HORIZON RAMDISK

User Operating Manual

3 May 2020

V8.32/8.38/8.42c Updates 2020 T. Tesch

Horizon Ramdisk wiki and source:

https://github.com/horizonramdisk/Horizon-Ramdisk-ti994a/wiki

```
:###############################:
::#############################::
:::###########################:::
::::#########################::::
::::                         ::::
::::      RAMBO/HORIZON      ::::
::::                         ::::
::::  USER OPERATING MANUAL  ::::
::::                         ::::
::::  FOR SOFTWARE SERIES 8  ::::
::::                         ::::
::::#########################::::
:::###########################:::
::#############################::
:###############################:
```

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

Horizon 4000B hardware, layout, and design ownership
Transferred from Bud Mills Services
to J. Fetzner (Tekumel Software) 2018

The ROS 8.14 manual was expanded and originally adapted from different manuals by John Johnston and Bud Mills Services. The version 8.14 manual was copyrighted by Bud Mills Services and by OPA Oasis Pensive Abacutors.

Version 8.42 and all subsequent revisions of this manual as well as all software, publications, and material associated with the HORIZON ramdisk and Ramdisk Operating system may be reproduced solely for non-commercial usage.

Selling or otherwise charging for the reproductions and/or distribution of this package, in part or in totality, and regardless of distribution method, is prohibited. This manual and all associated files and documentation are not salable items.

T. Tesch, J. Fetzner, and D. Warren reserve the right to make changes at any time to improve the overall design and supply the best possible product for the community.

TABLE OF CONTENTS

TABLE OF CONTENTS

If you have any questions about this package or have any problems in using the software, please feel free to contact the following individuals via the TI forum at **AtariAge.com**

T. Tesch	-	InsaneMultitasker
J. Fetzner	-	Ksarul
D. Warren	-	FALCOR4

1.0 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 CFG, ROS, and MENU programs. The CFG program with this package can configure any Horizon card built using 8K*8, 32K*8, 128K*8, or 512K*8 STATIC RAM chips, and also any HORIZON card upgraded with the RAMBO hardware.

**** NOTE! ****

ROS and CFG Version 8.42 should NOT be mixed and matched with older versions of ROS and CFG. While the ROS itself is compatible with previous SERIES 8 software, the drive allocation calculations may corrupt drives if ROS versions are co-mingled. SERIES 8 software versions are still cross-compatible within their respective packages. MENU 7.38/7.39 is still compatible with current ROS and CFG versions.

The SERIES 8 ROS perating system software requires the CFG, ROS, and MENU programs in this package to be used together. Absolutely no mix and match compatibilities with previous 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, CFG 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 ROS operating system.

****** CAUTION ******

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 configured using any lower versions of operating systems like 7.3*. Furthermore, the same clearing of data may occur if updating from older 8.x versions to the most recent, depending on the current drive configuration.

It is a best practice to keep a ‘pristine’ copy of your files and utilities. Since ramdisks are volatile by nature, always load from disk/known good versions when possible. Periodic backup of any files you create on the ramdisk may save you heartache later.

8-bit and 16-bit TERMINOLOGY:

Software and hardware documentation over the years references both 8-bit and 16-bit ramdisks.

The first ramdisks used an 8-bit (1 byte) CRU address space, limiting the total addressable memory. Thus older cards were often referred to as ‘8-bit ramdisks’. Later versions expanded the addressing space to 16-bits, allowing for much more capacity and addressable memory. These cards are often referred to as ’16-bit ramdisks’.

Horizon Ramdisk Wiki and Git repository

A “simple” set of the instructions and the source code for ROS and CFG is found in the Github repository referenced on the cover page.

The following enhancements have been introduced to the 8.14F (1990) software:

ROS 8.42:

- Reconciled source code to 8.14F release
- Unified ROS for all controllers
- Embedded ROS CRC to protect against file transmission/copy errors
- Increased the maximum disk size from 1600 sectors to 3200 sectors, supporting 8MB maximum
- Modified INPUT (>14) and OUTPUT (>15) and Catalog routines to mask "undefined" MYARC bits (e.g., Bit Remover).
- Removed VDP interference caused by drive number tracking routines
- Removed all "I Gotcha" code
- Enhanced ROS powerup to ensure proper page startup for UberGROM cartridges
- File allocation bug-fixes to inhibit file corruption:
  - if the file exceeds current available disk space
  - When the file exceeds the cluster allocation limit i.e., severely fractured files
  - Cluster and disk over-run into the next formatted disk under certain conditions
- Forced assembled size to 8192 bytes to inhibit TI Floppy Controller LOAD bug
- Fixed fractured file Deletion routine
- Fixed SAVE and OUTPUT (>15) opcode allocation issues
- Removed old OPA/BMS title screen code and moved CALL-related code into 2k space; necessary step to correct allocation bugs
- Added edge-case tests to allocation and bitmap cluster routines. Reduced some occurrences of LONG waits when cataloging an undefined drive number. (Level 2 IO error return operates differently from level 3, causing some of these issues)
- An offset pointer was added to the 2K section of ROS so that MENU and other programs can determine the location from which to load their characters. MENU739C has been fixed; other MENU programs may require modification.

CFG 8.42

- Reconciled source code to 8.14F release
- Warnings and confirmations prior to executing the Format and Delete actions
- Embedded CRC check and validation
- Enhanced device detection to include additional devices such as the TIPI peripheral and Classic99 emulation.
- Save and restore user-defined CALLS, colors, and powerup on/off flag during the ROS load process.

MENU 7.39c

- Updated character definition routine due to changes in ROS.

1.2 OVERVIEW OF THE CFG PROGRAM

The CFG program is designed to allow you to load the ROS into your Horizon cards and configure each card to your liking. The CFG 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 Horizons 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 CFG program allows you to select the drive numbers, disk names, and size of each of your ramdisks. It is also possible to select the default colors and enable/disable the auto-start program, and to configure the custom subprogram CALLs accessible from BASIC and EXTENDED BASIC.

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 CFG 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.

1.4 OVERVIEW OF THE AUTO-START MENU PROGRAM

One of the main things users liked about the Horizon software was John Johnston’s MENU program.

With version 8.14F, 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 through Z; but more on that later. For those not familiar with JJ’s MENU program, it is what is called an “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.

The 80-column “9640 Menu” program and other versions of the MENU may be used with series 8 ROS.

Please note: If an incorrect version of MENU is called or if MENU is loaded from an EA option 5 loader, the menu itself may DISABLE the ramdisk autostart feature. If this happens, replace the offending files and/or re-enable the autostart option. When this happens the first time do not panic as your data should still be intact.

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 DSRs which don’t have to have 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.

1.6 GENEVE CONSIDERATIONS

The Geneve 9640 computer contains its own master DSR. It does not utilize ROS for any file IO. Therefore, ROS and CFG may be used only if you intend to use the Geneve in TI mode and activate the ROMPAGE option. You will not be able to access CFG-formatted ramdisk from native MDOS.

To use a Horizon RAMdisk with the Geneve, please use FORM v1.23 or FORM3MEG and MDOS versions 2.21 or higher. Loading the OS from ramdisk is also possible with these utilities. See MDOS release updates for more information regarding CRU configuration and REMAP requirements.

WARNING: Geneve OS versions below 6.70RC2 do not always properly manage cluster allocations and fragmented files in drives larger than 800K. File corruption may eventually occur. Release of OS version 7.00 is pending identification of a hard drive corruption issue.

1.7 DISTRIBUTION FILES

The following files are typically distributed for use with the Horizon ramdisk. Some files in this distribution have been renamed for compatibility with non-TI file systems.

Primary files

README	TI File readme document and distribution permission
CFG1	The “CFG” Configuration Program (file 1 of 2)
CFG2	The “CFG” Configuration Program (file 2 of 2)
ROS842C	Ramdisk Operating System (ROS); this file may ONLY be loaded via the CFG program
MENU	V7.39c, auto-start menu adjusted for use with ROS 8.42 and higher
ROSCRC	*CRC installer; not publicly distributed*

Supporting Files:

CHARA1	Alternate FONT that may be loaded via CFG
LOAD	Extended BASIC loader for CFG and memory test programs
MGR1	DM1000 (Note: included only as a starter program; use DM2K or DSKU)
MGR2	" "

Demonstrations

OP8DEMO	Scratch Record Demonstration
OP8DEMO-S	Demo source
SHOW	a RAMBO slideshow demo program
SHOW-S	demo source for SHOW
USERDSR	a RAMBO User DSR Demo
USERDSR-S	demo source for USERDSR

TEST Programs

MEMTEST	0-256K memory test (8-bit ramdisk)
TEST-O	Support routines for MEMTEST
MEGTEST	256K+ memory test, requires XB (16-bit ramdisk)
TEST-O32	Support routines for MEGTEST
TST	E/A option 5 TEST program for all horizons. Not associated with MEMTEST/MEGTEST

GENEVE-only ramdisk formatters (see Geneve OS Distribution for these files)

FORM123	Geneve ramdisk formatter, single 800K drive per CRU
FORM3MEG	Geneve ramdisk formatter, single 3.2MB drive per CRU

2.0 USING THE CFG Configuration PROGRAM

2.1 LOADING CFG (THE CONFIGURATION PROGRAM)

Extended BASIC:
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 “CFG1” from “DSK1.” If the disk is in DSK1 when XB is selected, the program will auto-load and give you a choice to run CFG.

Editor Assembler:
The CFG program is a PROGRAM image 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.CFG1”.

2.2 PERIPHERAL DETECTION 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 CONFIG  v8.42c  03.14.2020 :
:                                          :
: CARD   PERIPHERAL    : TOTAL  RAM-  RAMBO:
: ADDR. CARD DETECTED  : SIZE   DISK  RAM  :
: ----- -------------- - ----- ----- ----- :
: >1000 MYARC Ramdisk                      :
: >1100 Floppy Cntrlr                      :
: >1200 32k - 512k HRD * 8192K 3192K 2048K :
: >1300 RS232 and PIO                      :
: >1400  8K*8 HORIZON  *  192K  186K    0K :
: >1600 TIPI Device                        :
: >1700 PGRAM CARD                         :
: >1900 MYARC Ramdisk                      :
: >1B00 Axiom Printer                      :
: >1D00 Clock Card                         :
: ----------------------------------------:
: Hit any key for CONFIG or FCTN = to QUIT :
:###########################################:
```

The CFG program scans the full range of CRU addresses from >1000 to >1F00 and then displays a list of the cards found and at what CRU address they were enabled. The CFG then tries 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 with some exceptions e.g., CFG will detect the SAMS memory card and Geneve 9640. The P-GRAM card and IDE card will only be listed if their respective DSR software has already been loaded into the card.

Any Horizons found will be identified by their general memory classification. Horizons with RAMBO installed correctly will be listed as Horizon/RAMBO no matter the physical RAM chips installed on the card.

In all cases all of the RAM on-board will be tested and the size displayed under "TOTAL SIZE".

If the ROS is loaded on a Horizon card, the software will display a "*" between the card's definition and the "TOTAL SIZE".

The amount of RAM on-board used as ramdisk space will be displayed after "TOTAL SIZE", and if the card is a RAMBO/HORIZON with a SERIES 8 ROS loaded, the amount of RAM not used for the ramdisk part will be displayed under "RAMBO RAM"; this is the amount of space free for use by USER DSRs and programs which access the RAMBO memory space.

Press any key to bring up the first configuration screen of your first Horizon card (lowest CRU address), or press FCTN = (QUIT) to exit the CFG program.

If one or more Horizon cards are not displaying as expected, or their information is incorrect, then make sure all your Horizons are plugged in and the on/off switch is in the ON position (some later Horizon models have a disable/enable switch or jumper pack). Also make sure none of your Horizons are set to the same CRU address as another card in the expansion box.

2.3 CONFIGURATION SCREEN FOR A HORIZON CARD

After pressing any key, the screen will display the configuration for the first Horizon card (with the lowest CRU address) in your expansion box. The screen is divided in three parts:

TOP - The top section displays the CURRENTLY selected card.

MIDDLE - Displayed only if a valid ROS is loaded, and contains information like the custom call names, foreground and background color settings, and auto-start enable/disable. ROS Version is displayed when loaded.

BOTTOM - A list of options you can select with the keypress matching the first letter of the option name.

```
:##########################################:
: CARD  PERIPHERAL      : TOTAL  RAM-  RAMBO:
: ADDR. CARD DETECTED   : SIZE   DISK  RAM  :
: ----- -------------- - ----- ----- ----- :
: >1200 Horizon/RAMBO   * 8192K 3192K 2048K :
:                                          :
:                                          :
: Len/Name        Len/Name        Len/Name  :
: --------        --------        --------  :
:  4 MENU          2 U4            2 U7     :
:  3 CFG           2 U5            2 U8     :
:  2 U3            2 U6            2 U9     :
:                                          :
:  Screen Color: 5   Total Sectors: 32760  :
:                                          :
:    Text Color: 16     Powerup On? Y      :
:                                          :
:  User DSR:                               :
:                                          :
:  Version:  ROS842c    8.42c   02/23/2020 :
:                                          :
:  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 configuration and ROS to a device, change the built-in character font, and change the current configuration of your ROS and ramdisks. If your Horizon card contains the RAMBO hardware will you be able to load in a USER DSR file.

2.3.Q Quitting the CFG 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 RAMBO 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 CRU address, total size, etc. If you have only one Horizon card in your expansion box, the "N" keypress will be ignored *unless the card is logically split into two CRU addresses.*

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 existing disk information?"

If you would like to preserve this information to prevent having to re-initialize all your disks and reloading all your files onto your Horizon card, answer the prompt with a "Y". CFG will also save (and restore) your defined CALLs, the colors, and the powerup on/off selection.

Selecting "N" will display a secondary confirmation prompt "Not Saving data – C to continue; F9=exit" to ensure no accidental loss of drive data.

Next you are prompted for a filename for the ROS file. If the file you loaded was a valid ROS then the system will show a loaded status. If the file was not loaded, you will get an 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.

Version 8.42 introduces a 16-bit CRC validation to protect against ROS file copy and transmission errors. As soon as CFG loads ROS (and before updating the DSR), a CRC value is calculated and compared to the embedded CRC value. If the two values are different, CFG will inform you of the error and will **not** allow the file to be loaded into the ramdisk. If the two values match, CFG will load ROS into the ramdisk's DSR.

Prior releases do not contain an embedded CRC; CFG will alert you to this condition. ROS files saved with CFG will do not contain an embedded CRC. (see next section).

2.3.S Saving the configured ROS

By pressing "S", you are then prompted for the filename and device to save the configured 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 an 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.

NOTE: *The SAVE option does not generate a new CRC value for the saved ROS. The embedded value is cleared, allowing the saved ROS to be re-loaded. This is intentional and intended to inhibit users from re-distributing unofficial/modified versions of ROS with a valid CRC.*

2.3.F Changing the built-in character font

The ROS contains a built-in character set (characters 32-126) which it uses when an auto-start program like the MENU is loaded, as well as anytime an 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 an 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, you can use a program like Character Sets and Graphics Design, by Dave Rose, to create your own CHARA1 file.

2.3.E Changing the configuration of a Horizon card

By pressing "E", you will then be able to edit the configuration of your custom calls, color settings, and enable/disable the auto-start program. The down and up arrow keys are active when in edit mode allowing you to easily move thru the fields until you get the configuration 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 (LEN):
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 "PTERM" with a length of 5; you would then have to type all five characters to be able to load the program called "PTERM" on your ramdisk; 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 "PTERM" on your ramdisk.

CUSTOM CALL NAME (Name):

This is the filename of the first file to start loading from the first ramdisk 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 maximum 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.

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 CFG text and screen colors will change too, allowing you to see the look of the colors you have just selected, plus the CFG 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 some MENU programs.

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 an "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 (Drive or Dsk)

Based on screen space limitations, CFG will show you "Dsk" when RAMBO is available or "Drive" when RAMBO is not available.

By pressing "D", you will be able to edit the configuration of the ramdisk "drives" on your Horizon card. *See sample screen below*. The down and up arrow keys are active when in edit mode allowing you to easily move through the fields until you get the configuration 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.

```
:##########################################:
:   #  Disk Name    W/P    Format   Size   :
: ---------------------------------------- :
:   5   DSK5-name  N                3200   :
:                                          :
:   6   Pictures   N                3200   :
:                                          :
:   7   Utilities  N                3200   :
:                                          :
: -----------------------------------------:
: Input new drive number.  1 to 9 & A to Z :
:##########################################:
```

DRIVE NUMBER/LETTER ("#"):

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 **strongly** recommend you don't have more than one ramdisk defined with the same drive number or letter. Drive numbers should not be duplicated on multiple physical ramdisk cards, nor should you duplicate a drive that is present on a floppy controller, TIPI device, or other similar cards.

DISK 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 disk name for each ramdisk on your Horizon card. Disk names can use any mixture of symbols, numbers, and letters. Unformatted ramdisks on your Horizon card will be listed on the screen "*" under the diskname field. It is recommended that you type in a diskname; some users like to use "DSKx" where X represents the drive number you selected.

WRITE PROTECTION ("W/P"):

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 an "N"; all other key presses are ignored.

FORMATTING A RAMDISK:

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

WARNING: Formatting a ramdisk drive causes all data to be lost on that ramdisk All ramdisk drives below the one just formatted *on the screen* will be made unformatted, causing their data to be lost!

Confirmation is required whenever the user selects a Format or Delete option.

SIZE OF A RAMDISK:

This field displays the size of a ramdisk in the total number of diskette sectors. If you selected "Y" to format a ramdisk, then the size will display the total number of sectors still not formatted for ramdisk space on your Horizon card, and the cursor will be awaiting your input of the new size of the ramdisk.

The maximum valid ramdisk drive size has been increased from 1600 sectors to 3200 sectors. The sector 0 drive parameters for drives >1600 sectors will reflect double-sided, double-density, 80 tracks and the sector 0 bitmap now represents two sectors per allocation unit. Some direct-sector access programs are not capable of manipulating drives >1600 sectors, so be certain your programs are compatible.

Valid range of the size is from 8 sectors to 3200 sectors in 8 sector increments. The size after input is adjusted to make sure it is within the nearest 8 sector increment. If the size is over 3200 sectors or the maximum number still unformatted, then the system will redisplay the amount still unformatted and await a new input of the size and/or adjust the value for you.

NOTE: If your Horizon card has more than 3200 sectors (800K) available for use, to be able to use all your memory you will have to have more than one ramdisk drive formatted on your card. For those Horizons with the RAMBO hardware installed, the amount left for RAMBO space is the amount of memory which has not been formatted into a ramdisk. To help in calculating RAM K into sectors: there are 4 disk sectors per each 1K of memory.

NOTE: The RAMBO page map is currently limited to 256 pages * 8K (2MB maximum). This is the maximum amount of memory CFG is able to allocate with this version of hardware and ROS, regardless of the maximum available free ram on the card.

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 CFG 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 an 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 CFG main screen showing in the “USER DSR:” line the DSR name (if any), and maybe the words POWER UP and/or INTERRUPT, depending if these 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 CFG 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.

3.0 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 interpreter 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 custom calls which can be configured through the CFG program to load your favorite programs with just a simple CALL. Below, you will find three different ways to access these calls:

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 set up one of the custom calls as MENU and the MENU program files are on the first ramdisk on 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 the name of the call you would like to run and press ENTER. The ROS will now load and run the program named JKL on your ramdisk, providing you have set up 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 are 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 configuration can be found in section 2.3.E of this manual.

3.2.1 What is an "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. 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 power on.

An example of an “auto-start” program is the MENU program supplied with this package. For more information on the MENU program, see section 4.0 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 an error message or nothing at all. In any case, you will be returned to the program you were 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 WO.B” This call would turn on the write protection of the ramdisk whose drive number is “B”. 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.AO Enable the auto-start program (AO.x)

The call “AO” 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 (AF.x)

The call “AF” allows you to disable the suto-start program. This is the same as typing an “N” in the “Powerup On?” field in the CFG program.

3.3.WO Turn on write-protection (WO.x)

The call “WO” 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.

3.3.WF Turn off write-protection (WF.x)

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

3.3.DN Changing a ramdisk drive number (DN.o.n)

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 the new drives specified have no OPEN files.

3.3.LD Load a program directly (LD.x.name)

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.c.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 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 and a new mode allowing data to be transferred directly to and from CPU memory without the need of going thru VDP RAM.

3.4.1 AUTOMATIC ATTRIBUTES (ramdisk only)

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.

WARNING: if you intend to write programs that will work on devices other than a ramdisk, specify the attributes for compatibility.

3.4.2 CPU RAM BUFFER

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 the LSBit). If the bit is set to 1, then the data buffer 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 cpu buffer addressing used by the Myarc floppy controller, Myarc hard and floppy controller, IDE controller, and HDX controller. 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.

Some floppy disk controllers e.g., TI and CorComp, use a VDP RAM buffer to keep track of the OPEN files and buffers. The number of files allowed OPEN at once is controlled thru the CALL Files(x) command whose default is 3 files.

The SERIES 8 ROS uses an improved internal buffer system which besides speeding up the operation of the Horizon card, allows you to have up to 16 files OPEN at once on each Horizon card in your system without the need to use a CALL FILES(x) command.

ROS 8.14F updated certain legacy VDP RAM pointers in the old VDP RAM buffer, used primarily for disk number tracking hacks not officially supported by the floppy disk controllers.

Starting with ROS 8.38, all VDP interference has been removed from ROS; VDP pointers are no longer manipulated. Boot tracking using these VDP pointers will no longer function.

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 removal of a record is managed within ROS, 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.

Note: SCRATCH RECORD opcode 8 is still available in ROS 8.42, however, it may be removed to improve the DSR at a future date.

3.4.9 The STATUS opcode

The regular disk controller system includes an 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.

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 the actual loading and execution of the BASIC program up to the EXTENDED BASIC module. This being so, the program which activated the 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”.

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 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 OPCODE >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 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 Monty Schmidt and published by Ryte Data. Other good resources include the MYARC Hard and Floppy controller manual, the IDE controller manual, and the TIPI Wiki.

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 >834C. If the bit is set to 1 then all data buffer addresses point to CPU RAM locations. Otherwise, if the bit is reset to 0, then all data buffer addresses point to VDP RAM locations.

This format is compatible with the type used by both the MYARC floppy controller card and their hard drive/floppy controller card. For example: if the location >834C contains a >B5 instead of the normal >05, then the ROS will use CPU RAM when reading or writing to drive #5 instead of using VDP RAM.

The standard low-level sub-programs use the CPU PAB byte at >834C to store the drive number the sub-programs will be using for its operation. The normal range of this byte is from >01 for drive #1 to >09 for drive #9, but now the SERIES 8 software allows the letters A thru Z to be used as drive numbers.

This being so, the byte at >834C 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). This will give you the correct value to use.

3.5.B0 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, OPA/BMS 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 is now possible to write large 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 destroying 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 transferred 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.

HORIZON/RAMBO CRU INTERFACE ACCESS:

The original Horizon interface has a CRU address range from >1000 thru >1700. The 8- or 16-bit CRU bits relate to the pages, banks, and SRAM/DSR selections. See the “Anatomy of the HRD4000B”, prior revision hardware construction guides, and previous ROS operating manual for more details.

The following information was retained from the 1990 version of the user operating manual. It has not been fully updated to reflect 512K*8 chip usage or more than one HCT chip for dual layer operation. Additional review and investigation is necessary to encompass both the HRD4000B and earlier hardware revisions.   Use as a loose reference only.

HORIZON MODE CRU INTERFACE:

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.
14	= DSR Space (only w/32K dsr chip in U9)
15	= Reset to 0: used to change to RAMBO mode, on original Horizons it was undefined.

RAMBO MODE CRU INTERFACE DIRECT ACCESS:

It is highly recommended that all RAMBO memory access is performed via the DSR routines.

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=ROS OFF, 1=ROS ON) Used by ROS and DSRLNK.
1 thru 2	= Not used when in RAMBO mode, recommend both be reset to 0.
3 thru 7	= 8K >6000 Page number 0 thru 31, (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.
14	= DSR Space (only w/32K dsr chip in U9)
15	= Set to 1: used to change to original Horizon mode.

RAMBO DSR ACCESS:
ACCESS MODE #1 FOR SUB-PROGRAM >B0 (Gets Max number of free 8K pages):

```
>834A  IN   Must be cleared before access to get true number of free pages.
       OUT  Max.  number of free 8K pages for all RAMBOs in the TI system.
>834C  IN   >6000 = gets the Max.  number of free 8K 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 errors detected.
```

Accessing >B0 with >834A and >834C cleared will, upon exit, return to the program in >834A the max. number of free 8K pages for all RAMBOs in 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 BLWPs.
GETMAX CLR  @>834A       Clear Max. number of pages.
       CLR  @>834C       Ask for Max. number of free 8K pages.
       LI   R0,VDPPAB    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.
       BLWP @VMBW        Write the PAB (>01B0) 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.
CALLBO DATA >01B0        PAB data for sub-program >B0.
```

ACCESS MODE #2 FOR SUB-PROGRAM >B0 (Selects an 8K page and turns on RAMBO):

```
>834A  IN   Not used for input to the sub-program >B0.
       OUT  CRU address of the RAMBO/HORIZON card that is currently on.
>834C  IN   A page number from 1 to max.  number of free 8K pages.
       OUT  The 16 bit CRU value used to select the page and turn RAMBO on.
>8350  IN   Not used for input to the sub-program >B0.
       OUT  >0000 if no errors, or >FFFF if errors detected.
```

Accessing >B0 with >834C 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 >834A and >834C. 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 BLWPs.
SETPAG MOV  *R11+,@>834C Get number of the page to select.
       LI   R0,VDPPAB    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.
       BLWP @VMBW        Write the PAB (>01B0) 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.
CALLBO DATA >01B0        PAB data for sub-program >B0.
```

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

```
>834A  IN   Not used for input to the sub-program >B0.
       OUT  Not changed, will be the same value as IN.
>834C  IN   >FFFF = Tell all cards to switch back to original Horizon mode.
       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 errors detected.
```

Accessing >B0 with >834C 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 ROS will do a complete reset any time the TI Title Screen appears. The previous assembly examples should be enough to figure out how to use this mode of the sub-program >B0.

4.0 USING THE AUTO-START MENU PROGRAM

The following documentation is retained for this release. MENU and BOOT program documentation is generally included with the respective program files. There are multiple versions of the MENU program including the common v7.39 (40 column) and the 9640Menu (80 column) programs. This section may be removed in future publications.

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 configure one of the custom calls to “MENU” with a length of “4”. If you want MENU to appear on power-up, the first custom call must be the one that is configured 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:
@	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 the diskette.
G	Cycles through all the GROM names; each name appears next to “C”.
I	Toggles the I/D color interrupt routine on and off.
K	Calls a routine – does a CALL.
P	Changes the print device name for SHIFT 1 and 2.
S	Displays the ROM cartridge header at >6000 (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.
FCTN 5	Enters edit mode for the “user options”.
FCTN 9	Goes to the next power-up program or where you originally came from.
FCTN =	Resets the computer system.

Key	Active keys during viewing of a file:
SPACE	Displays the next 22 lines of the file being viewed.
FCTN 5	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.

Key	Active keys on the main MENU screen:
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 space bar, 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 (REDO) will bring you back to the drive number/letter selection prompt, or pressing FCTN 9 (BACK) will bring you back to the 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.

The directory feature in some programs is sector-based and will not work with TIPI disk devices nor with hard drive devices.

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 also be 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.

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 space bar. If you would like to rapidly scroll through the file, press FCTN 8 (REDO). Viewing will then continue until the end of the file is reached or until you press the space bar. FCTN 8 (REDO) is most useful if you are sending the file to your printer. By merely pressing FCTN 8 (REDO), the complete file will be output 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 an 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 "G" 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 "@", 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 (BACK) will abort the delete operation.

By pressing "I", you can toggle on and off the action of the default colors as selected in the CFG program to be carried over the 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 at >1000 would be. 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 configured 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 activated 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 activated with the keys 1 thru 9. You can toggle thru the three screens by using the space bar. Each time the space bar is pressed, the system will switch to the next menu screen, and when it’s at the end of the screens, the system will loop back around to the main MENU screen.

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 space bar. 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 configured for a CALL. The following examples show the many ways a user option can be used:

To run a program from the ramdisk 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.filename
```

In the above examples, the user option could load either an 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 an "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 use as long as it is preceded by "X" and a space.

You can even setup your user options to call one of the custom calls, 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 samples to get you going. After you have finished changing the user options to your liking, red 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.

5.0 NOTES ON DIFFERENT THINGS

5.1 BASIC DEMO OF THE NEW OPCODE #8

“OP8DEMO” is a simple Extended BASIC program with Assembly support that demos the new opcode 8 (SCRATCH RECORD). Before running it, setup your Horizon card with two ramdisk drives numbered 6 and 7. Then copy the Assembly object file “OP8DEMO-O” to drive #7 and erase all files from drive #6. Now load and run the BASIC program file “OP8DEMO”.

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 makeup 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 maximum 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 “OP8DEMO-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 CFG 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 “L” to start the loading of the diskette into the RAMBO program space. On the top 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 errors 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.

6.0 CREDITS/ACKNOWLEDGEMENTS FOR THIS PACKAGE:

ROS 8.14/SERIES 8:

Texas Instruments, for supporting OPA with many needed technical manuals and books on the TI and the many support chips and cards.

Thanks to MYARC 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 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 Ballantyne and Charles Earl 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 padded 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 >6000 so it can be able to page in the HORIZON RAM at >6000 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 8 software finished. Also, many thanks to John Birdwell for his help in fixing some bugs in the original ROS.

J. 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 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 had not been designed, I am sure the TI world would have been smaller today! Many thanks to Ron Gries and his great team!

ROS/CFG 8.32 - 8.42c:

Bud Mills, for the generosity and support offered to me and so many others in the community.

Tony Knerr, for the XB27 Suite that is has been so invaluable for testing bug fixes, enhancements, and new programs.

D. Warren and J. Fetzner, for their invaluable support and enthusiasm with this project effort.

G. Bowser, OPA, for the series 8 manual and software, and for thoroughly commenting the source code making it possible to update the ROS and CFG files.

M. Brent, Jon Guidry, and J. Fetzner, for creating and supporting the *UberGROM* cartridge that I use daily.

M. Splett, creation and support of the *TIPI* hardware and software.

R. Benzinger, creation of the *xdt99* development tools, used to assemble and link ROS/CFG 8.42.

B. Carmany and C. Schneider, for subjecting their systems to interim (and at times catastrophic) versions of the ROS and CFG software.

F. Kaal and M. Zapf, for their respective valuable file manipulation tools: *TI99Dir* and *TIImageTool.*

D. Eicher, manual review and editing.

Albert, for hosting AtariAge.com, a forum that has enabled the TI community to come together for the discussion and implementation of many new hardware and software efforts.

And to everyone in the TI community who participates, creates, and endeavors to do more than what the TI could have ever imagined.

***** EOF PI Day, 3.14.2020 *****