BITS,BYTES&PIXELS

LIMA 99/4A USERS GROUP

NOVEMBER 1987 Volume 3, Number 9

CORRECTION

In the Oct 87 issue of BB&P we stated in an editorial note contained within an article by J Peter Hoodie that those with a CorComp disk controller could only use one Horizon Ramdisk. We are informed by David Romer of Horizon Computer Ltd that this is not correct. Multiple Horizon ramdisks may be used the CorComp disk controller. "You may, in fact operate as many Horizon cards as you have empty slots in your PE box regardless of your floppy controller" according to Romer. **!!DONE!!**

A REVIEW OF GENIAL COMPUTERWARE'S EPROM FOR THE HORIZON RAMDISK

by Charles Good

Although this article describes my experiences with the product listed in the title, the article should also be of interest to those considering use of John Johnson's MENU v6.3, which is very similar in many of its features. This article is based on my use of a TI disk controller , 32K, and two Horizon Ramdisks equipped with the Genial eprom ramdisk operating system.

AN OPERATING SYSTEM IN ROM:

One of the most annoying things that can happen to users of Horizon Ramdisks is to lose the ramdisk operating system (ROS). You know this has happened when, among other wierd things, a disk directory lists thousands of sectors used and free. Often files on the ramdisk are undisturbed by a blown ROS, but sometimes files are lost. Occasional ROS loss has occurred to all members of the Lima Area user group who have used the operating system that is supplied by the folks at Horizon Ramdisk. Although I have no user experience with the alternative RAM based "MENU v6.3" ROS of Johnson and Ballmann (in the public domain; reviewed in Micropendium Aug. 87; available in user group libraries or by sending a disk and postage paid return mailer to Miami User Group, 19301 N.E. 19th Ave., North Miami Beach, FL 33179), I have carefully read the MENU docs and the Micropendium review. I suspect that MENU v6.3 too is subject to occasional loss. This is because both the "official" and the MENU v6.3 ROS are housed in RAM. Although the operating system should always be there since the RAM is battery backed, sometimes a stray bit of data can get into the ROS during ramdisk access and mess things up. This can happen, for example, if the ramdisk is set for DSK1 and extended basic (XBASIC) fails to load properly. The XBASIC module initially accesses DSK1.LOAD when XBASIC is selected. I know from personal experience that if XBASIC doesn't load properly it may send bad data to a RAM based ROS blowing the ROS before the computer locks up.

Another problem I have very occasionally had with the "official" RAM based horizon ROS is the partial overwriting of existing ramdisk files when dumping new data files to the Horizon ramdisk. The ROS memory map sometimes gets confused and the new and old files get thrown together in the same ramdisk sectors. Neither the preexisting or newly dumped files can be used, and it is often necessary to reinitialize the ramdisk in order to clear up the mess. I don't know if this problem also occurs with MENU v6.3.

A solution to RAM based ROS unreliability or total ROS loss is the Horizon Ramdisk eprom sold by Genial Computerware. This eprom provides an operating system very similar to MENU v6.3 except that it is locked up in ROM and cannot be damaged by bad data being sent to it. This eprom is (as I understand from scanning other group's newsletters) required to use a Horizon Ramdisk with the 9640 computer. On the 99/4A the eprom is most suitable for systems with only one Horizon Ramdisk that is set at a CRU address of >1000. Things get complicated for ramdisks at other CRU addresses and when using multiple ramdisk cards. The Genial eprom ROS is specifically designed to be easily compatable with FUNNELWEB files on the ramdisk, more so than the two other Horizon operating systems.

Installation of the Genial eprom is easy. All you do is pull one of the original Horizon RAM chips out of its socket and replace it with the Genial eprom. Then you solder one wire already attached to the eprom to a particular pin on an adjacent chip.

As with the MENU v6.3 ROS, the Genial eprom provides an extra 4K (16 sectors) of data storage compared with the "official" ROS. This means that 90K, 180K, and 256K Horizon Ramdisks will actually store 94K, 184K, and 260K of data respectively with the Genial eprom installed.

SPECIAL CALLS:

The eprom ROS includes many CALL's all of which can be used without any module. In multiple Horizon card systems these calls usually only affect the card set at the lowest CRU address. The calls will work from command mode in TI BASIC and XBASIC, and from a running TI BASIC program. They DO NOT work from a running XBASIC program. The "official" ROS and MENU v6.3 have provisions for executing their calls from running XBASIC programs. CALL's are available to set the drive emulation number and drive emulation name. You can CALL HDDIR to get a directory of your ramdisk with the lowest CRU address even if it has no name or drive number. You can also load alot of assembly language software that would normally require use of the EA or XBASIC module. You can

next page

load and run program image software with CALL EA5. As UTIL1, FUNNELWEB can be loaded with no module at all using this call. You can also from TI BASIC do a CALL ILD (same as CALL INIT), then CALL LD (same as CALL LOAD) a D/F80 assembly program, and CALL LLD (same as CALL LINK) to this program from command mode or a running TI BASIC program.

CALL DM will load DM1000 as it does using the other operating systems. However, the Genial eprom is supposed to be compatable with FUNNELWEB's DM1000 and thus loads files named MG and MH rather than MGR1 and MGR2, the names usually given to DM1000 files. With the "official" ROS and with FUNNELWEB on the ramdisk you have to have double sets of DM1000 files on your ramdisk if you want DM1000 to load from both CALL DM (needs files named MGR1 and MGR2) and from FUNNELWEB (needs files named MG and MH). From the Genial eprom, files named MG and MH will load either way. But.....THERE ARE PROBLEMS here with DM1000 v3.5 software. If you use the MG and MH files that come with FUNNELWEB V3.4 (May 14,87 update) they will work fine CALL DMed from TI BASIC, but are not easily usable when CALL DMed from XBASIC. The FUNNELWEB MG and MH DM1000 files do load and execute from XBASIC but the foreground and background colors are both black and you can't see anything on the screen. There seems to be no easy way to change these colors, although you can from the totally black screen press 3 (Misc Utilities) and then still with a totally black screen press 4 and/or 5 to alter foreground and background colors which will then make the text visible. If you take regular DM1000 v3.5 MGR1 and MGR2 files (available from the Ottawa User Group or from most user group libraries) and rename the files MG and MH these files will load from XBASIC command mode, TI BASIC, and FUNNELWEB with a white on blue background. However, you loose the ability to directly return to FUNNELWEB from within DM1000. Perhaps more importantly, you loose the ability to access with DM1000 any Horizon Ramdisk whose CRU dip switch setting is not at >1000. Even the DM1000 v3.5 you get directly from the Horizon folks when you buy a ramdisk (at least as of Feb 87) will not recognize a ramdisk with the Genial eprom if the ramdisk is set up with a CRU address other than >1000. This is a VERY SIGNIFICANT problem to those of us with more than one Horizon Ramdisk in the PE box. The FUNNELWEB DM1000 MG and MH files have been altered by the FUNNELWEB authors to access ramdisks at any CRU address, but if you use these files you have the black screen difficulty mentioned above if you CALL DM from XBASIC. Probably the best solution if you have a ramdisk that is set to a CRU other than >1000 is to use the special DM1000 v3.5 MGR1 and MGR2 files that come on the MENU v6.3 operating system disk (see above for an address to obtain this). Rename these files MG and MH. They search the DSR's starting at CRU >1200 going up to >1F00 and then back to >1000 before finally stopping at the real disk controller at >1100. These particular DM1000 files will find Horizon Ramdisks at any CRU setting, and will load properly and run from TI BASIC, XBASIC, and FUNNELWEB, although you still loose the ability to directly return to FUNNELWEB from within DM1000.

CALL MD will load MoDem software (ie. a terminal emulator) or any other program image file series whose first file is named MD. This is compatable with the MODEM loader of FUNNELWEB.

THE BOOT PROGRAM:

CALL BOOT will load a program image file called BOOT, and a version of John Johnson's MENU program named BOOT is included with the Genial eprom. When BOOT is loaded you get this very versatile menu:

1. Show Directory
2. View a file
3. Run a program
4. Disk Manager

5-9 Your PROGRAM options

C. Cartridge Name (if present)

With item #1 you can get a directory of any drive or any ramdisk at any CRU setting. You have the option of printing this directory to a printer in 4 columns all across an 80 column page with no wasted paper. With the space bar you can mark a file which will then be displayed on the screen if you select #3. Item #2 displays and optionally prints to a printer any D/V80 text file.

Items 3-9 will load and run anything that lists as PROGRAM on a disk directory. Loading from 4-9 requires the PROGRAM files to be on the same Horizon card that contains BOOT if you are using more than one Horizon card. These can be program image files (BOOT will run those without any module being required and from a ramdisk set at any CRU address), TI BASIC, or XBASIC programs. In order to run a program in either BASIC, the XBASIC module must be pluged in and the BASIC program must be on a floppy (item #3) or on a ramdisk set at CRU >1000 (items 3-9). That's right, you need the XBASIC module to run TI BASIC programs from the BOOT menu, and you cannot run TI BASIC or XBASIC programs from BOOT on ramdisks with a CRU address that is greater than >1000. (I am not sure, but I suspect that the MENU program in the MENU v6.3 system probably has similar limitations. Previous versions of MENU totally lacked the ability to load BASIC programs from the powerup MENU display.) BOOT will not run large XBASIC programs that show up as I/V254 on a disk directory. Also, you can't load D/F80 assembly code from the BOOT menu, although you can with the CALL's discussed above. Only PROGRAM files work. For item 3 you either type in your own file name and drive number or you use the one you flagged from item 1 (Show Directory). The screen display and file names for items 4-9 are customized within the BOOT program. Source code is provided so you can reassemble BOOT with your own item 4-9 customizations. It is really easy. Even easier is to use a disk sector editor and directly edit BOOT. Each screen display name is allowed 15 spaces and each file name is allowed 10 spaces in the BOOT code as it appears on disk or ramdisk sectors. Just type over what is already there. You can use the trailing blank spaces as long as your text doesn't exceed 15 or 10 characters. For file names to be

next page

loaded from items 4-9 it is not necessary to specify "DSKx." when you customize file names in BOOT. The eprom ROS expects each file to be on the same ramdisk as the BOOT program, irrespective of the drive number assigned to that ramdisk. Item #4 is preconfigured to load DM1000, but you can change this to something else if you want.

There are other options from the BOOT menu, some not visually indicated. Among these are pressing C which loads the module that is plugged in. If no module is present a "C" isn't displayed on the BOOT menu. Pressing B takes you to TI BAISC. If you press M you can run the CorComp disk manager. Pressing T will display a clock with the correct time if you have a real time clock installed. All of the various clock devices made for the TI as of May 87 are supported here.

CORRECT FILE I/O

The "official" ROS has a bug which does not permit correct overwriting of single record Display/Variable files on the ramdisk. This bug is corrected in the Genial eprom ROS. Single item D/V files appear to perform correctly in all respects.

SPECIAL POWERUP OPTIONS ARE AVAILABLE:

One of the really neat options available to Genial eprom users is the ability to customize what happens when you turn on the computer. The eprom, programmed with a powerup option, takes control of the computer when it is first turned on or reset (as in inserting a new module). The various custom powerup options that can be loaded by the Genial eprom bypass the color bar TI title screen, and also bypass any custom powerup menu that you may have created with a GRAM KRACKER or other gram device. From either BASIC command mode type DELETE "PU.x" where x is a letter. "PU.D" powers up to DM1000 (loads MG and MH). "PU.M" powers up to MoDem software (loads file MD, MF, etc.). "PU.U" loads a file called UTIL1. "PU.B" loads the BOOT menu. You can temporarily bypass the powerup option if you press <ENTER> while the computer is turned on or reset. You then get the color bar title screen or your GRAM KRACKER menu. You can permanently turn off power up options by typing DELETE "PU" from either BASIC command mode.

Of these powerup options, by far the most useful are those that load BOOT and UTIL1. I can't imagine why anyone would want to immediately boot DM1000 or terminal emulation software each and every time the computer is turned on. It is, however, really neat to be able to start directly with the BOOT menu. One of the reasons for owning a GRAM KRACKER is to be able to create a custom powerup menu. The BOOT menu does more kinds of things than can be done at power up with a GRAM KRACKER! My own system is configured to boot UTIL1 of FUNNELWEB. Every time my computer is turned on, the screen shows the following slightly modified intermediate FUNNELWEB v3.4 menu:

1. Editor
2. Formatter
3. DM1000
4. MODEM
5. 1st Menu (changed by me from DATA BASE)
6. USER LIST

Since I do alot of word processing and disk editing for our user group library, I can do most of my computing directly from this menu. I can get from this intermediate FUNNELWEB menu to the first FUNNELWEB v3.4 menu (the one with the title screen and 16 configurable choices) by using a little program called "XB" published in vol. 1, #6 of Barry Traver's Diskazine. "XB" says in assembly RUN "DSK1.LOAD". I modified the program to "DSK3" and named it DB so that it loads where FUNNELWEB normally expects to load Data Base. I then used a sector editor to alter the text of #5 in the above FUNNELWEB menu in FUNNELWEB's LOAD and UTIL1 programs. My own feeling is that FUNNELWEB is more powerful and flexible than BOOT. By selecting 5 or 6 from the above FUNNELWEB intermediate menu, I can have quick menu access to many more ramdisk or floppy stored programs than can be handled by BOOT's menu.

The Genial eprom powerup options won't work when using the CorComp Memory Plus card. This card (even without the CorComp disk controler) captures control of the computer at powerup and generates its own title screen. If you try to set up a DELETE "PU.x" option the computer locks up at powerup unless you simultaneously press <ENTER> to bypass the powerup option. All other Genial eprom features seem to work properly with CorComp Memory Plus.

COMPLICATIONS WITH MULTIPLE EPROM EQUIPPED CARDS:

It is obvious from my experience that the Genial eprom is designed primarily for systems with only one Horizon Ramdisk. If two or more Horizon cards are used with a Genial eprom on each card, things get complicated. You can only set the drive emulation number and name with one card in the PE box at a time. Also, the power up options need to be set up on only the card with the highest CRU address. Put this card in the box and use CALL's to set up its drive emulation number and name. Use DELETE "PU.x" to specify a powerup option. Remove this card and insert another Horizon Ramdisk and use CALL's to set up its name and number. Continue configuring each card individually, then reinsert all cards in the box. The "official" and MENU v6.3 (but not earlier version) operating systems do this easily from software with all Horizon cards remaining in the PE box. With the Genial eprom it is a big pain to have to pull cards in and out of the PE box every time you want to change configurations in a multicard system.

next page

Other problems with multicard systems have been mentioned earlier in this article. DM1000 may not recognize cards set at a CRU other than >1000 (and with multiple cards, only one card can have this address). The BOOT program looses much of its usefulness since it must be placed on the Horizon card with the highest CRU address and as such cannot boot BASIC programs.

DOCUMENTATION:

The documentation does an excellent job explaining installation and how to use the special CALLs on systems with only one Horizon card. What the documentation does NOT do is give any hints of the problems encountered when using multiple Horizon Ramdisks each equipped with a Genial eprom. I had to figure out all the multiple ramdisk oddities (such as configuring each card one at a time with the powerup option on the highest CRU card; special CALLs only affecting the card at the lowest CRU address) by myself. Statements in the docs concerning multiple card configuration, DM1000 use, and BOOT use with more than one Horizon ramdisk would certainly be appropriate. The only mention of multiple Horizon cards in the docs is this statement "The reason for changing the names of all the CALLs (from CALL names that do the same thing in the "official" ROS) is to allow for both the eprom operating system and the original operating system to co-exist in the same expansion box (although in separate cards)." This statement does point to a solution to the problem of multiple Horizon card configuration, but is not much of an indication to a potential buyer of the fact that it is somewhat difficult to use the eprom on multiple cards.

RECOMMENDATIONS:

The Genial eprom is RELIABLE!! In two months of extensive use I have not lost any data due to ramdisk memory map mixups, nor have I lost a ramdisk operating system. This reliability is the best feature of the Genial eprom as compared to the RAM based Horizon operating systems. I also think the ability to boot FUNNELWEB each and every time the computer is turned on is fantastic. If you have only one Horizon Ramdisk, than the Genial eprom deserves very serious consideration, although it costs lots more than the other two Horizon operating systems, both of which are free. Not ever losing the operating system makes the files stored on your ramdisk much more secure. You can have an almost identical operating system if you use MENU v6.3, but MENU will in all probability sometimes get messed up and require rebooting. If you are using more than one Horizon Ramdisk you need consider cost. You also need to compare the inconveniences of configuring cards one at a time and the fact that BOOT with 2 cards will not do all of the things it can do with only one card with the benefit of having a secure ROS on both cards.

next column

REPORT CARD

Performance.....A
Ease of use.....A
Documentation...B
Value.....B
Final Grade.....A-

Cost: $25.00
Manufacturer: Genial Computerware
Box 183, Grafton MA 01519
Requirements: Horizon Ramdisk, disk system.

$$DONE$$

"DISK INFORMATION MANAGER" CAN LOAD FROM FUNNELWEB

DISK INFORMATION MANAGER was reviewed in the June 87 issue of BB&P. In this review it was stated that you need the EA module to load DIM and that DIM will not load from FUNNELWEB. A recent letter from the co-author of FUNNELWEB, Tony McGovern, tells how to modify DIM so that it will load as an EA program file from within FUNNELWEB. Using a sector editor examine the first sector of DIM. Change A0CC (the 5th word in the sector) to A0FA. That is all there is to it! However, if you make this change you will no longer be able to load DIM using the EA module. Oh well, thats the price of progress.

$$DONE$$

BITS, BYTES & PIXELS

Published by Lima OH

99/4A User Group

Material contained herein may be copied by any user group as long as credit is given. Individual and dealer advertisments are published at no charge at the editors' discretion.

ADDRESS- P.O. Box 647
Venedocia Ohio
45894

Published monthly except July and August

SOMETHING IS WRONG AT RYTE DATA

by Michael J. Martinko
Lima Area User Group

I had read with a great deal of interest about Ryte Data's new modification for TI's disk drive controller! Quad density was more than I could hope for since I was then stuck with SS/SD drives. New drives would be needed to achieve the quad density, but that would come later. So in mid-April (1987) I sent a check for $55 to Ryte Data along with the TI disk controller from my expansion box. Speedy service and return had been promised. Six weeks later the controller finally arrived. The check, to their credit, was not cashed until the controller had been placed in the return mail. With great anxiety and extreme care the controller was removed from the packaging with instructions and the old chips. The instructions, a poor copy of a single sheet, are a joke! The instructions, however, aren't needed as the controller doesn't work PERIOD! After carefully installing the controller in the PE box and powering up the system I could not access the drive in either basic or extended basic! Perhaps something in my box had gone haywire. Careful not to jump to conclusions I wrote to Ryte Data about my problems. A month later, still no answer. I left messages on their phone answering system with a plea to call collect! No reply. Having a non-functioning computer system was almost unbearable. Since I had a controller I could not use I reasoned nothing ventured, nothing gained, so I decided to snip the controller back to Ryte Data for testing. Maybe they are just behind in letter writing.

July came along and CorComp advertised their controller. Having purchased and received a CorComp controller in just ten days I plug it in, and everything whirrles like a top! The instructions are terrific (in comparison) although I would have liked some more examples from the utility tools.

Since there is nothing wrong with my expansion system or drives, and since I can get absolutely no reply from Ryte Data, I hereby proclaim Ryte Data to be "WRONG DATA" and warn any curious users who might be tempted to purchase their quad density system.

P.S. It is now mid September, 1987 and I have still received no reply from WRONG DATA.

EDITOR'S NOTES: Mike ordered his disk drive modification based on a very complementary article we published in Bits Bytes Pixels in March 1987. This BB&P article quoted from Ryte Data's own commercial newsletter in which VERY RAPID turn around time of a few days was promised if you sent your TI disk controller to Ryte Data and had them install the new chips. Obviously, Mike did not get such rapid turn around time. As of now (early November 87) Ryte Data STILL HAS MIKE'S TI CONTROLLER AND HIS $55.

Based on Ryte Data's 1987 catalog there seems no doubt that with the chips installed, Mike's modified controller should have been able access his original 40 track drive in SD format. The catalog (pages 8 and 9) states, "You can mix drives and formats i.e. one original 40 track drive and one or two 80 track drives. ...It DOES remain compatable with STANDARD format (SS/SD - DS/SD). ...60 day guarantee on parts and labour (if performed in our shop)." Since the modified controller would not access Mike's 40 track drives, something obviously was not right, and that is why Mike sent the controller back to Ryte Data for checking. How else could one expect Mike to take advantage of the 60 day guarantee, since Ryte Data still refuses to acknowledge any of his letters and phone calls.

As newsletter editor I sent Ryte Data a copy of Mike's article on Sept 20, 1987 and indicated that it would be published in our November newsletter. I invited Ryte Data to reply to Mike's article and promised to publish this reply along with Mike's article. That was 6 weeks ago and I have received no reply. Nothing personal about that, I guess. Mike gets no replies to his letters or phone calls either.

Mike's case is not the only example of Ryte Data taking money and not delivering the goods. Through a well informed and highly respected source I have knowledge of other such cases including that of a British gentlemen who many months ago ordered over $400 worth of goodies from Ryte Data. They CASHED HIS CHECK, SENT HIM NOTHING, and failed to respond in any way to his enquiries. This gentlemen now has my source and the Canadian postal authorities looking at his case.

Taking money for defective goods or for goods not delivered, and then failing to respond to letters and phone calls from several sources (including this newsletter) is not a good way to run a business. I advise caution in dealing with WRONG DATA.

Charles Good
BB&P editor

FOR SALE

99/4A console with 2 cartridges and dual cassette cable. Only $35 or trade even for an Atari modem. Call Mark Wagner evenings at 419-339-5853. (Ed Note: Mark told me he will deal and accept a lower offer if he can't get his price. He is very anxious to sell.

INTRODUCING THE GENIAL TRAVelER:

A Magazine-on-Disk for the TI-99/4A!

If you have a TI-99/4A with a disk system, you'll be interested in the Genial TRAVelER, an exciting magazine-on-disk which provides over 700 sectors of programs and articles in each issue.

TRAVelER is edited by Barry Traver, whose programs have been published in various TI magazines, including 99'er Home Computer Magazine, Craig Miller's Smart Programmer, Rich Mitchell's Super 99 Monthly, and MICROpendium. He is also the author of ARCHIVER, the standard file archiving procedure used on CompuServe, Delphi, GEnie, and the Source. He was the founder and first President of the Philadelphia Area TI Users Group. He is now a Sysop on both CompuServe and GEnie. He keeps in touch with the TI community in other ways as well, such as attending TI Faires in San Francisco, Chicago, Milwaukee, New Jersey, Boston, Providence (R.I.), Los Angeles, Ottawa, Seattle, and Washington, D.C.

The "diskazine" is actually priced less than some "freeware": you get a six-issue subscription for only $36 (that's over 4000 sectors, so that you are paying less than a penny per sector!).

Authors whose programs and/or articles have already appeared in TRAVelER include Ron Albright, John Behnke, Barry Boone, Paul Charlton, Mike Dodd, Tom Freeman, Herman Geschwind, J. Peter Hoddie, Todd Kaplan, Jim Peterson, Wayne Stith, Mike Wright, Jonathan Zittrain, and others, including Barry Traver, whose ARCHIVER first appeared in TRAVelER.

The "diskazine" is being published on commercially-made SS/SD "flippies," so that the same format will work on everyone's disk system. It is assumed that subscribers will have at least a 32K memory card, Extended BASIC, and Editor/Assembler. (Warning: the publishing schedule can be somewhat erratic: it took about two years to complete Volume 1! EDITOR'S NOTE: The "XB" Funnelweb enhancement noted on p.3 of this newsletter is part of volume 1 of Genial Traveler.

TRAVelER SUBSCRIPTION FORM

Check one: ___ Volume One @ $36.00 (six issues plus free bonus disks)

___ Volume Two @ $36.00 (six issues)

___ Volume One and Volume Two @ $65.00

Name _____

Address _____

City _____ State _____ Zip Code _____

Please make your check payable to Barry Traver, and send it - along with this form - to Barry Traver, editor, Genial TRAVelER, 835 Green Valley Drive, Philadelphia, PA 19128. Thank you.

DEBUGGING

by Jim Peterson

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often.

So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps you to debug better than some that cost ten times as much.

There are really three types of bugs. The first type will prevent the program from running at all - it will crash with an error message. The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a Basic program, which uses character sets 15 and 16, in Extended Basic.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand - such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance, that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and give you an error message - probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong, and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark).

```
100 PRINT "Program must be s
aved in:"merge format."
```

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program; that line may actually be correct but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that - Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended Basic manual list almost all the error codes, and almost all the causes of each one - it will pay you to consult these pages rather than guessing what is wrong.

You may create some really bad bugs when you try to modify a program that was written by someone else - especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray - you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-looking command !@P- up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the program to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the

memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE IN 680, and you bring line 680 to the screen and find it reads
CALL HCHAR(R,C,CH)
just type PRINT R;C;CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended Basic, you can even enter and run a multi-statement line in immediate mode (that is, without a line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by
FOR J=1 TO 100::PRINT A(J);:: : NEXT J - or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped but A will have a value of 100.

You can temporarily stop a program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and restart it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as 1 BREAK 960,970,980 Note that in this case the program breaks just BEFORE those listed line numbers. You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement, which will cause each line number to print to the screen as it is executed. If used as a command, it will trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multi-statement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE( ) with the line number between the parentheses, and a subprogram after everything else
30000 SUB TRACE(X)::DISPLAY AT(24,1):X :: SUBEND

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen - it may have invisible characters at one or both ends. Try PRINT LEN(M$) to see if it contains more characters than are showing; or
PRINT "*"&M$&"*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything - and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program. (Believe me, I speak from experience!)