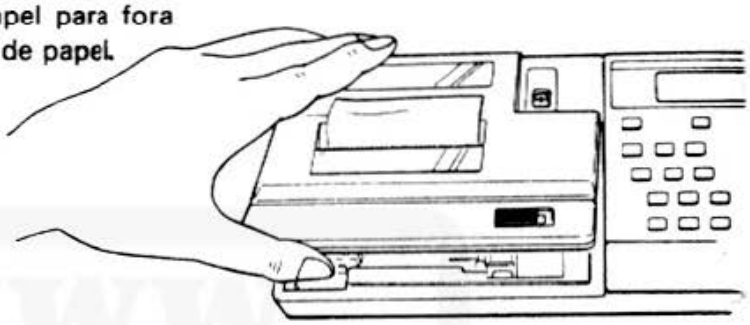
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

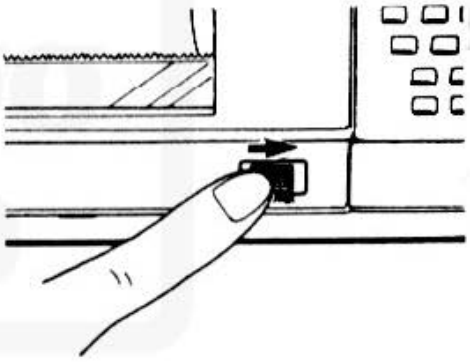
(4) Insira o eixo dentro da bobina de papel e coloque-o no compartimeno apropriado.



(5) Recoloque a tampa da impressora na posição, passando a ponta do papel para fora da tampa através do cortador de papel.

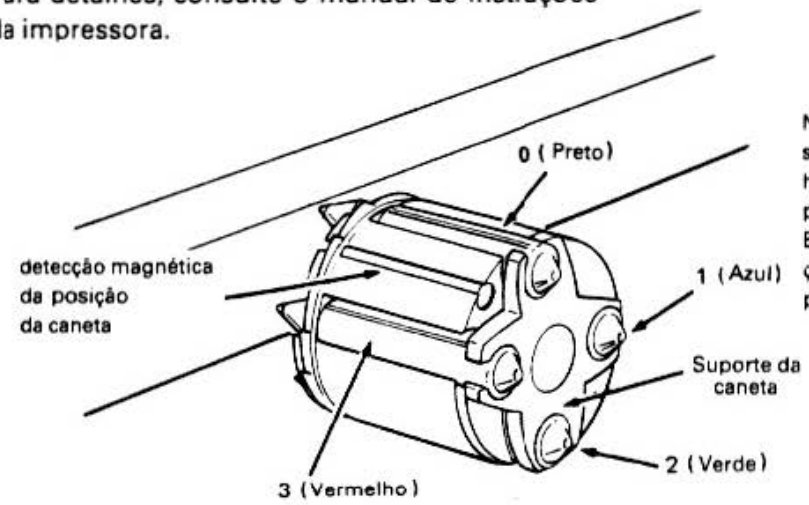


(6) Trave a tampa da impressora.



5. SUBSTITUINDO AS CANETAS

Quatro cores de canetas podem ser instaladas nesta unidade. As posições de instalação das canetas são conforme ilustrado abaixo:
Para detalhes, consulte o manual de instruções da impressora.



No desenho, as saliências das canetas são numeradas (0,1,2 e 3) no sentido horário do magneto detector de posição.
Estes números correspondem às posições adequadas quando selecionadas pelo comando de cor.

Do not sell this PDF !!!

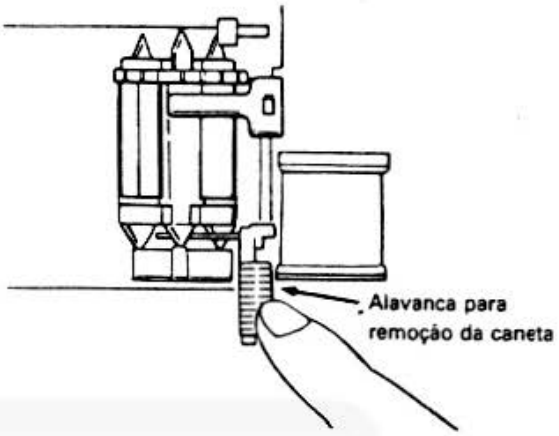
All and more about Sharp PC-1500 at http://www.PC-1500.info

Para a instalação ou substituição das canetas, siga as instruções abaixo:

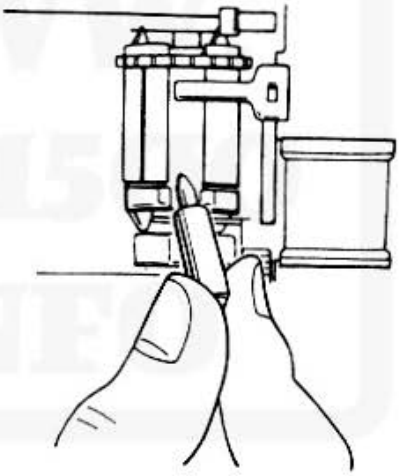
(1) Com a tecla [Ø] do computador pressionada, pressione a tecla [↑] da impressora. Isto faz com que a impressora coloque-se no modo de substituição de caneta, quando o suporte da caneta se mover para a esquerda e girar. (Solte a tecla quando o suporte da caneta começar a se mover).

(2) Para remover a caneta, pressione a alavanca de remoção. Isto fará com que a que está no topo seja solta.

(NOTA): Quando estiver removendo a caneta, segure-a suavemente para evitar que ela caia dentro da impressora.



(3) Instale uma nova caneta.



(4) Para instalar ou remover a próxima caneta, pressione a tecla [↑]. O suporte da caneta retorna para a esquerda, e gira para que a próxima caneta venha para cima, e desloca-se para a direita novamente. Remova a caneta e substitua-a por uma nova conforme o passo (2) e (3).

(5) Após a substituição ou instalação das canetas, pressione a tecla [↑] da impressora com a tecla [CL] do computador pressionada. Isto fará com que a impressora seja liberada do modo de substituição, e a caneta volte para a esquerda.

(NOTA): Para usar esta unidade, instale as quatro canetas no suporte. Operações com a falta de pelo menos uma das canetas pode causar falhas nas mudanças de cor.

MANUSEANDO AS CANETAS:

As canetas são instaladas na impressora, quando as mesmas forem utilizadas e removidas após o uso. Tampe-as e coloque-as no tubo apropriado para guardar.
Deixando as canetas instaladas na impressora por um longo período de tempo, ou expostas a tinta poderá secar.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

B. USANDO UM GRAVADOR DE FITA CASSETE

1. Operação do Gravador

Recomendamos que você siga este procedimento usando um programa simples e pequeno. Será mais fácil reefetuar a operação no caso de ocorrer problemas. Agora, introduza o programa.

Você deve preparar o gravador para uma transferência de programas e dados. Para fazer isto, efetue os seguintes passos:

1. Desligue a chave "remote" na interface.
2. Coloque a fita no gravador (um passo importante).
3. Ache uma parte vazia da fita. Se a fita for nova, avance-a até passar a parte guia. Se o seu gravador possui um contador, marque o número de voltas. Isto é muito útil para localizar o programa que você está guardando.
4. Se o seu gravador possui um controle de volume automático, posicione no automático. Se ele for de controle manual, coloque o nível do volume entre o meio e o máximo (aproximadamente 3/4)
5. Ligue a chave "remote" da interface novamente. Se o seu gravador não tiver a característica "remote" (isto significa que não há lugar para conectar um dos cabos) utilize a tecla PAUSE para parar a gravação temporariamente.
   Se o seu gravador também não tiver a tecla PAUSE, nós recomendamos que você compre um que tenha esta função. Melhor ainda se este tiver os dois, a característica remote, e a chave PAUSE.
6. Pressione as teclas RECORD e PLAY simultaneamente.
   Se você está usando um gravador sem a chave "PAUSE", você deve digitar o comando de gravação no computador, e somente digitar [ENTER] após pressionar as teclas de gravação.

2. O comando CSAVE

Tudo preparado? Pois então façamos uma verificação.

1. Gravador preparado e esperando? sim_____ não_____
2. A fita está no gravador? sim_____ não_____

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

3. O computador está ligado? sim_____ não_____
4. O programa está no computador? sim_____ não_____
5. Fios, todos colocados corretamente? sim_____ não_____
6. O contador está posicionando em zero? sim_____ não_____
7. Você está vestido? irrelevante_____

Se você respondeu "Não" para qualquer destas questões, leia novamente a secção anterior para sanar o seu problema.

Com o mesmo comando que guarda o seu programa, você deve dar um "nome para arquivo" do programa. Isto é para ser utilizado como referência. O nome para arquivo não pode possuir mais do que 16 caracteres. Para guardar o programa com o nome, digite:

CSAVE [SHIFT] " PROG.-1 [SHIFT] "

O programa será guardado com o nome "PROG.-1".

É possível atribuir qualquer nome que se deseje. Observe também que existe um limite de 16 caracteres para este nome. Se o nome possuir mais 16 caracteres, a parte excedente é ignorada. Uma prática desejável é manter uma anotação do programa que inclua o seu nome, o ponto de partida e parada da fita (use os números do contador) e uma breve descrição do que o programa faz.

Pressione a tecla [ENTER] . Neste momento, você deve ouvir um som agudo, e a fita deve estar girando. Também o indicador "BUSY" deve estar aceso. Isto indica que o computador está transferindo o programa da memória para a fita. Se isto não acontecer, comece novamente desde o início da secção.

Quando o computador chegar ao fim do programa, o indicador "BUSY" apagará, o gravador parará, e o "prompt" reaparecerá no visor. Congratulações, seu primeiro programa foi guardado para uso futuro. A fim de assegurar que isto foi conseguido, podemos ler outra vez o programa da fita, conforme explicamos na próxima secção

3. O comando CLOAD

Agora que o seu primeiro programa está guardado na fita, sem dúvida você irá querer ver se realmente ele está lá.

Para fazer isto, é realmente simples; use o comando CLOAD?

"O que ele faz?" Bem, após guardar o seu programa, simplesmente digite CLOAD?. O computador compara o programa guardado com um outro que está na sua memória. Se tudo foi bem, ele visualizará o seu nome de arquivo e termina a verificação. Caso contrário, uma mensagem de erro será visualizada, geralmente "ERROR 43". Este aponta que o programa da fita está de certa forma diferente do programa que está na memória do computador. Apague esta parte da fita e comece novamente. Verifique todas as suas conexões, e tente colocar o volume e a tonalidade um pouco mais altos.

As instruções seguintes mostram como recarregar um programa para o PC-1500.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

1. Desligue a chave "remote" da inteface.

2. Volte a fita para o início do programa, usando novamente o contador de voltas. (Veja como o contador é útil).

3. Pare a fita.

4. Ligue a chave "remote".

5. Pressione a tecla PLAY.

6. Digite:

   CLOAD [SHIFT] "PROG.-1 [SHIFT] "

   e pressione a tecla [ENTER]

(Lembre-se que "PROG.-1" é o nome do arquivo que nós demos ao seu programa. Se você guardou o programa com um outro nome, você deve usar aquele nome no lugar de "PROG.--1".)

7. O indicador "BUSY" acende e o programa será "trazido de volta" para a memória do computador.

8. Digite RUN e o programa será executado. A fita retém uma cópia do programa. Desta forma você pode carregar o mesmo programa várias vezes!
   Durante o carregamento, quando uma mensagem de erro "ERROR 43" ou "ERROR 44" for visualizada recomece no passo (1) acima.

4. Comandos PRINT # e INPUT

PRINT #:

Agora que nós ilustramos o uso de CSAVE e CLOAD, nós gostaríamos de apresentar dois comandos similares. O comando PRINT # guarda o valor de uma variável ou posição das variáveis na fita. Isto é diferente do CSAVE que guarda um programa. Por exemplo, no programa abaixo, a variável T$ está em uso:

```
10 : PRINT "QUAL É O SEU NOME?"
20 : INPUT T$
30 : PRINT T$
```

Se você quiser guardar o valor de T$ para usar em outro programa, você deve emitir o comando PRINT # para guardá-lo na fita. Você pode fazer isto de duas maneiras:

1. Manualmente
2. Através de um programa

NOTA: Esta operação requer o uso do gravador, então prepare-o para receber dados.

1. **O Método Manual**
   O método manual oferece várias opções:

   OPÇÃO 1:
   Após rodar um programa, posicione no modo RUN e digite:

   [P] [R] [I] [N] [T] [SHIFT] [#] A, B, C (pressione [ENTER])

   O gravador entrará em ação e guardará o valor de todas as variáveis na fita.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

OPÇÃO 2:

Se você quiser identificar somente algumas variáveis a serem guardadas, digite o seguinte:

[P] [R] [I] [N] [T] [SHIFT] [#] "nome de arquivo"; A, B, C

Com isso, você acaba de especificar as variáveis A, B e C a serem guardadas na fita sob o nome de arquivo dado.

OPÇÃO 3:

Você pode especificar também, todos os valores das variáveis relatadas, digitando:

[P] [R] [I] [N] [T] [SHIFT] [#] "nome de arquivo"; B (*)

O símbolo * guardará todas as variedades de "B", incluindo B (1) e o próprio B.

2. Através de um Programa

Para fazer isto, simplesmente atribua um número de linha para o comando PRINT # no seu programa. Você pode usar qualquer um dos formatos acima descritos. Quando o computador encontrar este número de linha, automaticamente ativará o gravador de fita e começará a transferir os dados para a fita. Experimente mais uma vez. Se não funcionar, volte para a seção anterior, pois algo deve ter ficado omitido.

NOTA:

O nome de arquivo representa os conteúdos de um cordão de caracteres ou uma variável de caracter colocada entre aspas.

Assim, se a primeira variável a ser gravada ou armazenada for colocada sem um nome de arquivo, esta variável de caracter será considerada como um nome de arquivo. Portanto, precavenha-se em colocar um nome de arquivo.

Exemplo 1: 10 A$ = "FITA"
20 PRINT # A$; X$
Isto é idêntico ao PRINT # "FITA"; X$

Exemplo 2: PRINT # X
Este grava somente os conteúdos da variável numérica X, sem um nome de arquivo.

Exemplo 3: PRINT # X$
Este não grava somente os conteúdos do cordão de caracteres X$ sem um nome de arquivo, mas o conteúdo de X$ é considerado como um nome de arquivo. Se não foi colocado o ponto e vírgula(;) após o nome de arquivo, causará um erro gramatical (ERROR 1).
Assim, se a primeira variável for uma variável de caracter, coloque um nome de arquivo como: PRINT # "AAA" X$.

INPUT #

Este comando permite os mesmos formatos que PRINT #. A diferença é que INPUT # transfere dados da fita para o computador. (PRINT # transfere dados do computador para a fita). Você pode usar o compando INPUT # manualmente ou como parte de um programa. Lembre-se de preparar o gravador de fita antes de começar a usar este comando.

NOTA:

- Se o número de dados gravados for menor do que o de memória de dados utilizadas, ocorre um "ERROR 43 ", depois que todos os dados são transferidos pela declaração INPUT #.
- Se o número de dados gravados for maior do que o de memórias de dados utilizadas, os dados são transferidos até que as memórias estejam carregadas. O restante será ignorado.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

5. Uso do Comando MERGE

O comando MERGE permite armazenar, ao mesmo tempo, alguns programas na memória do computador. Por exemplo, vamos supor que a memória do computador contém o seguinte programa:

```
10 : PRINT "DEPRECIAÇÃO DA PENSÃO"
20 : INPUT "INTRODUZA MÉTODO:"; A
```

Neste momento, subitamente você lembra que pode ter uma parte de um programa similar na fita com o nome de arquivo "DEP 1". Evidentemente, você vai querer ver se o programa tem alguma parte útil para o programa que você está construindo agora. O primeiro passo é encontrar a fita que contém DEP 1. Posicione a fita no início de "DEP 1".

Agora digite: [M] [E] [R] [G] [E] [SHIFT] ["] [D] [E] [P] [1] [SHIFT] ["] e pressione [ENTER].

Agora, o computador carregará "DEP 1" na memória, em adição ao programa acima. Após "DEP 1" ser carregado, você poderá encontrar alguma coisa na memória, similar a isto:

```
10: PRINT "DEPRECIAÇÃO DA PENSÃO"
20: INPUT "INTRODUZA MÉTODO"; A
10: "DEP 1" REM >> SEGUNDO MÉTODO <<
20: PRINT "VARIAÇÃO DE JUROS"
30: INPUT "QUANTIDADE APLICADA:"; B
  :
  :
(etc)
```

Observe que, diferente do comando CLOAD, o novo programa NÃO troca o que já existe e que, alguns números de linhas são duplicadas. Note também que foi usado um "rótulo" na primeira linha do módulo intercalado. Isto permite "Ligação" de módulos (veja ligação de módulos intercalados – abaixo).

É importante que você leia a seguinte informação antes de prosseguir com qualquer edição ou programação.

NOTAS IMPORTANTES:

Sendo efetuada uma INTERCALAÇÃO (MERGE), não são permitidas INSERÇÕES, CANCELAMENTOS ou SUBSTITUIÇÕES nas linhas de programa previamente existentes.
Exemplo:

```
10 "A" REM "Este é o PROGRAMA existente"
20 FOR T = 1 TO 100
30 LPRINT T
40 NEXT T
  :
  :
(etc)
```

Antes de fazer uma INTERCALAÇÃO (MERGE) no próximo programa, faça as substituições necessárias neste programa.
Então, INTERCALE o próximo programa: MERGE "PROG.2"
Exemplo:

```
10 "B" REM "Este é programa INTERCALADO"
20 INPUT "Introduza DEPRECIAÇÃO:      "; D
30 INPUT "Número de ANOS:             "; Y
40 Etc
```

Agora você pode efetuar as mudanças no programa acima desde que ele seja a última parte INTERCALADA. Caso você necessite efetuar mudanças no primeiro programa, então siga este procedimento:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

1. CSAVE o que você fez até este ponto; use um nome novo quando gravar na fita, isto é, "PROG 3" (Exemplo).
2. CLOAD o programa gravado no passo anterior na memória.
3. Agora, faça qualquer das modificações ou substituições desejadas. Entretanto, guarde isto em mente: mudanças podem ser feitas SOMENTE no primeiro programa ou na PRIMEIRA OCORRÊNCIA de qualquer número de linha duplicada. Isto é, se a linha do programa 10 aparecer tanto no primeiro programa quanto no segundo, as mudanças afetarão somente a primeira ocorrência da linha 10. Então se você tentar editar a segunda ocorrência da linha 10, a mudança será refletida errôneamente, no primeiro programa.

ACRESCENTANDO AS LINHAS DE PROGRAMAS ADICIONAIS:

As linhas adicionais podem ser acrescentadas no fim do programas existentes, SOMENTE SE ELES TIVEREM NÚMEROS DE LINHA MAIORES DO QUE AQUELES USADOS ANTERIORMENTE.

Observe que as linhas adicionais de programa não precisam necessariamente aparecer no "fim" da listagem.

Visto que os módulos devem ser ligados por rótulos (veja abaixo) isto não deve causar preocupação.

LIGAÇÃO DE MÓDULOS INTERCALADOS (programas) JUNTOS

Devido ao processador executar as linhas de um programa em seqüência lógica, ele irá parar quando encontrar uma interrupção na seqüência de numeração das linhas, isto é, se os números de linha 10, 20, 30 são seguidos por números de linhas duplicadas num segundo módulo, o processador cessará a execução após a linha 30 no primeiro módulo. Para "LIGAR" seus vários módulos, as seguintes técnicas são válidas: GOTO "B", GOSUB "B", IF...THEN "B"(B é usado somente para exemplo, você pode usar qualquer rótulo exceto as palavras reservadas ou as letras que aparecem na fileira # 2 do teclado, isto é, de Q à P).

NOTA:

Quando um outro programa é chamado como uma sub-rotina, através da declaração GOSUB, dentre vários programas intercalados, programe uma declaração GOTO (um número de linha maior do que o número da linha a ser executado) após a declaração GOSUB.

Exemplo:
```
10 "A" : INPUT A
20 GOSUB "B" : GOTO 30
30 GOTO 10
10 "B" PRINT A
20 RETURN
```

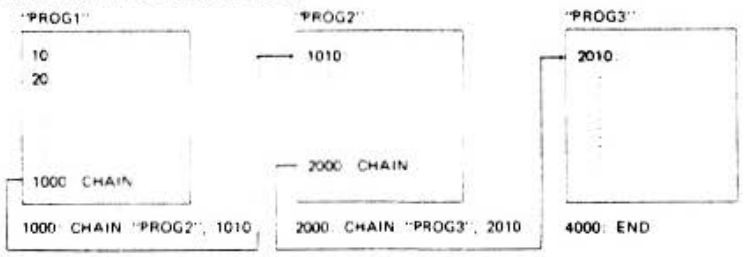
6. A Declaração CHAIN

CHAIN é uma instrução de programa; ela pode ser usada somente dentro de um programa. Ela não pode ser usada manualmente como CSAVE CLOAD e MERGE.

A declaração CHAIN permite a você rodar um programa muito grande, fazendo com que caiba na memória, todo ao mesmo tempo. Estes programas "grandes" devem ser divididos em partes com uma declaração CHAIN no fim de cada parte. Estas partes podem ser guardadas na fita.

Por exemplo, vamos supor que você tenha três partes do programa, chamadas PROG 1, PROG 2, PROG 3. Cada uma destas seções terminam com uma declaração CHAIN.

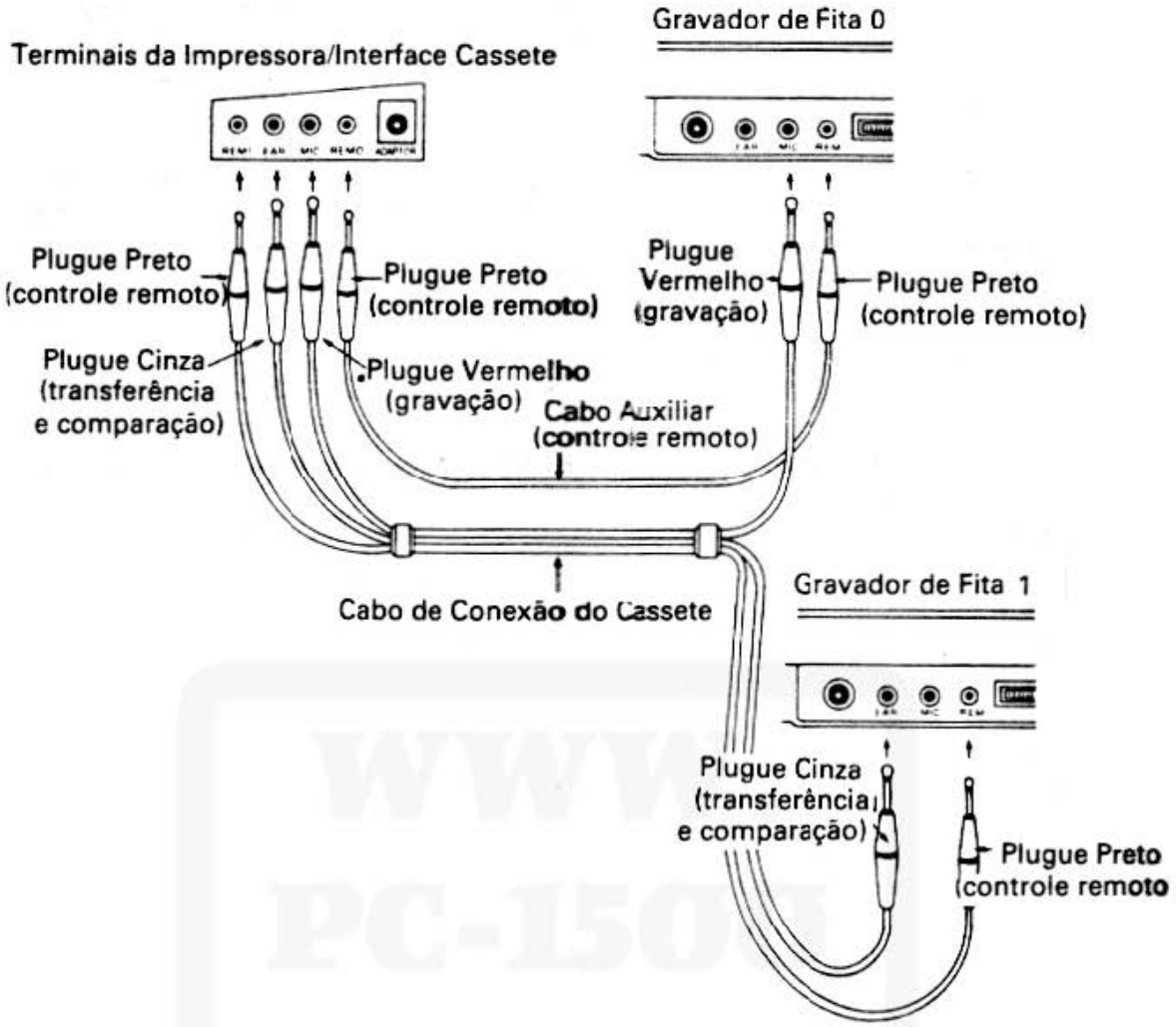
Durante a execução, quando o computador encontrar a declaração CHAIN, a próxima seção é chamada da memória e executada.



Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

7. Usando Dois Gravadores de Fita



Quando utiliza-se dois gravadores de fita, um deles é usado para gravação e o outro para transferência. Conforme ilustrado, a Impressora/Interface Cassete e os dois gravadores estão conectados através dos cabos de conexão e o cabo auxiliar para controle remoto.

- A Impressora/Interface Cassete está equipada com dois terminais de controle remoto REM 0 e REM 1, qualquer um dos dois pode ser usado. Na operação de programa (sem ser manual), entretanto, o programa irá designar o gravador conectado para o terminal REM 0 ou REM 1. Logo, estes terminais de controle remoto devem ser conectados de acordo com o programa.

Nesta seção, é explicado como operar o segundo gravador conectado no terminal REM 1 da interface.

1. Procedimentos para Guardar:

(1) Digite RMT OFF e pressione [ENTER] para desligar a função do segundo controle remoto. (controle do gravador 1 na ilustração acima).

(2) Coloque a fita no gravador.

(3) Digite RMT ON e pressione [ENTER] para posicionar a função do segundo controle remoto.

(4) Posicione os controles de volume e tonalidade da mesma forma como foi explicado anteriormente, para um único gravador.

(5) Pressione as teclas RECORD e PLAY simultaneamente.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

(6) Execute a instrução RECORD.
Programa: CSAVE – 1 "nome de arquivo" [ENTER]
Dados: PRINT # – 1 "nome de arquivo"; variável, variável...
(exemplo) Designe os modos PRO ou RUN
CSAVE – 1 "PR – 1" [ENTER]

Após guardar o programa, o "PRONTO" reaparece no visor e a fita pára. Volte a fita para verificação.

2. Comparação entre os conteúdos da Fita e do Computador
Procedimentos:

(1) Digite RMT OFF [ENTER] para limpar as funções do controle remoto.

(2) Volte a fita para o início do programa, novamente usando o contador de voltas.

(3) Introduza RMT ON e a tecla [ENTER] para posicionar as funções do controle remoto.

(4) Posicione os controles de volume e tonalidade da mesma forma como foi explicado anteriormente.

(5) Pressione a tecla de reprodução

(6) Execute a instrução COLLATION (comparação)
CLOAD ? – 1 "nome de arquivo" [ENTER]
(Exemplo) Designe o modo PRO ou RUN
CLOAD ? – 1 "PR-1" [ENTER]

A execução termina quando ambos conteúdos se igualam, resultando na visualização de "PRONTO".

3. Transferência da Fita
Procedimentos:

(1) Introduza RMT OFF e a tecla [ENTER] para desativar as funções de controle remoto.

(2) Coloque uma fita gravada no gravador.

(3) Digite RMT ON e pressione o [ENTER] para ativar as funções de controle remoto.

(4) Posicione os controles de volume e tonalidade da mesma forma como foi explicado anteriormente.

(5) Pressione a tecla de reprodução.

(6) Execute a instrução TRANSFER (transferência)
Programa: CLOAD – 1 "nome de arquivo" [ENTER]
Dados: INPUT # – 1 "nome de arquivo"; variável, variável,... [ENTER]
(EXEMPLO) Designe o modo PRO ou RUN
CLOAD – 1 "PR-1" [ENTER]

Após a transferência, aparecerá "PRONTO" no visor.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C. USANDO A IMPRESSORA

NOTA: Supõe-se que você já tenha providenciado os itens abaixo:

1) Conexão apropriada entre o Computador PC-1500 e a interface/Impressora Cassete.
2) Baterias da interface devidamente carregadas, mediante o uso do adaptador AC, EA-150 da Sharp.
3) As canetas colocadas na impressora.
4) Papel na impressora.

Se você não tomou estas providências, volte para a secção A deste capítulo para as instruções.

C.1. ESPECIFICAÇÕES DA IMPRESSORA

Caracteres/linha: 4, 5, 6, 7, 9, 12, 18 ou 36 dependendo do tamanho de caracter escolhido.

Tamanho do Caracter: 1.2 × 0.8 mm a 10.8 × 7.2 mm

Velocidade de Impressão: Máximo 11 caracteres por segundo, quando imprimindo os caracteres menores.

Rotação: Os caracteres podem ser impressos em ambas as direções em qualquer um dos dois eixos.

Cores: 4 – vermelho, azul, verde, preto.

Sitema gráfico: X – Y

Alimentação de Papel: Manual ou Programável.

C.2. O comando TEST

A primeira coisa que você irá querer fazer, é testar o funcionamento da interface/impressora cassete. Com o computador e interface ligados, digite:

TEST (e pressione ENTER)

A impressora desenhará agora 4 caixas, cada uma de cor diferente. A cor das caixas, da esquerda para a direita, será de acordo com o que você escolheu como cores de 0 à 3 quando inseriu as canetas.

Do not sell this PDF !!!

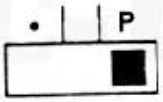
All and more about Sharp PC-1500 at http://www.PC-1500.info

C.3. Impressão de Cálculos

Usando a Interface Impressora/Cassete, é possível fazer uma cópia impressa da série de cálculos manuais efetuados no Computador PC-1500, como na figura 1:

```
12000*.065
                    780

780+25.56
                  805.56

SIN 30
                    0.5

TX=.065
                  0.065

P=20000*TX
                   1300

P/12
           108.3333333
```

Para fazer isto, simplesmente posicione a chave de impressão (PRINT) da interface para a posição "P".



Para evitar a impressão automática de cálculos manuais, a chave de impressão deve estar na posição "•". Com a chave nesta posição você ainda pode imprimir resultados selecionados prefaciando a computação com um comando LPRINT. (veja LPRINT). Outros comandos que causam impressão ou figuras, tais como: LLIST, LINE e OUTROS, são também funcionais neste modo.

Quando estiver imprimindo, a cor utilizada será a mesma que foi especificada anteriormente. Se você acaba de ligar o computador, esta será de cor correspondente à cor 0. Para mudar a cor, você deve emitir o comando COLOR (veja a secção apropriada).

O tamanho do caracter usado para imprimir cálculos manuais, depende de especificação prévia. Se o tamanho do caracter especificado for de tamanho 1 ou 2, este permanece. Se o tamanho anterior for maior que 2, será usado o tamanho 2.

A impressão automática faz com que o modo da impressora seja fixado em TEXT. Se você estava no modo GRAPH e deseja retornar para ele, emita o comando GRAPH.

(Os modos da impressora são explicados na próxima seção).

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.4. Modos da Impressora

Com a impressora em operação, ela pode estar em um dos dois modos; TEXT ou GRAPH.

Devido a maioria dos comandos funcionarem somente em um dos modos, é importante selecioná-lo apropriadamente antes de emitir as instruções.

O modo TEXT é usado para imprimir números e caracteres.

A largura do papel da impressora é dividida em colunas.

O número de colunas está relacionado ao tamanho do caracter especificado. Comandos de tabulação vertical e horizontal são fornecidos para a informação do formato do texto.

No modo GRAPH, uma variedade de figuras, tabelas e gráficos podem ser criados. Existem comandos para desenhar linhas sólidas e tracejadas, utilizando um sistema de coordenadas diretas ou relativas. Todos os desenhos utilizam um esquema normal de coordenadas X-Y.

Para especificar o modo TEXT, a declaração:

TEXT

é suficiente. Certos comandos (discutidos mais tarde) causam um chaveamento automático para o modo TEXT.

Especificar o modo GRAPH é igualmente simples. A declaração:

GRAPH

iniciará este modo, posicionando a caneta para o lado esquerdo do papel.

C.5. Listagem de Programas

O comando LLIST causa a impressão total ou parcial do programa. O Comando LLIST é extremamente útil durante o processo de desenvolvimento do programa, porque é possível imprimir partes selecionadas do mesmo.

A forma do comando LLIST é similar à forma do comando LIST. Como o LLIST é mais versátil, tem o subtítulo diferente. As formas do LLIST são as seguintes:

LLIST

- Imprime todas as linhas do programa existente na memória de programa.

LLIST expressão

- Imprime somente a linha do programa especificada pela expressão.

LLIST, expressão

- Imprime todas as linhas do programa até e inclusive a linha cujo o número é dado pela expressão.

LLIST expressão,

- Imprime as linhas do programa começando com a linha cujo número é dado pela expressão.

LLIST expressão 1, expressão 2

- Imprime as linhas do programa, começando com a linha cujo número é dado pela expressão 1 e termina com a linha cujo número é dado pela expressão 2. Assim, se o comando é:

  LLIST 100, 150

  então todas as linhas entre 100 e 150 (se existirem) serão listadas.

LLIST "label"

- Imprime a linha do programa contendo o rótulo dado.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

LLIST "label",

- Imprime as linhas do programa, iniciando com a linha que contém o rótulo dado, continuando até o final.

  NOTA: Especificação de um rótulo que não existe será assinalada por uma mensagem de "ERROR 11".

O comando LLIST faz com que a impressora seja colocada no modo TEXT. Se você estava no modo GRAPH e deseja retornar a ele, emita o comando GRAPH.

Enquanto estiver listando um programa, o Computador PC-1500 tenta ajustar as linhas do programa para uma melhor leitura. Isto é feito deixando espaço nas linhas numeradas. Os números de linha que têm de 1 a 3 dígitos serão ajustados à direita dentro de um campo de 3 caracteres.

Os números de linha que tem 4 à 5 dígitos serão impressos em um campo de 5 caracteres.

```
  10 : REM LARGURA 3
  20 : REM LARGURA 3
 300 : REM LARGURA 3
2001 : REM LARGURA 5
2010 : REM LARGURA 5
```

WWW.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.6. Controle Programável da Impressora

C.6.1. CSIZE

O comando CSIZE especifica o tamanho dos caracteres para toda impressão subseqüente. Existem 9 tamanhos disponíveis, que vão de 36 à 4 caracteres por linha impressa.
A forma do comando CSIZE é:

CSIZE expressão
(qualquer modo)

A expressão deve conter um número dentro de um limite de 1 à 9. A largura e a altura dos caracteres para cada tamanho é dado na seguinte tabela:

TABELA 1:

CSIZE	1	2	3	4	5	6	7	8	9
Caracteres por linha impressa	36	18	12	9	7	6	5	4	4
Altura de cada caracter (mm)	1.2	2.4	3.6	4.8	6.0	7.2	8.4	9.6	10.8
Largura de cada caracter (mm)	0.8	1.6	2.4	3.2	4.0	4.8	5.6	6.4	7.2

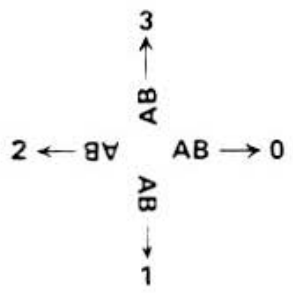
C.6.2. ROTATE

O comando ROTATE é usado somente no modo GRAPH, para especificar a direção na qual a impressão ocorre. São possíveis quatro direções; para cima, para baixo da esquerda para a direita, e da direita para a esquerda (com a letra virada para baixo). As direções são ilustradas na figura 1. A forma do comando ROTATE é:

ROTATE expressão
(somente modo GRAPH)

A expressão deve possuir um número dentro de um limite de 0 à 3. ROTATE 0 designa a forma normal de imprimir caracteres, ou seja, da esquerda para a direita.

Figura 1



Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.6.3. COLOR

O comando COLOR permite a especificação da caneta a ser utilizada para todas as impressões e desenhos subseqüentes. Como cada posição contém uma caneta de cor diferente, o comando COLOR pode ser utilizado para mudar as cores das mesmas. A forma do comando COLOR é:

COLOR expressão
(qualquer modo)

A expressão deve possuir um número inteiro de 0 à 3. Cada número, correspondente a uma caneta diferente. A cor representada pelo número variará dependendo da ordem em que as canetas estão colocadas. A correspondência pode ser determinada pelo comando TEST (mencionado acima).

Números não inteiros, mas que estão entre 0 à 3, serão trocados por um inteiro. Todos os outros números causarão um "ERROR 19". Quando o computador PC-1500 é desligado e ligado novamente, a caneta que corresponde ao zero é selecionada.

No modo TEXT, a execução do comando COLOR causará o reposicionamento da caneta no lado esquerdo do papel.

No modo GRAPH, a caneta volta para a sua posição anterior.

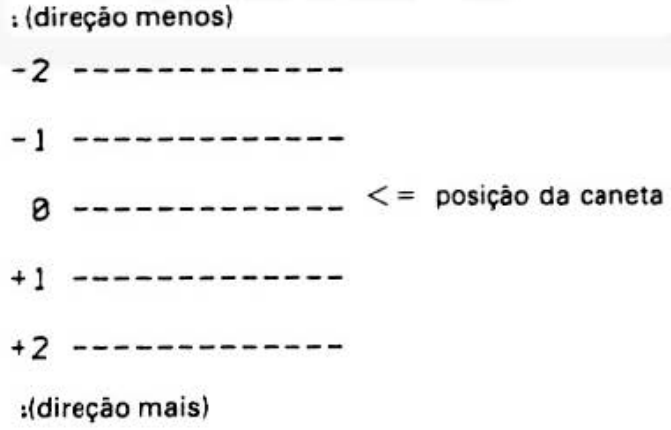
C.6.4. LF

O comando LF faz com que o papel, na impressora, seja movido para frente ou para trás. A forma do comando é:

LF expressão
(somente modo TEXT)

Se a expressão possui um número positivo, o papel avança o número de linhas especificado pela expressão. Uma expressão que possui um número negativo, fará o papel voltar o número de linhas especificado pela expressão. Isto é ilustrado pela figura 2:

Figura 2:



A distância real que o papel é movido, está relacionada ao tamanho do caracter em efeito quando a instrução LF for especificada.

Quando o papel está em movimento na direção Reversa (isto é, está voltando) um contador interno previne o retorno maior do que 10.24 centimetros.

NOTA: Não tente inserir papel enquanto o mecanismo de alimentação do papel estiver operando. Isto poderá danificar a impressora.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.6.5. LPRINT

O comando LPRINT é o principal comando para visualizar o texto de informação na impressora. Ele é similar ao comando LPRINT do PC-1500, e ambos compartilham de diversas formas iguais. Entretanto, devido as características adicionais disponíveis na impressora, as ações da instrução LPRINT são mais complexas. Desta forma, nós nos concentramos em subtítulos implicados no uso da declaração LPRINT.

A discução do comando LPRINT abaixo, assume somente o modo TEXT. Embora as formas dadas possam funcionar no modo GRAPH, suas operações serão diferentes.

LPRINT item

onde item é uma expressão, cordão de caracteres, ou nome de uma variável, cujos conteúdos serão impressos. Usualmente, os caracteres são ajustados à esquerda e os números são ajustados à direita.

Da mesma forma que o cursor no visor, se a caneta não estiver posicionada à esquerda do papel, a impressão começará de onde ela estiver. A posição da caneta pode ser mudada pela declaração LCURSOR ou pela cláusula TAB (vide abaixo).

Um problema ocorre quando se tenta imprimir um item que é grande demais para ser ajustado em uma linha de impressão, devido ao tamanho do caracter. Se o item é um número, resultará em um "ERROR 76". Se o item forma um cordão de caracteres. o cordão será prosseguido na linha seguinte.

O tamanho de um item impresso é também importante para a declaração LPRINT com dois itens, cuja a forma é:

LPRINT item 1, item 2

Usando CSIZE 1 garante que dois itens numéricos serão impressos na mesma linha. Neste caso, os itens serão ajustados de forma usual dentro das duas metades da linha impressa. Para a declaração LPRINT que envolve cordões, isto não é tão simples. Se os itens couberem, eles são ajustados da forma usual e impressos na mesma linha. Se não couberem, os resultados geralmente são divididos em duas linhas. Para caracteres de maior tamanho, os dois itens são impressos em duas linhas sucessivas.

O ponto e vírgula também pode ser usado na declaração LPRINT. Ele serve tanto para indicar o mínimo espaçamento de itens, como no final de uma declaração, para agrupar itens impressos na mesma linha. Nesse caso, se o comprimento total dos itens exceder a capacidade da linha de impressão, os itens serão impressos em linhas sucessivas (tantas quanto forem necessárias). A LPRINT com o ponto e vírgula tem a forma:

LPRINT item 1; item 2; ... (etc)

ou a forma:

LPRINT item – lista;

Várias das formas mencionadas até aqui, são utilizadas no programa de demonstração abaixo:

```
10 A$ = "ABCDEFG"
20 B = 123456
30 FOR I = 1 TO 3
40 CSIZE I
50 LPRINT A$
60 LPRINT A$, B
70 LPRINT A$; B
80 LF 5
90 NEXT I
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.6.7. TAB

A declaração TAB é idêntica à declaração LCURSOR, exceto que pode ser usada dentro de uma declaração LPRINT. Este tipo de declaração LPRINT tem a forma:

LPRINT TAB posição; item – lista

Os mesmos comentários que se aplicam para a expressão de posição de LCURSOR aplica-se para TAB. Se a lista de itens estiver vazia,o resultado da instrução acima será um avanço de papel.

C.6.8. SORGN

O comando SORGN é usado para estabelecer a origem do sistema de coordenadas X–Y, para os comandos gráficos subseqüentes. A declaração SORGN faz com que a posição atual da caneta seja considerada como origem. Portanto, esta instrução é, geralmente usada logo após uma instrução que move a caneta para um determinado ponto no papel. A forma do comando SORGN é simplesmente:

SORGN
(somente modo GRAPH)

NOTA: A impressora permite que a posição da caneta seja especificada fora da margem das posições desenháveis. Neste caso, a caneta move-se tanto quanto possível e "pára". Se a caneta for movida para esta "faixa imaginária" e o comando SORGN for emitido, as declarações de impressão e os desenhos subseqüentes não terão efeito. Ela apreciaria então, como se o programa tivesse um erro ou como se a interface estivesse danificada.

O programa abaixo posiciona a origem em 100 unidades para cima e 100 unidades para à direita da posição atual da caneta. Ele desenha uma caixa de 10 unidades com um dos cantos da caixa na nova origem.

```
10 GRAPH
20 LINE (0, 0) – (100, 100), 9
30 SORGN
40 LINE (0, 0) – (10, 10), 0, 0, B
50 TEXT
60 END
```

C.6.9. GLCURSOR

A declaração GLCURSOR move a caneta para qualquer coordenada X-Y sem traçar uma linha.

A forma da declaração GLCURSOR é:

GLCURSOR (expressão 1, expressão 2)

Ambas expressões devem possuir um número na faixa de –2047 à +2047. Expressão 1 representa a distância X para o ponto de destino e expressão 2 representa a distância Y para o ponto de destino.

NOTA: Se o ponto de destino estiver fora da margem de pontos na qual a caneta pode realmente ser posicionada, esta vai parar em um dos lados do papel. Internamente, o contador que controla a caneta irá contar até o fim.

Exemplos de uso da declaração GLCURSOR são os seguintes:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

O que produz a saída:

```
ABCDEFG
ABCDEFG                    123456
ABCDEFG 123456
```

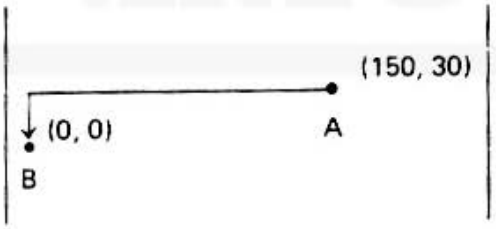
```
ABCDEFG
ABCDEFG
                  123456
ABCDEFG 123456
```

```
ABCDEFG
ABCDEFG
          123456
ABCDEFG 1234
56
```

O comando LPRINT pode ser usado sem uma especificação de item, tanto no modo TEXT como no GRAPH.

```
LPRINT
```

Usado desta maneira, causará o retorno da caneta e um avanço simples de papel. Entretanto, ele não reposiciona o contador do modo GRAPH. Portanto, no seguinte exemplo, embora a declaração LPRINT seja usada para mover a caneta do Ponto A para o Ponto B, a impressora "acredita" que está nas coordenadas (150, 30) e executará todos os comandos subseqüentes como se lá estivesse.



A declaração LPRINT incorpora também uma cláusula USING que opera da mesma maneira que a declaração PRINT. A cláusula USING pode ocorrer somente na declaração LPRINT que é executada no modo GRAPH.

C.6.6. LCURSOR

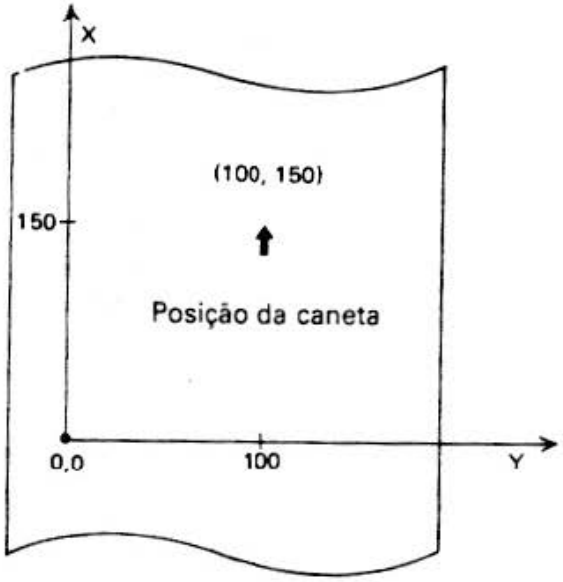
A declaração LCURSOR permite o posicionamento da caneta de forma análoga à declaração CURSOR do visor. A forma da declaração LCURSOR é:

```
LCURSOR posição
(somente modo TEXT)
```

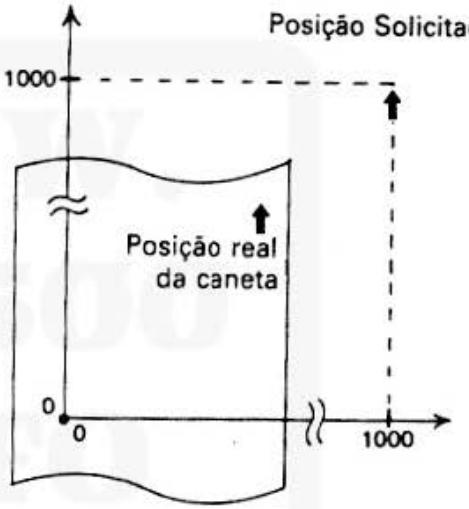
A posição do caracter para que a caneta possa ser movida é, evidentemente, dependente do tamanho do caracter em efeito. Em geral, a caneta pode ser posicionada em um espaço menor do que o máximo para o tamanho do caracter.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info



No exemplo à direita, a caneta foi movida para a posição (100, 150).



Neste exemplo, a caneta é posicionada incorretamente na região "imaginária", ou seja, na posição (1000, 1000).
A caneta na realidade move-se para a direita e enrola o papel para trás. Ao alcançar o lado direito, ela continua subindo até alcançar o limite de 10 cm., e para.

C.6.10. LINE

O comando LINE é o comando primário do modo GRAPH. Ele especifica o movimento da caneta de uma ponta para a outra. Se a caneta for colocada para baixo, quando em movimento, uma linha é traçada. O comando LINE, também permite linhas tracejadas que podem ser traçadas com até oito comprimentos diferentes do tracejado.

A primeira forma do comando LINE é:

LINE(X1, Y1) – (X2, Y2), tipo de linha, cor.

O ponto inicial da linha é determinado pelos valores das expressões X1 e Y1. O final da linha é determinado pelos valores da expressão X2 e Y2. Ambos valores de X e Y devem estar na faixa de −2048 e +2047. Uma especificação do valor que exceda este limite produzirá um erro.
O tipo de linha e os parâmetros de cor são opcionais. Se forem omitidos, os valores usados serão os que estavam em efeito antes do comando. A cor é, evidentemente, uma das cores na faixa de 0 à 3. O tipo da linha deve ser uma expressão que determina um número na faixa de 0 à 9. A tabela 2 identifica o significado de cada opção:

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

TABELA 2:

Tipo de linha Valor	Resultado Tamanho da Linha
0	Linha sólida (contínua)
1	0.4 mm pontilhada
2	0.6 mm pontilhada
3	0.8 mm pontilhada
4	1.0 mm pontilhada
5	1.2 mm pontilhada
6	1.4 mm pontilhada
7	1.6 mm pontilhada
8	1.8 mm pontilhada
9	caneta p/ cima (sem linha)

Usado de outra forma, o comando LINE interpreta as especificações de ponto, como sendo os pontos finais de uma diagonal. Ele prosseguirá então, com o desenho da caixa representada pela diagonal. A forma para esta declaração LINE é:

LINE (X1, X2) – (Y1, Y2), tipo de linha, cor, B

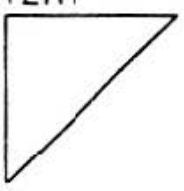
A letra maiúscula B indica que é um comando para desenhar uma caixa. Os outros parâmetros são os mesmos que para a forma de comando anterior.

A forma final do comando LINE permite múltiplas especificações de pontos. Cada ponto, após o primeiro, representa um ponto final de destino do próximo segmento de linha a ser desenhado. A posição atual é assumida como sendo o início do ponto final para o segmento de linha. Esta forma do comando LINE é:

LINE (X1, Y1) – (X2, Y2) ... (X6, Y6), tipo de linha, cor

Os três pontos na forma acima são usados para indicar que uma série de especificações de ponto (até seis em uma linha) podem ser dados. Observe que o parâmetro B não pode ser usado nesta forma de comando.
Damos abaixo, um exemplo de programa, para desenhar um triângulo usando quatro linhas de segmento.
A linha 15 serve simplesmente para estabelecer a origem, enquanto o triângulo é desenhado pela linha 20:

```
10:GRAPH
15:LINE (0,0)-(10
   0,0),9:SORGN
20:LINE (0,0)-(50
   ,50)-(-50,50)-
   (-50,-50)-(0,0
   ),0,0
30:TEXT
```



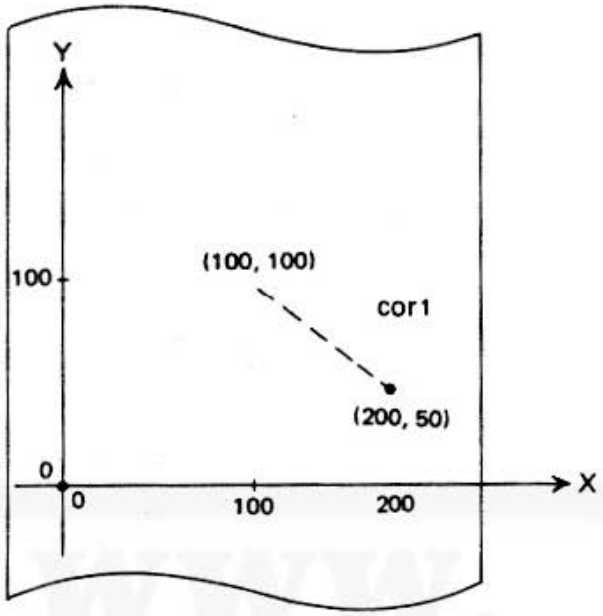
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

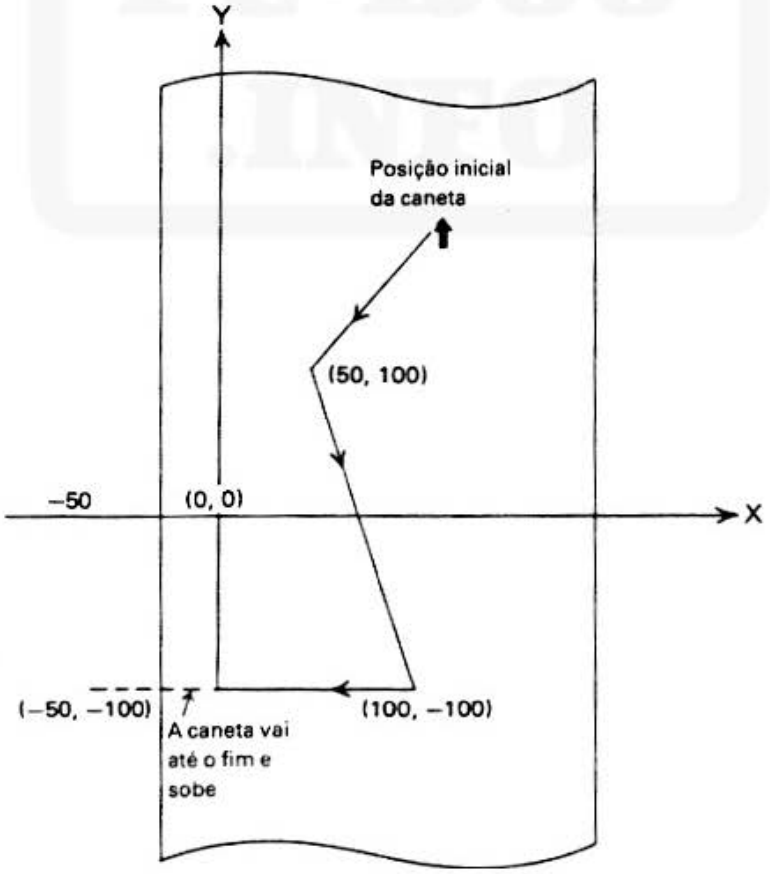
C.6.11. RLINE

O comando RLINE é o mesmo que o comando LINE, exceto que todas as especificações dos pontos representam uma posição relativa para a posição atual da caneta, um pouco antes da origem. As formas dos comandos RLINE são as mesmas dos comandos LINE com a substituição da palavra RLINE por LINE.

Seguem os exemplos: LINE (100, 100) – (200, 50), 2, 1



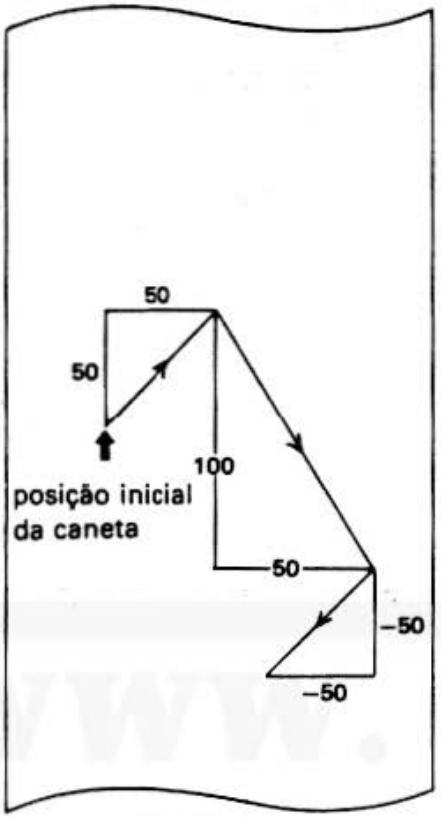
LINE – (50, 100) – (100, –100) – (–50, –100)



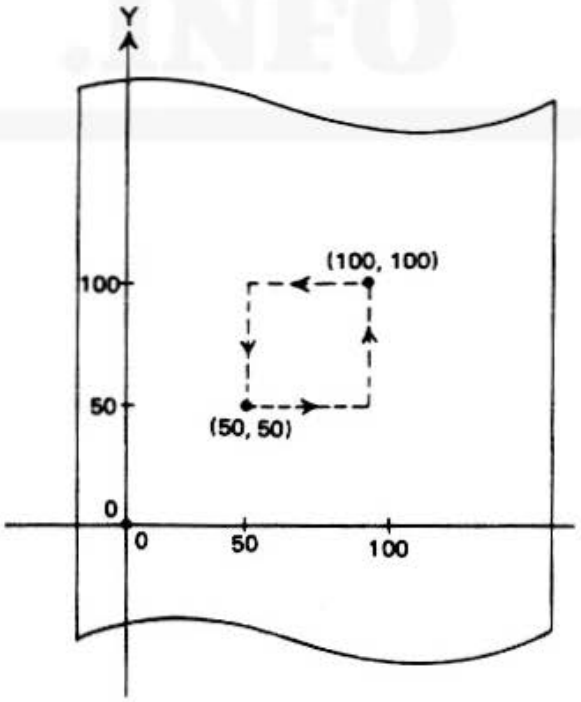
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

RLINE – (50, 50) – (50, –100) – (–50, –50)



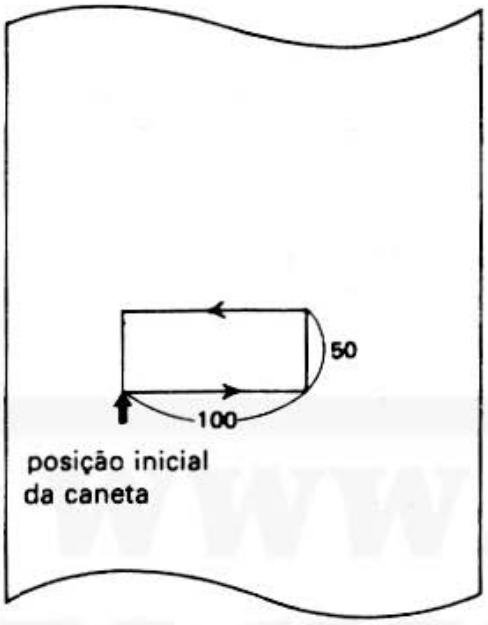
LINE (50, 50) – (100, 100), 2, , B



Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

```
R LINE – (100, 50), , , B
```



VII. MODO RESERVE

A. Definição e Seleção das Teclas de Reserva

Uma importante característica para poupar o trabalho do PC-1500 é a função das seis teclas que podem ser redefinidas. Estas teclas de reserva permitem ao usuário, especificar frases escritas frequentemente ou palavras chaves, que são chamadas com um toque de botão. As teclas de Reserva são as seis teclas, na fila superior do teclado que são !, ", #, $, % e &. Cada uma destas teclas podem ser atribuidas com até três frases ou palavras-chave, com um total de 18 frases armazenadas.

A atribuição de frases para as teclas é efetuada no terceiro modo do Sharp; o modo RESERVE. Para introduzir o modo RESERVE pressione:

SHIFT MODE

O indicador de modo na parte superior do visor deve agora indicar RESERVE. Para sair do modo RESERVE, simplesmente pressione a tecla MODE uma vez.

Devido cada tecla de Reserva ter capacidade de armazenar até três frases, existe um método de selecionar qual frase é chamada quando uma tecla de RESERVA é pressionada. Este método é uma tecla no lado inferior esquerdo marcada com o símbolo (▲▼).

Chamada a Tecla de Seleção de Reserva, esta irá selecionar qual frase armazenada corresponde às teclas reservadas. É importante observar que a Tecla de Seleção de Reserva muda a correspondência para todas as teclas reservadas ao mesmo tempo. Isto quer dizer que a Tecla de Seleção de Reserva seleciona um grupo de frases, cada uma das quais correspondente a uma tecla de reserva única. O grupo selecionado é indicado pelos algarismos romanos (I, II e III) da parte superior do visor.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Para atribuir uma frase a uma Tecla de Reserva, introduza o modo RESERVE (pressione [SHIFT] [MODE]) e use a tecla de seleção de Reserva para selecionar o grupo (I, II ou III) no qual a frase será armazenada. Em seguida, pressione a tecla de Reserva apropriada (!, ", #, $, % ou &). Dependendo de qual tecla de Reserva você pressionou, no visor aparecerá algo parecido com o seguinte:

```
                              RESERVE      I      •
F6 :
```

(O número que segue o F, seis no nosso exemplo, representa a tecla que você pressionou). Quando este "Prompt" aparece, você pode digitar a frase a ser armazenada. Como exemplo, digite o seguinte:

[R] [U] [N] [1] [0] [0] [ENTER]

Esta frase é agora associada com a tecla de Reserva que você selecionou.

Vamos testar isto. Volte para o modo RUN pressionando a tecla MODE. Pressione a tecla de Reserva que você usou no nosso exemplo. O visor mostrará agora, o comando que acabamos de armazenar:

```
                              RUN          I      •
RUN 100_
```

Se você pressionar a tecla ENTER, o computador tentará executar um programa na linha 100. A vantagem da tecla de Reserva é que você não tem que digitar um comando inteiro cada vez que você desejar emití-lo.

Uma notação especial permitida no modo RESERVE poderia ter nos ajudado em alguns problemas dos exemplos anteriores.

Esta notação é o uso de @ para representar o comando ENTER.

Se nós tivessemos atribuído a frase "RUN 100 @ " para a nossa tecla de Reserva, a execução do programa teria começado logo que nós voltamos para o modo RUN e pressionamos a tecla de Reserva.

Para demonstrar isto, introduza as seguintes declarações:

```
222 BEEP 5,50: END
```

Agora, introduza o modo RESERVE e defina uma das teclas de Reserva usando esta digitação de teclas: [G] [O] [T] [O] [2] [2] [2] [•] [ENTER]

(Note que a digitação da tecla [ENTER] ainda é necessária para definir a tecla de Reserva).

Volte para o modo RUN e pressione a tecla de Reserva que acaba de ser definida.

Você observará que a digitação da tecla [ENTER] após a digitação da tecla de Reserva não será necessária.

Realmente, o nosso exemplo poderia ter sido conseguido através da atribuição da frase:

```
BEEP 5,50@
```

diretamente para uma tecla de Reserva. Tente.

B. Identificação das Teclas de Reserva

Conforme você vai aumentando a utilização das teclas de Reserva, você necessitará lembrar qual tecla foi atribuída à qual função.

O PC-1500 permite que você armazene três cordões de caracteres (uma para cada grupo de teclas de Reserva) que identificam as funções das teclas. Estes cordões são análogos aos comentários no modo PRO.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Cordões de identificação (chamada "gabaritos") são criadas no modo RESERVE.

Posicione para este modo e selecione o grupo apropriado de teclas de Reserva, usando a tecla de Seleção de Reserva. No lugar de pressionar uma tecla de Reserva, como você faria normalmente, digite um gabarito (template) e pressione [ENTER]. O gabarito é armazenado em associação com o grupo.

Como um exemplo, suponhamos que nós atribuímos teclas de Reserva de um à seis (no grupo I) com os nomes de Funções Trigonométricas (Seno, Coseno, Tangente, Arco-Coseno, Arco-Seno e Arco-Tangente). A fim de lembrar qual tecla corresponde a qual função, nós especificaremos um gabarito.

Para fazer isto, posicione no modo RESERVE (pressione [SHIFT] [MODE]), e use a tecla de Seleção de Reserva para selecionar o grupo I (um algarismo Romano I aparecerá no visor).

Agora digite:

"SIN COS TAN ACS ASN ATN"

Digitação de Teclas

[SHIFT] ["] [S] [I] [N] [SPACE] [C] [O] [S] [SPACE]
[T] [A] [N] [SPACE] [A] [C] [S] [SPACE] [A] [S] [N]
[SPACE] [A] [T] [N] [SPACE]
[SHIFT] ["] [ENTER]

O gabarito foi agora definido e armazenado.

Volte para o modo RUN pressionando a tecla [MODE] . Para chamar o significado das teclas de Reserva, simplesmente pressione a tecla [RCL] e lá estão eles! Pressione a tecla [RCL] mais uma vez e o "prompt" retorna.

Os gabaritos podem ser criados para cada grupo de teclas de reserva, e pode ser de até 26 caracteres. Observe que o gabarito é somente para lembrarmos das palavras ou letras que armazenamos. Não tem significado para o computador.

C. Cancelamento de Programas de Reserva

1) Como você sabe, as teclas [N] [E] [W] [ENTER] limpam todas as memórias de reserva. Observe que a operação da tecla acima deve ser feita no modo RESERVE.

2) Para cancelar a memória de Reserva, utilize as teclas [SPACE] ou [SHIFT] [DEL] como descritas abaixo:

Exemplo: Limpe A * A que está reservada na tecla de número F3 ([#]) no grupo I.

Opção	Visor	Observação
[↑] [#]		Visor mostra o símbolo I
[F3]	F3: A * A	
[►] or [◄]	F3: A * A	
[SPACE] [SPACE] [SPACE]	F3: _	
or [SHIFT] [DEL] [SHIFT] [DEL] [SHIFT] [DEL]	(F3: _)	
[ENTER]	F3:	

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

VIII. INICIANDO A EXECUÇÃO DO PROGRAMA

A. A Tecla DEF

A tecla DEF (abreviação de definir) fornece várias formas de poupar tempo.

A.1. Execução de Programas Definíveis

A tecla DEF é o terceiro modo de iniciar um programa. Conforme vimos na seção do comando RUN, um programa pode ser "rotulado" com uma letra. A tecla DEF pode ser usada para iniciar rapidamente um programa rotulado. Isto é feito através do pressionamento da tecla DEF, seguida pela tecla que corresponde ao rótulo do programa.

As seguintes teclas podem ser usadas desta maneira:

A, S, D, F, G, H, J, K, L, Z, X, C, V, B, N, M, SPACE e =

Como exemplo, introduza as seguintes declarações para criar três programas rotulados:

Listagem do Programa:

```
10 "Z": GOSUB 500
20 PRINT "TECLA Z"
30 END
140 "A": GOSUB 500
150 PRINT "TECLA A"
160 END
270 " ": GOSUB 500
280 PRINT "TECLA SPACE"
290 END
500 CLS: PAUSE "VOCE PRESSIONOU "
510 RETURN
```

No modo RUN, tente iniciar cada programa com o uso da tecla DEF. Observe que se especificarmos uma letra, cujo programa rotulado correspondente não existe, ocorrerá um "ERROR 11".

A.2. Palavras-Chave pré-atribuídas

Algumas das palavras-chave mais frequentemente utilizadas foram atribuídas permanentemente a uma tecla alfabética única na segunda fila do teclado. Estas palavras-chave podem ser reativadas em qualquer modo, pressionando a tecla DEF seguida por uma destas teclas alfabéticas. Por exemplo, para recuperar a palavra-chave USING pressione: DEF E

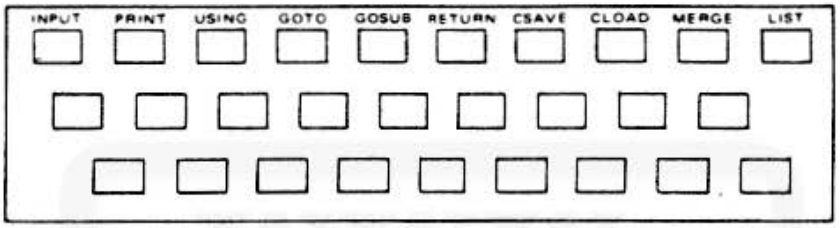
Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

As palavras-chave disponíveis e as suas teclas alfabéticas correspondentes são:

Tecla Alfabética	Palavra Chave	
Q	INPUT	
W	PRINT	
E	USING	
R	GOTO	
T	GOSUB	
Y	RETURN	
U	CSAVE	Pode ser utilizado quando a interface impressora/cassete está conectada ao computador
I	CLOAD	
O	MERGE	
P	LIST	

Gabarito



São fornecidos dois gabaritos com o seu computador. Use-os para identificar a operação funcional atribuída à tecla de definição.

A.3. A declaração AREAD

Programas rotulados, que são iniciados usando a tecla DEF, podem ter um valor único cada vez que o programa é rodado sem usar a declaração INPUT. A leitura do valor é efetuada através da declaração AREAD, que deve seguir o "rótulo" do programa na mesma linha. A declaração AREAD tem a forma:

AREAD nome da variável

onde o nome da variável é um numérico legal ou nome da variável de caracter.

Para passar um valor para um programa que incorpora uma declaração AREAD, o usuário digita o valor, pressiona a tecla DEF, e em seguida pressiona a tecla alfabética correspondente do rótulo do programa.

Como um exemplo, introduza os dois programas abaixo:

Listagem do Programa

```
10 "X": AREAD TM
20 TIME = TM
30 PRINT "HORA ACERTADA EM "; TM
40 END
100 "Z": AREAD D$
110 PRINT "HOJE É"; D$
120 END
```


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Volte para o modo RUN e comece o programa rotulado X digitando o mês, dia, hora e DEF X:

[1] [2] [3] [1] [0] [2] [.] [4] [0] [0] [0] [DEF] [X]

Este programa posicionará o sistema de relógio na hora em que você especificou, antes da digitação da tecla DEF.

(Veja a função TIME).

Para iniciar o programa rotulado Z, digite um dia da semana e DEF Z:

[F] [R] [I] [D] [A] [Y] [DEF] [Z]

B. Iniciando o Programa Automaticamente

Os programas podem ser iniciados rapidamente e facilmente, não somente utilizando a tecla DEF, como também podem ser iniciados automaticamente quando você ligar o PC--1500.

Para fazer isto, você usa a declaração ARUN. Esta declaração deve ser a primeira declaração na memória do programa ou ela será ignorada. Além disso, várias outras condições são necessárias para a declaração ARUN funcionar. São elas; que o PC-1500 seja desligado no modo RUN, e que nenhum erro seja detectado quando o PC-1500 é ligado.

O programa abaixo usa a declaração ARUN para saudar o operador do computador:

Listagem do Programa:

```
10 ARUN
30 CLS
50 BEEP 5, 50
70 PRINT "MUITO BEM"
90 END
```

C. Comparação entre Métodos de Iniciação

Embora vários métodos de iniciação de um programa consigam superficialmente o mesmo resultado, a sua operação interna difere. A fim de explorar estas diferenças para se obter vantagens, é necessário discutir o armazenamento dos dados dentro do computador. Também está incluso, uma seção comparando as várias preparações internas que são feitas pelo PC-1500 antes de rodar um programa.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C.1. Área de Memória Fixa

Embora todas as variáveis do mesmo tipo sejam utilizadas da mesma forma, elas não são tratadas internamente da mesma maneira.

O PC-1500 inclui uma área de "memória fixa" com espaço de armazenamento suficiente para 26 variáveis numéricas e 26 variáveis do cordão de caracteres (tamanho do cordão de 16 caracteres). Conseqüentemente, as variáveis de A à Z, e as variáveis de A$ à Z$ são distribuidas permanentemente dentro desta área.

Todas as outras variáveis, incluindo nomes com dois caracteres, são distribuídas dentro da área de memória principal do computador. Esta área de memória principal é compartilhada com as instruções do programa, embora as variáveis sejam distribuídas iniciando do extremo oposto da memória de instruções. Devido as instruções e dados do programa compartilharem a mesma área, é possível que utilizem todo o espaço de armazenamento disponível. Neste caso, um "ERROR" na faixa de 177 à 181 ocorrerá.

É importante compreender que as duas áreas de memória não são tratadas igualmente depois de iniciar o programa. Isto é explicado na tabela da próxima seção. Basicamente, as variáveis na área de memória fixa nunca são removidas exceto por uma explícita declaração CLEAR. Aquelas que estão na memória principal são removidas toda vez que um programa inicia com o comando RUN.

Uma outra observação sobre a área de memória fixa é que os dados nesta área podem ser redefinidos como uma matriz cujo nome é @ (para variáveis numéricas, e @$ para variáveis do cordão . Assim, a designação @ (1) é a mesma localização de armazenamento que a variável A e @ (26) é a mesma localização de armazenamento que a variável Z. A designação @ $ (5) refere-se a mesma localização que E$, e a designação @$ (20) refere-se à mesma localização que T$. Por razões óbvias, subscritos acima de 26 não são permitidos. Observe que os arranjos @ e @$ não necessitam ser dimensionados antes do uso.

C.2. Tabela Comparativa do Programa

Métodos de Iniciação

	RUN	GOTO	DEF
O visor é limpo	S	S	N
O cursor retorna para a primeira coluna	S	N	N
O intervalo WAIT é posicionado para infinito	N	N	N
O modo TRACE é alterado.	N	N	N
Área de Memória Fixa é limpa	N	N	N
Área de Memória Principal é limpa	S	N	N
O "stack" interno do FOR-NEXT, GOSUB é limpa	S	S	S
ON ERROR GOTO é cancelado.	S	N	N
Apontador de dados (DATA) para a operação READ é rearmazenada	S	N	N
Formato USING é cancelado	S	N	N

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

www.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

IX. APÊNDICE

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

A. Uma breve tabela de abreviações.

Comandos de impressora

COLOR	COL.
	COLO.
CSIZE	CSI.
	CSIZ.
GLCURSOR	GL.
	GLC.
	GLCU.
	GLCUR.
	GLCURS.
	GLCURSO.
GRAPH	GRAP.
LCURSOR	LCU.
	LCUR.
	LCURS.
	LCURSO.
LF	LF.
LINE	LIN.
LLIST	LL.
	LL.
	LLS.

LPRINT	LP.
	LPR.
	LPRI.
	LPRIN.
RLINE	RL.
	RLI.
	RLIN.
ROTATE	RO.
	ROT.
	ROTA.
	ROTAT.
SORGN	SO.
	SOR.
	SORG.
TAB	TAB
TEST	TE.
	TES.
TEXT	TEX.

Comandos de Cassete

CHAIN	CHA.
	CHAI.
CLOAD	CLO.
	CLOA.
CLOAD?	CLO.?
	CLOA.?
CSAVE	CS.
	CSA.
	CSAV.
INPUT #	I #
	IN #
	INP #
	INPU #

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Declarações

AREAD	A. AR. ARE. AREA.
ARUN	ARU.
BEEP	B. BE. BEE.
CLEAR	CL. CLE. CLEA.
CLS	– –
CURSOR	CU. CUR. CURS. CURSO.
DATA	DA. DAT.
DEGREE	DE. DEG. DEGR. DEGRE.
DIM	D. DI.
END	E. EN.
ERROR	ER. ERR. ERRO.
FOR	F. FO.
GCURSOR	GCU. GCUR. GCURS. GCURSO.
GOSUB	GOS. GOSU.
GOTO	G. GO. GOT.
GPRINT	GP. GPR. GPRI. GPRIN.
GRAD	GR. GRA.
IF	– –
INPUT	I. IN. INP. INPU.
LET	LE.
LOCK	LOC.
NEXT	N. NE. NEX.
ON	O.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PAUSE	PA. PAU. PAUS.
PRINT	P. PR. PRI. PRIN.
RADIAN	RAD. RADI. RADIA.
RANDOM	RA. RAN. RAND. RANDO.
READ	REA.
REM	– –
RESTORE	RES. REST. RESTO. RESTOR.
RETURN	RE. RET. RETU. RETUR.
STEP	STE.
STOP	S. ST. STO.
THEN	T. TH. THE.
TRON	TR. TRO.
TROFF	TROF.
UNLOCK	UN. UNL. UNLO. UNLOC.
USING	U. US. USI. USIN.
WAIT	W. WA. WAI.

Comandos

CONT	C. CO. CON.
LIST	L. LI. LIS.
NEW	– –
RUN	R. RU.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Funções

ABS	AB.	MEM	M. ME.
ACS	AC.	MID$	MI. MID.
AND	AN.	NOT	NO.
ASC	– –	OR	– –
ASN	AS.		
ATN	AT.	PI	– –
CHR$	CH. CHR.	POINT	POI. POIN.
COS	– –	RIGHT$	RI. RIG. RIGH. RIGHT.
DEG	– –	RND	RN.
DMS	DM.	SGN	SG.
EXP	EX.	SIN	SI.
INKEY$	INK. INKE. INKEY.	SQR	SQ.
INT	– –	STATUS	STA. STAT. STATU.
LEFT$	LEF. LEFT.	STR$	STR.
LEN	– –	TAN	TA.
LOG	LO.	TIME	TI. TIM.
LN	– –	VAL	V. VA.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

B. SUBSTITUIÇÃO DAS PILHAS PARA O PC-1500

Quando substituir as pilhas, estas precauções evitarão problemas:

* Substitua sempre as quatro (4) pilhas ao mesmo tempo.
* Não misture as pilhas novas com as velhas.
* Use somente: pilhas (tipo AA, R6 ou SUM-3, 1.5V) × 4.

PROCEDIMENTOS PARA A SUBSTITUIÇÃO DAS PILHAS

Antes do computador ser utilizado, ou toda vez que o indicador de bateria desaparecer, é necessário substituir as pilhas. Siga estes procedimentos para a substituição:

1. Desligue o computador pressionando a tecla [OFF]
2. Remova o parafuso da tampa da bateria com uma moeda ou uma chave de fenda pequena (vide figura 1).
3. Substitua as 4 pilhas (figura 2).

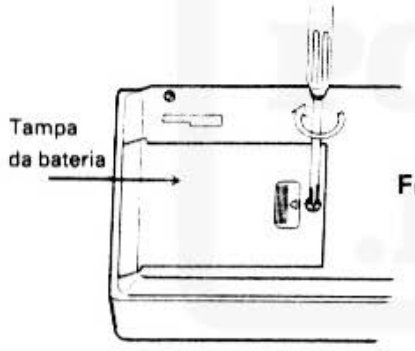


Fig. 1

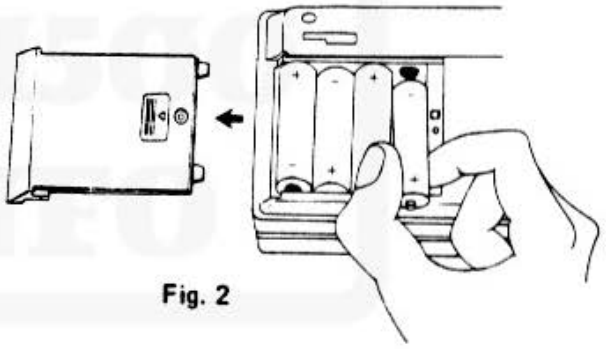
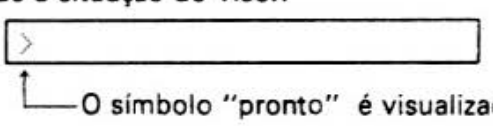


Fig. 2

4. Empurre a tampa da bateria levemente quando recolocar o parafuso.
5. Para prosseguir, pressione as teclas [ON] e [CL]. (verifique se o computador está em Modo PRO. Caso não esteja pressione a tecla [MODE] até que no visor apareça o símbolo PRO. Digite NEW 0 e [ENTER].

   Após isso pressione [SHIFT] [MODE] e digite NEW [ENTER]. Estes procedimentos limparão as memórias do computador e as áreas de memória reservadas.

6. Verifique a situação do visor.



O símbolo "pronto" é visualizado.

Se o visor estiver vazio, ou visualizar qualquer símbolo diferente de ">", remova as pilhas, instale-as novamente, e verifique o visor.

Do not sell this PDF !!!

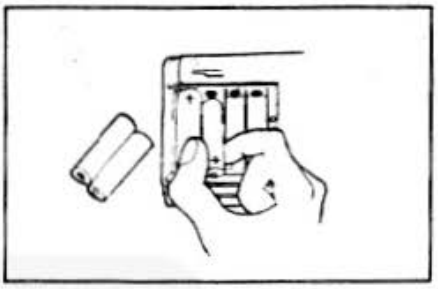
All and more about Sharp PC-1500 at http://www.PC-1500.info

NOTAS

* Se o computador não for ser operado por um longo período de tempo, remova as pilhas para evitar possíveis danos causados por vazamentos das mesmas.
* Conservar pilhas gastas, poderá causar danos no computador devido ao derramamento de compostos químicos das mesmas. Remova-as prontamente.
* Baterias recarregáveis não podem ser utilizadas no PC-1500.
* O adaptador AC, EA-150 para Interface Impressora/Cassete é usado também como um adaptador AC para o PC-1500, quando este for utilizado separadamente.
  (Não conecte o EA-150 no computador PC-1500, quando este estiver conectado na interface impressora/cassete).

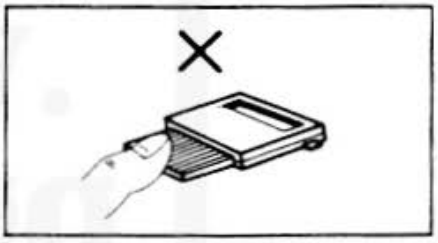
MÓDULO DE EXPANSÃO DA MEMÓRIA

Antes de instalar ou remover o módulo, retire as pilhas e pressione a chave ALL RESET

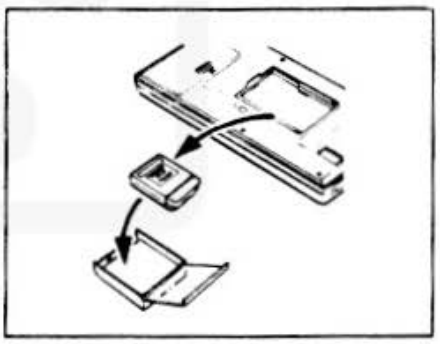


O módulo pode ser danificado por descargas eletrostáticas.

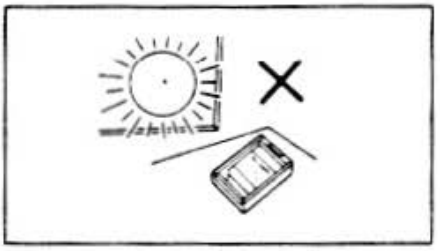
Nunca toque os contatos do módulo.



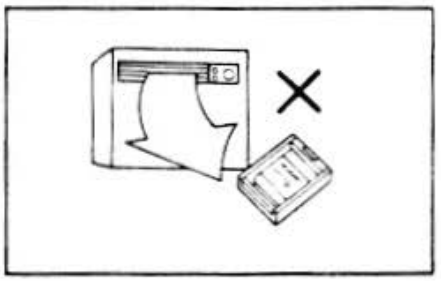
Quando você tiver com o módulo removido do computador, coloque a tampa de contatos no mesmo, colocando-o no estojo de proteção.



Nunca deixe o módulo exposto a luz solar direta.



Não deixe o módulo perto de fontes de calor, tais como aquecedores e outros.

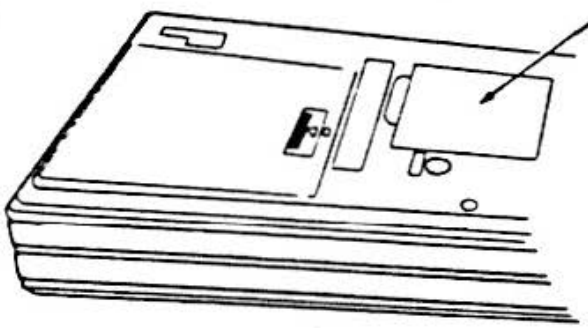


Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PARA INSTALAR O MÓDULO

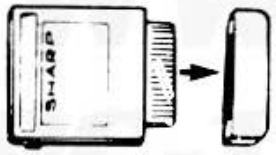
1) Desligue o computador pressionando a tecla [OFF] e remova as baterias.
2) Pressione a chave ALL RESET da parte traseira da unidade por aproximadamente 15 segundos com a tecla [ON] pressionada.
3) Remova a tampa do módulo.



Compartimento do módulo de expansão da memória.

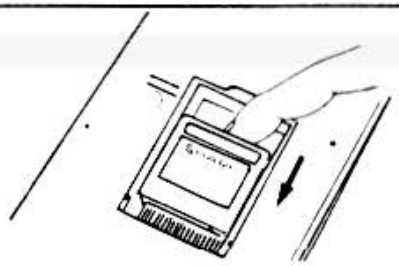
Este compartimento é para acomodar o módulo de expansão de memória.
Não toque no conector dentro do compartimento, pois isto poderá causar danos ao computador devido à eletricidade estática.

4) Retire o módulo do estojo de proteção, e remova a tampa dos contatos do mesmo.



Tampa de contatos

5) Instale o módulo no compartimento adequado, pressionando-o na direção da seta para conexão no conector do computador.



6) Recoloque a tampa do módulo e as pilhas.
7) Ligue o computador e pressione as teclas [CL] [N] [E] [W] [Ø] e [ENTER] para limpar os seus conteúdos.

PRECAUÇÕES:

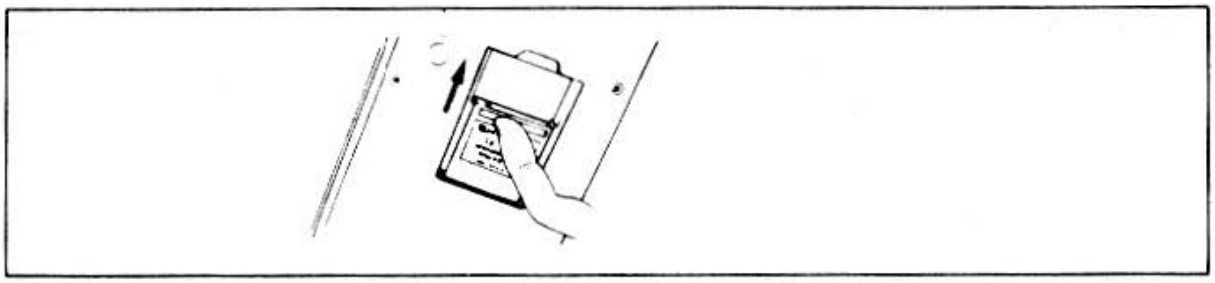
- Para evitar possíveis danos ao módulo, causados pela eletricidade estática, toque em um objeto de metal (uma maçaneta da porta, abajur, etc...) antes de manuseá-lo. É mais provável que forme eletricidade estática durante os meses de inverno (ou em áreas com clima muito seco), quando a umidade relativa do ar é baixa.
- Certifique-se de remover as pilhas antes de retirar ou fixar o módulo no computador.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

PARA REMOVER O MÓDULO

1) Desligue o computador, pressionando a tecla [OFF] e remova as baterias.
2) Pressione a chave ALL RESET da parte traseira da unidade por aproximadamente 15 segundos com a tecla [ON] pressionada.
3) Remova a tampa do módulo na parte traseira do computador.
4) Remova o módulo cuidadosamente



5) Coloque a tampa de contatos no mesmo colocando-o imediatamente no seu estojo de proteção.

6) Recoloque a tampa do módulo e instale as pilhas.
7) Ligue o computador e pressione as teclas [CL] [N] [E] [W] [Ø] e [ENTER] .

ESPECIFICAÇÕES:

TIPO:	módulo de expansão da memória
CONTEÚDOS:	8192 Bytes (RAM)
TEMPERATURA DE OPERAÇÃO:	0° - 40°C
DIMENSÕES:	31,4 x 33.2 x 7mm
PESO:	8g.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

C. CÓDIGO DE CARACTERES ASC II PARA O PC-1500.

Posições dos Bits Superiores b7, b6, b5 →

Posições dos Bits Inferiores b4, b3, b2, b1 ↓

	Hexa decimal	000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0			SPACE	0	@	P		p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	√	k	{
1100	C			,	<	L	¥	l	¦
1101	D			-	=	M	π	m	}
1110	E			.	>	N	^	n	~
1111	F			/	?	O	_	o	■

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

E: LISTA DE CÓDIGOS DE ERROS DO PC-1500

Código de Erro **Explanação**

1. Erro de Sintaxe: Este resulta de erros de digitação, tais como:

   informação incompleta:

   ```
   10: GOTO
   ```

   ou comandos inválidos.

   ```
   10: NEW
   ```

   (NEW não pode ser uma declaração).

   <Visor> ERROR 1 IN 10

2. Este erro ocorre quando não existe correspondência entre comandos FOR e NEXT, ou entre GOSUB e RETURN.

   EX.
   ```
   10: FOR A = 1 TO 10
    :
    :
   100: NEXT B
   ```

   <Visor> ERROR 2 IN 100

4. Este erro ocorre quando não existe correspondência entre DATA e comando READ

   Ex.
   ```
   10: READ X, Y
   20: DATA 10
   30: END
   ```

   <Visor> ERROR 4 IN 10

5. Este erro ocorre quando uma variável de matriz foi declarada com o nome de uma variável existente.

   Ex.
   ```
   10: DIM A (10, 10)
   20: DIM A (5)
   ```

   <Visor> ERROR 5 IN 20

6. Este erro ocorre quando uma variável de matriz foi usada sem uma declaração DIM (dimensão).

   EX.
   ```
   10: CLEAR
   20: A (3) = 1
   ```

   <Visor> ERROR 6 IN 20

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

7. Este erro ocorre quando o nome da variável não é apropriado.

EX. 10: A$ = 10 ou

10: FOR A$ = 1 TO 10

<Visor> ERROR 7 IN 10

8. Este erro ocorre quando a dimensão excede 3 na declaração de uma variável de matriz.

EX. 10: DIM A (3, 4, 5, 6)

<Visor> ERROR 8 IN 10

9. Este erro ocorre quando o número subscrito de uma variável de matriz excede o tamanho da matriz determinada pelo comando DIM.

EX. 10: DIM A (3)

20: A (4) = 1

<Visor> ERROR 9 IN 20

10. Este erro ocorre quando não existir disponível, memórias suficientes para criar mais variáveis.

EX.	Operação de tecla	Visor
	MEM [ENTER]	7
	AB = 10 [ENTER]	ERROR 10

11. Este erro ocorre quando a linha especificada não existe no programa.

EX. 10: PRINT "X="; X: GOTO 5

<Visor> ERROR 11 IN 10

12. Este erro ocorre quando o comando USING especifica algo de formato incorreto.

EX. 100: PRINT USING "###,A #"; 10

<Visor> ERROR 12 IN 100

13. Este erro ocorre quando um programa excede a capacidade da memória de programa ou quando a especificação da tecla de Reserva, excede a capacidade da área de Reserva.

EX.	Operação de Tecla	Visor
	MEM [ENTER]	7
	15 A = A + 1 [ENTER]	ERROR 13

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

14. (1) As declarações FOR, foram demasiadamente encadeadas e a capacidade do "stack" foi excedida.
(2) Enquanto estiver analisando uma expressão e o espaço de "buffer" (armazenamento temporário) exceder.

15. (1) As declarações GOSUB foram demasiadamente encadeadas e a área do "stack" foi excedida.
(2) Enquanto estiver analisando uma expressão e o espaço do "buffer" (armazenamento temporário) do cordão exceder pelo manuseio das mesmas.

16. (1) O valor especificado for superior à 1E100 ou inferior à -1E100.

EX. 123E99

(2) O valor fixado por hexadecimais exceder 65535.

EX. & 1FFAB

17. O tipo de dados (números, cordões de caracteres) é inadequado para a expressão de cálculos.

EX. 1.+ "A" ENTER

18. Número de argumentos inadequados para a expressão.

EX. LEFT$ ("ABC") ENTER

SIN (30, 60) ENTER

19. Valor numérico especificado fora da faixa permitida.

EX. 10: DIM A (256)

<Visor> ERROR 19 IN 10

20. Quando as variáveis da matriz da área de memória fixa são especificadas e não existir "(" seguindo '@' ou '@$'.

EX. 100: @$ = "A"

<Visor> ERROR 20 IN 100

21. Uma variável é requisitada na expressão.

EX. 10: FOR 1 = 0 TO 10

<Visor> ERROR 21 IN 10

22. Este erro ocorre quando o programa é carregado, e não existe espaço de memória disponível para o carregamento.

23. Este erro ocorre quando a hora é fixada incorretamente.

EX. TIME = 131005.10

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

26. Este erro ocorre quando um comando não pode ser executado no modo atual.

EX. <RUN MODE> NEW ENTER

27. Este erro ocorre quando a interface impressora/cassete não é conectada ou não existe programa que coresponda ao rótulo dado.

EX.

Operação de Tecla	Visor
DEF I ENTER	ERROR 27

28. Este erro ocorre quando um comando ou código de função foi inserido entre aspas " ", ou quando você tenta substituir os comandos INPUT ou AREAD por variáveis de caracter.

EX. 10 INPUT A$

Operação de Teclas	Visor
DEF W ENTER	ERROR 28

30. Este erro ocorre quando um número de linha excede 65539.
(65280 ~ 65539 : ERROR 1)

EX. 102235 A = 10 ENTER

32. Este erro ocorre quando o cursor gráfico está entre as coluna 152 ~ 155 durante a execução de comandos de entrada (o código de entrada não pode ser visualizado).

EX. 100: GCURSOR 152
110: INPUT X

<Visor> ERROR 32 IN 110

131. Existe um " + " ou " − ", que é considerado como sinal, antes do cordão de caracteres e/ou variável de caracter.

177 ~ 181 Durante a criação do programa, este reescreve a área de dados.
Ocorre a sobreposição destas duas áreas.

0.224 ~ 241 São fornecidos dados de entrada incorretos durante a execução de comandos INPUT ou AREAD.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

36. Dados ou caracteres não podem se visualizados de acordo com o formato especificado pelo comando USING.

EX. 10: USING " # # # # . # # "
20: PRINT 12345

A secção de números inteiros mais o seu sinal excedeu os 04 espaços para dígitos.

37. Este erro ocorre em cálculos numéricos, quando os resultados excedem 9.999999999E99.

38. Este erro ocorre quando efetuamos cálculos usando 0 como denominador.

EX. 10: PRINT 5/0

39. Este erro ocorre quando um cálculo ilógico é efetuado.

* cálculo logarítmico de número negativo Ex. LN (-10)
* ASN, ACS no caso de X = 1 Ex. ASN (1,5)
  ACS (100)
* Raiz quadrada de números negativos Ex. SQR (-10)

ERROS RELACIONADOS COM O CASSETE

40. Especificação inadequada para a expressão.

41. SAVE e LOAD foram especificados para a área de ROM.

42. Dados de arquivo do Cassete são grandes demais e não podem ser carregados (LOAD).

43. Durante a verificação de dados usando o comando CLOAD? o formato de dados a ser carregado não coincide com o formato de arquivo.

44. Ocorreu um erro de um CHECK SUM

COMANDOS RELACIONADOS COM A IMPRESSORA

70. (1) A caneta excedeu o limite de coordenadas de:
-2048 <= X, Y <= 2047
(2) A caneta excederá o limite ao executar outros comandos posteriores.

71. (1) O papel voltou mais de 10,24 cm no modo TEXT.
(2) O papel voltará mais de 10,24 cm se executar outros comandos (no MODO TEXT)

72. O valor dado é inadequado para o valor de TAB.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

73. Um comando, foi usado em modo de impressora errôneo. (GRAPH/TEXT).

74. O número de Vírgulas (,) no comando LINE ou RLINE é muito grande.
NOTA: Entrada de mais de sete vírgulas resulta em um erro. Também, se a primeira vírgula for omitida, mais de seis causará um erro.

76. Quando a impressão do resultado de cálculo não puder ser feita em uma linha no LPRINT (no modo TEXT).

78. (1) As canetas estão no processo de troca.
(2) O estado de LOW BATTERY não foi corrigido (vide nota 1).
Este erro ocorre quando, por qualquer das duas razões, os comandos que movem a caneta (tais como LPRINT e LINE) não estão capacitados para serem executados.

79. O sinal de cor não apareceu (vide nota 2)

80. Bateria Fraca. (vide nota 3).

NOTAS:

(1) Se o "ERROR 78" é devido ao estado de "LOW BATTERY" (bateria fraca da impressora), desligue a interface impressora/cassete.
Após recarregar a impressora, ligue-a novamente.
Agora, você pode continuar.

(2) O sinal de cor é para COLOR e aparece somente quando a caneta vem para o lado esquerdo.
Quando a caneta está nesta posição, é possível saber a cor da caneta que está posicionada.

(3) Após a recarga, pressione a tecla ON do PC-1500 e inicie a operação.

O: ORDEM DE AVALIAÇÃO DE EXPRESSÕES

Os cálculos são efetuados de acordo com a seguinte hierarquia; expressões entre parênteses, tem a maior prioridade e as operações lógicas a menor. Se duas ou mais operações de mesma prioridade forem encontradas na mesma expressão ou sub-expressão, serão avaliadas da esquerda para a direita.

1) Expressões entre parênteses

2) Recuperação de valores de variáveis, TIME, PI, MEM, INKEY$.

3) Funções (SIN, COS, LOG, EXP, etc...)

4) Exponenciação (Exemplo: 2A ^ 3 = 2 * (A ^ 3)

5) Sinal Aritmético (+, −)

6) Multiplicação, Divisão (*, /)

7) Adição, Subtração (+, −)

8) Operadores de comparação (<, <=, =, >=, >, <>)

9) Operadores lógicos (AND, OR, NOT).

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

NOTAS:

Quando sinais aritméticos e exponenciais são usados na mesma expressão, o expoente é avaliado antes do sinal.

Exemplo: -5^4 é avaliado para -625 ao invés de 625.

Cálculos entre parênteses serão avaliados primeiro. Nas expressões com vários "níveis" de parênteses, os cálculos iniciam com o par "mais interno" e prosseguem para os pares "mais externos" de parênteses.

Exemplo de avaliação.

$$7 \wedge 2 + 3 * \sqrt{144} / \sqrt{81} + \text{SIN}\,(120 + 150) * -3$$

$$7 \wedge 2 + 3 * \sqrt{144} / \sqrt{81} + \text{SIN}\,(270) * -3$$

$$7 \wedge 2 + 3 * 12 / 9 + -1 * -3$$

$$49 + 3 * 12 / 9 + -1 * -3$$

$$49 + 36 / 9 + 3$$

$$49 + 4 + 3$$

$$53 + 3$$

$$56$$

(CAPACIDADE DE CÁLCULO)

Funções	Capacidade de Cálculo
$y \wedge x$ (y^x)	$-1 \times 10^{100} < x \log y < 100$ ($y = 0, x \leqq 0$: ERRO 39; $y = 0, x > 0$: 0; $y < 0, x \neq$ inteiro: ERRO 39)
SIN x COS x TAN x	DEG: $\lvert x \rvert < 1 \times 10^{10}$. RAD: $\lvert x \rvert < \frac{\pi}{180} \times 10^{10}$ GRAD: $\lvert x \rvert < \frac{10}{9} \times 10^{10}$ Em TAN x, os seguintes casos são excluídos. DEG: $\lvert x \rvert = 90(2n-1)$ RAD: $\lvert x \rvert = \frac{\pi}{2}(2n-1)$ GRAD: $\lvert x \rvert = 100(2n-1)$ (n: inteiro)
$\text{SIN}^{-1} x$ $\text{COS}^{-1} x$	$-1 \leqq x \leqq 1$
$\text{TAN}^{-1} x$	$\lvert x \rvert < 1 \times 10^{100}$
LN x LOG x	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$
EXP x	$-1 \times 10^{100} < x \leqq 230.2585092$
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$

Outras funções além das mostradas acima, só podem ser calculadas quando X está dentro do seguinte limite.

$$1 \times 10^{-99} \leqq \lvert x \rvert < 1 \times 10^{100} \quad \text{e} \quad 0$$

(Ex.)
- Ø ^ Ø ENTER → ERRO 39
- Ø ^ 5 ENTER → Ø
- (−4) ^ Ø.5 ENTER → ERRO 39
- −4 ^ Ø.5 ENTER → −2

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

O PC-1500, para aumentar a precisão, utiliza 12 dígitos para cálculos internos da mantissa, e o resultado é arredondado para o 10.º dígito. Por exemplo, 5/9 e 5/9 * 9 são calculados conforme abaixo:

5 / 9 5.55555555555 E−Ø1

↑ é arredondado para o 10.º dígito.

5. 555555556E−01

5 / 9 * 9 4.99999999999 E−ØØ (interno)

↑ é arredondado para o 10.º dígito.

5.

Isto causa uma diferença entre os resultados de cálculos sucessivos e independentes.

Exemplo 1:

$3^2 - 9 =$

Cálculo sucessivo: 3 [SHIFT] [^] 2 − 9 [ENTER] → −9E−11

Cálculo independente: 3 [SHIFT] [^] 2 [ENTER] → 9

−9 → 0

Esta diferença pode também fazer com que o programa não funcione, conforme o planejado em alguns cálculos sucessivos em comparações na declaração IF:

Exemplo 2:

```
10 INPUT A
20 IF A^2 >= 9 THEN 90
```

Com A = 3, 3 ^ 2 resulta em 8.99999999991EØØ o que causará uma conclusão errada da declaração IF.

Para evitar isso, reprograme o cálculo, utilizando variáveis que sejam independentes.

```
10 INPUT A
15 B = A^2
20 IF B >= 9 THEN 50
```

O resultado de A ^ 2 será substituído por uma variável que é usada para o teste do "IF".

Cálculos de potências são baseados em log x e 10^x. Isto causa estas diferenças nos resultados internos.

$$A \wedge B \longrightarrow 10^{B \log A}$$

COMO REGRA, PODE-SE CONSIDERAR QUE O ERRO MÁXIMO NO CÁLCULO DE FUNÇÕES É +/ −1 NO DÍGITO MENOS SIGNIFICATIVO DO VALOR NUMÉRICO MOSTRADO NO VISOR (OU NO DIGITO MENOS SIGNIFICATIVO DA MANTISSA NO CASO DE NOTAÇÃO CIENTÍFICA), DENTRO DOS LIMITES DE CÁLCULO DE CADA FUNÇÃO.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

X. COMPARAÇÃO DE COMANDOS:

PC-1211 vs. PC-1500

X.1. Instruções do PC-1211 disponíveis no PC-1500

Funções

ABS
ACS
ASN
ATN
COS
DEG
DMS
EXP
INT
LOG
LN
π (PI)
SGN
SIN
√ (Raiz Quadrada)
TAN
^ (exponenciação)

3. Comandos

CONT
LIST
NEW
RUN

4. Comandos de Cassete

CHAIN
CLOAD
CLOAD?
CSAVE
INPUT #
PRINT #

2. Declarações

AREAD
USING
CLEAR
DEGREE
END
FOR-TO-STEP
GOSUB
GOTO
GRAD
IF
INPUT
LET
MEM
NEXT
PAUSE
PRINT
RADIAN
REM
RETURN
STOP
THEN
USING

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

X.2. COMANDOS SOMENTE PARA O PC-1500

1. Funções

AND
ASC
CHR$
INKEY$
LEFT$
LEN
MID$
NOT
OR
POINT
RIGHT$
RND
STATUS
STR$
TIME
VAL

2. Declarações

ARUN
BEEP (não PC-1211)
CLS
CURSOR
GCURSOR
GPRINT
DATA
DIM
LOCK
ON ERROR
ON GOSUB
ON GOTO
POINT
RANDOM
READ
RESTORE
TRON
TROFF
UNLOCK
WAIT

3. Comandos

(iguais para o PC-1211)

4. Instruções p/ gravador cassete

CHAIN
CLOAD
CLOAD?
CSAVE
INPUT#
MERGE
PRINT#
RMT OFF
RMT ON

5. Instruções p/ Impressora

COLOR
CSIZE
GCURSOR
GLCURSOR
GPRINT
GRAPH
LCURSOR
LF
LINE
LLIST
LPRINT
RLINE
ROTATE
SORGN
TAB
TEST
TEXT

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Z) TABELA DE REFERÊNCIAS DOS COMANDOS

1. Funções

Função	Abreviações	Observações
ABS	AB.	Valor absoluto
ACS	AC.	$\cos^{-1}$
AND	AN.	exp. AND exp. [lógica E]
ASC	AS.	Converte caracteres para código decimal ASC II. ASC "caracter" variável de caracteres
ASN	AS.	$\sin^{-1}$
ATN	AT.	$\tan^{-1}$
CHR$	CH. CHR.	Converte código ASCII para caracteres CHR$ ASCII valor numerico de código decimal
COS		
DEG		Converte grau, minuto, segundo para grau decimal
DMS	DM.	Converte grau decimal para grau, minuto, segundo
EXP	EX.	e^x
INKEY$	INK. INKE. INKEY.	Variável de caracter = INKEY$ Se uma tecla for pressionada durante a execução do comando INKEY$, o caracter ACII será lido dentro de uma variável de caracter.
INT		Converte o valor fracionário para inteiro, desprezando as casas decimais. INT (10/3) ENTER < Visor > 3
LEFT$	LEF. LEFT.	LEFT$ (variável de caracter, expressão numerica) Toma o número especificado de caracteres do lado esquerdo do cordão de caracteres especificado.
LEN		LEN "caracter" variável de caracter Efetua a contagem do n.º de caracteres do cordão de caracteres.
LOG	LO.	$\log_{10} X$
LN		$\log_e X$
MEM	M. ME.	Visualiza o número restante de passos disponíveis na memória. Igual ao STATUS 0.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Função	Abreviações	Observações
MID$	MI. MID.	MID$(variável de caracter,numérico exp 1, numérico exp 2); Toma caracter(es) do meio do cordão de caracteres especificado
NOT	·NO.	NOT exp. [Negação lógica]
OR		exp. OR exp. [Lógica OU]
π (PI)		Especifica a relação da circunferência: (= 3.141592654)
POINT	POI. POIN.	Retorna um número que representa a estrutura de pontos ativados dentro da coluna dada. POINT expressão numérica.
RIGHT$	RI. RIG. RIGH. RIGHT.	RIGHT$ (variável de caracter, exp. numérica) Toma o número especificado de caracteres do lado direito do cordão de caracteres especificado.
RND	RN.	RND expressão Comando para gerar números aleatórios.
SGN	SG.	Função do sinal
SIN	SI.	seno
√‾ (SQR)	SQ.	raiz quadrada
STATUS	STA. STAT. STATU.	STATUS 0 ou 1 (0) Número de passos de programa disponíveis. (1) Número de passos de programa utilizados.
STR$	STR.	STR$ expressão numérica Converte números para cordão de caracteres.
TAN	TA.	tangente
TIME	TI. TIM.	(1) TIME = mês, dia, hora, minuto, segundo. A função de tempo posiciona mês, dia, hora, minuto, segundo. (2) TIME [chama a data e hora]
VAL (valor)	V. VA.	VAL { "caracter" / variável de caracter } Converte os cordões de caracteres para números.
^		Exponenciação

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

2. DECLARAÇÕES:

Comando	Abreviação	Observação
AREAD (auto leitura)	A. AR. ARE. AREA.	AREAD variável Quando executa programas por teclas definidas, AREAD introduz o conteúdo do visor na variável especificada.
ARUN (auto execução)	ARU.	ARUN Comando para iniciar a execução do programa automaticamente quando o PC-1500 é ligado.
BEEP	B. BE. BEE.	BEEP exp1, exp2, exp3 Comando de som. Ativa as funções geradoras de som. ON/OFF, especifica o tom e a duração dos sons.
CLEAR	CL. CLE. CLEA.	Comando para limpar todos os dados (variáveis)
CLS (limpeza)		Comando para limpar o visor. Apaga o visor.
CURSOR	CU. CUR. CURS. CURSO.	(1) CURSOR exp. (0< = exp. < = 25) especificação da posição inicial do visor. (2) CURSOR cancela a especificação anterior.
DATA	DA. DAT.	DATA exp, exp,... Dados a serem lidos usando o comando READ.
DEGREE	DE. DEG. DEGR. DEGREE.	Especificação de modo angular. O grau [°] é designado.
DIM (dimensão)	D. DI.	(1) DIM nome de variável (exp) (2) DIM nome de variável (exp) * exp3 (3) DIM nome de variável (exp1, exp2) nomes de variáveis: A, B, C$, D$, etc.... ( ): especifica o tamanho e a dimensão da matriz. exp3: especificação do espaço de dígitos.
END	E. EN.	Comando que indica o fim do programa.
FOR TO STEP	F. FO. STE.	(1) Variável numérica = exp1 e exp2 Início do "loop" FOR-NEXT. Usada em correspondência com o comando NEXT. (2) FOR variável numérica = exp1 e exp2 STEP exp3 exp1: expressão inicial exp2: valor final exp 3: intervalo para aumentar com cada "loop".

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviação	Observações
GCURSOR (cursor gráfico)	GC. GCU. GCUR. GCURS. GCURSO.	Especifica a posição do visor através de unidades de pontos. GCURSOR expressão (0 < = expressão < = 155) ou (& 0 < = expressão < = & 9B)
GOSUB	GOS. GOSU.	GOSUB { expressão / "caracter" / variável de caracter } Comando de desvio para subrotina. Move a execução para linha ou rótulo especificado. Declarações neste ponto são executadas como uma subrotina. Usada em correspondência com o comando RETURN.
GOTO.	G. GO. GOT.	Comando de desvio. GOTO { expressão / "caracter" / variável de caracter } Move a execução para a linha ou rótulo especificado.
GPRINT (impressão gráfica)	GP. GPR. GPRI. GPRIN.	Visualiza na impressora, o conteúdo dado no visor. (1) GPRINT "00 00 0: ..." (dentro de " " estão os números hexadecimais) (2) GPRINT 0; 0; ... (3) GPRINT & 0; & 0; ...
GRAD	GR. GRA.	designação de modo angular. grado ([g]) é designado.
IF		(1) IF comando de execução de expressão condicional. (2) IF comando de execução de expressão aritmética. Avalia as condições dadas e também muda a execução para a próxima linha ou executa comandos.
INPUT	I. IN. INP. INPU.	(1) INPUT variável, variável, ... (2) INPUT "caracter"; variável, "caracter"; variável. (3) INPUT "caracter"; variável, "caracter"; variável.
LET	LE.	(1) LET variável numerica = expressão (2) LET variável de caracter = "caracteres" (3) LET variável de caracter = variável de caracter LET segue comandos IF. Ele pode ser omitido em outros casos.
LOCK	LOC.	Trava o computador no MODE que está.
NEXT	N. NE. NEX.	NEXT variável numerico Mostra o fim do "loop" FOR-NEXT. A variável numérica deve ser o mesmo que a variável numérica que segue o comando FOR.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviações	Observações
ON ERROR	O. ER. ERR. ERRO.	ON ERROR GOTO expressão Comando para detectar erro.
ON GOSUB	O. GOS. GOSU.	ON expressão GOSUB exp 1, exp 2, exp 3 Comando de desvio para subrotina. Especifica o lugar de desvio para (exp 1, exp 2, exp 3) pelo valor da expressão.
ON GOTO	O. G. GO. GOT.	ON expressão GOTO exp 1, exp 2, exp 3 Comando de desvio. Especifica lugar de desvio para (exp 1, exp 2, exp 3) pelo valor da expressão seguindo ON.
OPN	OP.	OPN "Nome do dispositivo." Este comando pode ser usado quando o PC-1500 é conectado com a Interface. Para detalhes, consulte o Manual de Instruções da Interface.
PAUSE	PA. PAU. PAUS.	Forma igual ao comando PRINT. Visualiza o conteúdo especificado cerca de 0.85 seg, e então executa o programa.
POINT	POI. POIN.	POINT expressão (0 < = expressão < = 155) (& 0 < = expressão < = & 9 B) Lê a estrutura de pontos da informação visualizado no ponto especificado. ex. A = POINT 56
PRINT	P. PR. PRI. PRIN.	(1) PRINT { expressão / "caracter" / variável de caracter } (2) PRINT { expressão / "caracter" / variável de caracter } { expressão / "caracter" / variável de caracter } (3) PRINT { expressão / "caracter" / variável de caracter } { expressão / "caracter" / variável de caracter } ; . . . ; { expressão / "caracter" / variável de caracter } { expressão / "caracter" / variável de caracter }
RADIAN	RAD. RADI. RADIA.	Designação de modo angular. O Radiano ([rad]) é designado.
RANDOM	RA. RAN. RAND. RANDO.	Muda a origem da geração de números aleatórios para uso do comando RND.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviações	Observações
READ	REA.	READ variável, variável ,... comanda a leitura de dado. Introduz dados da declaração DATA nas variáveis especificadas.
REM (observação)		REM...notas de documento...
		Especifica observações que não serão executados.
RESTORE	RES. REST. RESTO. RESTOR.	(1) RESTORE expressão Muda a origem da leitura de dados através dos comandos READ. (2) RESTORE Inicia no começo da primeira declaração.
RETURN	RE. RET. RETU. RETUR.	Volta a continuar a execução após a declaração GOSUB que invoca esta sub-rotina.
STOP	S. ST. STO.	Comando para parar a execução do programa.
THEN	T. TH. THE.	Define o comando de execução para declaração IF. Comandos de desvio são possíveis somente para definir comandos de execução para comando IF. THEN expressão "caracter" variável de caracter
TRON (trace on)	TR. TRO.	Especifica o modo para efetuar depuração.
TROFF (trace off)	TROF.	Cancela o modo para efetuar a depuração.
UNLOCK	UN. UNL. UNLO. UNLOC.	Cancela o modo LOCK.
USING	U. US. USI. USIN.	(1) USING "###•### " (2) USING "&&& ..... &&&" (3) PRINT USING "Formato";.... (4) USING (5) PRINT USING Especifica a formulação do visor através do valor da expressão.
WAIT	W. WA. WAI.	WAIT expressão (0 < = expressão < = 65535) Especifica a duração de tempo da visualização quando o comando PRINT é usado. WAIT sem argumento cancela a especificação anterior (duração = infinita).

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

3. COMANDOS

Comando	Abreviação	Observação
CONT (continue)	C. CO. CON.	Reinicia a execução de programas que tenha sido parada temporariamente. Efetivo no modo RUN.
LIST	L. LI. LIS.	Comando que efetua a listagem do programa. Efetivo no modo PRO.
NEW		(1) NEW (2) NEW 0 No modo PRO, limpa o programa de todas as variáveis.
RUN	R. RU.	(1) RUN (2) RUN expressão (3) RUN "caracter" variável de caracter Efetivo no modo RUN.

4. COMANDOS DE CASSETE

Comando	Abreviação	Observações
CHAIN	CHA. CHAI.	Comando de transmissão. Quando é usado no meio de um programa, lê os programas da fita (transmite) e executa o programa. (1) CHAIN "nome de arquivo" (CHAIN-1 "nome de arquivo") (2) CHAIN "nome de arquivo", expressão (CHAIN-1 "nome de arquivo", expressão) (3) formas abreviadas do "nome de arquivo" para (1) e (2).
CLOAD	CLO. CLOA.	Comando de transmissão. Este comando transmite programa ou conteúdo da reserva, da fita para a memória do computador. (1) CLOAD (CLOAD-1) (2) CLOAD "nome de arquivo" (CLOAD-1 "nome de arquivo")

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviação	Observações
CLOAD?	CLO.? CLOA?	Comando de comparação. Este comando compara o programa da memória ou o conteúdo da reserva com o conteúdo gravado na fita. (1) CLOAD? (CLOAD?-1) (2) CLOAD? "nome de arquivo" (CLOAD?-1 "nome de arquivo"
CSAVE guardar na fita	CS. CSA. CSAV.	Este comando grava na fita o conteúdo do programa e a memória de reserva. (1) CSAVE (CSAVE-1) (2) CSAVE "nome de arquivo" (CSAVE-1 "nome de arquivo")
INPUT#	.# N.# NP.# NPU.#	Comando de transmissão de dados. Este comando transmite dados gravados na fita para as variáveis especificados. Toma a mesma forma que o comando PRINT#.
MERGE	MER. MERG.	Comando de transmissão. Este comando transmite programas da fita para o computador. Toma a mesma forma que o comando CLOAD. Neste comando serão retidos programas gravados anteriormente como eles estão e os programas novamente lidos serão acrescentados.
PRINT#	P.# PR.# PRI.# PRIN.#	Comando de gravação de dados. Este comando grava na fita os dados armazenados no PC-1500. (1) PRINT# nome de variável, nome de variável,... (PRINT#-1, nome de variável, nome de variável),... (2) PRINT# "nome de arquivo"; nome de variável,... (PRINT#-1, "nome de arquivo"; nome de variável,...)
RMT OFF desligado)	RM. OF. RMTOF.	Este comando cancela a função remoto do terminal REM 1. (para o segundo gravador).
RMT ON ligado)	RM.O. RMTO.	Este comando, reposiciona a função remoto do terminal REM 1. (para o segundo gravador).

5. COMANDOS DA IMPRESSORA

Comando	Abreviação	Observações
COLOR	COL. COLO.	Especifica a cor dos caracteres. COLOR expressão (0 < = expressão < = 3)
CSIZE tamanho do caracter)	CSI. CSIZ.	Especifica o tamanho dos caracteres a serem impressos. CSIZE expressão (1 < = expressão < = 9)

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviação	Observações
GLCURSOR (cursor da linha gráfica)	GL. GLC. GLCU. CLCUR. GLCURS. GLCURSO	Comando que move a posição da caneta do ponto inicial para uma coordenada X e Y. Válido somente para o modo GRAPH. GLCURSOR (exp 1, exp 2)
GRAPH (gráfico)	GRAP.	Este modo é usado para traçar gráficos e ilustrações
LCURSOR (cursor de) linha)	LCU. LCUR. LCURS. LCURSO.	Move a caneta para a posição desejada na impressora.
LF (avanço de linha)		Efetua o avanço de papel até o número de avanço de linha mostrado pela expressão. Válido somente para o modo TEXT. LF expressão.
LINE	LIN.	Comando para traçar linha. Válido somente para modo GRAPH. (1) LINE (exp 1, exp 2) – (exp 3, exp 4) (2) LINE (exp 1, exp 2) – (exp 3, exp 4), exp 5, exp 6 (3) LINE (exp 1, exp 2) – (exp 3, exp 4), exp 5, exp 6 exp 5: especifica o tipo de linha exp 6: especifica a cor B: especifica um desenho de caixa (4) LINE (exp 1, exp 2) – (exp 3, exp 4) – ... ... (exp 11, exp 12)
LLIST	LL. LLI. LLIS.	Listagem do Programa.
LPRINT	LP. LPR. LPRI. LPRIN.	Imprime conteúdo especificado. Válido somente para modo text. Toma a mesma forma que os comandos PRINT.
RLINE (linha relativa)	RL. RLI. RLIN.	Comando que traça linhas com a posição da caneta como ponto inicial. Válido somente para modo GRAPH. Toma a mesma forma que os comandos LINE.
ROTATE (rotater)	RO. ROT. ROTA. ROTAT.	Especifica a direção dos caracters a serem impressos (direção de impressão). Válido somente para o modo GRAPH. ROTATE expressão (0 < = expressão < = 3)

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

Comando	Abreviação	Observações
SORGN	SO. SOR. SORG.	Este comando especifica a posição da caneta presente como o novo ponto inicial (ponto de origem). Válido somente para modo GRAPH.
TAB		Especifica a posição da caneta. Válido somente para modo TEXT. (1) TAB Expressão (2) LPRINT TAB Expressão;...
TEST	TE. TES.	Verificação da cor. Quando executado, testa traçando um quadrado de 5 mm por 5 mm em cada cor.
TEXT	TEX.	Especificação do modo TEXT. Este modo imprime caracteres e números.

www.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

CERTIFICADO DE GARANTIA

A Sharp do Brasil S.A. Indústria de Equipamentos Eletrônicos, nos limites fixados neste certificado, assegura, como fabricante, ao comprador-consumidor da calculadora ao lado identificada, garantia contra qualquer defeito de material ou de fabricação apresentado no prazo de 365 dias, contado a partir da data da emissão da nota fiscal de venda.

Limita-se a responsabilidade da Sharp do Brasil S.A. Indústria de Equipamentos Eletrônicos em substituir as peças defeituosas da calculadora, desde que seu técnico credenciado constate falha em condições normais de uso. Não estão incluídos nesta garantia: CABO MÓVEL / SUPORTE DE BOBINA / BOBINA DE PAPEL / ADAPTADOR AC/ CONJUNTOS DE CANETAS / GABARITOS / CAPAS.

A presente garantia ficará sem efeito, se a calculadora sofrer qualquer dano provocado por acidente, agentes da natureza, uso em desacordo com o manual de instruções, ou por ter sido ligada à rede elétrica imprópria ou sujeita a flutuações excessivas, ou ainda no caso de apresentar sinal de violação de seu selo de garantia, ajuste ou conserto por pessoa não autorizada, bem assim por defeito oriundo de caso fortuito ou força maior.

Obriga-se a Sharp do Brasil S.A. Indústria de Equipamentos Eletrônicos a prestar os serviços, tanto gratuitos como remunerados, exclusivamente nas localidades nas quais mantenha oficinas de serviços próprias ou devidamente autorizada para efetuar consertos em calculadoras.

O comprador-consumidor residente em outra localidade será, pois, o único responsável pelas cespesas e risco de transporte, de ida e volta, da calculadora à oficina mais próxima da Sharp do Brasil S.A. Indústria de Equipamentos Eletrônicos. Se a calculadora for transferida no período de garantia, esta ficará cecida de pleno direito, continuando em vigor até a expiração de seu prazo, contado da data da aquisição pelo primeiro comprador-consumidor.

A presente garantia somente será válida se devidamente preenchida pela Sharp do Brasil S.A. Indústria de Equipamentos Eletrônicos ou pelo revendedor no ato de aquisição da calculadora, e não apresentar rasuras ou modificações. É necessáro, ainda, que este certificado seja apresentado juntamente com a nota fiscal a cada solicitação de prestação de serviços de conserto.

CERTIFICADO DE GARANTIA

(Deve ser preenchido pelo revendedor no ato da venda).

A Sharp do Brasil S.A. Industria de Equipamentos Eletrônicos, da garantia a esta calculadora:

Modelo e n.º
de Fabricação

Adquirido pelo Sr.: .....

Endereço: .....

N.º ..... Bairro .....

Cidade ..... Estado .....

Nos termos contidos neste certificado de garantia.

Revendedor:

Carimbo e assinatura

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

É sabido que as empresas, como as pessoas, desenvolvem vocação e, quando atuam a partir dela atingem a forma mais completa do seu desempenho. A SHARP é, por vocação, um grupo empreendedor.

A SHARP é um conglomerado de empresas, atuando nas áreas industrial, comercial, financeira e de serviços, que tem como finalidade a fabricação e comercialização de TV em cores, vídeo-cassetes, calculadoras eletrônicas, faturadoras, aparelhos de som e outros. Para realização desses objetivos o grupo dispõe de modernas fábricas, sendo a unidade industrial (SHARP DO BRASIL S/A) sediada na Zona Franca de Manaus, uma das maiores do país e das mais modernas da América Latina; implantou unidades de comercialização e assistência técnica nas principais cidades brasileiras e desenvolveu empresas fornecedoras de suprimentos e prestadoras de serviços, gerando um processo de verticalização que se inicia na fabricação de componentes, abrangendo também todos os serviços auxiliares de comercialização, incluindo a atividade de artes gráficas e transportes.

O bom desempenho da empresa, em setores fortemente competitivos, deve ser atribuído, não somente à qualidade de seus produtos ou à consagração da marca, mas à prática consistente de preceitos empresariais em tota consonância com diretivas governamentais visando a substituição e nacionalização de produtos e insumos importados, máxima produtividade de cada cruzeiro investido e manutenção de um ritmo saudável de crescimento com máxima estabilidade possível de preços. Nosso crescimento sólido, equilibrado e na direção de objetivos previamente definidos consagra a competência dessa empresa genuinamente nacional.

Pioneira, agressiva, sensível e altamente contributiva para o progresso nacional, novos projetos, no campo da pesquisa tecnológica, encontram-se em avançada fase de desenvolvimento, oriundos de estudos feitos no Brasil e voltados para nossas necessidades.

Isto é a SHARP Assim é a SHARP no desempenho de sua vocação.

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

WWW.
PC-1500
.INFO

Do not sell this PDF !!!

All and more about Sharp PC-1500 at http://www.PC-1500.info

www.
PC-1500
.INFO

SHARP DO BRASIL S.A.
INDUSTRIA DE EQUIPAMENTOS ELETRÔNICOS

Do not sell this PDF !!!