

**HORIZON
RAMDISK
ROS 8.14**

COPYRIGHT 1990 DPH AND BUD HILLS SERVICES

```

#####;
: #####;
: : #####;
: : : #####;
: : : :
: : : :   RAMBO/HORIZON   : : : :
: : : :
: : : :   USER OPERATING MANUAL   : : : :
: : : :
: : : :   FOR SOFTWARE SERIES 0   : : : :
: : : :
: : : : #####< : : :
: : : : #####< : :
: : : : #####< :
#####<

```

ENHANCED part of the software by OPA Oasis Pensive Abacutors
ORIGINAL part of the software by John Johnston and Micheal Ballmann

MENU program by John Johnston and copyrighted by Bud Mills Services

RAMBO hardware by OPA Oasis Pensive Abacutors
HORIZON hardware by Horizon Computer Ltd. and Bud Mills Services

ENHANCED part of the software and RAMBO hardware
Copyrighted 1990 by OPA Oasis Pensive Abacutors

ORIGINAL part of the software and HORIZON hardware
Copyrighted 1988 by Bud Mills Services

This manual has been expanded and originally adapted from different manuals by John Johnston and Bud Mills Services and is copyrighted by Bud Mills Services and by OPA Oasis Pensive Abacutors.

No part of the manuals, publications or software associated with this package may be reproduced in any manner including storage in a retrieval system or transmittal via electronic means, or other reproduction in any form or by any method (electronic, mechanical, photocopying, recording, or otherwise) without prior written permission of either OPA Oasis Pensive Abacutors or Bud Mills Services.

OPA Oasis Pensive Abacutors and Bud Mills Services reserves the right to make changes at any time to improve the overall design and to supply the best possible product for the spectrum of users.

TABLE OF CONTENTS

1.0 INTRODUCTION TO HORIZON SYSTEM SOFTWARE

1.1	Notes on switching over to the latest version.....	3
1.2	Overview of the CF6 (ConFIGure program).....	3
1.3	Overview of the ROS (Randisk Operating System).....	3
1.4	Overview of the auto-start MENU program.....	4
1.5	Overview of the USER DSR system.....	4
1.6	Notes on the other programs included.....	4

2.0 USING THE CONFIGURE PROGRAM

2.1	Loading the configure program.....	5
2.2	CF6 test result screen.....	5
2.3	CF6 screen for a Horizon card.....	6
2.3.Q	Quitting the configure program.....	7
2.3.N	Displaying the next Horizon card.....	7
2.3.L	Loading of a ROS into a Horizon card.....	7
2.3.S	Saving the configed ROS.....	8
2.3.F	Changing the built-in character font.....	8
2.3.E	Changing the configure of a Horizon card.....	8
2.3.D	Setting up your different randisks.....	9
2.3.U	Loading of a USER DSR into a RAMBO/HORIZON card....	10

3.0 USING THE RANDISK OPERATING SYSTEM

3.1	What is a CALL?.....	11
3.2	The custom CALLs.....	11
3.2.1	What is a "auto-start" program?.....	12
3.2.2	Using the custom calls.....	12
3.3	The built-in CALLs.....	12
3.3.AQ	Enable the auto-start program.....	12
3.3.AF	Disable the auto-start program.....	12
3.3.WO	Turn on write-protection.....	12
3.3.WF	Turn off write-protection.....	13
3.3.DN	Changing a randisk drive number.....	13
3.3.LD	Load a program directly.....	13
3.4	The BSK. and BSKx. emulator.....	13
3.4.8	The SCRATCH RECORD opcode.....	14
3.4.9	The STATUS opcode.....	14
3.4.A	The ASSEMBLY LOADER opcode.....	15
3.4.B	The BASIC LOADER opcode.....	15
3.4.C	The CARTRIDGE LOADER opcode.....	16
3.5	What are low-level sub-programs?.....	16
3.5.00	BANK OPERATOR for RAMBO/HORIZON cards.....	17

TABLE OF CONTENTS

4.0 USING THE AUTO-START MENU PROGRAM

4.1	Show directory function.....	21
4.2	Display a file function.....	21
4.3	Run a program function.....	22
4.4	The cartridge function.....	22
4.5	Misc. commands/functions.....	22
4.6	The user options.....	23
4.6.1	Using the different user options.....	23
4.6.2	Changing the user options.....	24
4.6.3	Saving the changed MENU program.....	25

5.0 NOTES ON DIFFERENT THINGS

5.1	BASIC demo of the new opcode #B.....	26
5.2	Demo of the USER DSR functions.....	26
5.3	Slide-Show demo using the RAMBO program space.....	26
5.4	Where and how to market your programs for RAMBO.....	27
5.5	Where the support BBS's are for RAMBO/HORIZON owners.	27
5.6	Where and from what to get more information.....	27

6.0 CREDITS FOR THIS PACKAGE

If you have any questions about this package or have any problems in using the software, please feel free to write or phone us at the following addresses:

OPA OASIS PENSIVE ABACUTORS
432 Jarvis Street Suite 502
Toronto, Ontario Canada M4Y-2H3

416-960-0925 8am - 11pm EST Ask for Gary

BUD MILLS SERVICES
166 Dartmouth Drive
Toledo, OH. 43614

419-385-5946 8am - 8pm EST Ask for Bud

INTRODUCTION TO HORIZON SYSTEM SOFTWARE

1.1 NOTES ON SWITCHING OVER TO THE LATEST VERSION

This software is for use with all Horizon cards. It replaces the operating system diskettes you received with your Horizon and all previous versions of the CF6, ROS, and MENU programs. The CF6 program with this package can config any Horizon card built using 8K*8, 32K*8 or 128K*8 STATIC RAM chips, and also any HORIZON card upgraded with the RAMBO hardware.

*** NOTE! ***

The SERIES 8 operating system software requires the CF6, ROS, and MENU programs in this package to be used together. Absolutely no mix and match compatibilities with previous lower versions of programs can be tolerated! To insure you have matched files, take any and all Horizon files, disks, books or rumors to another room and store it' Get them off your desk' As an additional precaution, CF6 won't load anything except the new SERIES 8 of ROS, and MENU won't run off a Horizon with anything other than a SERIES 8 operating system.

*** CAUTION ***

As the SERIES 8 operating system is radically different from any previous versions, the initial loading of this operating system into your Horizon card will wipe out all files which existed on your Horizon card at the time of loading if that Horizon card was configed using any lower versions of operating systems like 7.3. We suggest in this case, or in case you are unsure of the version you are using, to make a backup of all files on all your Horizon cards before loading the new CF6 program.

1.2 OVERVIEW OF THE CONFIGURE PROGRAM

The CF6 program is designed to allow you to load the ROS into your Horizon cards and configure each card to your liking. The CF6 program scans the cards in your expansion box upon loading, allowing it to detect the type of cards contained in your system and to auto-modify the ROS on each of your Horizon's to match your particular system in the best possible way. This allows you to be able to configure your Horizon card the same way no matter what model or size your Horizon is and no matter how many fancy or different cards you have in your computer system. The CF6 program allows you: To select the drive numbers, disknames and size of each of your ramdisks. To select the default colors and enable/disable the auto-start program. To configure the custom calls.

1.3 OVERVIEW OF THE RAMDISK OPERATING SYSTEM

The ROS program is the main control DSR (Device Service Routine) for all your ramdisk drives, each Horizon card must have a proper ROS loaded into it by the CF6 program for your system to be able to access the ramdisk and/or RAMBO functions. The ROS interfaces with the main computer's operating system and becomes active each time a program you are using scans for a certain device like DSK, PIO, RS232, etc. The ROS defines the type and size of each of your ramdisks and handles the data requested by a program. The new SERIES 8 of the ROS is so far the fastest ramdisk operating system

INTRODUCTION TO HORIZON SYSTEM SOFTWARE

written for your Horizon card and adds many more features to your system than merely emulating a floppy disk drive; in fact, if you have RAMBO on one of your Horizons and together with the new USER DSR feature, your Horizon may soon reach outside the limits of your mind and be able to control new future devices.

1.4 OVERVIEW OF THE AUTO-START MENU PROGRAM

One of the main things users liked about the Horizon ramdisk software, was John Johnston's MENU program. Well, of course we wouldn't want you to give up this great MENU program, so we did some minor changes to allow the MENU program to catalog and load programs from the expanded range of drive numbers, whose range now includes the letters A thru Z; but more on that later. For those not familiar with JJ's MENU program, it is what is called a "AUTO-START" program which means if your Horizon card has auto-start enabled and this program is on your first ramdisk drive, you will see a powerful MENU program appear each time you turn on your computer or RESET it. The MENU program allows you to load programs from any storage device like floppy, ramdisk, hard drives at the touch of one key press; allows cataloging of the different ramdisks and marking files to run, view, delete, etc. The MENU program even allows you to access the Horizon CALLs which allow you to change ramdisk drive numbers, etc. We think this feature was the only one which did not need any improving, because it is already the best MENU program out there.

1.5 OVERVIEW OF THE USER DSR SYSTEM

With the addition of RAMBO to a Horizon card, the operating-system-reserved-space was expanded from the normal 8K to 16K, but, with all the byte-saving we did in bringing you the new SERIES 8 software, we decided that, instead of leaving over 1K of space not doing anything, we would add a feature allowing the USER to load in a file containing up to three things: POWER-UP DSR, INTERRUPT DSR, and ONE DSR NAME. This USER DSR function opens up a whole new world, allowing programmers to add new features or unique functions to the ROS, or write new DSR's which don't have to anything to do with the Horizon system or RAMBO; in fact, they could control new devices like mice, MIDI, digitizers, etc.; devices which normally must contain their own DSR in each program that is able to use them. Watch for third-party USER DSR files or future news from OPA on our own USER DSR files.

1.6 NOTES ON THE OTHER PROGRAMS INCLUDED

Besides the main CFG, ROS and MENU programs, at least three extra programs are included in this package, they are as follows: OPBDemo, it is an Extended BASIC program which demos the new opcode #B (Scratch Record) which until now was not supported by any device on the market. The other two work only on Horizons with the RAMBO hardware installed, they are: SHDW, a high-speed TI-ARTIST slide show program which demos the high-speed of the RAMBO bank switching; and USRDSR, a sample USER DSR file which demos a power-up, interrupt and a DSR name. All the above mentioned DEMO programs include fully commented assembly source code for those programmers at heart out there, (I hope you get some great ideas!). Any other programs that may be included in this package are described in the 'READ-ME' file contained on each diskette in this package.

USING THE CONFIGURE PROGRAM

2.1 LOADING THE CONFIGURE PROGRAM

BASIC: To load the CFG program from a EXTENDED BASIC system, run the program file called "LOAD" on the main diskette in this package; this file is a simple E/A #5 loader which will, upon running, try to load a file called "CFG" from "DSK1.". Since this loader is called "LOAD" upon selection of the EXTENDED BASIC and if the main diskette is in drive #1, the Extended Basic system will auto-load and run the "LOAD" file which in turn will load and run the CFG program.

OTHER: The CFG program is a normal E/A #5 type file which means it will load from most loaders on the market like MENU, BOOT, FUNNELWEB, or of course from the Editor/Assembler module option #5 (RUN PROGRAM FILE). Select the RUN PROGRAM FILE option or the similar option in your favorite loader, and load the CFG file from a floppy or other type of device. I.E. "DSK1.CFG"

2.2 CONFIGURE TEST RESULT SCREEN

After loading the CFG program, the screen will clear and display a list of valid CRU addresses and the definition of the card DSR software found at each CRU address. The screen will look very much like the one below, from any system, except that you may have a different set of cards.

```

#####;
: RAMBO/HORIZON CONF16 TEST RESULT SCREEN :
: Copyrighted 1990 by OPA and Gary Bowser :
: :
: CARD DEFINITION OF : TOTAL RAM- PROG :
: ADDR. CARD SOFTWARE : SIZE DISK RAM :
: ----- : ----- :
: >1000 NYARC RANDISK : : :
: >1100 DRIVE CONTROL : : :
: >1200 RAMBO/HORIZON # 256K 184K 72K :
: >1300 RS232 and PIO : : :
: >1400 8K88 HORIZON # 192K 186K 0K :
: >1600 32K88 HORIZON # 384K 384K 0K :
: >1700 PERAM CARD : : :
: >1900 NYARC RANDISK : : :
: >1800 AXION PRINTER : : :
: >1D00 CLOCK CARD : : :
: ----- : : :
: CARDS FOUND: 10 TOTAL PROG RAM: 72K :
#####<

```

What the CFG program has done was to scan the full range of CRU addresses from >1000 to >1F00 and then display a list of the cards found and at what CRU address they were enabled; the CFG then tried to define the type of DSR software which is used to interface your cards to the CPU; cards which are rare or new on the market will either be listed as "UNDEFINABLE" or as "UNKNOWN CARD". Cards which don't have any on-board DSR will not be listed at all.

USING THE CONFIGURE PROGRAM

```
: 4 MENU      2 U4      2 U7      :  
: 3 CFG       2 U5      2 U8      :  
: 2 U3        2 U6      2 U9      :  
:                                     :  
: Screen Color: 5   Total Sectors: 1016 :  
:                                     :  
:   Text Color: 16   Powerup On? Y      :  
:                                     :  
: User DSR:                                     :  
:                                     :  
: Version:  ROS   CFG   0.14   04/25/90 :  
:                                     :  
: Quit Next Load Save Font Edit Dsk User :  
#####
```

Without a valid ROS loaded, you will only be able to quit the CFG program, step to the next Horizon card in your expansion box, or load a proper valid ROS into your Horizon card. If a valid ROS is loaded, you will also be able to save the current configure and ROS to a device, change the built-in character font, and change the current configure of your ROS and raadisks; but only if your Horizon card contains the RAMBO hardware will you be able to load in a USER DSR file.

2.3.Q Quitting the configure program

By pressing 'Q', the CFG program will re-display the TEST RESULT screen. All memory on your Horizon cards will be re-tested, and the PROGRAM SPACE for your RAMBO/HORIZON cards will be updated. From the TEST RESULT screen you can then press FCTN = (QUIT) to exit the CFG program.

2.3.N Displaying the next Horizon card

By pressing 'N', the next Horizon card in your expansion box will be displayed, and the top status line will show its total size, etc. If you have only one Horizon card in your expansion box, the 'N' keypress will be ignored.

2.3.L Loading of a ROS into a Horizon card

By pressing 'L', you can then load in a valid ROS into the current Horizon card displayed. If the Horizon card already contains a valid ROS in memory you are first asked if you would like to keep your old diskette information; the reason for this is that when the ROS is loaded, it completely wipes out the full 10K of DSR memory in the card. Included in this 10K of memory is information on how you had previously formatted (initialized) or partitioned your raadisk. If you would like to preserve this table of data to prevent having to re-initialize all your disks and reloading all your files onto your Horizon card, answer the prompt with a 'Y'. Next you are prompted for a filename. Press ENTER when you are finished changing the pathname to the correct filename and device. If the file you loaded was a valid ROS then the system will display the new configure for your Horizon card, if not, you will get a ERROR message asking you to press any key and your Horizon card will be returned to the state it was before you tried to load the ROS.

USING THE CONFIGURE PROGRAM

2.3.5 Saving the configed ROS

By pressing 'S', you are then prompted for the filename and device to save the configed ROS from your Horizon card to. We recommend you never overwrite the original ROS which is supplied with this package; you never can tell when you might need the original untouched file to correct a problem which might have slipped into a saved ROS. If any errors occurred during saving, the system will display a ERROR message asking you to press any key, and your Horizon card will be returned to the state it was before you tried to save the ROS.

2.3.F Changing the built-in character font

The ROS contains a built-in character set which it uses when a auto-start program like the MENU is loaded, as well as anytime a ASSEMBLY program is loaded thru one of the custom calls or thru the LD (Load Directly) device. By pressing 'F', you are then prompted for the filename and device of a CHARA1 compatible character definition file; this allows you to change the default character set built into the ROS which is currently loaded on your Horizon card. If the loading of the CHARA1 file is successful, and the new CHARA1 file just loaded was different than the one in memory, you will notice a change in the appearance of the characters on the screen. If any errors occurred during loading, the system will display a ERROR message asking you to press any key, and your Horizon card will be returned to the state it was before you tried to change the character set. CHARA1 compatible files can be found with most TI-WRITER type programs and some other programs on the market; in fact, there is a program out there which even allows you to make and edit your own CHARA1 files.

2.3.E Changing the configure of a Horizon card

By pressing 'E', you will then be able to edit the configure of your custom calls, color settings, and enable/disable the auto-start program. The down and up arrows keys are active when in edit mode allowing you to easily move thru the fields until you get the configure to your liking. When you are finished with the editing, pressing FCTN 9 (BACK) will bring you back to the CFG main screen and menu. The following describes the fields which you can edit:

CUSTOM CALL NAME LENGTH:

The custom call name length can be any value from 1 to 5 and controls the number of characters you must type to activate the custom call. For example: Suppose you name a call 'PTERN' with a length of 5; you would then have to type all five characters to be able to load the program called 'PTERN' on your raedisk; but, by changing the length down to 1 you would then only have to type 'P' to activate the loading of the program, but the program would still be named 'PTERN' on your raedisk.

CUSTOM CALL NAME:

This is the filename of the first file to start loading from the first raedisk on your Horizon card when a user calls this program from BASIC or inside another program he/she is using. It can be any mixture of letters and numbers up to a max. of 5 characters long. The custom call loader in the ROS allows for three different types of programs, namely: ASSEMBLY, BASIC, and CARTRIDGE, all of which are better explained in section 3.4 of this manual.

USING THE CONFIGURE PROGRAM

COLOR SETTINGS:

The foreground (text) and background (screen) color settings can be set to any mixture of the 16 colors using values ranging from 1 to 16. When changing the color settings you will notice the CFB text and screen colors will change too, allowing you to see the look of the colors you have just selected, plus the CFB programming makes sure you can't select a foreground and background of the same value which would make the text invisible. These colors will be evident when a program that doesn't setup its own colors is loaded thru one of the custom calls or the LD device. These colors are also used by the MENU program.

AUTO-START ENABLE/DISABLE:

The final field is called 'Powerup On?' and allows you to enable auto-start on your Horizon card by typing a 'Y' or to disable it by typing a 'N'. The auto-start function allows you to have a program loaded and run each time the TI title screen appears. If auto-start is enabled the ROS will then use the first custom call name as the filename of the program to be loaded upon power-up. This program must be on the first ramdisk on your Horizon card, and can be any ASSEMBLY, BASIC, or CARTRIDGE program that will work as a custom call.

2.3.D Setting up your different ramdisks

By pressing 'D', you will be able to edit the configure of the ramdisks on your Horizon card. The down and up arrows keys are active when in edit mode allowing you to easily move thru the fields until you get the configure to your liking. When you are finished with the editing, pressing FCTN 9 (BACK) will bring you back to the CFB main screen and menu. The following describes the fields which you can edit:

DRIVE NUMBER:

Each ramdisk on your Horizon card can have its own unique drive number. This field allows you to select the drive number of each ramdisk on your Horizon card. Valid drive numbers range from 1 to 9 and the letters A to Z. All other symbols will not be accepted. We also recommend you don't have more than one ramdisk with the same drive number or letter.

DISKETTE NAME:

Each ramdisk on your Horizon card can have its own unique diskname. This field allows you to type in up to a 10 character diskname for each ramdisk on your Horizon card. Disknames can use any mixture of symbols, numbers, and letters. Unformatted ramdisks on your Horizon card will be listed on the screen as '?' under the diskname field.

WRITE PROTECTION:

This field shows if a ramdisk on your Horizon card is write protected or not; its function is much like putting a write protect tab on one of your floppy disks. This field allows you to enable write protection by typing a 'Y' or disabling it with a 'N'; all other key presses are ignored.

USING THE CONFIGURE PROGRAM

FORMATING A RANDISK:

This field allows you to format, reformat or delete a randisk from your Horizon card. By typing a 'Y' in this field, the system will format the randisk after asking for the new size of the randisk. Typing a 'N' will move the cursor down one line to the next randisk on the screen, leaving all randisks intact. A 'D' for DELETE only works if the randisk you want to delete is the last one on the screen which has been formatted; this allows you to remove a randisk without losing any files on the randisks above it.

NOTE: Formatting a randisk causes all data to be lost on that randisk; also, all randisks below the one just formatted on the screen will be made unformatted causing their data to be lost too.

SIZE OF A RANDISK:

This field displays the size of a randisk in the total number of diskette sectors. If you selected 'Y' to format a randisk, then the size will display the total number of sectors still not formatted for randisk space on your Horizon card, and the cursor will be awaiting your input of the new size of the randisk. Valid range of the size is from 8 sectors to 1600 sectors in 8 sector increments, allowing you to have a randisk of almost any size wanted. The size after inputted is adjusted to make sure it is within the nearest 8 sector increment. If the size is over 1600 sectors or the max. number still unformatted, then the system will redisplay the amount still unformatted and await a new input of the size.

NOTE: If your Horizon card has more than 1600 sectors (400K) available for use, to be able to use all your memory you will have to have more than one randisk formatted on your card. For those Horizons with the RAMBO hardware installed, the amount left for PROGRAM space is the amount of memory which has not been formatted into a randisk. To help in calculating RAM K into sectors:- There are 4 disk sectors per each 1K of memory.

2.3.U Loading of a USER DSR into a RAMBO/HORIZON card

You will only be able to select the 'User' option from the CF6 main screen if your Horizon has the RAMBO hardware installed on-board. If so, by pressing 'U' you are then prompted for a filename and device of a USER DSR DISPLAY/FIXED 80 UNCOMPRESSED OBJECT type file. If any errors occurred during loading, the system will display a ERROR message asking you to press any key, and your Horizon will be returned to the state it was before you tried to load a USER DSR file into your RAMBO/HORIZON card. If the loading was successful, then the system will redo the CF6 main screen showing in the 'USER DSR:' line the DSR name (if any), and maybe the words POWER-UP and/or INTERRUPT depending if those functions are included in the USER DSR file that you just loaded. To clear a USER DSR from memory, you must reload the ROS into your Horizon card by using the 'Load' option from the CF6 main screen. If you are using a USER DSR file other than the demo one supplied with this package, we recommend you follow the information supplied with the USER DSR file you are using for the proper loading and usage of the file. For more information on USER DSR files see section 5.2 of this manual. For information on how to write your own USER DSR files, write to OPA Oasis Pensive Abacutors for the complete and complex programming guide for RAMBO.

USING THE RAMDISK OPERATING SYSTEM

3.1 WHAT IS A CALL?

A CALL is a type of command to the ROS which can be accessed from any BASIC and most running programs. The ROS contains a series of built-in calls, which allow the user to perform functions like loading of programs, changing drive numbers and write protection, etc. The ROS also contains a bunch of custom calls which can be configured thru the CFG program to load your favorite programs with just a simple CALL. These calls can be accessed the following three different ways:

FROM ANY BASIC WHEN IN COMMAND MODE:

When any BASIC is just sitting there in command mode, not running any programs, you can access the calls by typing 'CALL' a space and then the name of the call you wish to access. For example: 'CALL MENU' will run the auto-start MENU program on your ramdisk, providing you have setup one of the custom calls as MENU and the MENU program files are on the first ramdisk in your Horizon card.

USING THE CALLS IN A BASIC PROGRAM:

To access one of the CALLs from a running BASIC program, just enter a program line as illustrated below:

```
100 DELETE "PTERM"      ! runs a program named PTERM on your Horizon card.
```

The statement DELETE can be used exactly like the CALL statement when you would like to make the call from a running BASIC program. Just don't forget the quotes around the call name.

USING THE CALLS FROM ANOTHER PROGRAM:

In any program which gives you the option of doing a SAVE, LOAD, DELETE, or OPEN and prompts you for the complete filename, you can enter your call name (for example JKL) instead of a real filename. For instance, go into the TI-WRITER editor command line, and type 'DF' and press ENTER for the Delete File function, then type 'JKL' (or whatever name of the call you would like to run) and press ENTER. The ROS will now load and run the program called JKL on your ramdisk, providing you have setup one of the custom calls as JKL and the JKL program files are on the first ramdisk in your Horizon card. As you can see, you can run just about all your programs from almost any program that gives you the capability to RUN, SAVE, DELETE, or OPEN a file. As another example: If you were using PRBASE, you could configure your printer name to 'SET', and as soon as you tell PRBASE to print something, 'SET' would be loaded and run.

3.2 THE CUSTOM CALLS

Each Horizon card can have up to nine different ASSEMBLY, BASIC or CARTRIDGE programs loaded via the custom calls which the user has setup thru the CFG program. These calls allow you to quickly load programs from any BASIC and from almost any running program. More information on custom calls and their configure can be found in section 2.3.E of this manual.

USING THE RAMDISK OPERATING SYSTEM

3.2.1 What is a "auto-start" program?

If auto-start has been enabled on your Horizon card, the ROS will, each time the TI Title Screen appears, try to run the program for which the first custom call has been configured to. This program can be any program which works correctly when loaded thru a custom call. If the program can't be found, or if there is an error during loading, the ROS will then try to select the cartridge which is currently plugged into the TI module port, if no cartridge is found, the system will then display the TI Title Screen. You can bypass the auto-start by holding down either SHIFT key during a reset or during turning on your computer system. An example of a "auto-start" program is the MENU program supplied with this package. For more information on the MENU program see section 4.6 of this manual.

3.2.2 Using the custom calls

After the custom calls have been configured thru the CFG program, before you can use them, you must copy all the files for each program, which is going to be loaded via a custom call, to the first ramdisk on your Horizon card. The first ramdisk is the top-most ramdisk listed on the 'Drives' screen of the CFG program. If there are any errors during loading of a program thru a custom call, you may get a error message or nothing at all, in any case you will be returned to the program you where using when you or the program did the call. For more information on using the calls see section 2.3.E of this manual.

3.3 THE BUILT-IN CALLS

The ROS contains a number of built-in calls which allow you to change drive numbers, control write protection, and enable/disable the auto-start program. For computer setups with more than one ramdisk or more than one Horizon card, you can tell the system what ramdisk the call is for by adding the drive number after the call name. For example: 'CALL WD.8' This call would turn on the write protection of the ramdisk whose drive number is '8'. If the drive number is not added after the call name, the system will then use the first ramdisk on the first Horizon card in your expansion box. The following describes the different calls in detail.

3.3.A0 Enable the auto-start program

The call 'A0' allows you to enable the auto-start program. This is the same as typing a 'Y' in the 'Powerup On?' field in the CFG program.

3.3.AF Disable the auto-start program

The call 'AF' allows you to disable the auto-start program. This is the same as typing a 'N' in the 'Powerup On?' field in the CFG program.

3.3.W0 Turn on write-protection

The call 'W0' allows you to turn on write-protection for a ramdisk. This is the same as typing a 'Y' in the 'W/P' field on the 'Drives' screen of the CFG program.

USING THE RAMDISK OPERATING SYSTEM

3.3.WF Turn off write-protection

The call 'WF' allows you to turn off write-protection for a ramdisk. This is the same as typing 'N' in the 'W/P' field on the 'Drives' screen of the CF6 program.

3.3.DN Changing a ramdisk drive number

The call 'DN' allows you to change the drive number of a ramdisk. The format for this call is: 'CALL DN.o.n' where 'o' is the old drive number or letter, and 'n' is the new drive number or letter. It is recommended that both drive numbers always be included in the call string to make sure the correct ramdisk is changed. The ROS will only allow the drive number to be changed, if both the old and new drive's specified have no OPENed files on them.

3.3.LD Load a program directly

The device 'LD' allows you to load an ASSEMBLY, BASIC, or CARTRIDGE program directly into CPU memory. The loading system used by this device is much like the custom call loader, but allows you to load a program from any ramdisk and is not limited to five character filenames. The format for this device is: 'LD.x.name' where 'x' is the drive number, and 'name' is the filename of the program to be loaded. The 'LD' is a device similar to 'DSK1.' or 'DSK.' and is not a call, so the proper way to access it in any Basic is: RUN"LD.x.name" or DELETE"LD.x.name".

3.4 THE DSK. AND DSKx. EMULATOR

This is the main part of the ROS allowing each ramdisk on your Horizon card to look and act just like a floppy disk drive, allowing all programs designed to load and run off floppys to run unchanged on your Horizon card. This section of the manual will only describe the changes and additions to the standard opcodes and functions listed in the Editor/Assembler and floppy disk controller manuals. Two of the major additions are four new opcodes, which until now have never been supported by any device, and a new mode allowing data to be transferred directly to and from CPU memory without the need of going thru VDP RAM. For more information on these opcodes and their uses see the Editor/Assembler manual.

In BASIC you no longer have to specify the file attributes when you open a file which already exists. For example: Let's say you have a file called 'DATAFILE' on your ramdisk that's an INTERNAL, FIXED 192 file. Normally to open it you would type: 'OPEN #1:"DSK9.DATAFILE",INTERNAL, FIXED 192' but now, with this operating system, you just have to type: 'OPEN #1:"DSK9.DATAFILE"'. All accesses to the file called 'DATAFILE' will be handled exactly like any other disk controller would handle it. Of course, to create a new file, you must still specify its attributes when you open it if it is going to be something other than the BASIC default of DISPLAY, VARIABLE 80.

In ASSEMBLY you can now specify what kind of memory the ROS will use as the data buffer for reading and writing of records and files. The control bit for this new mode is bit 1 of the opcode byte in the PAB, (where bit 0 is the MSBit and bit 7 is LSBit). If the bit is set to 1, then the data buffer

USING THE RAMDISK OPERATING SYSTEM

address specified in bytes 2 and 3 of the PAB points to a CPU RAM location, otherwise, if the bit is reset to 0, then the data buffer address specified in bytes 2 and 3 of the PAB points to a VDP RAM location. This format is compatible with the type used by both the MYARC floppy controller card and their new hard drive/floppy controller card. For example: If the opcode byte contains >42 instead of the normal >02, then the ROS will read the record into CPU RAM instead of into VDP RAM.

Due to the physical design of the floppy disk drive, the floppy disk controller DSR uses a VDP RAM buffer to keep track of the OPENed files, etc. The number of files allowed OPENed at once is controlled thru the CALL FILES(x) command whose default is 3 files. But now with the super-fast operation of the SERIES 8 software we found that emulating this part of the floppy disk controller slowed the ramdisk speed way down, so we decided to go with an improved internal buffer system which besides speeding up the operation of the Horizon card, allows you to have up to 16 files OPENed at once on each Horizon card in your system without the need to use a CALL FILES(x) command. This change will not effect the operation of programs or the floppy disk controller because the ROS still updates certain needed and useful pointers in the old VDP RAM buffer that a program or the disk controller may want to access.

3.4.8 The SCRATCH RECORD opcode

When TI drew up the master plan of the DSR system, they defined the opcodes 0 thru 9; but in the master design, opcode 8 (called 'SCRATCH RECORD') was impossible to include in any disk controller due to the physical design of the floppy diskette. But now with the new super-fast code in the ROS, we were able to finally support the 'SCRATCH RECORD' opcode.

The PAB format for opcode 8 is the same as reading or writing a record with opcodes 2 and 3, except that bytes 6 and 7 specify the record number to be removed from the DISPLAY or INTERNAL, FIXED file instead of the record number to be read or written. Since the removing of a record is an internal matter, the VDP/CPU flag in the opcode byte is ignored as well as the DATA BUFFER ADDRESS in the PAB. This opcode will only work if the file has a FIXED record length and if the file has been OPENed using the OPEN opcode; if not, or if any errors occur during the removal of the record the ROS will return the normal error code in byte 1 of the PAB. Removing a record from a file causes all higher numbered records to become one less in their number. For more information on the 'SCRATCH RECORD' opcode see section 5.1 of this manual.

3.4.9 The STATUS opcode

The regular disk controller system includes a opcode called 'STATUS' which returns in byte 8 of the PAB the status of the file specified in the PAB, but due to the design of the internal code of the ROS system, the current status of the file will always be returned in byte 8 of the PAB no matter what opcode is used. This change will not affect normal operation as byte 8 is not used for anything else besides the status of a file or operation. The 'STATUS' opcode still works as listed in the Editor/Assembler manual.

USING THE RAMDISK OPERATING SYSTEM

3.4.A The ASSEMBLY LOADER opcode

This opcode in the ROS allows any programmer to easily load Editor/Assembler memory image PROGRAM files without writing a special loader. This opcode is also used internally by the ROS to load your ASSEMBLY programs via custom calls and the LD device. Bytes 2 and 3 of the PAB control, if the opcode is going to auto-execute the program after loading or not, and also control the starting execution address. In all cases bytes 2 and 3 of the PAB will contain the normal execution address, after loading of the program is completed. This allows many uses in new programs, the main one is a type of overlay loading system similar to the program called 'TELCO' by Charles Earl. The PAB format for opcode >A is as follows:

PAB FORMAT FOR OPCODE >A (ASSEMBLY):

Byte 0 = Hex value >0A for opcode >A (ASSEMBLY).

Byte 1 = not used.

Byte 2 and 3 = Execute Address: If ZERO, then use the normal execution address.

If greater than ZERO, then use this value as the execution address.

If >FFFF, then don't execute the program after loading is completed.

Byte 4 and 5 = not used.

Byte 6 and 7 = Max. length: If ZERO, then don't check each file's length.

If greater than ZERO, then only load the file if the file's

length is less than or equal to the value in Bytes 6 and 7.

Byte 8 = not used.

Byte 9 = Device name length.

Byte 10 and up = Device name. I.E. 'DSK1.UTIL1'

3.4.B The BASIC LOADER opcode

This opcode in the ROS allows any programmer to easily load BASIC PROGRAM files and EXTENDED BASIC, INTERNAL/VARIABLE 254 files without writing a special loader. This opcode is also used internally by the ROS to load your BASIC programs via custom calls and the LD device. Control of the computer system is passed over to the EXTENDED BASIC module after the ROS makes sure the file specified in the PAB is a BASIC program and that a compatible EXTENDED BASIC module is plugged into the cartridge port. The ROS doesn't load the program, it leaves actual loading and execution of the BASIC program up to the EXTENDED BASIC module; this being so, the program which activated this opcode does not have any control over the execution of the program after loading is completed. This opcode is most useful in a user control menu type program, and is similar to the built-in loader in the MENU program supplied with this package. The PAB format for opcode >B is as follows:

PAB FORMAT FOR OPCODE >B (BASIC):

Byte 0 = Hex value >0B for opcode >B (BASIC).

Byte 2 thru 8 = not used.

Byte 9 = Device name length.

Byte 10 and up = Device name. I.E. 'DSK1.LISTER'

USING THE RAMDISK OPERATING SYSTEM

3.4.C The CARTRIDGE LOADER opcode

This opcode in the ROS allows any programmer to easily load GRAM/KRACKER CARTRIDGE FORMAT PROGRAM files into the Horizon P-GRAM card without writing a special loader. This opcode is also used internally by the ROS to load your CARTRIDGE programs via custom calls and the LD device. Before loading of the cartridge whatever module (if any) that is plugged into the module port is overridden (or turned off), and during loading of the new cartridge the current cartridge in the P-GRAM may be overwritten; this being so, the programmer must be careful not to use this opcode from any ASSEMBLY or GPL program code within the cartridge or P-GRAM card. After loading of the CARTRIDGE is successfully completed, the P-GRAM card is turned on, and the ROS returns to the program that activated this opcode. This opcode is most useful in designing a very large CARTRIDGE type program which overlays many different module files similar to the ASSEMBLY overlay done in the program called 'TELCO' by Charles Earl. For more information on the CARTRIDGE format and how the loading system works, see the P-GRAM or the GRAM/KRACKER manuals. The PAB format for opcode >C is as follows:

PAB FORMAT FOR OP CODE >C (CARTRIDGE):

Byte 0 = Hex value >0C for opcode >C (CARTRIDGE).

Byte 2 thru 8 = not used.

Byte 9 = Device name length.

Byte 10 and up = Device name. I.E. 'DSK1.SXB'

3.5 WHAT ARE LOW-LEVEL SUB-PROGRAMS?

The second main part of the ROS has the low-level sub-programs which allow disk managers to copy and rename files, plus the more advanced things like sector editing and copying. This section will only describe the changes and additions to the low-level sub-programs. For more complete information on the different sub-programs you can use in your programs, read the book called 'TECHNICAL DRIVE' by Rye Data.

You can now specify what kind of memory the ROS will use as the data buffer for the low-level sub-programs. The control bit for this new mode is the MSBit of the drive number stored in >B34C. If the bit is set to 1 then all the data buffer addresses point to CPU RAM locations; otherwise, if the bit is reset to 0, then all the data buffer addresses point to VDP RAM locations. This format is compatible with the type used by both the NYARC floppy controller card and their new hard drive/floppy controller card. For example:- If the location >B34C contains a >B5 instead of the normal >B5 then the ROS will use CPU RAM when reading or writing to drive B5, instead of using VDP RAM.

The standard low-level sub-programs use the CPU PAB byte at >B34C to store the drive number the sub-program will be using for its operation. The normal range of this byte is from >01 for drive 01 to >09 for drive 09, but now the SERIES B software allow the letters A thru Z to be used as drive numbers, this being so, the byte at >B34C now also accepts the range of numbers from >11 for drive letter A thru >2A for drive letter Z. To calculate the exact value, take the ASCII value for the number or letter and subtract >30 (decimal 48) from it, this will give you the correct value to use.

USING THE RAMDISK OPERATING SYSTEM

3.5.00 BANK OPERATOR for RAMBO/HORIZON cards

This section of the manual will try to explain the memory management sub-program called >B0 and how to interface with it using assembly code. This low-level sub-program will only work on Horizon cards with the RAMBO hardware installed. If you need more help on using this sub-program see section 5.6 of this manual.

The original Horizon hardware only allowed the Horizon memory to be accessed in 2K blocks at the DSR space from >5800 thru >5FFF. The rest of the DSR space from >4000 thru >57FF was reserved for the Horizon ROS. This limited access to the memory on the Horizon card was fine for uses like a ramdisk, but to write a program which really executed from and used the Horizon memory was almost impossible. There needed to be a new way of accessing the Horizon memory, but without losing the wonderful and powerful ramdisk features and functions. With this in mind, we decided to design a simple upgrade kit for all Horizon cards which would allow the upgraded Horizon to work the same in the original Horizon mode; but when accessed thru a new mode it would switch to a whole new paging system which would be better designed for writing large programs, etc. The ROS would also have to be upgraded to support the new mode by adding a super memory management sub-program to ease in the writing of large programs as well as allowing both ramdisk files and program space to share the same memory.

Finally, with the RAMBO hardware and the SERIES 8 software, it was now possible to write large programs or programs which used the RAMBO program space for DATA storage very easily. The SERIES 8 software added a sub-program called >B0 to handle the complex memory management needed to make sure all programs run perfectly without destorying any files or other programs which may be sharing the memory on-board the Horizon card.

All the values for the sub-program >B0 are transfered thru CPU PAD addresses, which will be explained later on. Access to the sub-program >B0 is similar to the normal DSRLNK type >8 which is for DSR devices like "DSK1.", and "RS232."; except the DSR only needs a 2 byte VDP PAB of >01B0, the first byte being the length, and the second byte being the sub-program name >B0. The following section is divided into five different parts, the first two explaining the CRU interface bits, and the last three explaining the three different ways to access the sub-program >B0.

ORIGINAL HORIZON CRU INTERFACE ACCESS:

The original Horizon interface has a CRU address range from >1000 thru >1700, and the 8 or 16 CRU bits are defined as follows:

CRU BITS : Definition of the CRU interface bits (15=MSBit / 0=LSBit)

-
- 0 = 6K >4000 OFF/ON (0=ROS OFF, 1=ROS ON) Used by ROS and DSRLNK.
 - 1 thru 7 = 2K >5800 Page number 0 thru 127. (Max. of 256K for 8K*8 cards).
 - 8 thru 9 = Select one of four banks of 256K of memory. (Max. of 1024K for 32K*8 cards).
 - 10 thru 11 = Select one of four banks of 1024K of memory. (Max. of 4096K for 128K*8 cards).
 - 12 thru 14 = Undefined, may be used on a future series of Horizon cards.
 - 15 = Reset to 0: Used to change to RAMBO mode, on original Horizons it was undefined.

USING THE RAMDISK OPERATING SYSTEM

RAMBO CRU INTERFACE ACCESS:

A Horizon card with RAMBO installed can work the same as an original Horizon as detailed above or when bit 15 of the CRU interface is set to 1, it can be accessed as follows:

CRU BITS : Definition of the CRU interface bits (15=MSBit / 0=LSBit)

0	=	8K >4000 OFF/ON (0=RDS OFF, 1=RDS ON) Used by RDS and DSRLNK.
1 thru 2	=	Not used when in RAMBO mode, recommended both be reset to 0.
3 thru 7	=	8K >6000 Page number 0 thru 31. (Max. of 256K for 8K00 cards).
8 thru 9	=	Select one of four banks of 256K of memory. (Max. of 1024K for 32K00 cards).
10 thru 11	=	Select one of four banks of 1024K of memory. (Max. of 4096K for 128K00 cards).
12 thru 14	=	Undefined, may be used on a future series of Horizon cards.
15	=	Set to 1: Used to change to original Horizon mode.

ACCESS MODE #1 FOR SUB-PROGRAM >B0 (Gets Max. number of free BK pages):

834A	IN	Must be cleared before access to get true number of free pages.
	OUT	Max. number of free BK pages for all RAMBOs in the TI system.
834C	IN	>0000 = Gets the Max. number of free BK pages in the system.
	OUT	Not changed, will be the same value as IN.
8350	IN	Not used for input to the sub-program >B0.
	OUT	0000 if no errors, or >FFFF if where errors detected.

Accessing >B0 with >834A and >834C cleared will, upon exit, return to the program in >834A the max. number of free BK pages for all RAMBOs in the your expansion box. This access causes all cards to be switched to original Horizon mode, and being so can't be accessed from within the >6000 thru >7FFF space. The following is an assembly example which is accessed like so: BL @GETMAX.

REF	VMBW, DSRLNK	Normal VMBW and DSRLNK E/A BLMP's.
GETMAX	CLR @>834A	Clear Max. number of pages.
	CLR @>834C	Ask for max. number of free BK pages.
	LI R0, VDPPAB	Address of the PAB in VDP memory.
	LI R1, CALLB0	Address of the PAB in CPU memory.
	LI R2, 2	Length of the PAB to be written to VDP memory.
	BLWP @VMBW	Write the PAB (>0100) to VDP RAM.
	MOV R0, @>8356	Make >8356 point to the length byte of the PAB in VDP.
	BLWP @DSRLNK	Execute the sub-program >B0.
	DATA 'A	
	MOV @>8350, R0	Check for errors. EQUAL BIT IS SET IF NO ERRORS'
	RT	Return to program.
CALLB0	DATA 0100	PAB data for sub-program >B0.

USING THE RAMDISK OPERATING SYSTEM

ACCESS MODE #2 FOR SUB-PROGRAM >B0 (Selects an BK page and turns on RAMBO):

>B34A IN Not used for input to the sub-program >B0.
OUT CRU address of the RAMBO/HORIZON card that is currently on.
>B34C IN A page number from 1 to max. number of free BK pages.
OUT The 16 bit CRU value used to select the page and turn RAMBO on.
>B350 IN Not used for input to the sub-program >B0.
OUT >0000 if no errors, or >FFFF if where errors detected.

Accessing >B0 with >B34C containing the number of the page you want will turn on the correct RAMBO/HORIZON card and then select the page you specified, and upon exit will return the CRU address and CRU value used to select the page in >B34A and >B34C. This command also overrides any cartridge which may be plugged into the module port, and will also turn off the P-GRAM card if it was on. This function makes sure no other device is responding at the same >6000 space of memory. The following is an assembly example which is accessed like so: BL @SETPAG with the next line containing a line of DATA with the number of the page you want.

```
REF VMBW,DSRLNK Normal VMBW and DSRLNK E/A BLMP's.
SETPAG MOV #R11+,@>B34C Get number of the page to select.
LI R0,VDP PAB Address of the PAB in VDP memory.
LI R1,CALLBO Address of the PAB in CPU memory.
LI R2,2 Length of the PAB to be written to VDP memory.
BLMP @VMBW Write the PAB (>0100) to VDP RAM.
MOV R0,@>B356 Make >B356 point to the length byte of the PAB in VDP.
BLMP @DSRLNK Execute the sub-program >B0.
DATA >A
MOV @>B350,R0 Check for errors. EQUAL BIT IS SET IF NO ERRORS'
RT Return to program.
CALLBO DATA >0100 PAB data for sub-program >B0.
```

ACCESS MODE #3 FOR SUB-PROGRAM >B0 (Turns off all RAMBO/HORIZON cards):

>B34A IN Not used for input to the sub-program >B0.
OUT Not changed, will be the same value as IN.
>B34C IN >FFFF = Tell all cards to switch back to original Horizon mode.
OUT Not changed, will be the same value as IN.
>B350 IN Not used for input to the sub-program >B0.
OUT >0000 if no errors, or >FFFF if where errors detected.

Accessing >B0 with >B34C containing >FFFF will tell all the Horizon cards in your computer system to switch back to original Horizon mode. This call, since it will remove any page at >6000, can't be accessed from the >6000 space. If for some reason a RAMBO program doesn't switch all the cards back to original Horizon mode, the RDS will do a complete reset any time the II Title Screen appears. The previous assembly examples should be enough to figure how to use this mode of the sub-program >B0.

USING THE AUTO-START MENU PROGRAM

The MENU program is a utility which we think you will find useful. To use it, copy it over to the first ramdisk on your Horizon card using a disk manager program. MENU is designed to be loaded via a custom call or thru the auto-start feature. To be able to CALL MENU, you must first use the CFG program to config one of the custom calls to 'MENU' with a length of '4'. If you want to MENU appear on power-up, the first custom call must be the one that is configed to 'MENU' and auto-start must be enabled. Upon calling MENU or upon power-up the following screen will be displayed:

```

#####;
: 1 Show directory :
: 2 Display a file :
: 3 Run a program :
: 4 user option 1 :
: 5 user option 2 :
: 6 user option 3 :
: 7 user option 4 :
: 8 user option 5 :
: 9 user option 6 :
: :
: C (cartridge name) :
#####<
```

If you have a CorComp, MBP, P-GRAM or John Clulow's clock attached, the time is displayed in the upper right corner of your screen. The 'C' option will only be displayed if you have a cartridge installed in your module port. The following quickly describes the active keys and their functions:

Key	Active keys on the main MENU screen:
0	Turns the screen off until another key is pressed.
1	Does a disk directory.
SHIFT 1	Does a disk directory and sends it to your printer.
2	Views any non-program file.
SHIFT 2	Views any non-program file and sends it to your printer.
3	Runs any Assembly file or Extended Basic PROGRAM file.
4-9	Three screens of user options, which are toggled thru with SPACE.
C	Runs the cartridge name shown next to 'C'.
D	Deletes a file from a diskette.
G	Cycles through all the GROM names; each name appears next to 'C'.
I	Toggles the I/O color interrupt routine on and off.
R	Calls a routine - does a CALL.
P	Changes the print device name for SHIFT 1 and 2.
S	Displays the ROM cartridge header at 16000 (if any), next to 'C'.
V	Displays the version number, date, and author's initials.
X	Same as 3, but only runs Extended Basic INTERNAL/VARIABLE 254 files.
SPACE	Switches display to the next screen of user options.
FCIN 5	Enters edit mode for the 'user options'.
FCIN 9	Goes to the next power-up program or where you originally came from.
FCIN =	Resets the computer system.

USING THE AUTO-START MENU PROGRAM

<u>Key</u>	<u>Active keys during viewing of a file:</u>
SPACE	Displays the next 22 lines of the file being viewed.
FCTN 8	Continues viewing the file until the end, or until a SPACE is pressed.
FCTN 9	Abort the file viewing. Return to the main MENU screen.

<u>Key</u>	<u>Active keys during viewing of a disk directory:</u>
1-9,A-Z	Does a disk directory of the drive selected.
ARROWS	Use the arrow keys E, S, D, X to move around the disk directory.
ENTER	Marks the highlighted file as the one to view, delete, or run.
FCTN 8	Brings you back to disk drive selection prompt.
FCTN 9	Returns you to the main MENU screen.

4.1 SHOW DIRECTORY FUNCTION

By pressing '1', the screen is cleared, and you are then prompted for a disk drive number or letter. Press 1-9 or A-Z for a disk directory, or FCTN 9 (BACK) to return to the main MENU screen. If you would like the disk directory to also go to your printer, instead of just the screen, then press SHIFT 1 instead of '1' from the main MENU screen. A reverse video 'P' will appear at the top-right hand corner of your screen to remind you the disk directory is also going to your printer in full eighty column format. Make sure your printer is on-line before selection of the drive number or letter.

Since we display two columns of filenames per screen, we use a small 'd' to indicate a DISPLAY/VARIABLE file type, and a large 'D' to indicate a DISPLAY/FIXED file type. The same is true for INTERNAL file types with either a small 'i' for INTERNAL/VARIABLE or a large 'I' for INTERNAL/FIXED. If there is more than one page of files, you are informed, and by pressing 'S', you can stop the directory and then use the arrow keys to highlight a file, or, by pressing the spacebar, the system will continue the disk directory.

Once a disk directory is displayed on the screen, you can then highlight a filename by using the arrow keys (E,S,D,X). Pressing FCTN 8 (REDD) will bring you back to the drive number/letter selection prompt, or pressing FCTN 9 (BACK) will bring you back to main MENU screen. Once you have highlighted the filename you want, pressing ENTER will cause that filename to be moved into the filename prompt for the View, Delete, and Run options, which is discussed later on.

4.2 DISPLAY A FILE FUNCTION

By pressing '2', you are then prompted for the filename of a non-program file. Type the filename you want to view, or mark the file using the 'Show Directory' function. While you are typing in the filename, all lowercase entries will be changed to uppercase. If you would like the file to be also sent to your printer instead of just the screen, then press SHIFT 2 instead of '2' from the main MENU screen. A reverse video 'P' will appear at the lower-right hand corner of your screen to remind you the file you are going to view will also go to your printer. Make sure your printer is on-line before pressing ENTER to start the viewing process.

USING THE AUTO-START MENU PROGRAM

After selecting the file to be viewed, and pressing ENTER, the screen will clear and the first 22 lines of the file will be displayed. To display another screen of 22 lines, just press the spacebar. If you would like to rapidly scroll through the file, press FCTN B (REDO); viewing will then continue until the end of the file is reached or until you press the spacebar. FCTN B (REDO) is most useful if you are sending the file to your printer. By merely pressing FCTN B (REDO), the complete file will be outputted to your printer without the need of manually paging thru it.

NOTE: The printer device is always opened in APPEND mode with a file type of DISPLAY/VARIABLE x, where 'x' equals the length of the fixed or variable record length of the file being viewed.

4.3 RUN A PROGRAM FUNCTION

By typing '3', you are then prompted for a filename of a Assembly program file, or an Extended Basic PROGRAM file. Type the filename you want to run, or mark the file using the 'Show Directory' function. While you are typing in the filename, all lowercase entries will be changed to uppercase. If you try to run an Extended Basic PROGRAM, and you don't have the Extended Basic cartridge plugged into the module port, the request will be ignored and you will be informed 'no ex basic', and returned to the main MENU screen. For running those large Extended Basic INTERNAL/VARIABLE 254 files, press 'X' instead of '3'. This option works just like the option '3', but is designed only to run the Extended Basic INTERNAL/VARIABLE 254 files.

4.4 THE CARTRIDGE FUNCTION

By pressing 'C', the cartridge whose name is next to the 'C' option will be selected. For those cartridges that have more than one entry name, by pressing 'B' you can cycle thru the entry names which will be displayed one at a time after the option 'C' on the main MENU screen. For those third-party ROM only cartridges, or super-cart programs, you can display the ROM entry name after option 'C' on the main MENU screen by pressing 'S'.

4.5 MISC. COMMANDS/FUNCTIONS

By pressing '0', the screen will go blank, pressing any other key afterwards will turn the screen display back on. After 10 minutes of keyboard inactivity, the built-in TI screen blanker will turn the screen off, but this option allows you to turn it off anytime.

By pressing 'D', you are then prompted for a filename. Type in the filename you want to delete, or mark the file using the 'Show Directory' function. While you are typing in the filename, all lowercase entries will be changed to uppercase. Pressing ENTER will cause the system to delete the file on the device you specified, but only if both the file and the device are unprotected. Pressing FCTN 9 (BAC) will abort the delete operation.

USING THE AUTO-START MENU PROGRAM

By pressing 'I', you can toggle on and off the action of the default colors as selected in the UFS program to be carried over to Extended Basic upon loading of an Extended Basic program or upon selection of the Extended Basic cartridge. This carry over program sets up an interrupt that changes the text and screen colors of Extended Basic to those of the MENU program, and keeps them set, even if Extended Basic tries to reset the colors back to the normal Extended Basic BLACK on CYAN. Even if this option is turned off, you can still carry over the MENU colors by pressing and holding either SHIFT key down immediately after pressing your option to run an Extended Basic program or after selecting the Extended Basic cartridge.

By pressing 'K', you are then prompted to type in a call name and any parameters. You can use this option to access the built-in calls and the custom calls in the ROS, just like doing a CALL in BASIC.

By pressing 'P', you are then prompted to type in the device name for your printer. While you are typing in the device name, all lowercase entries will be changed to uppercase. The printer device is always opened in APPEND mode, this means you could assign a disk file as your printer and append files or directories to it.

By pressing 'V', the MENU program will display, at the upper-left hand corner of the screen, the version and date of the MENU program and the author's initials.

By pressing 'FCTN 9' (BACK) from the main MENU screen, you will be returned to the calling environment, (the BASIC) you came from, or, if the MENU program was loaded via auto-start, then FCTN 9 will bring up the next auto-start program on the next lower CRU addressed card in your expansion box. For example, if you have two Horizon cards in your expansion box, and both have auto-start enabled, the highest CRU addressed one will be the first to appear on power-up, and upon pressing FCTN 9 the next lower CRU addressed one will appear. In the computer systems which have a CorComp floppy controller, CorComp's menu screen will always be the last (lowest CRU addressed) screen to appear, even though CorComp's CRU address is)1100, which is higher than a Horizon can be at)1000. If there are no more auto-start programs, then FCTN 9 will bring you back to the TI Title Screen.

4.6 THE USER OPTIONS

The MENU program contains 24 user configurable options. These options can be configured to reflect any titles that you want to see on your MENU. Using these options, you can, with one keypress load and run your most used utilities and programs. These options can even be configed to do calls.

4.6.1 Using the different user options

The 24 user options are divided between three screens of menus, the first six are actived with the keys 4 thru 9 from the main MENU screen, the rest are divided equally on two screens of nine options each and are actived with the keys 1 thru 9. You can toggle thru the three screens, by using the spacebar. Each time the spacebar is pressed, the system will switch to the next menu screen, and when it's at the end of screens, the system will loop back around to the main MENU screen.

USING THE AUTO-START MENU PROGRAM

4.6.2 Changing the user options

To change the default filenames and options of the MENU program, press FCTN 5 (BEGIN) after selecting one of the three screens by using the spacebar. After pressing FCTN 5 (BEGIN), the filenames will be printed under the user options on that screen, and the cursor will be awaiting your input at the first user option title. At the title prompts, both uppercase and lowercase keys are accepted, whereas the actual filename line will only accept uppercase entries, so feel free to leave the alpha lock up while editing.

You have room for sixteen characters in the title line, and twenty-six characters in the filename line. The large filename area is provided for hard drive users. Any text you would like to see can be entered in the title line, but filenames require a certain format to work properly. The MENU program can run files from either the Horizon card it was loaded from, or any other device. Also any user option can be configed for a CALL. The following examples show the many ways a user option can be used:

To run a program from the raddisk, which loaded the MENU program, enter your filename in the following format:

DSK#.filename

The '#' character instead of the normal drive number or letter, tells the MENU program to use the same drive that MENU was loaded from when loading the program you specified.

Let's say that you have a diskette that you keep a bunch of secondary utilities on, and you keep that diskette in drive 3 to load programs from it; you would enter the filename like this:

DSK3.filename or DSK.diskname.filename

Easy enough, now for those users with hard drives, they can try things like this:

WDS1.dir1.dir2.filename or WDS1.dir1.dir2.dir3.name

In the above examples, the user option could load either a assembly program, or an Extended Basic PROGRAM file, just like using option '3' (Run a Program); but to load and run those large Extended Basic INTERNAL/VARIABLE 254 files, you must add a 'X' and a space before the filename like so:

X DSK#.filename or X DSK3.filename

X DSK.diskname.filename or X WDS1.dirname1.dirname2.filename

And now with the 'X' and a space added before the filename, when you select that option, it will automatically go into the Extended Basic load routine just like using option 'X' from the main menu. Any filename can be used as long as it is preceded by 'X' and a space.

USING THE AUTO-START MENU PROGRAM

You can even setup your user options to call one of the custom calls, or change a drive number, etc., by adding a 'C' and a space in front of the call name, like so:

C PTERM or C DN.9.1

This option works like the 'K' option from the main menu, and allows you to have any custom or built-in call as a user option on the MENU. Any valid call name can be used, as long as it is preceded by 'C' and a space. For more information on the different calls you can access, see section 3.0 of this manual.

Well, that should be enough examples to get you going. After you have finished changing the user options to your liking, read the next part on how to save your changes.

4.6.3 Saving the changed MENU program

After you have finished changing the user options to your liking, press FCTN 9 (BACK) to exit edit mode, and bring up the save or abort prompt. Pressing FCTN 9 (BACK) again will not save the changes to the ramdisk, but they will still be there in memory until you exit the MENU program or run another program. Pressing any other key will cause the system to save the MENU program back to the ramdisk it was loaded from, overwriting the existing MENU file.

Everything is saved, including all three user option screens, the filename prompts for View, Delete and Run, as well as your printer name. Since the filename prompts are saved, if you have a file that you are constantly viewing, deleting or running, place its name in the view, delete or run prompt prior to saving and that will be the new default for that option. Also the status of the 'I' option (Extended Basic colors on/off) is saved.

NOTES ON DIFFERENT THINGS

5.1 BASIC DEMO OF THE NEW OPCODE #8

This is a simple Extended Basic program with Assembly support that demos the new opcode 8 (SCRATCH RECORD). Before running the Basic program called 'OPBDENO', setup your Horizon card with two ramdisk drives numbered 6 and 7. Then copy the Assembly object file 'OPBDENO-O' to drive #7, and erase all files from drive #6. Now load and run the Basic program file 'OPBDENO'. The program will first ask for the number of records. This value controls the number of fixed length records the program will write to a dummy file called 'TEST' on drive #6. After inputting a value, the program will then make up the dummy file on drive #6. After a short wait the program will now ask you for the record number to be scratched. Type in any number from 0 to the max. number of records in the dummy file. After pressing ENTER the system will then remove (scratch) the record number you just specified from the dummy file. After it is finished removing the record, the program will then ask if you want to scratch another record. Answering 'Y' will bring the program back to asking for a record number to be scratched, whereas answering 'N' will cause the program to list out all the dummy records until it reaches the end of the file, and then it will delete the dummy file 'TEST' from drive #6. This simple demo program is useful in seeing how fast the system can remove a record from a very large data file. For more information on the scratch record opcode, see section 3.4.8 of this manual and also print out the included Assembly source file called 'OPBDENO-S'.

5.2 DEMO OF THE USER DSR FUNCTIONS

If you have bought RAMBO, then included in this package is a sample USER DSR file called 'USRDSR'. This file when loaded via the CFB program will perform three simple demos. The first demo will appear on power-up even if the auto-start function is disabled on your Horizon card. This power-up demo will cause the Speech Synthesizer (if attached) to say "THE TEXAS INSTRUMENTS HOME COMPUTER IS READY TO START" each time you turn on or reset your computer system. The second part demos a USER DSR NAME. This name is preset to 'TEST' in the sample file, and when called (similar to the custom calls), causes the Speech Synthesizer (if attached) to say "THE TIME DEVICE IS NOW ON". The CALL TEST command also turns on the third part of the demo, which is a USER INTERRUPT. This interrupt will occur about once a minute while you are using your computer, and cause the computer's sound system to play a short high-speed chime sound. Calling 'TEST' again will cause the Speech Synthesizer (if attached) to say "THE TIME DEVICE IS NOW OFF", and of course the interrupt will be turned off. Even the best programmers may find this demo a little hard to understand, but to help you out, the Assembly source file 'USRDSR-S' is included in this package. For more information on USER DSRs see section 2.3.U of this manual.

5.3 SLIDE-SHOW DEMO USING THE RAMBO PROGRAM SPACE

If you have bought RAMBO, then included in this package is a little program called 'SHOW'. This file when loaded via the MENU program or other E/A #5 type loader will allow you to load a series of TI-ARTIST pictures in the RAMBO program space, and then display them at different speeds. This program is fully menu driven and very easy to operate. In short, after loading the program press the number key corresponding to the number of the disk drive containing a disk full of TI-ARTIST picture files. Then press 'C' to start the loading of the diskette into the RAMBO program space. On the top

NOTES ON DIFFERENT THINGS

of the screen you will notice the amount of total memory, used memory, and free memory is displayed. The bottom of the screen will show the current drive and filename being loaded. If any error occurred during loading of the diskette or if there was no free RAMBO program space, then the program will exit to the TI Title Screen. After loading is completed, press 'R' to run the slide-show. The SHOW program's main menu lists the active keys during the slide-show, for delay, pause, and exit. Also included in this package for programmers is the complete fully commented Assembly source code file for 'SHOW'. This file should be most helpful for those programmers wishing to start learning how to access and use the RAMBO program space.

5.4 WHERE AND HOW TO MARKET YOUR PROGRAMS FOR RAMBO

This section of the manual is designed to give you some information on where to market the programs that you have written for use with RAMBO/HORIZON cards. In short, there are at least three ways to get your program out there in the TI world.

The first method is PUBLIC DOMAIN; this means you give out copies of the program to the public and you don't keep any rights to the program, anyone can modify it, add to it, etc. This method is best for little programs or utilities that you would like to see other TI'ers using, but without the want of any pay for your work, and without the responsibility of updating the program. It is very easy to get a PUBLIC DOMAIN program out there, you can upload it to TI BBS's, give copies of it to friends, ask your local user group to add it to their library.

The most common method is SHARE-WARE, FAIRWARE, or TRI-WARE. There are many different names for this, but in short it is much like Public Domain in the way it is marketed, except that you keep all the rights to your work, and you expect users of your programs will send you something in payment for your work, which is normally \$10 to \$20 in the TI world. This method is best for small to medium programs which are simple enough or user-friendly enough that printed manuals are not needed, and the necessary manuals can be included with the program on diskette.

The last method is COMMERCIAL. This means you sell your program for a fixed price, and you normally must include things like nice printed manuals, warranty/registration cards, etc. This is the hardest method to market yourself and requires the most work, plus buyers of your program expect that you will fix any errors in programs you sold them, and be updating the programs with new features over the years. On the other hand, it is the best method to use if you want the most money you can get for your programs, and your programs on the medium to large side. Marketing firms like ASGARD and JP SOFTWARE, they take most of the workload needed to successfully market and sell your program to the TI world; you just sit back and wait for the monthly royalty check.

For more information on these methods here is a list of people you can contact:

919 USERS GROUP
109-2356 Gerrard St. E.
Toronto, Ontario Canada M4E-2E2

NOTES ON DIFFERENT THINGS

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH. 43213

JP SOFTWARE
2390 El Camino Real, #107
Palo Alto, CA. 94306

ASGARD SOFTWARE
P.O. BOX 10306
Rockville, MD. 20850

5.5 WHERE THE SUPPORT BBS'S ARE FOR RAMBO/HORIZON OWNERS

There are currently two BBS's operated by OPA Oasis Pensive Abacutors and Bud Mills Services, which are support boards for RAMBO and Horizon owners. Their 24 hour phone numbers are as follows:

OPA Oasis Pensive Abacutors (TI TOWER BBS)	416-921-2731	300/1200/2400bd BNI
Bud Mills Services	419-385-7484	300/1200bd BNI

5.6 WHERE AND FROM WHAT TO GET MORE INFORMATION

There is a number of good books and/or manuals that should be read, if you want to understand more about DSRs, CALLs, OPCODEs, and SUB-PROGRAMs. These books and manuals are recommended reading for those really serious about writing programs for RAMBO and Horizon cards; programs using the full power the SERIES 8 software can offer. The following lists a few titles and address of where to get these books and manuals:

EDITOR/ASSEMBLER software package and manuals by Texas Instruments:

This package is still available from many TI dealers, and most user groups can point you in the right direction for the best and closest dealer in your area.

TECHNICAL DRIVE by Monty Schmidt. Published in North America by RYTE Data:

This book may still be available from RYTE Data. If you get no response from the below address, write to OPA Oasis Pensive Abacutors, and we will see what we can do in finding you a copy of this wonderful book.

RYTE DATA
210 Mountain St.
Haliburton, Ontario Canada K0M-1S0

NOTES ON DIFFERENT THINGS

GRAPHICS PROGRAMMING LANGUAGE USER'S GUIDE by Texas Instruments:

This guide is mainly for GPL programming use, but the section 5.0 on the disk peripheral is very good programming guide no matter what language you are using. If you can't find a copy anyplace, write to OPA Oasis Pensive Abacutors and we will mail you an original print of the complete guide.

TI-99/4A CONSOLE AND PERIPHERAL EXPANSION SYSTEM TECHNICAL DATA by Texas Instruments:

Another great data book from Texas Instruments, if you can't find a copy anyplace, write to OPA Oasis Pensive Abacutors and we will mail you an original print of the complete data.

INTERFACE STANDARD & DESIGN GUIDE for TI99/4A PERIPHERALS by Tony Lewis, PE:

This a fairly new book in the TI world, and is mainly towards the hardware use of the different peripherals, but it has a section on the Horizon system and contains some different programming examples. This book should be useful for those programmers wanting to write USER DSR files. You can order this book from the following address:

TONY LEWIS
409 Drolmond Dr.
Raleigh, NC. 27615

PROGRAMMING FOR THE FUTURE by OPA Oasis Pensive Abacutors:

This book currently is still being written, but will be a complete programming guide for the TI world, covering all known cards including a large section on the RAMBO/HORIZON and the USER DSR function. This book will cover all the tips and tricks for programming the 9900 series of processors, and will included a library of useful sub-programs and utilities that you can use in your own programs. Write to following address for more information on this wonderful new book:

OPA DASIS PENSIVE ABACUTORS
432 Jarvis Street Suite 502
Toronto, Ontario Canada M4Y-2H3

CREDITS FOR THIS PACKAGE

Lots of thanks to Texas Instruments, for supporting my company with many needed technical manuals and books on the TI and the many support chips and cards.

Thanks to NYARC INC. for finally writing a technical manual on their hard drive/floppy controller card that described their format of having CPU data transfer instead of VDP data transfer.

Thanks to Monty Schmidt and Tony Lewis for their all the useful information in their wonderful books.

Many thanks to the RAMBO owners who reported back with the different problems they were having. Without their help some of the major errors in the new software may have never been found.

Jim Gallantyne and Charles Eari helped in writing the three new opcodes A, B and C for loading ASSEMBLY, BASIC, and CARTRIDGE programs.

Cecil Chin and all the rest of the staff at OPA for helping in ideas, beta-testing, and those many long nights before the Washington D.C. '89 show.

Bud Mills and Ron Walters for the original work on having 8K paged Horizon memory that started the RAMBO project at OPA.

John Guion and Robert Jones for their knowledge of the TI module hardware which allows RAMBO to override any memory at 16000 so it can be able to page in the HORIZON RAM at 16000 without the need of removing the module.

Many thanks to John Johnston, Michael Ballmann and Bud Mills for the original 7.3 Horizon software, and their help in getting the first version of the SERIES B software finished. Also many thanks to John Birdwell for his help in fixing some bugs in the original ROS.

Jim Schroeder did an outstanding job of adding full arrow key functions to the Show Directory routine in the MENU program. The directory function was originally adapted from a directory program written by Clint Pulley.

The program loader in the MENU program was adapted from a program written by Todd Kaplan. Paolo Bagnaresi further modified the program loader to allow it to load programs into low memory. I can't find a program it won't load! Paolo also modified the Extended Basic interrupt loader and now it works flawlessly too. Much thanks to him!

To all the Horizon users out there who offered many suggestions or improvements - a big thank you! Without your help, there wouldn't have been a new version.

And finally of course to the great designers of the first Horizon card, for if this card have not been designed, I am sure the TI world would had been smaller today! Many thanks to Ron Gries and his '83 team!

==== THE END! ====