



**ACESSÓRIOS**

# CCE RAM CARD

**64K/128K**

---

## **MANUAL DE INSTRUÇÕES**

---

**RAM 64K AS-4080**

**RAM 128K AS-4090**





CCE  
RAM CARD  
64K/128K

---

*MANUAL DE INSTRUÇÕES*

---

1ª EDIÇÃO

Todos os direitos desta publicação são reservados, portanto este manual não pode ser reproduzido parcialmente ou integralmente, sejam quais forem os meios empregados, eletrônicos, mecânicos ou quaisquer outros, sem prévia autorização por escrito da CCE .

CCE - Indústria e Comércio de Componentes Eletrônicos S/A - DIVISÃO DE MANUAIS DE INFORMÁTICA Av. Otaviano Alves de Lima, 2724 CEP 02501 - São Paulo- SP - Brasil

# SUMÁRIO

## CAPÍTULO 1

1 - INTRODUÇÃO

## CAPÍTULO 2

2 - INSTALAÇÃO

## CAPÍTULO 3

4 - UTILIZAÇÃO

4 - SOFTWARE

4 - MOVEDOS

5 - RAMEXPAND

5 - PSEUDO-DISCO

## CAPÍTULO 4

6 - MOVEDOS -Software de Relocação

6 - PROPRIEDADES DO DOS RELOCADO

7 - UTILIZAÇÃO COM CDOS

7 - UTILIZAÇÃO COM RAM CARD 16

8 - LIMITAÇÕES

## CAPÍTULO 5

- 9 - RAMEXPAND
- 9 - CARACTERÍSTICAS
- 10 - ORGANIZAÇÃO DE DADOS
- 10 - ALOCAÇÃO DE MEMÓRIA
- 13 - USO DE RAMEXPAND
- 14 - IMPLANTAÇÃO DOS COMANDOS
- 15 - CHAMADA DE RAMEXPAND
- 15 - INICIALIZAÇÃO
- 16 - ACOPLAMENTO DE BLOCOS DE MEMÓRIA
- 17 - PRECAUÇÕES
- 19 - MANIPULAÇÃO DE ERROS
- 19 - RESUMO DOS ERROS
- 20 - RESUMO DAS FUNÇÕES DISPONÍVEIS
- 21 - SINTAXE E DESCRIÇÃO DOS COMANDOS
- 25 - TABELA DE PARÂMETROS

## CAPÍTULO 6

- 28 - PSEUDO DISCO
- 29 - CARACTERÍSTICAS E REQUISITOS

- 29 - AJUSTE
- 32 - INSTALAÇÃO DO PSEUDO-DISCO
- 32 - RECONEXÃO
- 32 - PRECAUÇÕES E OBSERVAÇÕES
- 33 - UTILIZAÇÃO DO PSEUDO-DISCO
- 34 - DESCRIÇÃO DO FUNCIONAMENTO

## CAPÍTULO 7

- 35 - PSEUDO-DISCO CP/M
- 36 - AJUSTE DO PSEUDO-DISCO
- 39 - INSTALAÇÃO DO PSEUDO-DISCO
- 39 - UTILIZAÇÃO DO PSEUDO-DISCO
- 40 - REQUISITOS E LIMITAÇÕES

## CAPÍTULO 8

- 41 - INFORMAÇÕES TÉCNICAS
- 41 - CONTROLE DE PLACAS 64 E 128K
- 43 - UTILIZAÇÃO DA PLACA DE RAM





# CAPÍTULO 1

## INTRODUÇÃO

As placas RAM CARD-64/128 são expansões de memória que dão ao seu EXATO 64k ou 128K bytes adicionais ao sistema já existente.

Sua configuração em bancos de 16K asseguram total compatibilidade ao Software que exige expansão de memória.

Junto com a RAM CARD-64/128, é fornecido 1 disco com programas que permitem:

- liberar o espaço ocupado pelo sistema operacional CDOS, transferindo-o para RAM CARD 64/128;
- substituir acionadores de disco tornando os acessos a drives mais rápidos;
- armazenar matrizes e sub-rotinas.  
(Tanto em CDOS como em CP/M)

## INSTALANDO A RAM CARD-64/128K

Se você possui uma expansão de RAM no slot 0, a RAM CARD-64/128 pode ser instalada em qualquer slot disponível, mas, se o slot 0 estiver vazio, deverá ocupar o mesmo, pois a maioria do software existente espera encontrar a expansão de memória nesse slot.

Atenção: nunca instale ou remova placas do seu MICRO com o equipamento ligado, pois este e a placa poderão ser danificados!

Com a chave de alimentação desligada remova a tampa do gabinete. Selecione um dos slots (0-7) (de preferência o 0). Descarregue a eletricidade estática (que você pode ter acumulado) colocando a mão na carcaça da fonte de alimentação (caixa metálica preta à sua esquerda). Isso evitará danos ao equipamento, pois este é sensível a cargas eletrostáticas.

Remova a placa de expansão de sua embalagem .

Insira a placa no slot escolhido, com a parte mais longa apontando para você, ligeiramente inclinada para cima, com a parte posterior sendo encaixada primeiro; pressione delicadamente até que ela

se encaixe totalmente. Não ligue o equipamento até ter certeza que a inserção da placa está correta.

Os mesmos cuidados devem ser tomados por ocasião da inserção de qualquer placa adicional de expansão.

## CAPÍTULO 3

### UTILIZAÇÃO DAS PLACAS CCE RAM CARD 64/128 K

A presença da placa de expansão no SLOT 0 é reconhecida automaticamente pelo Software, que em sua maioria está hoje disponível para esta configuração.

Eis alguns exemplos de Software que reconhecem automaticamente os primeiros 16K Bytes das expansões 64/128:

FORTRAN	SUPER VISICALC
PASCAL	CP/M 60
JANELA MÁGICA II	FORTRAN 80
DBASE II	COBOL 80
	etc.

### SOFTWARE

Daremos em seguida uma descrição geral do Software que é fornecido com as placas de expansão CCE RAM CARD 64/128.

### MOVEDOS

Reloca o sistema operacional CDOS para a memória da expansão, liberando assim espaço (10kb) na memória do micro para outras aplicações.

## RAMEXPAND

É um conjunto de sub-rotinas, acessíveis via CCE BASIC , que permite transferências rápidas de dados do espaço de endereçamento normal para o da placa de expansão e vice-versa. Opera com o CDOS normal ou relocado.

## PSEUDO-DISCO OU DISCO VIRTUAL

É um programa que faz a placa de 64/128K emular um acionador de disco. Sendo este um processo puramente eletrônico, livre de dispositivos mecânicos, o tempo de resposta para um comando do CDOS é muito menor.

## CAPÍTULO 4

### MOVEDOS— Software de Relocação

O programa descrito neste capítulo libera uma parte da memória (aproximadamente 10K Bytes) do endereçamento do microprocessador, movendo o CDOS para a expansão.

### PROPRIEDADES DO CDOS RELOCADO

O sistema operacional relocado retém todas as capacidades do original, incluindo a capacidade de inicializar discos. Quando um disco é inicializado através do sistema relocado, ele é transferido, resultando num disco que, quando carregado, moverá automaticamente o sistema para a expansão.

Assim, o programa que efetua a relocação só precisa ser efetuado uma vez. E os respectivos discos já terão esta característica.

O programa foi projetado para operar com as placas RAM CARD-64/128 em qualquer slot e reloca apropriadamente o sistema operacional nesta posição.

A busca das placas é efetuada a partir do SLOT 0 para cima, até que uma placa de 64/128K seja encontrada. Caso exista uma placa de 16K no Slot 0, ocorrerão problemas, pois neste programa o EXATO encontrará primeiro esta placa, e tentará carregar nela o CDOS, sem conseguir. O uso da versão MOVEDOS.RAM 16 evita este problema.

## UTILIZAÇÃO COM CDOS

Insira o disco CCE RAMCARD 64/128K (que acompanha a placa) com o lado 1 voltado para cima e ligue o equipamento. Selecione a opção <1> do menu (MOVEDOS); aparecerá na tela a mensagem:

```
MOVEDOS - VER 1.0
CCE SOFT, MARÇO /85
CARREGANDO CDOS (C/INIT) NO SLOT 0
```

## UTILIZAÇÃO COM RAM CARD-16

Caso haja CCE RAM CARD-16K no sistema, utilize a opção <2>, MOVEDOS.RAM 16.

# LIMITAÇÕES

O Pseudo-Disco pode utilizar até 256K Bytes de memória (RAM) distribuída pelas diversas placas de expansão.

O espaço no bloco de configuração de E/S não estará disponível após a instalação do Pseudo-Disco.

PLACA PRINCIPAL	1.BANCO	2.BANCO
\$FFFF=====	\$FFFF=====	\$FFFF=====
! ROM'S !	!Monitor!	!Monitor
!da placa !	!CCE- !	!
! do !	!Integer!	!Reloca -
! EXATO !	!PR.AID !	! ção dos
! !	! !	!Arquivos
!(CCE - !	! !	!** Anti-
! BASIC) !	! !	! gos.
\$D000=====	\$D000=====	\$D000=====
! EXATO !		
! E/S !		
\$C000=====		
! CDOS !		
\$BE00=====	=HIMEM (Somente mostra 2	
! LIVRE !	bancos)	
! PARA !		
! USO !		
\$9600!		



## RAMEXPAND

O RAMEXPAND compõe-se de uma série de funções que tem o objetivo de aumentar a capacidade da memória para uso com BASIC. O RAMEXPAND permite armazenar programas ou dados que poderão ser usados a qualquer hora.

## CARACTERÍSTICAS

## 1- TERMINOLOGIA

Usaremos o termo RAM expandida para designar a memória da placa CCE RAMCARD 64/128K ou placas de 16K (estas não necessariamente usadas).

## 2- UTILIZAÇÃO DA MEMÓRIA

A maior parte do RAMEXPAND reside na placa de 64/128K com o Sistema Operacional relocado, ocupam o banco de 4K denominado 2 B. O RAMEXPAND utiliza a página \$BD00 até \$BDFF, e usa-se o indicador HIMEM para proteção desta área.

O RAMEXPAND consiste de 3 arquivos que sempre trabalham em conjunto:

- a) RAMEXPAND
- b) RAMEXPAND.LOW
- c) RAMEXPAND.LOAD

Todos os 3 são carregados através da opção <3> do menu (RAMEXPAND).

## ORGANIZAÇÃO DE DADOS

O RAMEXPAND pode guardar e recuperar programas inteiros (ou dados), ou apenas parte deles, armazenados na RAM expandida. Costuma-se atribuir um nome ao segmento desta RAM que armazena uma informação. Estes "nomes" são números inteiros positivos (diferentes de zero) e menores que 32767.

## ALOCACÃO DE MEMÓRIA

Obs.: E/S = ENTRADA E SAÍDA em porticos.

A parte da memória ocupada pelo firmware é também ocupada pelos outros bancos de memória da placa de 64/128K. O firmware, a área de E/S da máquina, a entrada e a saída podem ser desativadas sem perdas para a capacidade de armazenamento. Portanto, a memória expandida, Monitor e E/S são separáveis, podendo ser ativados somente um de cada vez. Consequentemente a memória expandida não pode ser operada diretamente.

A E/S deve ser usada para acesso ou armazenamento considerado secundário.

Organização:

Subdivide-se a placa de 64K em 4 bancos de 16K e a de 128K em 8 bancos. O banco de 16K é dividido em 1 bloco de 12K e um de 4K. A representação desta distribuição está na figura da página seguinte.

Para se especificar determinado bloco de memória, deve-se:

1- Especificar a seção de 32K.

256 = 1ª seção

512 = 2ª seção

1024 = 3ª seção

2048 = 4ª seção (correspondente ao "nibble" mais significativo).

2- Especificar o bloco:

8 = 1º bloco de 12K

16 = 2º bloco de 12K

32 = bloco de 4K associado ao 1º de 12K

64 = bloco de 4K associado ao 2º de  
12K (correspondente aos dois  
"nibbles" menos significativos  
na figura).

# Organização dos bancos de memória para 128K

\$FFFF=====		Designação do banco de 32K
! ! ! !		
! ! ! !		
!12K ! !12K !		
! ! ! !		
\$E000!(8) !====--!(16)!=====		
! !4K ! ! !4K !	256	
! ! (32)! ! ! (64)!		
\$D000=====		
16K BANCO 1 16K BANCO 2		
\$FFFF=====		
! ! ! !		
! ! ! !		
!12K ! !12K !		
! ! ! !		
\$E000!(8) !====--!(16)!=====		
! !4K ! ! !4K !	512	
! ! (32)! ! ! (64)!		
\$D000=====		
16K BANCO 3 16K BANCO 4		
\$FFFF=====		
! ! ! !		
! ! ! !		
!12K ! !12K !		
! ! ! !		
! ! ! !		
\$E000!(8) !====--!(16)!=====		
! !4K ! ! !4K !	1024	
! ! (32)! ! ! (64)!		
\$D000=====		
16K BANCO 5 16K BANCO 6		

(continuação)

	Designação do banco de 32k
\$FFFF=====	
! ! ! !	
! ! ! !	
!12K ! !12K !	
! ! ! !	
\$E000!(8) !====--!(16) !====	
! !4K ! !4K !	2048
! ! (32) ! ! (64) !	
\$D000!=====	
16K BANCO 7 16K BANCO 8	

## USO DO RAMEXPAND

Obs.: Quando nos referimos a slot queremos dizer um dos oito conectores de expansão.

### 1) INSTALAÇÃO DO RAMEXPAND NO SISTEMA

Carregue o RAMEXPAND através da opção <4> do menu, que executa o RAMEXPAND.LOAD. Este carrega os outros segmentos do RAMEXPAND (inclusive o que reside no espaço entre \$BD00 e \$BDFF, sendo que o indicador HIMEM é baixado para proteger este segmento), usando o comando CALL 10, ou através da funçãoUSR(CCE BASIC). Em seguida o RAMEXPAND.LOAD pergunta em qual slot deve-se carregar o RAMEXPAND (ENTRE COM O SLOT-DESTINO DE RAMEXPAND). Especifique o nº do slot em que está a placa de 64/128K.

TIPO DE PLACA (especificar devidamente)

- a) 128K ou 64K
- b) 32K
- c) 128K ou 64K, com RAM CARD-16
- d) 32K com RAM CARD-16

Isto feito, a instalação está pronta. Para se mover dados ou programas, são necessários 2 procedimentos de inicialização, que serão descritos mais adiante.

## IMPLANTAÇÃO DOS COMANDO

A variável de comando para chamar o RAMEXPAND deve obrigatoriamente ser chamada CM\$; ela contará com 5 valores, separados por vírgula, podendo estes ser um número ou uma variável.

Ex: CM\$ = "V,W,X,Y,Z" As variáveis representam:

V - Função      nº inteiro, variando de 0 a 14; identifica a parte a ser executada.

W - Segmento - nº inteiro de 1 a 32767, é o código do segmento de memória.

- X - Código de - É uma variável, obrigatoriamente, sob a qual o RAMEXPAND responde (em código) se a função tem erros ou não.
- Y - Início - Representa o nome de uma matriz ou o início de uma área de programa.
- Z - Fim - Pode representar o fim de uma área de programa.

## CHAMADA DE RAMEXPAND

Use a variável de comando CM\$ seguida de CALL 10. Repita este comando a cada chamada do RAMEXPAND.

## INICIALIZAÇÃO

Após a instalação do RAMEXPAND ele deverá ser inicializado. Isto é feito através de um comando com código de função 14. Essa string deve ser da forma:

CM\$ = "14,0,A,0,0"

em que A é uma variável de erro, definida antes da edição do comando.

Obs.: VARIÁVEL STRING : é uma sequência ou cadeia de caracteres alfanuméricos.

## ACOPLAMENTO DE BLOCOS DE MEMÓRIA

Você deverá especificar a RAM expandida após a inicialização. Existe uma função de correção para acoplar blocos de memória. O procedimento para acoplagem é o seguinte:

- a) Calcule o parâmetro do bloco adicionando 128 ao nº do slot e a designação da seção de 32K.

Ex: O parâmetro do primeiro bloco de 12K no slot 3 é dado por:

3--- Nº do slot

128--- Placa de expansão 64/128K

8--- Indicativo do 1º bloco de 12K

256--- Indicativo do 1º bloco de 32K

---

395

Este parâmetro (395) deve ser incluído na segunda posição da string de comando. Esta será, então:

CM\$ "9,395,A,0,0" (A - variável de erro)



## PRECAUÇÕES

### PROTEÇÃO DE RAMEXPAND

O bloco ocupado pelo RAMEXPAND (4k com designação 64K) não deve ser especificado. Tampouco deve ser especificado o bloco que contém o CDOS.

### AJUSTE DE MAXFILES

O RAMEXPAND.LOW não pode ser movido depois de instalado. Portanto, o MAXFILES, depois de instalado o RAMEXPAND não pode ser aumentado.

### RECARGA DO RAMEXPAND

O HIMEM é reduzido para a proteção do endereço inferior do RAMEXPAND, quando o programa RAMEXPAND.LOAD for usado. Um comando MAXFILES evita uma redução maior do HIMEM.

### REPOSIÇÃO DOS PONTOS DE ENTRADA DE RAMEXPAND

O valor de HIMEM frequentemente coincide com o indicador de entrada de RAMEXPAND. Este último é primeiramente armazenado nas posições 11 e 12, portanto se

elas forem alteradas, torna-se necessário restabelecê-las antes de usar o RAMEXPAND.

## DECLARAÇÃO DE VARIÁVEIS

É necessário declarar antecipadamente as variáveis que serão utilizadas na string de comando; basta para isso atribuir um valor qualquer à variável antes de chamar RAMEXPAND.

# MANIPULAÇÃO DE ERROS

## DESCRIÇÃO

O resultado de uma execução é descrito por um código devolvido por RAMEXPAND (que para isso altera uma das variáveis da string de comandos). Só haverá indicação de erro se a string de comando CMS for encontrada. A função será executada caso não haja erros. E se houver, este só será indicado através do código, se a variável de erro for especificada.

Caso a variável não definida anteriormente for especificada como variável de erro, este, mesmo que exista, não será acusado. Se nenhuma variável for encontrada, a variável erro será alterada, e nenhuma ação executada.

Normalmente, caso seja encontrada uma condição de erro, nenhuma ação será executada.

## RESUMO DE ERROS

1. Função ilegal
2. Variável não encontrada
3. Bloco de memória já em uso
4. Bloco de memória não encontrado
5. Parâmetro de seleção de bloco ilegal
6. O bloco de memória não é RAM
7. Nome do segmento já em uso

8. Segmento não encontrado
9. Nome do segmento ilegal
10. Programa não especificado
11. Matriz não especificada
12. Espaço insuficiente para armazenar dados na memória expandida
13. Espaço insuficiente no endereçamento real para recuperar o programa
14. Espaço insuficiente no endereçamento real para carregar a string.

## RESUMO DAS FUNÇÕES DISPONÍVEIS

Cada uma das seguintes funções é indicada pelo primeiro valor da string de comando:

- 0 - Função nula
- 1 - Armazena todo o programa na RAM
- 2 - Busca o programa na RAM expandida e o executa
- 3 - Armazena sub-rotina ou segmento de programa na RAM expandida
- 4 - Procura sub-rotinas ou segmento de programas na RAM expandida
- 5 - armazena o conteúdo de uma matriz na RAM expandida
- 6 - Busca o conteúdo de uma matriz na RAM
- 7 - Troca o conteúdo de uma matriz entre a memória normal e a RAM expandida
- 8 - Apaga segmento da RAM expandida
- 9 - Conecta novo bloco de RAM expandida
- 10 - Desconecta o bloco de RAM expandida

- 11 - Localiza e traz espaços disponíveis na RAM expandida
- 12 - Localiza o nome do próximo segmento
- 13 - Sem definição
- 14 - Fiscaliza o RAMEXPAND
- 15 - Armazena setor de CCE BASIC na RAM expandida
- 16 - Localiza um setor de CCE-BASIC na RAM expandida

## SINTAXE E DESCRIÇÃO DOS COMANDOS

A string de comando tem a forma

CM\$ = "A,B,C,D,E"

Se A=0- FUNÇÃO NULA - O comando nada executa.

A=1- FUNÇÃO PROGRAMA - Armazena o programa existente em outro segmento da memória;

B = Nome do novo segmento de RAM expandida

C = Erros

D = 0 (não usada)

E = 0 (não usada)

Erros ativados: 0,1,2,7,9,12

A=2- FUNÇÃO BUSCA E EXECUTA - Recupera e executa o programa no segmento endereçado por B e ali residente;

B = Nome do segmento de RAM  
expandida  
C = Erros  
D = 0  
E = 0

Erros ativados: 0,1,2,8,9,13

A=3- FUNÇÃO ARMAZENA SUB-ROTINA OU  
SEGMENTO DE PROGRAMA NA RAM  
EXPANDIDA - O nome do segmento só  
existirá após a execução. Este  
programa será integralmente  
armazenado.

B = Nome do segmento de RAM  
expandida  
C = Erros  
D = Nº da linha inicial  
E = Nº da linha final

Erros ativados: 0,1,2,7,9,10,12

A=4- FUNÇÃO BUSCA SUB-ROTINA OU  
SEGMENTO DE PROGRAMA - Recupera a  
sub-rotina ou segmento de programa  
no segmento denominado por B e  
o superpõe ao programa atual,  
iniciando pela linha especificada  
por D.

B Nome do segmento de RAM  
expandida  
C = Erros  
D = Nº da linha inicial  
E = 0

Erros ativados: 0,1,2,8,9,13

A=5- FUNÇÃO ARMAZENA VETOR CCE  
INTEGER- Especificar o vetor com  
uma variável qualquer, que não  
seja um número.

B = Nome do segmento de RAM  
expandida  
C = Erros  
D = Nome de vetor ou string  
E = 0 (não utilizada)

Erros ativados: 0,1,2,7,9,11,12

Obs.: O CCE INTEGER é uma linguagem já  
ultrapassada e em desuso; caso  
voce a utilize, colocamos aqui as  
funções necessárias.

A=6- FUNÇÃO RECUPERA VETOR - Recupera  
e armazena (sob seu nome atribuído pela função) a string alfa-  
numérica ou vetor (CCE INTEGER).  
Quando definido o seu comprimento  
o RAMEXPAND devolve o nº de bytes  
que estão realmente armazenados.  
O nº de elementos transferidos se  
obtem dividindo o nº de bytes  
transferidos por 2(no caso de  
serem inteiros) ou por 6 (se  
forem reais).

B = Nome do segmento de RAM  
expandida  
C = Erros  
D = Nome do vetor  
E = Comprimento do vetor (mostra  
nº de bytes transferidos)

Erros obtidos: 0,1,2,8,9,11

A=7- FUNÇÃO TROCA DE CONTEÚDO DE VETORES - Com exceção da troca é igual à função anterior, inclusive o valor retornado.

B = Nome do segmento de RAM expandida

C = Erro

D = Nome do vetor

E = Comprimento do vetor (nº de bytes transferidos)

Erros ativados: 0,1,2,8,9,11

A=8- FUNÇÃO APAGAR SEGMENTO - Disponibiliza o espaço na RAM expandida para outros usos, cancelando o segmento denominado por B.

B = Nome do segmento

C = Erro

D = 0 (não utilizado)

E = 0 (não utilizado)

A=9- FUNÇÃO ACOPLAR BLOCOS-

B = Parâmetro de seleção de bloco

C = Erro

D = 0 (não utilizada)

E = 0 (não utilizada)

Erros ativados: 0,1,2,3,5,6





A=11- FUNÇÃO DETERMINAR ESPAÇO  
DISPONÍVEL - Computa o espaço  
disponível na RAM, na variável  
E, e devolve o nº de bytes.

B = 0 (não utilizada)  
C = Erro  
D = 0 (não utilizada)  
E = Espaço disponível

Erros ativados: 0,1,2

A=12- FUNÇÃO RECUPERAR NOME DO  
PRÓXIMO SEGMENTO - Esta função  
percorre a RAM expandida para  
verificar quais segmentos foram  
utilizados. A variável B mostra  
o nome do próximo segmento. O  
comprimento do novo segmento é  
devolvido na variável D. A  
variável B (com valor 0) dá o  
primeiro nome de segmento. O 0  
indica que não há mais segmentos  
livres.

Erros ativados: 0,1,2,8,9

A=13- FUNÇÃO NÃO IMPLANTADA

A=14- FUNÇÃO INICIALIZAR RAMEXPAND -  
Desacopla todos os blocos quando  
acionada, e os dados anteriores  
são perdidos.

B = 0  
C = Erro  
D = 0  
E = 0

Erros ativados: 0,1

A=15- FUNÇÃO ARMAZENAR - Armazena o conteúdo do vetor, identificado por uma variável.

B = Nome do segmento

C = Erro

D = Nome do vetor

E = 0 (não utilizada)

Erros ativados: 0,1,2,7,9,11,12

A=16- FUNÇÃO BUSCAR O CONTEÚDO DE VETOR NA RAM EXPANDIDA

B = Nome do segmento

C = Erro

D = Nome do vetor

E = 0 (não utilizado)

Recupera o vetor na RAM expandida e o armazena na variável designada em CCE BASIC.

Erros ativados: 0,1,2,8,9,11,14

## PSEUDO-DISCO

As placas de expansão CCE RAMCARD-64/128K podem ser usadas de modo semelhante aos acionadores de discos flexíveis com a utilização deste pacote de Software. Os programas que se utilizam do CDOS, que aqui é modificado, verão a placa de expansão como um acionador de discos extra.

Para possibilitar essas modificações no CDOS, foi incluído no disco que acompanha a placa um pacote de 3 rotinas para a expansão:

PSEUDO.DISCO - Usado para iniciar e instalar o parâmetro "PSEUDO DISCO".

PSEUDO.NRM - Programa em Assembly que contém o driver e as modificações do CDOS.

PSEUDO.MVD - Tem as mesmas características do PSEUDO.NRM. Usa-se com o Sistema Operacional Relocado.

PSEUDO.PARAMETROS - É gerado na entrada do PSEUDO.MVD e contém parâmetros do PSEUDO DISCO, posições de memória e placas de memória adicionais.

## CARACTERÍSTICAS E REQUISITOS DOS SISTEMAS PSEUDO DISCO

As placas de expansão CCE RAMCARD 64/128k podem ser utilizadas até um máximo de 256kbytes. A sequência para uso do PSEUDO DISCO é a seguinte:

1. Iniciar o PSEUDO DISCO
2. Instalar o PSEUDO DISCO, realizadas nesta ordem.

## AJUSTE DO PSEUDO DISCO

Este ajuste permite ao sistema reconhecer a posição das placas de expansão que serão utilizadas. Execute o PSEUDO DISCO (lembre-se que o PSEUDO.NRM e PSEUDO.MVD devem estar nos discos). Em seguida aparecerá um novo "menu" na tela:

---

### PSEUDO-DISCO

EXAMINAR AJUSTE	<1>
AJUSTAR	<2>
INSTALAR	<3>
RECONECTAR	<4>
FIM	<5>

OPÇÃO:

Selecione a opção ( 2 ).  
A tela mostrará o seguinte:

SLOT 5	DRIVE 1	SETORES DIR 4
	BANCOS	BANCOS
SLOT	12345678	SLOT 12345678
0		4
1		5
2		6
3		7

R = BASIC U= USADO N=NAO USADO

```
MUDAR CONTEUDO DO SLOT <1>
MUDAR PARAMETROS <2>
FIM <3>
```

Selecione a opção <1> para indicar ao sistema onde as placas se encontram. Descreve-se a seguir o diálogo com o micro:

QUAI SI NT?

Responda com o slot do primeiro cartão de memória.

```

      (0) VAZIO
<1> RAM CARD 128      <3> RAM CARD 32k
<2> RAM CARD 64k      <4> RAM CARD 16k
0 QUE HA AGORA NESSE SLOT?

```

Responda com o número correspondente ao conteúdo da placa.

QUER DEIXAR ESPACO PARA O BASIC?

Respondendo Sim, o 1º banco será reservado para o BASIC.

QUER USAR A PLACA INTEIRA?

Responda S em caso afirmativo. O "menu" de ajuste atualiza automaticamente o conteúdo do slot. Caso a resposta seja negativa, será editada pela máquina a seguinte pergunta:

QUER UTILIZAR O BANCO 1? (S/N)

Responda S ou N e repita o procedimento até que todos os bancos da placa tenham sido definidos.

Obs.: Se você reservou espaço para o BASIC e a capacidade de memória de sua placa for maior que 16k, ele começará o procedimento a partir do banco 2.

Através da opção <2>, você pode alterar o slot e drive associado ao PSEUDO-DISCO, e o nº de setores ocupados pelo diretório (máximo de 15).

Concluída esta operação, selecione a opção <3>, para sair do modo ajuste. O sistema responde a esta ação acionando o disco e atualizando o arquivo PSEUDO PARAMETROS. Lembre-se de que o disco não deve estar protegido contra escrita (com selo de proteção).

## INSTALAÇÃO DO PSEUDO DISCO

Depois de definidos os parâmetros de disco, tipo de placa e locação, o PSEUDO DISCO é instalado selecionando-se a opção <3>. O sistema carregará o PSEUDO NRM (parâmetro com CDOS normal) ou o PSEUDO.MVD (parâmetro com CDOS relocado) e o PSEUDO-DISCO inicializará com diretório vazio. Agora, o acesso ao PSEUDO DISCO pode ser feito como se fosse um acionador real.

Exemplo: CATALOG,S5,D1 mostra o diretório do PSEUDO-DISCO instalado no slot 5.

## RECONEXÃO DO PSEUDO DISCO

O PSEUDO DISCO é desconectado do Sistema Operacional toda vez que este é carregado. Exemplo: através de um comando PR#6 ou CALL-1370 ou IN#6. As informações nele contidas se tornam inacessíveis, mas não são perdidas se o micro não for desligado.

Para reconectar o PSEUDO-DISCO, utilize a opção <4> do "menu" principal, que recuperará os arquivos.

## PRECAUÇÕES E OBSERVAÇÕES

Não é recomendado o uso do PSEUDO DISCO junto com a memória expandida.



Se, por exemplo, for carregado um programa no banco 1 do slot 0 (quando este estiver sendo usado pelo PSEUDO DISCO), os dados armazenados nele poderão ser destruídos. Uma cópia do CDOS (que será transferida quando um disco real for inicializado sem a presença do PSEUDO DISCO) conterá as modificações necessárias para operar com o acionador virtual.

Como o Sistema Operacional está modificado, o disquete real não poderá efetuar "bootstrap".

É desnecessário o comando RUN para o programa de ajuste do PSEUDO-DISCO. Sempre que o acesso for instalado, o item "AJUSTE DO PESUDO-DISCO" pode ser ignorado, caso não haja alterações nas configurações das placas do micro, pois a mesma está armazenada no arquivo PSEUDO PARÂMETROS.

## UTILIZAÇÃO DO PSEUDO DISCO

A quantidade de arquivos a serem armazenados no PSEUDO-DISCO depende da capacidade da placa instalada. Para uma placa CCE RAMCARD- 64/128k com 128k o "disco" equivalente tem uma capacidade de mais ou menos 115Kbytes, excluídos os 16K para o CDOS (no caso de este ser relocado). Esta é aproximadamente a capacidade de um disco de 16 setores por trilha. Programas normais de cópia (a nível de

trilha de "nibble") não poderão ser usados para cópia de dados do PSEUDO DISCO. Todavia, o programa FIDCCE.MOD (opção <5> do menu principal), fornecido junto com a placa CCE RAM CARD 64/128k pode ser usado com o PSEUDO DISCO. O FIDCCE.MOD é idêntico ao FIDCCE, e pode ser usado com o CDOS e com o CDOS RELOCADO, indistintamente.

## DESCRIÇÃO DO FUNCIONAMENTO DO PSEUDO DISCO

O PSEUDO DISCO consiste nos seguintes programas:

PSEUDO DISCO - Funciona como interface entre o usuário e os programas PSEUDO.NRM e PSEUDO.MVD que operam as alterações no CDOS.

PSEUDO.NRM - é o vetor de instalação do PSEUDO DISCO. A inicialização é efetuada por ele mesmo. As rotinas de controle estão localizadas entre o "buffer" de arquivos e o início do CDOS com o HIMEM ajustado na descendente.

PSEUDO.MVD - Mesmas funções descritas anteriormente; é acionado automaticamente. Suas rotinas estão abaixo das rotinas do CDOS relocado (\$BE00) com o indicador HIMEM acionado para proteção contra escrita.

## PSEUDO DISCO CP/M

Este software prepara a expansão CCE RAMCARD 64/128k para funcionamento como acionador virtual no sistema operacional CP/M.

Permite utilizar expansões de 16k em conjunto com as de 64/128k.

O "software" consta dos seguintes programas:

PSEUDO.BAS - Programa em MBASIC que indica onde estão e quais são as expansões presentes, e também para alterar parâmetros do acionador.

PSEUDO.COM - Programa em linguagem ASSEMBLY, carregado e executado pelo PSEUDO.BAS. Origina os comandos INIT.COM e INIT2.COM.

INIT.COM e

INIT2.COM - Estes comandos inicializam o PSEUDO-DISCO. O comando INIT2 funciona como INIT, só que não inicializa a trilha de diretório depois de instalar as rotinas de acionador do PSEUDO DISCO.

PARAM.DAT - Arquivos que contém os parâmetros ajustados pelos programas acima.

Além disso, são fornecidos os programas PIP.COM, CAT.COM e CPM60.COM. Para maiores referências sobre estes programas, consulte seu Manual de CP/M.

Dois procedimentos são usados para a instalação do PSEUDO DISCO:

- a) Ajuste de parâmetro
- b) Instalação dos acionadores e inicialização do Sistema.

## AJUSTE DO PSEUDO DISCO

Esta sequência deve ser seguida para se obter a indicação de quais placas de expansão estão no sistema, onde estão e quais porções destas devem ser utilizadas. Os arquivos e programas anteriormente descritos devem estar disponíveis para uso em um disco que contenha o Sistema CP/M, instalado no acionador A.

Para ajustar o PSEUDO DISCO use o programa PSEUDO conforme indicado:

MBASIC PSEUDO <CR>.Aparecerá na tela:

---

PSEUDO-DISCO CP/M

- <1> EXAMINAR AJUSTE DO PSEUDO-DISCO
- <2> AJUSTAR O PSEUDO-DISCO
- <3> ATUALIZAR INIT.COM E INIT2.COM
- <4> FIM

QUAL E A SUA OPCAO?

---

Selecione neste ponto a opção <2>.O sistema mostrará a seguir:

---

# PSEUDO DISCO CP/M

DRIVE: C		BLOCOS DE DIRETORIO 1
SLOT	BANCO	SLOT BANCO
	12345678	12345678

0	4
1	5
2	6
3	7

---

U = USADO      N= NAO USADO      C= RESERVADO CP/M

---

<1> MUDAR CONTEUDO DO SLOT  
 <2> MUDAR PARAMETROS  
 <3> FIM

QUAL SUA OPCAO?

---

Selecione agora a opção <1>. O sistema perguntará a seguir:

QUAL SLOT?

A resposta deve ser dada conforme o nº do slot a ser alterado. O sistema perguntará então qual o conteúdo do slot para identificar a placa. a resposta deve ser dada com o nº associado ao tipo de placa que está sendo utilizada. Feito isto, o sistema perguntará quais bancos serão utilizados. No caso do slot ser o 0, o sistema perguntará também se deve ser reservado espaço para CP/M. Esta sequência deve ser repetida até aparecer na tela uma configuração que seja cópia fiel daquela que estiver sendo usada. Selecione então a opção (2) e modifique os parâmetros do "disco" conforme necessário. As opções são:

---

<1> MODIFICAR NOME DO ACIONADOR  
<2> MOD. BLOCOS PARA DIRETORIO  
<3> FIM

---

Caso se deseje um nome diferente para C, use a opção <1.> para alterar os blocos alocados ao armazenamento da trilha de diretório, use a opção <2>.

Obs: Cada bloco tem uma capacidade máxima de 32 entradas de diretório.

Os comandos INIT e INIT2 são atualizados (para seleção de opção (3) do menu principal) e ficam capacitados a reconhecer o Sistema Operacional, o que lhe permite reconhecer a configuração recém-alterada, após o que o sistema volta ao controle do Sistema Operacional.

## INSTALAÇÃO DO PSEUDO DISCO

Depois destes ajustes o acionador virtual já pode ser instalado. Para isto é suficiente executar (a partir do Sistema Operacional) o comando INIT ou INIT2, que completará a instalação, conforme descrito; o INIT2 não inicializa o Diretório do "disco" há pouco instalado. Sua utilização é evidente no caso do aparecimento de um "bootstrap", sem que a alimentação seja desligada. Assim sendo, os acessos aos arquivos da placa CCE RAMCARD 64/128k podem ser mantidos. Isto seria impossível se o bloco do diretório fosse reinitializado, pois seus indicadores seriam irremediavelmente perdidos, impedindo o acesso aos dados armazenados.

## UTILIZAÇÃO DO PSEUDO DISCO

Utilizando a sua denominação atribuída, o PSEUDO DISCO, após a instalação, é utilizado como se fosse um acionador, e a transferência de arquivos e dados pode ser efetuada pelos comandos de programas de cópia, por exemplo, o comando PIP (INIT PIP C:=B: \*.\*).

Um arquivo SUBMIT pode ser usado para:

- a) Instalar o PSEUDO DISCO
- b) Copiar vários arquivos com apenas um comando.

Ex: O arquivo XFER.SUB instalará o PSEUDO-DISCO (acionador C:) e copiará arquivos do acionador B:,digitando apenas SUBMIT XFER.

## REQUISITOS E LIMITAÇÕES

O PSEUDO DISCO pode utilizar no máximo 256kbytes RAM, distribuídos por várias placas de expansão. As rotinas vetores do PSEUDO DISCO residem no bloco de configuração de E/S; este espaço não estará disponível depois da instalação do PSEUDO DISCO, pois é reservado a vetores do usuário.



## INFORMAÇÕES TÉCNICAS

Conforme já foi descrito, a placa CCE RAMCARD 64/128k é subdividida em blocos de 16k; tem-se acesso aos 4k inferiores de cada bloco quando se especifica:

1. Qual o banco de 16k
2. O bloco inferior de 4k a ser utilizado

Por convenção o banco inferior de 4k de cada bloco de 12k é chamado de A e o bloco adjacente de B.

CONTROLE DE PLACAS  
64 E 128K

Os endereços (dezesseis) utilizados pelo controle de placa estão entre \$C0n0 e \$C0n (n=8+número do slot). Eles permitem:

1. Selecionar RAM ou ROM da placa principal
2. Selecionar o banco de 16k
3. Selecionar a página de 4k (inferior)
4. Habilitar ou desabilitar escrita na RAM

Para pôr em determinado estado a placa 64/128k, basta acessar o endereço correspondente a este estado. Consegue-se

isto lendo ou escrevendo no endereço (Via CCE BASIC através de PEEK e POKE).

O modo de operação e qual dos bancos de 16k será acessado é determinado pelos 4 Bits menos significativo do endereço de controle.

A linha A2 é utilizada como controle de estados, quando em 0. A linha A2 em 1, habilita a seleção de até 8 bancos de 16k (no caso da placa de 128k) com as linhas A0, A1 e A3 indicando o endereço do banco. Com a linha A2 em 0, o significado das linhas A0, A1 e A3 é o seguinte:

A0 - Habilita (A0=0) ou desabilita escrita na RAM expandida

A1 - Em conjunto com a linha A0 é utilizada para desabilitar a RAM da placa RAM CARD

A3 - Seleciona a página de 4k dentro do banco de 16k (A3 = 0, página A).

A2	A3	A1	A0	LEITURA ESCRITA		
0	0	0	0	seleciona banco 4k A	RAM	**
0	0	0	1	seleciona banco 4k A	ROM	RAM
0	0	1	0	seleciona banco 4k A	ROM	*
0	0	1	1	seleciona banco 4k A	RAM	RAM
0	1	0	0	seleciona banco 4k B	RAM	**
0	1	0	1	seleciona banco 4k B	ROM	RAM
0	1	1	0	seleciona banco 4k B	ROM	*
0	1	1	1	seleciona banco 4k B	RAM	RAM
1	0	0	0	seleciona banco 16k número 1		
1	0	0	1	seleciona banco 16k número 2		
1	0	1	0	seleciona banco 16k número 3		
1	0	1	1	seleciona banco 16k número 4		

A2	A3	A1	A0		LEITURA	ESCRITA
1	1	0	0	seleciona banco 16k	número 5	
1	1	0	1	seleciona banco 16k	número 6	
1	1	1	0	seleciona banco 16k	número 7	
1	1	1	1	seleciona banco 16k	número 8	

\* - desabilita a RAM  
 \*\* - "write protect" (protegido  
 contra escrita).

Esta tabela mostra que com qualquer combinação pode-se acessar qualquer segmento da RAM.

## UTILIZAÇÃO DA PLACA DE RAM

CDOS RELOCADO - Ocupa o banco 2 do bloco 16k e a 2A do 4k (espaço da memória \$D000 - \$DFFF total 12k). É capacitado a usar o controle do endereço \$C0n3, e selecionar o banco de endereço \$C0n5. O 4k do banco 2B não é usado para DOS RELOCADO.

BASIC ALTERNATIVO - Quando o BASIC ALTERNATIVO (CCE/INTEGER ou CCE BASIC) está presente na placa 64/128k, ele ocupa 16k do banco 1 e 4k do banco 1A, e é habilitado para leitura usada no endereço de controle \$C0n0 e seleciona o banco \$C0n4. O 4k do banco 1B é usado para este BASIC.

LEDS DE INDICAÇÃO - Quando a RAM está habilitado para a escrita, o LED mais afastado teclado (D4) acende-se. Os restantes (D3,D2,D1) dão a indicação binária do nº de banco acessado (bit menos significativo = D3).

```

$FFFF =====
!      !      !      !
!      !      !      !
! 8K   !      ! 8K   !
!      !      !      !
!      !      !      !
$E000 !-----!-----!-----!-----
! 4K   ! 4K   ! 4K   ! 4K   !
!      !      !      !      !
!      !      !      !      !
$D000 =====
16K BANCO 1 16K BANCO 2

```

```

$FFFF =====
!      !      !      !
!      !      !      !
! 8K   !      ! 8K   !
!      !      !      !
$E000 !-----!-----!-----!-----
! 4K   ! 4K   ! 4K   ! 4K   !
!      !      !      !      !
!      !      !      !      !
$D000 =====
16K BANCO 3 16K BANCO 4

```

(continuação)

```
$FFFF =====
!         !         !         !
!         !         !         !
!  8K  !         !  8K  !
!         !         !         !
$E000 !-----!-----!-----!-----
!         !         !         !
!  4K  !  4K  !  4K  !  4K  !
$D000 =====
16K BANCO 5  16K BANCO 6

$FFFF =====
!         !         !         !
!         !         !         !
!  8K  !         !  8K  !
!         !         !         !
$E000 !-----!-----!-----!-----
!         !         !         !
!  4K  !  4K  !  4K  !  4K  !
!         !         !         !
$D000 =====
16K BANCO 7  16K BANCO 8
```

# MANUAL DO RAM CARD 64/128

## SUGESTÕES DO LEITOR

1- Você encontrou erros neste manual? Se a resposta for afirmativa, especifique citando a página.

.....

.....

.....

2- Você achou este manual fácil de entender? Faça sugestões para torná-lo melhor.

.....

.....

.....

3- Por favor, indique o tipo de leitor/usuário  
você melhor se enquadra:

- ☐ Programador
- ☐ Usuário com pouca experiência em  
programação
- ☐ Estudante
- ☐ Não programador
- ☐ Outro. Qual?\_\_\_\_\_

4- A que tipo de aplicação se destina seu pro?

- ☐ Educação
- ☐ Engenharia
- ☐ Processamento de Dados
- ☐ Estatísticas
- ☐ Aplicação Comercial
- ☐ Finanças
- ☐ Lazer
- ☐ Outro. Qual? \_\_\_\_\_

DATA \_\_\_\_ / \_\_\_\_ / \_\_\_\_

NOME

ENDEREÇO \_\_\_\_\_

TELEFONE\_\_\_\_\_

CIDADE \_\_\_\_\_ ESTADO \_\_\_\_ CEP\_\_\_\_\_

Impresso pela  
Companhia Melhoramentos de São Paulo  
Indústrias de Papel -- Rua Tito, 479, -- São Paulo





**INFORMÁTICA**