

**300**  
**POKES**  
**PEEKS 'N EXECS**  
**FOR THE COCO III**

**Kishore M. Santwani**

# 300

POKES  
PEEKS 'N EXECES  
for the  
COCO III

by  
Kishore M. Santwani

Copyright (C) 1987 by MICROCOM SOFTWARE  
1988 PO BOX 214  
FAIRPORT, NY 14450  
PH: (716) 223-1477

ALL RIGHT RESERVED. The text of this publication or any part thereof, may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, downloading, storage in an information retrieval system, or otherwise, without the written permission of the publisher.

Printed in the United States of America

First Printing May 1987

Second Printing Jan. 1988

Every effort has been made to insure the accuracy of the contents of this book. Microcom Software does not guarantee the accuracy of or the information contained in this book, nor does it assume any liability arising out of the application or the use of the information herein.

## P R E F A C E

The advent of the Color Computer III has necessitated certain modifications/additions of different POKEs, PEEKs and EXECs for a more effective utilization of the capabilities of the CoCo. The popularity and usefulness of our books "500 POKES PEEKS 'N EXECs" and "SUPPLEMENT TO 500 POKES PEEKS 'N EXECs" has created an unprecedented demand for an exclusive repository of the POKEs, PEEKs and EXECs for the CoCo III. Most of the POKEs, PEEKs 'N EXECs in the earlier books are also applicable to CoCo III. This book includes a number of POKE, PEEK 'N EXEC commands, as well as some routines that will give you more programming power.

Some of the commands included in this book are:

40/64/80 Column Text Screen Dump  
Save Text/Graphics Screens to tape/disk  
Command/Function disables  
**Enhancements for CoCo III Basic.**  
128K/512K Ram Test Program  
HPRINT character modifier

We hope the CoCo III users will be delighted with the information given in this book. We shall be glad to receive your views on the book and any changes/additions to be made to make it even more useful.

GOOD LUCK!!

## HOW TO USE THIS BOOK

Before typing in any COMMAND, please read its ensuing RESULT and REMARKS. This will give you a better understanding of the command and whether or not it will be compatible with your system.

If the COMMAND is a PEEK, for example PEEK (341), precede it with a PRINT command. For example PRINT PEEK (341) and press <ENTER>. The computer will return (or display) a value. Then read the RESULT and REMARKS to see what that value stands for. If you wish to use the PEEK command in a Basic Program, you may precede it with a variable, for example: A = PEEK (341)

If the COMMAND contains any DATA statements, it must be preceded by a statement # and RUN. If you do not precede such commands with a statement #, you will get an ?00 ERROR. Always make BACKUPS of your program that contain POKES, PEEKs and EXECs as a slight error can wipe out your entire program.

Please read Page 43 of this book before using this book.

Some abbreviations used in this book are:

Col	-->	Column
CR + LF	-->	Carriage Return
LF	-->	Linefeed
hi-res	-->	high resolution

## C O N T E N T S

Reset Coldstart .....	1
ALT,CTRL,F1,F2 Key Check .....	2
True Lowercase .....	2
Inverse Video .....	3
Double PEEK routine .....	3
Better Monitor Display .....	3
Save new DOS to disk .....	3
Change vertical spacing .....	4
Border Change for 40 cols .....	5
Border Change for 80 cols .....	6
64 Col Text Screen! .....	6
Command Disables/Enables .....	6
BREAK Key Disable .....	11
200 rows for hi-res screen .....	13
Cursor modifications .....	14
Character blink rate .....	17
Hi-res graphic modes .....	17
Border color changes .....	19
Palettes! .....	22
High Speed POKE .....	26
Screen character attributes .....	27

### UTILITY ROUTINES

Text Screen Dump .....	29
Text Screen Save .....	30
Text Screen Load .....	31
Graphics Screen Save .....	32
HPRINT Controls/ Character modifier .....	34
Scroll Protection .....	36
Disk Search Routine .....	37
512K RAM TEST .....	37
128K RAM TEST .....	38
Printer Linefeed Routine .....	38
Define Printer Left Margin .....	39
Smooth vertical scrolling .....	39
Memory Management Registers .....	39
ROM Change Preserver .....	43

COMMAND: POKE 113,0:EXEC &H8C1B  
RESULT: Performs a cold start  
REMARKS: \*\* WARNING \*\* Erases any Basic  
program in memory

COMMAND: PEEK (&HE6)  
RESULT: Returns 1 if HSCREEN 1 is used  
REMARKS: Returns 0 if no hi-res graphics  
is used

COMMAND: PEEK (&HE6)  
RESULT: Returns 2 if HSCREEN 2 is used  
REMARKS: See previous remarks

COMMAND: PEEK (&HE6)  
RESULT: Returns 3 if HSCREEN 3 is used  
REMARKS: See previous remarks

COMMAND: PEEK (&HE6)  
RESULT: Returns 4 if HSCREEN 4 is used  
REMARKS: See previous remarks

COMMAND: PEEK (&HE7)  
RESULT: Returns 0 With 32x16 text screen  
REMARKS: None

COMMAND: PEEK (&HE7)  
RESULT: Returns 1 with 40 col text screen  
REMARKS: None

COMMAND: PEEK (&HE7)  
RESULT: Returns 2 with 80 col text screen  
REMARKS: None

COMMAND: PEEK (341)  
RESULT: Returns 191 if ALT key is pressed  
REMARKS: None

COMMAND: PEEK (342)  
RESULT: Returns 191 if CTRL key is pressed  
REMARKS: None

COMMAND: PEEK (343)  
RESULT: Returns 191 if F1 key is pressed  
REMARKS: None

COMMAND: PEEK (344)  
RESULT: Returns 191 if F2 key is pressed  
REMARKS: None

COMMAND: EXEC &H8C1B  
RESULT: Performs a warm-start  
REMARKS: All changes made to "ROM" are lost

COMMAND: POKE &H95AC,57:POKE &HFF22,  
PEEK (&HFF22) OR &H10  
RESULT: Gives true lowercase in 32 col  
REMARKS: Lowercase includes descenders!

COMMAND: POKE &H95AC,57:POKE &HFF22,  
PEEK (&HFF22) OR &H20  
RESULT: Inverse Video for 32 col screen  
REMARKS: Better visual display!

COMMAND: POKE &H95AC,57:POKE &HFF22,  
PEEK (&HFF22) OR &H40  
RESULT: Changes border of 32 col  
screen to green  
REMARKS: None

COMMAND: POKE &H95AC,57:POKE &HFF22,  
PEEK (&HFF22) AND (255-64)  
RESULT: Restores normal border  
REMARKS: For 32 col screen only

COMMAND: POKE &H95AC,52  
RESULT: Restores to normal after previous 4  
          POKEs  
REMARKS: None

COMMAND: POKE &HB752,236:POKE &HB756,&HF4  
RESULT: Replaces PEEK command with DPEEK  
REMARKS: DPEEK gives you 16-bit value of  
          consecutive memory locations.  
          Very useful!!

COMMAND: EXEC &HE010  
RESULT: Executes the present ROMPAK  
REMARKS: This command should be used in  
          place of the EXEC 49152 command to  
          execute a ROMPAK.

COMMAND: PALETTE 13,255:POKE &HE033,PEEK  
          (&HE033) OR 16  
RESULT: Disables colorburst for 32 col  
          text screen  
REMARKS: Better visual display for  
          monochrome monitors

COMMAND: POKE &HE033,PEEK(&HE033) AND  
          (255-16)  
RESULT: Re-enables colorburst  
REMARKS: For 32 col screen only

COMMAND: SAVEM "DOS",PEEK(&HE038)\*256,  
          &HFEDF,0  
RESULT: Saves the EXB DOS to disk  
REMARKS: All changes are saved also

COMMAND: CSAVEM "DOS",PEEK(&HE038)\*256,  
          &HFEDF,0  
RESULT: Saves the EXB DOS to tape  
REMARKS: All changes are saved also



COMMAND: LOADM "DOS",PEEK (&H8A)  
RESULT: Loads the EXB DOS from disk  
REMARKS: Loads back all changes to DOS

COMMAND: EXEC 42136 "DOS"  
RESULT: Loads the EXB DOS from tape  
REMARKS: Loads back all changes to DOS

COMMAND: POKE &HE03C,3  
RESULT: Sets 1 line spacing between  
vertical rows on screen  
REMARKS: For 40 col screen only. This is  
default spacing

COMMAND: POKE &HE03C,4  
RESULT: Sets 2 line spacing between  
vertical rows on screen  
REMARKS: For 40 col screen only

COMMAND: POKE &HE03C,5  
RESULT: Sets 3 line spacing between  
vertical rows on screen  
REMARKS: For 40 col screen only

COMMAND: POKE &HE03C,6  
RESULT: Sets 5 line spacing between  
vertical rows on screen  
REMARKS: For 40 col screen only

COMMAND: PALETTE 13,255:POKE &HE03C,PEEK  
(&HE03C) OR 16  
RESULT: Disables colorburst for 40 col  
text screen  
REMARKS: Better visual display for  
monochrome monitors

COMMAND: POKE &HE03C,PEEK(&HE03C) AND 239  
RESULT: Re-enables colorburst  
REMARKS: For 40 col screen only

COMMAND: POKE &HE03E,x  
RESULT: Sets border around 40 col text  
screen to color x  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HE045,3  
RESULT: Sets 1 line spacing between  
vertical rows on screen  
REMARKS: For 80 col screen only. This  
is default spacing

COMMAND: POKE &HE045,4  
RESULT: Sets 2 line spacing between  
vertical rows on screen  
REMARKS: For 80 col screen only

COMMAND: POKE &HE045,5  
RESULT: Sets 3 line spacing between  
vertical rows on screen  
REMARKS: For 80 col screen only

COMMAND: POKE &HE045,6  
RESULT: Sets 5 line spacing between  
vertical rows on screen  
REMARKS: For 80 col screen only

COMMAND: PALETTE 13,255:POKE &HE045,PEEK  
(&HE045) OR 16  
RESULT: Disables colorburst for 80 col  
text screen  
REMARKS: Better visual display for  
monochrome monitors

COMMAND: POKE &HE045,PEEK(&HE045) AND 239  
RESULT: Re-enables colorburst  
REMARKS: For 80 col screen only

COMMAND: POKE &HE046,17:WIDTH80:POKE  
&HFE04,64:POKE &HF871,&H80:  
POKE &HF875,&H2B:POKE &HF876,  
&H80:POKE &HFE06,&H2C  
RESULT: Sets up 64 col text screen  
REMARKS: Especially useful for TV users!

COMMAND: POKE &HE046,21:WIDTH80:POKE  
&HFE04,80:POKE &HF871,&HA0:  
POKE &HF875,&H2E:POKE &HF876,  
&H60:POKE &HFE06,&H2F  
RESULT: Restores normal text screen  
REMARKS: For use after previous command

COMMAND: POKE &HE047,x  
RESULT: Sets border of color x around the  
80 col text screen  
REMARKS: x is any color between 0 & 63

#### Command Disables/Enables

COMMAND: POKE &HE1C6,0  
RESULT: Disables the WIDTH command  
REMARKS: None

COMMAND: POKE &HE1C6,&H49  
RESULT: Re-enables WIDTH command  
REMARKS: For use after previous command

COMMAND: POKE &HE1CC,0  
RESULT: Disables the PALETTE command  
REMARKS: None

COMMAND: POKE &HE1CC,&H4C  
RESULT: Re-enables the PALETTE command  
REMARKS: For use after previous command

COMMAND: POKE &HE1D6,0  
RESULT: Disables the HSCREEN command  
REMARKS: None

COMMAND: POKE &HE1D6,&H45  
RESULT: Re-enables the HSCREEN command  
REMARKS: For use after previous command

COMMAND: POKE &HE1D8,0  
RESULT: Disables the LPOKE command  
REMARKS: None

COMMAND: POKE &HE1D8,&H4C  
RESULT: Re-enables the LPOKE command  
REMARKS: For use after previous command

COMMAND: POKE &HE1DE,0  
RESULT: Disables the HCLS command  
REMARKS: None

COMMAND: POKE &HE1DE,&H43  
RESULT: Re-enables the HCLS command  
REMARKS: For use after previous command

COMMAND: POKE &HE1E2,0  
RESULT: Disables the HCOLOR command  
REMARKS: None

COMMAND: POKE &HE1E2,&H43  
RESULT: Re-enables the HCOLOR command  
REMARKS: For use after previous command

COMMAND: POKE &HE1EC,0  
RESULT: Disables the HPAINT command  
REMARKS: None

COMMAND: POKE &HE1EC,&H54  
RESULT: Re-enables the HPAINT command  
REMARKS: For use after previous command

COMMAND: POKE &HE1FO,0  
RESULT: Disables the KCIRCLE command  
REMARKS: None

**COMMAND:** POKE &HE1F0,&H52  
**RESULT:** Re-enables the HCIRCLE command  
**REMARKS:** For use after previous command

**COMMAND:** POKE &HE1F4,0  
**RESULT:** Disables the HLINE command  
**REMARKS:** None

**COMMAND:** POKE &HE1F4,&H48  
**RESULT:** Re-enables the HLINE command  
**REMARKS:** For use after previous command

**COMMAND:** POKE &HE1FB,0  
**RESULT:** Disables the HGET command  
**REMARKS:** None

**COMMAND:** POKE &HE1FB,&H45  
**RESULT:** Re-enables the HGET command  
**REMARKS:** None

**COMMAND:** POKE &HE1FD,0  
**RESULT:** Disables the HPUT command  
**REMARKS:** None

**COMMAND:** POKE &HE1FD,&H48  
**RESULT:** Re-enables the HPUT command  
**REMARKS:** For use after previous command

**COMMAND:** POKE &HE205,0  
**RESULT:** Disables the HBUFF command  
**REMARKS:** None

**COMMAND:** POKE &HE205,&H46  
**RESULT:** Re-enables the HBUFF command  
**REMARKS:** For use after previous command

**COMMAND:** POKE &HE20D,0  
**RESULT:** Disables the ON ERR GOTO command  
**REMARKS:** None

COMMAND: POKE &HE20D,&H52  
RESULT: Re-enables the ON ERR GOTO command  
REMARKS: For use after previous command

COMMAND: POKE &HE20F,0  
RESULT: Disables the ON BRK GOTO command  
REMARKS: None

COMMAND: POKE &HE20F,66  
RESULT: Re-enables the ON BRK GOTO command  
REMARKS: For use after previous command

COMMAND: POKE &HE215,0  
RESULT: Disables the LOCATE command  
REMARKS: None

COMMAND: POKE &HE215,65  
RESULT: Re-enables the LOCATE command  
REMARKS: For use after previous command

COMMAND: POKE &HE218,0  
RESULT: Disables the HSTAT command  
REMARKS: None

COMMAND: POKE &HE218,&H48  
RESULT: Re-enables the HSTAT command  
REMARKS: For use after previous command

COMMAND: POKE &HE21D,0  
RESULT: Disables the HSET command  
REMARKS: None

COMMAND: POKE &HE21D,&H48  
RESULT: Re-enables the HSET command  
REMARKS: For use after previous command

COMMAND: POKE &HE223,0  
RESULT: Disables the HRESET command  
REMARKS: None

COMMAND: POKE &HE223,&H45  
RESULT: Re-enables the HRESET command  
REMARKS: For use after previous command

COMMAND: POKE &HE22A,0  
RESULT: Disables the HDRAW command  
REMARKS: None

COMMAND: POKE &HE22A,65  
RESULT: Re-enables the HDRAW command  
REMARKS: For use after previous command

COMMAND: POKE &HE22C,0  
RESULT: Disables the CMP command  
REMARKS: None

COMMAND: POKE &HE22C,&H43  
RESULT: Re-enables the CMP command  
REMARKS: For use after previous command

COMMAND: POKE &HE22F,0  
RESULT: Disables the RGB command  
REMARKS: None

COMMAND: POKE &HE22F,&H52  
RESULT: Re-enables the RGB command  
REMARKS: For use after previous command

COMMAND: POKE &HE233,0  
RESULT: Disables the ATTR command  
REMARKS: None

COMMAND: POKE &HE233,&H54  
RESULT: Re-enables the ATTR command  
REMARKS: For use after previous command

COMMAND: POKE &HE267,0  
RESULT: Disables the LPEEK function  
REMARKS: None

COMMAND: POKE &HE267,&H45  
RESULT: Re-enables the LPEEK function  
REMARKS: For use after previous command

COMMAND: POKE &HE26A,0  
RESULT: Disables the BUTTON function  
REMARKS: None

COMMAND: POKE &HE26A,&H55  
RESULT: Re-enables the BUTTON function  
REMARKS: For use after previous command

COMMAND: POKE &HE274,0  
RESULT: Disables the HPOINT function  
REMARKS: None

COMMAND: POKE &HE274,&H54  
RESULT: Re-enables the HPOINT function  
REMARKS: For use after previous command

COMMAND: POKE &HE275,0  
RESULT: Disables the ERNO function  
REMARKS: None

COMMAND: POKE &HE275,&H45  
RESULT: Re-enables the ERNO function  
REMARKS: For use after previous command

COMMAND: POKE &HE279,0  
RESULT: Disables the ERLIN function  
REMARKS: None

COMMAND: POKE &HE279,&H45  
RESULT: Re-enables the ERLIN function  
REMARKS: For use after previous command

COMMAND: POKE &HE414,0:POKE &HE42A,0  
RESULT: Disables the BREAK KEY  
REMARKS: None



COMMAND: POKE &HE414,3:POKE &HE42A,3  
RESULT: Enables the BREAK KEY after  
previous command  
REMARKS: None

COMMAND: POKE &HE47B,18:POKE &HE47C,18  
RESULT: Disables error trapping  
REMARKS: None

COMMAND: POKE &HE47B,38:POKE &HE47C,54  
RESULT: Re-enables error trapping after  
previous command  
REMARKS: None

COMMAND: POKE &HE649,16  
RESULT: Fixes the RGB/CMP bug  
REMARKS: All 16 (instead of 15) palette  
registered are copied now

COMMAND: FOR I = &HE654 TO &HE663: PRINT  
PEEK(I):NEXT  
RESULT: Gives initial palette color  
settings for composite monitors  
REMARKS: None

COMMAND: FOR I=&HE664 TO &HE673:PRINT PEEK (I):NEXT  
RESULT: Gives initial palette color  
settings for RGB monitors  
REMARKS: None

COMMAND: POKE &HE6C6,18:POKE &HE6C7,18  
RESULT: Prevents HSCREEN from clearing  
the hi-res graphics screen  
REMARKS: None

COMMAND: POKE &HE7BA,199:POKE &HE7BE,198:  
 POKE &HEB75,198:POKE &HEFDF,25-1:  
 POKE &HF521,199:POKE &HF526,198  
 RESULT: Allows 200 rows instead of 192 on  
 a hi-res graphics screen  
 REMARKS: One of the following 4 POKEs (for  
 the appropriate graphics screen)  
 must be set before using HSCREEN

COMMAND: POKE &HE06C,&H35  
 RESULT: Patches HSCREEN 1 to set 200 rows  
 instead of 192  
 REMARKS: To be used with previous command

COMMAND: POKE &HE06C,&H3E  
 RESULT: Patches HSCREEN 2 to set 200 rows  
 instead of 192  
 REMARKS: See remarks for previous command

COMMAND: POKE &HE06C,&H34  
 RESULT: Patches HSCREEN 3 to set 200 rows  
 instead of 192  
 REMARKS: See remarks for previous command

COMMAND: POKE &HE06C,&H3D  
 RESULT: Patches HSCREEN 4 to set 200 rows  
 instead of 192  
 REMARKS: See remarks for previous command

COMMAND: POKE &HEF19,&HCC-8  
 RESULT: Fixes the HPUT "NOT" bug  
 REMARKS: None

COMMAND: POKE &HF655,18:POKE &HF656,18:  
 POKE &HF657,18  
 RESULT: Prevents WIDTH command from  
 erasing text screen  
 REMARKS: For 32 col text screen only

COMMAND: POKE &HF670,18:POKE &HF671,18  
RESULT: Prevents WIDTH command from erasing the text screen  
REMARKS: For 40/80 column screens

COMMAND: POKE &HF6BC,16  
RESULT: Allows you to CLS 1 - 16 instead of CLS 1 - 8  
REMARKS: For hi-res text screens only

COMMAND: POKE &HF787,198:POKE &HF788,1  
RESULT: Creates a steady cursor  
REMARKS: Only for hi-res text screens

COMMAND: POKE &HF787,10:POKE &HF788,148  
RESULT: Restores normal cursor  
REMARKS: For use after previous command

COMMAND: POKE &HF78C,x  
RESULT: Changes cursor blink rate  
REMARKS: x is any # between 0 & 255

COMMAND: POKE &HF78C,11  
RESULT: Restores normal cursor  
REMARKS: For use after previous command

COMMAND: PEEK (&HFE00)\*256+PEEK(&HFE01)  
RESULT: Returns cursor position in hi-res text screen  
REMARKS: Not compatible with 32 col screen

COMMAND: PEEK (&HFE02)  
RESULT: Returns X cursor position on screen  
REMARKS: Not compatible with 32 col screen

COMMAND: PEEK (&HFE03)  
RESULT: Returns Y cursor position on screen  
REMARKS: Not compatible with 32 col screen

COMMAND: PEEK (&HFEO4)  
RESULT: Returns 40 with 40 column screen  
REMARKS: None

COMMAND: PEEK (&HFEO4)  
RESULT: Returns 64 with 64 column screen  
REMARKS: None

COMMAND: PEEK (&HFEO4)  
RESULT: Returns 80 with 80 column screen  
REMARKS: None

COMMAND: POKE &HFEO4,20  
RESULT: Divides 40 col screen into 2 cols  
REMARKS: None

COMMAND: POKE &HFEO4,40  
RESULT: Divides 80 col screen into 2 cols  
REMARKS: None

COMMAND: PEEK (&HFEO5)  
RESULT: Returns # of rows in current hi-res  
text screen  
REMARKS: Default value should be 24

COMMAND: PEEK (&HFEO6)\*256+PEEK (&HFEO7)  
RESULT: Returns end address of current  
hi-res text screen  
REMARKS: None

COMMAND: PEEK (&HFEO8)  
RESULT: Returns the current cursor  
attribute  
REMARKS: Not for 32 col screens

COMMAND: PEEK (&HFEOA)  
RESULT: Returns current foreground color  
REMARKS: Only for HSCREEN graphic modes

COMMAND: PEEK (&HFE0B)  
RESULT: Returns current background color  
REMARKS: Only for HSCREEN modes

COMMAND: PEEK (&HFE0C)\*256+PEEK (&HFE0D)  
RESULT: Returns current ON BRK GOTO line #  
REMARKS: None

COMMAND: PEEK (&HFE0E)\*256+PEEK (&HFE0F)  
RESULT: Returns current ON ERR GOTO line #  
REMARKS: None

COMMAND: PEEK (&HFE10)  
RESULT: Returns # of most recent error  
REMARKS: Returns 255 if no error occurred

COMMAND: PEEK (&HFE11)\*256+PEEK (&HFE12)  
RESULT: Returns Basic Program line # which  
contains the ON ERR GOTO statement  
REMARKS: None

COMMAND: PEEK (&HFE13)\*256+PEEK (&HFE14)  
RESULT: Returns line # where error occurred  
REMARKS: None

COMMAND: PEEK (&HFE15)\*256+PEEK (&HFE16)  
RESULT: Returns Basic Program line # which  
contains the ON BRK GOTO statement  
REMARKS: None

COMMAND: PEEK (&HFE18)  
RESULT: Returns # of characters to be  
HPRINTed to hi-res graphics screen  
REMARKS: None

COMMAND: FOR I=1 TO PEEK (&HFE18):PRINT  
CHR\$(PEEK(I+&HFE18));:NEXT I  
RESULT: Displays characters that were  
last HPRINTed on the screen  
REMARKS: None

COMMAND: POKE &HFF01,0  
RESULT: Disables keyboard input  
REMARKS: \*\* WARNING \*\* Save any programs  
before using this POKE. Great for  
demos!

COMMAND: POKE &HFF01,4  
RESULT: Restores to normal after previous  
POKE  
REMARKS: Same as for previous command

COMMAND: POKE &HFF94,x  
RESULT: Controls the blinking rate of  
characters on the screen  
REMARKS: x is any number between 0 & 100

COMMAND: POKE &HFF94,126  
RESULT: Restores to normal after  
POKE &HFF94,x  
REMARKS: None

#### Hires-graphic modes

The next few commands show you how to set various graphics modes. Many of these modes are not supported by Basic directly. Some knowledge of ML is required to access these graphic modes. Hint for Basic Programmers: Try LPOKEing values starting at address \$60000!

COMMAND: HSCREEN 4:POKE &HFF99,0  
RESULT: Sets 128x192 graphics Mode  
REMARKS: 2 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,9  
RESULT: Sets 128x192 graphics Mode  
REMARKS: 4 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,18  
RESULT: Sets 128x192 graphics Mode  
REMARKS: 16 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,4  
RESULT: Sets 160x192 graphics Mode  
REMARKS: 2 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,22  
RESULT: Sets 160x192 graphics Mode  
REMARKS: 16 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,8  
RESULT: Sets 256x192 graphics Mode  
REMARKS: 2 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,17  
RESULT: Sets 256x192 graphics Mode  
REMARKS: 4 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,26  
RESULT: Sets 256x192 graphics Mode  
REMARKS: 16 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,21  
RESULT: Sets 320x192 graphics Mode  
REMARKS: 4 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,30  
RESULT: Sets 320x192 graphics Mode  
REMARKS: 16 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,16  
RESULT: Sets 512x192 graphics Mode  
REMARKS: 2 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,25  
RESULT: Sets 512x192 graphics Mode  
REMARKS: 4 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,20  
RESULT: Sets 640x192 graphics Mode  
REMARKS: 2 colors are available

COMMAND: HSCREEN 4:POKE &HFF99,29  
RESULT: Sets 640x192 graphics Mode  
REMARKS: 4 colors are available

The following border changes will work only in hi-res modes. Also, they will not work from direct mode. To change border color in direct mode, see contents. Colors depend on Monitor settings and may vary slightly.

COMMAND: POKE &HFF9A,x  
RESULT: Creates a red border around the current hi-res text/graphics screen  
REMARKS: For RGB monitors only. Values for x = 4,32-39,60 for various red types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a red border around the current hi-res text/graphics screen  
REMARKS: For composite monitors only. Values for x = 6-8,21,23,24,38,39,54 for various red types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a blue border around the current hi-res text/graphics screen  
REMARKS: For RGB monitors only. Values for x = 1,8-15,57 for various blue types



COMMAND: POKE &HFF9A,x  
RESULT: Creates a blue border around the current hi-res text/graphics screen  
REMARKS: For composite monitors only. Values for x = 10-13,27,29,43,44,57 for various blue types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a green border around the current hi-res text/graphics screen  
REMARKS: For RGB monitors. Values for x= 2,16-23,58 for various green types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a green border around the current hi-res text/graphics screen  
REMARKS: For composite monitors. Values for x= 1-3,17-19,33,34,49,50 for various green types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a yellow border around the current hi-res text/graphics screen  
REMARKS: For RGB monitors. Values for x= 48-55,62 for various yellow types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a yellow border around the current hi-res text/graphics screen  
REMARKS: For composite monitors. Values for x= 4,20,35.51-53,63 for various yellow types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a brown border around the current hi-res text/graphics screen  
REMARKS: For RGB monitors. x = 48 or 6

COMMAND: POKE &HFF9A,x  
RESULT: Creates a brown border around the  
current hi-res text/graphics screen  
REMARKS: For composite monitors. x = 20 or 5

COMMAND: POKE &HFF9A,x  
RESULT: Creates a cyan border around the  
current hi-res text/graphics screen  
REMARKS: For RGB monitors. Values for x=  
3,24-31,59 for various cyan types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a cyan border around the  
current hi-res text/graphics screen  
REMARKS: For composite monitors. Values for  
x= 14,15,30,31,45-47,60-62 for  
various cyan types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a magenta border around the  
current hi-res text/graphics screen  
REMARKS: For RGB monitors. Values for x=  
5,40-47,61 for various magenta  
types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a magenta border around the  
current hi-res text/graphics screen  
REMARKS: For composite monitors. Values for  
x= 9,24-26,40-42,56-58 for various  
magenta types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a white border around  
the current hi-res text/graphics  
screen  
REMARKS: For RGB monitors. Values for x=  
7,56,63 for various white types

COMMAND: POKE &HFF9A,x  
RESULT: Creates a white border around  
the current hi-res text/graphics  
screen  
REMARKS: For composite monitors. Values for  
x= 16,32,48 for various white types

COMMAND: POKE &HFF9A,0  
RESULT: Creates a black border around the  
current hi-res text/graphics screen  
REMARKS: None

COMMAND: POKE &HFF7F,0  
RESULT: EXECutes ROMPAK in multipak slot 1  
REMARKS: Requires a multipak

COMMAND: POKE &HFF7F,17  
RESULT: EXECutes ROMPAK in multipak slot 2  
REMARKS: Requires a multipak

COMMAND: POKE &HFF7F,34  
RESULT: EXECutes ROMPAK in multipak slot 3  
REMARKS: Requires a multipak

COMMAND: POKE &HFF7F,51  
RESULT: EXECutes ROMPAK in multipak slot 4  
REMARKS: Requires a multipak

COMMAND: PEEK (&HFFB0)-64  
RESULT: Returns color # in Palette 0  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB1)-64  
RESULT: Returns color # in Palette 1  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB2)-64  
RESULT: Returns color # in Palette 2  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB3)-64  
RESULT: Returns color # in Palette 3  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB4)-64  
RESULT: Returns color # in Palette 4  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB5)-64  
RESULT: Returns color # in Palette 5  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB6)-64  
RESULT: Returns color # in Palette 6  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB7)-64  
RESULT: Returns color # in Palette 7  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB8)-64  
RESULT: Returns color # in Palette 8  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFB9)-64  
RESULT: Returns color # in Palette 9  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBA)-64  
RESULT: Returns color # in Palette 10  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBB)-64  
RESULT: Returns color # in Palette 11  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBC)-64  
RESULT: Returns color # in Palette 12  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBD)-64  
RESULT: Returns color # in Palette 13  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBE)-64  
RESULT: Returns color # in Palette 14  
REMARKS: Color # is between 0 & 63

COMMAND: PEEK (&HFFBF)-64  
RESULT: Returns color # in Palette 15  
REMARKS: Color # is between 0 & 63

COMMAND: POKE &HFFB0,x+64  
RESULT: Stores color x in Palette 0  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB1,x+64  
RESULT: Stores color x in Palette 1  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB2,x+64  
RESULT: Stores color x in Palette 2  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB3,x+64  
RESULT: Stores color x in Palette 3  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB4,x+64  
RESULT: Stores color x in Palette 4  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB5,x+64  
RESULT: Stores color x in Palette 5  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB6,x+64  
RESULT: Stores color x in Palette 6  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB7,x+64  
RESULT: Stores color x in Palette 7  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB8,x+64  
RESULT: Stores color x in Palette 8  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFB9,x+64  
RESULT: Stores color x in Palette 9  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBA,x+64  
RESULT: Stores color x in Palette 10  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBB,x+64  
RESULT: Stores color x in Palette 11  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBC,x+64  
RESULT: Stores color x in Palette 12  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBD,x+64  
RESULT: Stores color x in Palette 13  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBE,x+64  
RESULT: Stores color x in Palette 14  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBF,x+64  
RESULT: Stores color x in Palette 15  
REMARKS: x is any color between 0 & 63

COMMAND: POKE &HFFBC,x+64  
RESULT: Changes foreground color of 32  
col screen to color x  
REMARKS: x is any # between 0 & 63

COMMAND: POKE &HFFBC,64  
RESULT: Restores to normal after previous  
command  
REMARKS: None

COMMAND: POKE &HFFBD,x+64  
RESULT: Changes background color of 32  
col text screen to color x  
REMARKS: x is any # between 0 & 63

COMMAND: POKE &HFFBD,82  
RESULT: Restores to normal after previous  
command  
REMARKS: Only for 32 col screens

COMMAND: POKE &HFFD9,0  
RESULT: High speed poke  
REMARKS: Doubles the speed of all operations  
Affects disk/printer I/O

COMMAND: POKE &HFFD8,0  
RESULT: Normal speed poke  
REMARKS: Restores to normal after the high  
speed poke

COMMAND: POKE &HFFDE,0  
RESULT: Switches to CoCo Compatible mode  
REMARKS: Disables all CoCo 3 commands

COMMAND: POKE &HFFDF,0  
RESULT: Switches to CoCo 3 mode  
REMARKS: Restore to normal after previous  
command.

COMMAND: PEEK(&HFFFE)\*256+PEEK (&HFFFF)  
RESULT: Returns 35867 with CoCo 3  
REMARKS: Helps ascertain if CoCo 2/3 is  
installed

COMMAND: PEEK(&HFFFE)\*256+PEEK (&HFFFF)  
 RESULT: Returns 40999 with CoCo 2  
 REMARKS: Same as for previous command

COMMAND: A=LPEEK(442368+((y-1)\*80)+((x-1)  
 \*2)+1):A=A AND 128  
 RESULT: Returns A=128 if character at  
 column x, row y is blinking  
 REMARKS: For 80 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*80)+((x-1)  
 \*2)+1):A=A AND 64  
 RESULT: Returns A=64 if character at  
 column x, row y is underlined  
 REMARKS: For 80 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*64)+((x-1)  
 \*2)+1):A=A AND 128  
 RESULT: Returns A=128 if character at  
 column x, row y is blinking  
 REMARKS: For 64 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*64)+((x-1)  
 \*2)+1):A=A AND 64  
 RESULT: Returns A=64 if character at  
 column x, row y is underlined  
 REMARKS: For 64 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*40)+((x-1)  
 \*2)+1):A=A AND 128  
 RESULT: Returns A=128 if character at  
 column x, row y is blinking  
 REMARKS: For 40 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*40)+((x-1)  
 \*2)+1):A=A AND 64  
 RESULT: Returns A=64 if character at  
 column x, row y is underlined  
 REMARKS: For 40 col screen only



COMMAND: A=LPEEK(442368+((y-1)\*80)+((x-1)\*2)+1):A=A AND 56:A=A/8  
 RESULT: Returns in A, the palette # which contains the foreground color of the character at column x, row y  
 REMARKS: For 80 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*80)+((x-1)\*2)+1):A=A AND 7  
 RESULT: Returns in A, the palette # which contains the background color of the character at column x, row y  
 REMARKS: For 80 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*64)+((x-1)\*2)+1):A=A AND 56:A=A/8  
 RESULT: Returns in A, the palette # which contains the foreground color of the character at column x, row y  
 REMARKS: For 64 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*64)+((x-1)\*2)+1):A=A AND 7  
 RESULT: Returns in A, the palette # which contains the background color of the character at column x, row y  
 REMARKS: For 64 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*40)+((x-1)\*2)+1):A=A AND 56:A=A/8  
 RESULT: Returns in A, the palette # which contains the foreground color of the character at column x, row y  
 REMARKS: For 40 col screen only

COMMAND: A=LPEEK(442368+((y-1)\*40)+((x-1)\*2)+1):A=A AND 7  
 RESULT: Returns in A, the palette # which contains the background color of the character at column x, row y  
 REMARKS: For 40 col screen only

## UTILITY ROUTINES

COMMAND: 10 R=0:FORI=442368 TO 444288 STEP 2  
20 A= LPEEK(I):R=R+1  
30 A=A AND 127:IF A>95 THEN A=A-64  
40 PRINT #-2,CHR\$(A);  
50 IF R>39 THEN PRINT #-2:R=0  
60 NEXT

RESULT: Text Screen Dump for 40 col screen  
REMARKS: Make sure correct baud rate is set  
before RUNNING this routine

COMMAND: 10 R=0:FORI=442368 TO 445440 STEP 2  
20 A= LPEEK(I):R=R+1  
30 A=A AND 127:IF A>95 THEN A=A-64  
40 PRINT #-2,CHR\$(A);  
50 IF R>63 THEN PRINT #-2:R=0  
60 NEXT

RESULT: Text Screen Dump for 64 col screen  
REMARKS: Make sure correct baud rate is set  
before RUNNING this routine

COMMAND: 10 R=0:FORI=442368 TO 446208 STEP 2  
20 A= LPEEK(I):R=R+1  
30 A=A AND 127:IF A>95 THEN A=A-64  
40 PRINT #-2,CHR\$(A);  
50 IF R>79 THEN PRINT #-2:R=0  
60 NEXT

RESULT: Text Screen Dump for 80 col screen  
REMARKS: Make sure correct baud rate is set  
before RUNNING this routine

COMMAND: LPOKE &H60000,3:IF LPEEK(&H40000)=3  
AND LPEEK(&H20000)=3 THEN ?"128K"

RESULT: Tests if 128K RAM is present  
REMARKS: Prints 128K if 128K RAM is present

COMMAND: LPOKE &H40000,33:IF  
 LPEEK(&H00000)=3 THEN ?"256K"  
 RESULT: Tests if 256K RAM is present  
 REMARKS: Prints 256K if 256K RAM is present

COMMAND: CLEAR 200,&H6000:POKE  
 &HFFA3,&H36:SAVEM "40COL",  
 24576,&H7FFF,44539:POKE &HFFA2,123  
 CLEAR 200,&H7FFF  
 RESULT: Saves the current 40 col text  
 screen to disk  
 REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36  
 :CSAVEM "40COL", 24576,&H7FFF,  
 44539:POKE &HFFA2,123:POKE  
 &HFFA3,123:CLEAR 200,&H7FFF  
 RESULT: Saves the current 40 col text  
 screen to tape  
 REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36:  
 CSAVEM "64COL",24576,27648,0:POKE  
 &HFFA2,123:CLEAR 200,&H7FFF  
 RESULT: Saves the current 64 col text  
 screen to tape  
 REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36:  
 SAVEM "64COL",24576,27648,0:POKE  
 &HFFA2,123:CLEAR 200,&H7FFF  
 RESULT: Saves the current 64 col text  
 screen to disk  
 REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36:  
SAVEM "80COL", 24576,&H7FFF,  
44539:POKE &HFFA2,123:CLEAR 200,  
&H7FFF

RESULT: Saves the current 80 col text  
screen to disk

REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE  
&HFFA3,&H36:CSAVEM "80COL",  
24576,&H7FFF,44539:POKE &HFFA3,123:  
CLEAR 200,&H7FFF

RESULT: Saves the current 80 col text  
screen to disk

REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE  
&HFFA3,&H36:LOADM"40COL":POKE  
&HFFA3,123:CLEAR 200,&H7FFF

RESULT: Loads a previously saved 40 col  
text from disk

REMARKS: None

COMMAND: CLEAR 200, &H6000:POKE  
&HFFA3,&H36:CLOADM"40COL":POKE  
&HFFA3,123:CLEAR 200,&H7FFF

RESULT: Loads a previously saved 40 col  
text from tape

REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36:  
CLOADM "64COL":POKE &HFFA3,123:  
CLEAR 200,&H7FFF

RESULT: Loads a previously saved 64 col  
text screen from tape

REMARKS: None

COMMAND: CLEAR 200,&H6000:POKE &HFFA3,&H36:  
 LOADM "64COL":POKE &HFFA3,123:  
 CLEAR 200,&H7FFF  
 RESULT: Loads a previously saved 64 col  
 text screen from disk  
 REMARKS: None

COMMAND: CLEAR 200, &H6000: POKE  
 &HFFA3,&H36:LOADM"80COL":POKE  
 &HFFA3,123:CLEAR 200,&H7FFF  
 RESULT: Loads a previously saved 80 col  
 text from disk  
 REMARKS: None

COMMAND: CLEAR 200, &H6000:POKE  
 &HFFA3,&H36:LOADM"80COL":POKE  
 &HFFA3,123:CLEAR 200,&H7FFF  
 RESULT: Loads a previously saved 80 col  
 text from disk  
 REMARKS: None

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 2:POKE &HFFA3,48+I-1:SAVEM A\$+  
 CHR\$(I),24576,32767,0:POKE &HFFA3,  
 123:CLEAR 200,&H7FFF  
 RESULT: Saves HSCREEN 1,3 screen to disk  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 2:POKE &HFFA3,48+I-1:CSAVEM A\$+  
 CHR\$(I),24576,32767,0:POKE &HFFA3,  
 123:CLEAR 200,&H7FFF  
 RESULT: Saves HSCREEN 1,3 screen to tape  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 4:POKE &HFFA3,48+I-1:SAVEM A\$+  
 CHR\$(I),24576,32767,0:POKE &HFFA3,  
 123:CLEAR 200,&H7FFF  
 RESULT: Saves HSCREEN 2,4 screen to disk  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 4:POKE &HFFA3,48+I-1:SAVEM A\$+  
 CHR\$(I),24576,32767,0:POKE &HFFA3,  
 123:CLEAR 200,&H7FFF  
 RESULT: Saves HSCREEN 2,4 screen to tape  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 2:POKE &HFFA3,48+I-1:LOADM A\$+  
 CHR\$(I):POKE &HFFA3, 123:CLEAR  
 200,&H7FFF  
 RESULT: Loads HSCREEN 1,3 screen from disk  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 2:POKE &HFFA3,48+I-1:CLOADM A\$+  
 CHR\$(I):POKE &HFFA3, 123:CLEAR  
 200,&H7FFF  
 RESULT: Loads HSCREEN 1,3 screen from tape  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 4:POKE &HFFA3,48+I-1:LOADM A\$+  
 CHR\$(I):POKE &HFFA3, 123:CLEAR  
 200,&H7FFF  
 RESULT: Loads HSCREEN 2,4 screen from disk  
 REMARKS: Filename(fname) must <=6 characters

COMMAND: CLEAR200,&H6000:A\$="fname":FORI=1  
 TO 2:POKE &HFFA3,48+I-1:CLOADM A\$+  
 CHR\$(I):POKE &HFFA3, 123:CLEAR  
 200,&H7FFF  
 RESULT: Loads HSCREEN 2,4 screen from tape  
 REMARKS: Filename(fname) must <=6 characters

## HPRINT CONTROLS

This routine will allow you to modify the HPRINT graphics character set of the CoCo 3. First, design the character you want on the following grid:

128	64	32	16	8	4	2	1

Then take each row, add up the #'s of the "boxes" that contain a pixel. Do this for all 8 rows. Then determine which character you want to replace with your character. Then use the following command (s):

COMMAND: INPUT "CHARACTER TO REPLACE:";C\$;  
FOR I=0 TO 7:"# OF ROW ";I:  
INPUT A(I): POKE &HF09D+  
((ASC(C\$)-32) \*8)+I,A(I): NEXT I  
RESULT: Modifies HPRINT character C\$  
REMARKS: None

COMMAND: INPUT "CHARACTER";C\$:FOR I=0 TO 7:  
PRINT PEEK(&HF09D+((ASC(C\$) -32)  
\*8)+I): NEXT  
RESULT: Prints the 8 row #'s of character  
C\$

COMMAND: CSAVEM "NEWSET",&HF09D,&HF39C,  
PEEK(&H8A)  
RESULT: Saves the new character set to tape  
REMARKS: None

COMMAND: SAVEM "NEWSET",&HF09D,&HF39C,  
PEEK(&H8A)  
RESULT: Saves the new character set to disk  
REMARKS: None

COMMAND: CLOADM "NEWSET",PEEK (&H8A)  
RESULT: Loads a previously saved character  
set (for HPRINT) from tape  
REMARKS: None

COMMAND: LOADM "NEWSET",PEEK (&H8A)  
RESULT: Loads a previously saved character  
set (for HPRINT) from disk  
REMARKS: None



COMMAND: WIDTH 40: X=8192+(x-1)\*80:  
A=INT(X/256): B=X-(A\*256): POKE  
&HF866, A: POKE &HF867,  
B: X=X+80: A=INT(X/256): B=X-(A\*256):  
POKE &HFE06, A: POKE &HFE07, B

RESULT: Scroll Protects a portion of the  
40 col text screen

REMARKS: Scroll Protects 24-'x' lines from  
the bottom

COMMAND: POKE &HF866, 39: POKE &HF867, 48:  
WIDTH 40

RESULT: Restores normal scroll for 40 col

REMARKS: See previous command

COMMAND: WIDTH 80: X=8192+(x-1)\*160:  
A=INT(X/256): B=X-(A\*256): POKE  
&HF875, A: POKE &HF876, B: X=X+160:  
A=INT(X/256): B=X-(A\*256): POKE  
&HFE06, A: POKE &HFE07, B

RESULT: Scroll Protects a portion of the  
80 col text screen

REMARKS: Scroll Protects 24-'x' lines from  
the bottom

COMMAND: POKE &HF875, 46: POKE &HF876, 96:  
WIDTH 80

RESULT: Restores normal scroll for 80 col

REMARKS: See previous command

```

COMMAND: 10 CLEAR 2000: A$="SEARCH"
          20 FOR I=0 TO 34: FOR J=1 TO 18:
            DSKI$ PEEK(&HEB), I, J, B$, C$:
            IF INSTR(B$+LEFT$(C$, 127), A$)
              <> 0 OR INSTR(LEFT$(B$, 127)+C$,
                A$) <> 0 THEN ?"TR" I "SEC" J
          30 NEXT J, I

```

RESULT: Searches thru the disk for a phrase

REMARKS: A\$ should contain the search string. Lists all tracks/sectors which contain the string

```

COMMAND: FOR I=&H1D1 TO &H225: READ A$: POKE
          I, VAL("&H"+A$): NEXT: EXEC 465: DATA
          1A, 50, BD, A9, 28, 4F, 81, 38, 26, 2, 86, 39,
          81, 3F, 22, 23, B7, FF, A3, BE, 60, 0, E6, 84,
          6F, 84, 6D, 84, 26, 10, 6C, 84, 6D, 80, 27, A,
          E7, 1F, 8C: DATA 7F, FF, 23,
          EB, 4C, 20, DB, CE, 2, 1B, 20,
          3, CE, 2, 14, 8E, 4, 0, A6, C0, 27, 4,
          A7, 80, 20, FB, 20, FE, 35, 31,
          32, 4B, 4F, 4B, 0, 4D, 45, 4D, 4F, 52, 59,
          20, 42, 41, 44, 0

```

RESULT: Tests 512K RAM

REMARKS: Use two lines for this command. Initially the screen is cleared. After about 1 minute the message 512K OK or MEMORY BAD will appear. Press Reset Button to return to Basic. \*\* WARNING \*\* Save any programs before using this test This routine will lock up the computer if it is RUN on a computer system with less than 512K of memory.

COMMAND: FORI=&H1D1 TO &H226:READ A\$:POKE  
I,VAL("&H"+A\$):NEXT:EXEC 465:DATA  
1A,50,BD,A9,28,86,30,81,38,  
26,2,86,39,81,3F,22,23,B7,  
FF,A3,8E,60,0,E6,84,6F,84,6D,84,  
26,10,6C,84,6D,80,27,A,E7,1F,8C,7F,  
FF,23,EB,4C,20,08,CE,2:DATA  
1C,20,3,CE,2,15,8E,4,0,A6,CO,27,4,  
A7,80,20,F8,20,FE,31,32,  
38,4B,4F,4B,0,4D,45,4D,4F,  
52,59,20,42,41,44,0

RESULT: Tests 128K RAM

REMARKS: Initially the screen is cleared.  
After about 1 minute the message  
128K OK or MEMORY BAD will appear.  
Press Reset Button to return to  
Basic. \*\* WARNING \*\* Save any  
programs before using this test  
Use two lines for this command.

COMMAND: POKE 243,PEEK(&H168):POKE  
244,PEEK(&H169):X= PEEK(39) \*  
256+PEEK(40)-100: CLEAR 200,X:  
X=PEEK(39) \*256+PEEK(40)+1: FORI=X  
TO X+&H2B:READ A\$:POKE I,  
VAL("&H"+A\$):NEXT:EXEC X:DATA  
BE,1,68,AF,8D,0,13,30,8D,0,4,BF,1,  
68,39,34,2,96,6F,81,FE,27,5,35,2,7E  
,0,0,35,2,81,D,26,F7,86,D,BD,A2,BF,  
86,A,20,EE,0

RESULT: Linefeed routine for printer

REMARKS: Sends CR+LF with every CR code.  
Great for printers that seem to  
print on the "same line"

COMMAND: POKE &H168,PEEK(243):POKE  
&H169,PEEK(244)

RESULT: Restores to normal after previous  
command

REMARKS: CR no longer generates CR + LF

COMMAND: X= PEEK(39) \* 256+PEEK(40)-100:  
 CLEAR 200,X: X=PEEK(39)  
 \*256+PEEK(40)+1: FORI=X TO  
 X+&H33:READ A\$:POKE I,  
 VAL("&H"+A\$):NEXT:EXEC X:DATA  
 BE,1,68,AF,8D,0,13,30,8D,0,4,  
 BF,1,68,39,34,2,96,6F,81,FE,27,  
 5,35,2,7E,0,0,35,2,81,D,26,F7,  
 86,20,34,4,F6,1,14,5A,BD,A2,BF,  
 5A,26,FA,35,4,20,ES  
 RESULT: Allows you to define left margin  
 for your printer  
 REMARKS: See next command

COMMAND: POKE &H114,x  
 RESULT: Sets left margin for previous  
 command  
 REMARKS: x is the value of left margin

COMMAND: FORI=0 TO 8:POKE &HFF9C,I:FOR DE=  
 1 TO 250:NEXT:NEXT  
 RESULT: Smooth vertical scrolling  
 REMARKS: For 40 col screen only

COMMAND: FORI=0 TO 8:POKE &HFF9C,I:FOR DE=  
 1 TO 250:NEXT:NEXT  
 RESULT: Smooth vertical scrolling  
 REMARKS: For 80 col screen only

#### MEMORY MANAGEMENT

This section will describe the CoCo 3's  
 Memory Management Unit as simply as possible  
 as well as the different commands associated  
 with bank switching.

In 128K CoCo 3, there is 128K of RAM.  
 However, this RAM is not treated as one  
 block. It is divided into 8K sections.  
 Therefore in 128K computer, there are 16  
 (128K/8K) blocks and in 512K computer there

are 64 (512K/8K) blocks. Each of these blocks has a number, starting with 00 through 63. These blocks are "hidden" inside the computer. All these blocks can't be accessed at one time because the CoCo can only access 64K at one time. Here is where the MMU comes into play. The CoCo's 64K work space is also divided into eight 8K "empty" blocks. Each of these "empty" 8K blocks can be filled with the "hidden blocks" described above. How is it done? Simple. Each of the 8K "empty" blocks is assigned its own Memory Register. By POKEing the value of a "hidden" block into this Memory Register we are effectively taking the "hidden" block and storing it in the "empty" block to store information in. Here are the POKES for moving different blocks into the CoCo workspace. DO NOT attempt to use these POKES unless you have a thorough understanding of this process or you could crash your computer.

COMMAND: POKE &HFFA0,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$0000-\$1FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA1,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$2000-\$3FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA2,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$4000-\$5FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA3,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$6000-\$7FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA4,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$8000-\$9FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA5,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$A000-\$BFFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA6,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$C000-\$DFFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFFA7,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$E000-\$FFFF  
REMARKS: x = block # between 0 & 63

Tandy (R) also included another set of  
memory locations besides the ones listed  
above. Here they are!

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFA8,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$0000-\$1FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFA9,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$2000-\$3FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAA,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$4000-\$5FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAB,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$6000-\$7FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAC,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$8000-\$9FFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAD,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$A000-\$BFFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAE,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$C000-\$DFFF  
REMARKS: x = block # between 0 & 63

COMMAND: POKE &HFF91,PEEK(&HFF91) OR 1:  
POKE &HFFAF,x  
RESULT: Puts "hidden block" x into CoCo's  
memory locations \$E000-\$FFFF  
REMARKS: x = block # between 0 & 63

## RESET PROTECTION

Many of these COMMANDs given in this book allow you to make changes to the "ROM". However, as soon as you press the RESET, all the changes are lost. The next set of POKEs will allow you to RESET PROTECT your changes made to the "ROM". Do not use Cassette I/O when using RESET PROTECTION.

COMMAND: POKE 114,1:POKE 115,&HD1:FOR  
I=&H1D1 TO &H1D9:READ A\$:POKE  
I,VAL("&H"+A\$):NEXT:DATA 12,1A,  
50,7F,FF,DF,7E,CO,E7

RESULT: Reset Protection  
REMARKS: For Disk Basic 1.1 Only

COMMAND: POKE 114,1:POKE 115,&HD1:FOR  
I=&H1D1 TO &H1D9:READ A\$:POKE  
I,VAL("&H"+A\$):NEXT:DATA 12,1A,  
50,7F,FF,DF,7E,CO,D4

RESULT: Reset Protection  
REMARKS: For Disk Basic 1.0 Only

COMMAND: POKE 114,1:POKE 115,&HD1:FOR  
I=&H1D1 TO &H1D9:READ A\$:POKE  
I,VAL("&H"+A\$):NEXT:DATA 12,1A,  
50,7F,FF,DF,7E,80,CO

RESULT: Reset Protection  
REMARKS: For non-disk systems

COMMAND: POKE 114,&H80:POKE 115,&HCO  
RESULT: Restores normal RESET  
REMARKS: For non-disk systems

COMMAND: POKE 114,&HCO:POKE 115,&HE7  
RESULT: Restores Normal RESET  
REMARKS: For Disk Basic 1.1 Only  
For 1.0 use, POKE 114,&HCO:  
POKE 115,&HD4