MSP 99 USER GROUP

Vol. 10 No. 3
April 1987

THE MSP 99 NEWSLETTER

HANDLING DSRs IN ASSEMBLY LANGUAGE SUBROUTINES THAT ARE LINKED TO XB (or how the power of the user's group saved me mucho aggravation!!)

by: Ed Johnson 3/15/87

For some time I had been wrestling with a problem: I was working on a large project with Donn Granros (author of OLD DARK CAVES and a new MSP99 member) that required the use of loading and saving memory image files from/to disk. Well, in the beginning it seemed simple enough- just set up the PABs and buffer space that we needed, read or write the files, and return to XBasic.

I wrote the routines and found that they were working just fine after the fifth or sixth assembly (has

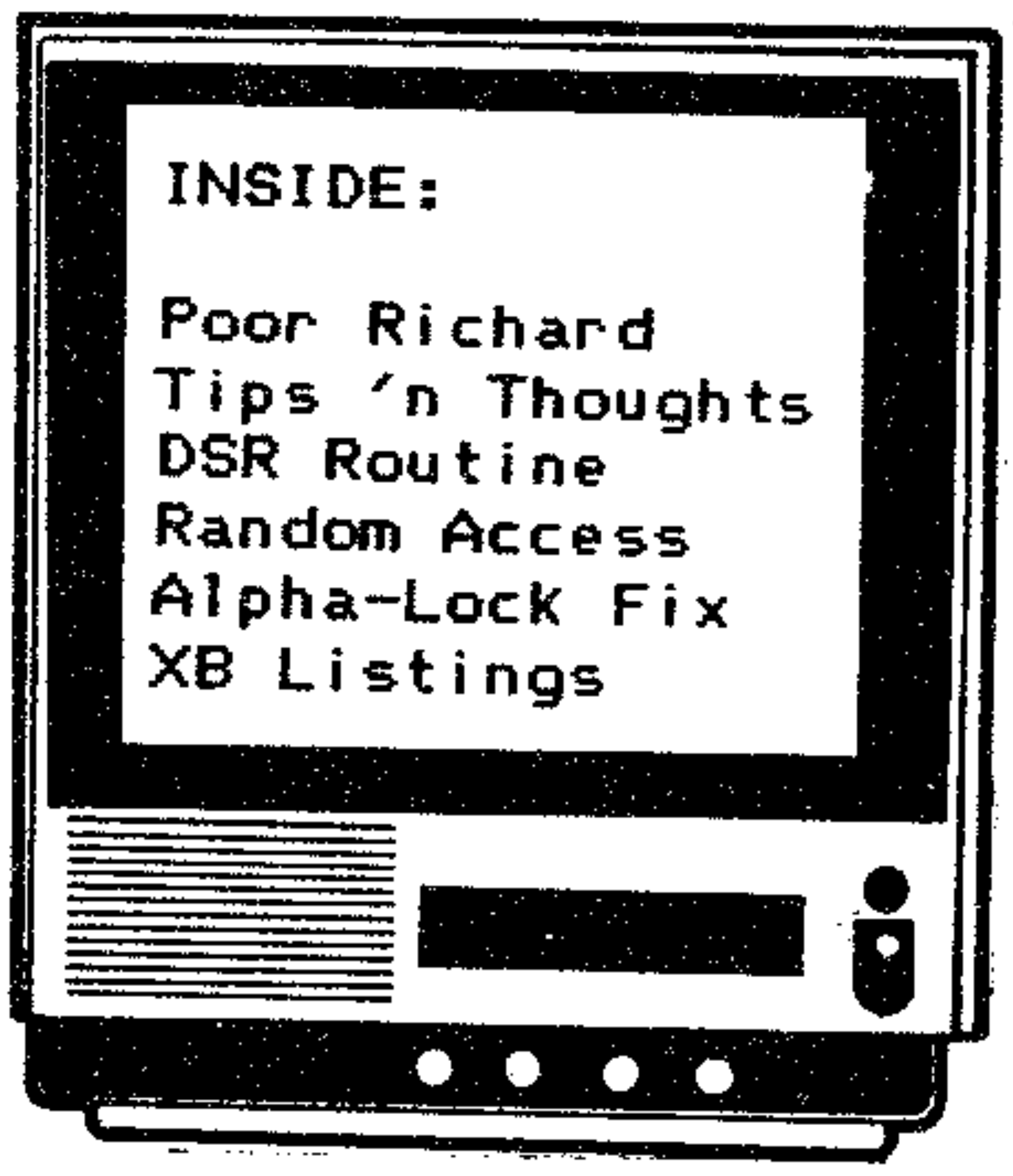
(Continued on Page 11)

MSP 99 NEWS

Last month's hardware meeting was a great success. It seemed to be one of the busiest meetings we've had in quite a while, yet we still found time to hold not only our monthly raffle but the Easter ham raffle as well.

For our monthly raffle we ended up with two prizes this time since one of our members kindly donated a handy little disk file on the spur of the moment. The TI power supply we originally had promised went to R. Greenwood while the disk file went to Donn Granros. Donn is the author of the popular TI game "Old Dark Caves" and has donated several copies of it for next months raffle, just so you know what's ahead.

Two Easter hams were sent off to Joe Syverson and Gerald Belfanz who assures me he never won anything before. See how much owning a TI can improve your life.



INSIDE:

The MSP 99 USERS GROUP meets each month for discussions and presentations that enable its members to be better informed about their computers. Users group members share and exchange information. Some members have a broad range of computer expertise. others are just beginning. We are not affiliated with or sponsored by any other group or company. Membership dues are $18 a year for a family or individual, and $50 for a sponsor member. You're welcome to visit a meeting as a guest before you join. Call or write for more information.

USERS GROUP MEETINGS are held the third tuesday of each month at Dunwoody Industrial Institute, 818 Wayzata Blvd., Minneapolis, MN 55403. Meetings start at 7:00 PM.

MSP 99 USERS GROUP
P.O. BOX 12351
ST.PAUL, MN 55112, U.S.A.

PRESIDENT: Dick Lauhead 429-5256
V. PRES: George Madline 341-3780
SECRETARY: Gary Gese 529-3989
TREASURER: Mark Tellevik 780-5856

The MSP 99 NEWSLETTER is published eleven times per year on a monthly basis, except during July, by the MSP 99 Users Group. Members are encouraged to contribute articles for publication. Opinions expressed are those of the writers and not necessarily those of the MSP 99 Users Group, its officers, editors, of mermbers. Materials accepted by the editors for publication in the MSP 99 Newsletter, including software listings, are believed to be in the public domain. Newsletter articles may be reproduced by other users groups if appropriate credit is given to the author (if one is listed), and to the Minneapolis, St. Paul 99 Users Group.

NEWSLETTER EDITOR

Gary Gese 529-3989

Articles intended for the next newsletter should be submitted NO LATER than the users group meeting on the month prior to publication. Articles submitted after this deadline are likely to appear in the following month's newsletter.

COMMITTEE VOLUNTEERS are sought for all of our committees. (Education, Equipment, Program, Publicity, Software, Newsletter)
If you would like to join one of these committees or have an idea for a monthly program, please contact one of the officers.

COMMERCIAL ADVERTISEMENT RATES:
Business firms that wish to communicate with our members may do so by placing an advertisement in the newsletter. Rates are:
Full page $40; Half page $30; Quarter page $22.

Each ad must be camera ready in one of the sizes indicated and paid in advance. Inserts (printed by the advertiser on 8 1/2 X 11 or 8 X 10) may be inserted in the newsletter at $20 per sheet. Contact the editor for more information.

CHANGE OF ADDRESS: Before you move, please mail a change of address tr the Users Group. DO NOT rely on the standard Post Office change of address card since the P.O. will not forward this Newsletter.

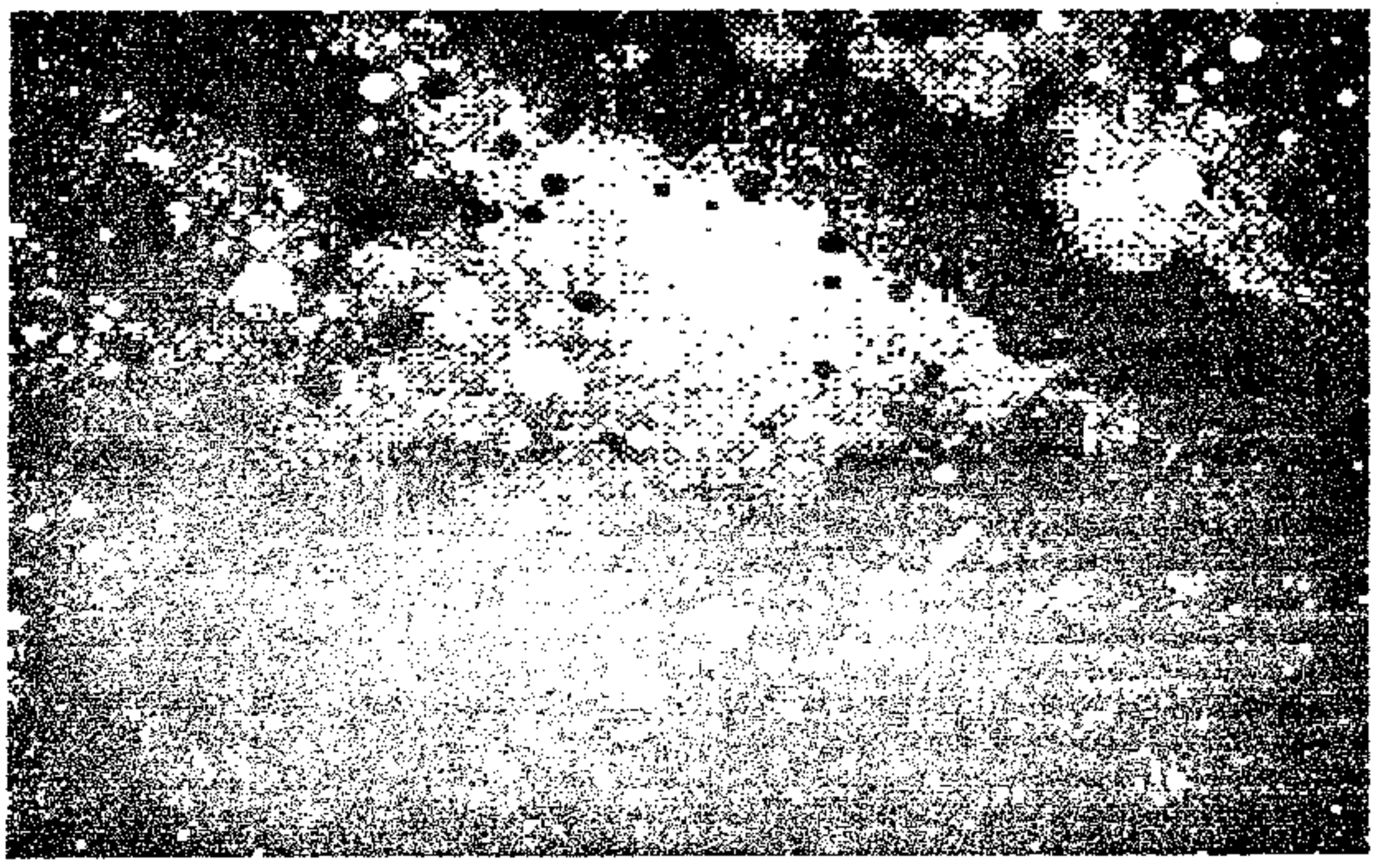
**** WARNING ****

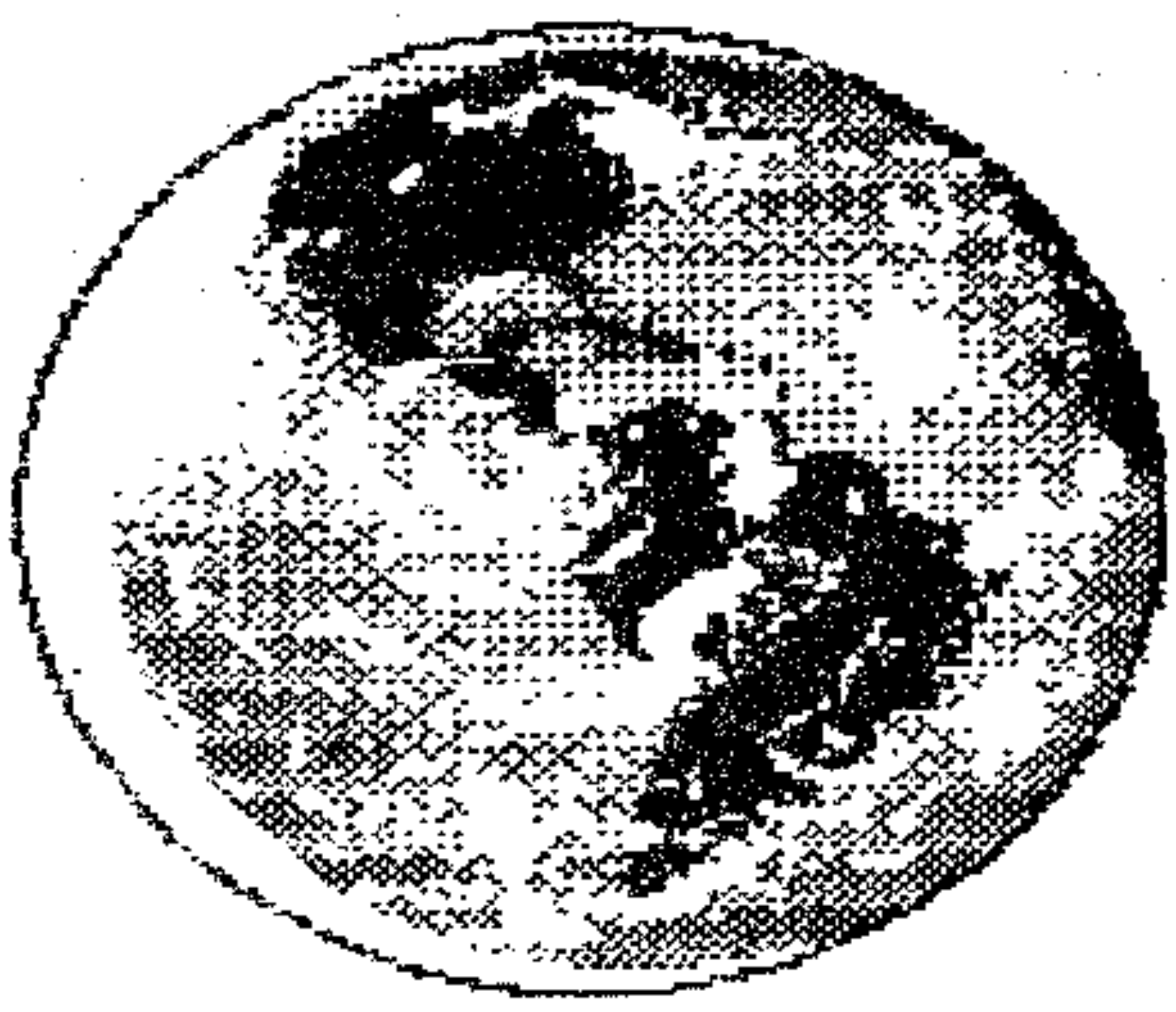
In last month's hardware issue you may have noticed a modification to the TI keyboard that would provide a fix for the Alpha Lock problem that has plagued so many of you joystick jockeys. The fix was to prevent the Alpha Lock key from interfering with the upwards movement of the joystick.

This fix does work just the way it is described, however there is one drawback. The Alpha Lock will no longer lock-in the capital letters the way it used to. In fact, the Alpha Lock is effectively removed from the circuit. The "fix", rather than correcting one problem, actually creates another, in my eyes more drastic problem. I would rather put up with having to release the Alpha Lock key when using the joystick than have to try to touch type in a filename while holding down the Shift Key.

I have explored the matter using the schematics for the TI console to see if there was an immediate solution for this but can find none. Placing the recommended diode into the circuit trace has the same effect as simply cutting the trace, thereby disabling the Alpha Lock switch. I can see no other need for the use of the diode other than as a channel for current flow.

So consider yourself warned. Unless you wish to be bothered with loosing the handy "Capitals Only" feature of the Alpha Lock key don't try this modification. At least until it has been explored a little deeper.





MSP 99 Calendar of Events

APR 21: DataBase Time -- This month we'll be looking closely at Navarone's DataBase Manager software package as seen through the eyes of a regular user Steve Gonnella. If you've ever had the need for a good data base you'll want to be there for sure for this one.

MAY 19: XB II -- Tonight we get a peek at the new Extended BASIC II from Myarc. We'll also be raffling off a TI99/4A computer console to some lucky person attending this month's meeting, so be there!

JUN 16: To Be Announced. (That means if there is anything in particular that you would like to see happen at one of our meetings, then let us know so we can fill this space.)

Subgroup Meetings

ASSEMBLY GROUP -- 1st Tuesday of month, 7:00 p.m.
Bryant Community Center
Bryant Ave and 31st St.

BUSINESS and APPLICATION SIG
Call Dick Clemetson (926-8083)

EDUCATION -- At monthly meetings

YOUTH GROUP - At monthly meetings

Committee Chairs

EQUIPMENT -- George Madline
(784-2395)

NEWSLETTER -- Gary Gese
(529-3989)

PUBLICITY -- Dave Wunderlin
(544-8266)

SOFTWARE -- Steve Gonnella
(533-8494)
6281 Winnetka Ave
Brooklyn Park, MN 55428

YOUTH GROUP --
Ed Johnson (690-3442)
Gordy Myers (377-6713)

*** ELECTION RESULTS ***

Well, it's all over but the shouting. This years election for the MSP 99 officers drew a large crowd to the March meeting (or was it the hardware modifications that were being done?). All those attending voted, but very few mail-in ballets were recieved. And, since there were no other candidates nominated in opposition, the results were somewhat predictable. The past officers will all be serving a second term in office.

For the unawares, they are:

Dick Lauhead - President

George Madline - Vice President

Gary Gese - Secretary

Mark Tellevik - Treasurer

I hope that everyone will continue to give us all the support and encouragment we've seen in the past. Thank You.

RANDOM ACCESS

By Dick Lauhead

"A Fix for the Horizon RAM Disk"

I have owned a Horizon RAM Disk for many months now, and it has become indispensible to me. I use it mostly to keep my "permanent" files handy for fast loading. Used in this manner, it has served me well. There is, however, a bug in V04 (and earlier versions) of the operating system which renders the disk useless for saving non-program type files if you have 32 files on the disk. If you have 32 or more files, the saving of files becomes 4 times or more slower than saving to a floppy disk! I am absolutely astounded that this quirk seems to have gone unnoticed by everyone. I have read dozens of reviews of the Horizon RAM Disk, and have yet to come across any reference to this problem. Perhaps it HAS been written up, but I have never seen a fix for the problem. Am I the only one who has more than 32 files on my disk?

Several months ago now, I wrote to Horizon pointing out the problem. To date I have heard absolutely nothing from them, and have no idea if they are working on the problem, have fixed it and not let me know, or have just ignored it. Therefore, I took it upon myself to track down the problem and fix it.

After spending 10-15 hours narrowing down the problem (thank God for Explorer!), I finally traced it to the code in the status routine in the DSR. I will not bore you with all the details, but suffice it to say that the code for checking for disk full was extremely inefficient. Since the first 32 sectors (after sectors 0 and 1) are reserved for file descriptor records, an empty sector was always found in that area if you had less than 32 files on disk; therefore the inefficiency was not noticeable. However, when all those sectors filled up, the next available sector was a much higher number because it was at the end of all the data on the disk. Then the inefficient code became very noticeable.

The changes described here to fix this problem are relatively simple, but the process of reassembling everything is not. It is impossible to reassemble the new DSR file with one SS/SD disk drive. You must have at least 2 SS/SD drives, or preferably a DS/DD drive to hold all the necessary files. (You can, of course, use the RAM disk also.)

The reassembled DSR alone is 120 sectors long, not to mention the source files! Therefore, if you do not have the equipment to undertake this project, or simply don't want to attempt it, I will send you the UTIL1, and the new PARTA_04, and PARTB_04 files. These files load all the routines into the RAM disk operating system memory. Just send me a disk and a couple dollars for postage and copying. Send it to the MSP99 P.O. Box. If you want all the changed source files, it will require 2 disks.

Before attempting this modification, please read through the entire procedure. DO NOT make these changes to your original disks. Make backups! The files are scattered across all 3 disks supplied by Horizon. I want to stress that these changes have been tested for V04 only. Some of the changes are necessary only to free up memory. There are not enough bytes of memory left in the 6K DSR space unless those changes are made. Be sure to unprotect all files, and save them with the same name after making the changes. These changes will work with all configurations of the RAM disk, including the 256K version. It is assumed that the 256K changes have been made already if you have a 256K disk. To begin, make the following changes to the file called PARTA.

Insert before CLR @FLAG1 (7 lines before label USED):

```
      DECT  @FLAG1
      JEQ   UP3
```

Insert after JMP UPRTN (1 line after label USED):

```
UP3   CLR   @FLAG1
      MOV   @FORSEC,R6
      AI    R6,7
      SRL   R6,4
      SETO  R7
UP4   C     *R8+,R7
      JNE   UPRTN
      DEC   R6
      JNE   UP4
      JMP   USED
```

Change LI R4,0 (2 lines after label SEARCH) to CLR R4.

Delete C R6,@ZERO (4 lines after label SRCH1).
Change LI R1,1 (at label SRCHGD) to MOV @ONE,@FLAG.
Delete the line MOV R1,@FLAG (one line after label SRCHGD).
Delete C R6,@ZERO (5 lines after label ADJST1).
Change LI R1,1 (7 lines after label ADJST3) to MOV @ONE,@FLAG.
Delete MOV R1,@FLAG (8 lines after label ADJST3).

Make the following changes to the file called PARTB.

```
Delete LI R10,>FF00    and
       MOVB R10,@FAC+3
(7 and 8 lines before label INP5).
Change  LI R2,>FFFF  to  SETO R2
(4 lines after label INP5).
Delete NOP (7 lines after label
OUTE).
```

Make the following changes to the file called PARTC.

```
Insert after MOV @ONE,@FLAG1 (1
line after label OP09L2)
INC @FLAG1.
Delete INC R0
       C   R0,@FORSEC
       JNE OP09L2 (6, 7, and 8
lines after label OP09L2).
Change MOV @ZERO,@INFO+8 (10 lines
after label OP00M8) to CLR @INFO+8.
```

Make the following change to the file called CALL/S.

```
Change  VRWA EQU >4290  to
        VRWA EQU >42B0.
```

Make the following change to the file called DOWNLD/S.

```
Change  VWWA EQU >428C  to
        VWWA EQU >42AC.
```

Make the following changes to the file called XB/S.

```
Change  VRWA EQU >4290  to
        VRWA EQU >42B0.
Change  VWWA EQU >428C  to
        VWWA EQU >42AC.
Change  INPT EQU 462A   to
        INPT EQU >4632.
          >463A for 256K cards).
```

Make the following change to SVXB/S:
Change LI R1,>3FC0 to LI R1,>3FB8 (3 lines after label SVL1).

Now use the E/A editor to create the following source file:

```
    COPY "DSK1.PARTA"
    COPY "DSK1.PARTB"
    COPY "DSK1.PARTB"
    COPY "DSK1.PARTC"
    COPY "DSK1.PARTD"
    COPY "DSK1.PARTE"
```

Save this file as COPY/S. Note that the files PARTA-E can be on different disks. In that case change DSK1 to whatever disk the file is on. The COPY directives start in the first tab column (hit FCTN/7).

Now that all the source files have been modified and saved (you did remember to save them didn't you?), you must assemble them. Assemble the following files: CALL/S, CHAR/S, COPY/S, CREATE/S, DOWNLD/S, LOADER/S, SVXB/S, and XB/S. For object file names, use the same name minus the /S (e.g., CREATE, or CALL), except use the object file name NEWDSR (DSR256 for 256K cards) when you assemble COPY/S. All files must be assembled uncompressed (assembly option R only) except LOADER and CREATE which can be compressed (assembler RC options). CHAR/S, CREATE/S, and LOADER/S, were not modified. Use the original files supplied by Horizon. COPY/S copies the files PARTA-E, so make sure those files are online on the specified drive.

You are now ready to load the new DSR into your RAM disk. You will not lose any files on the RAM disk by doing this, if all goes well. In other words back up the RAM disk files! Load the file called LOADER with E/A option 3. It auto starts. Then enter DSK1.NEWDSR (assuming the file is on disk 1). Use the name DSR256 for 256K cards. Note, DO NOT load any of the files from the RAM disk using LOADER!!! You must load them from a floppy drive. Enable block 1. The DSR will be loaded. You should see that the next available DSR address is >57F8 (>5800 for 256K cards), and the 2K block address is >5EBA. If not, you have entered something wrong or have a different version of the operating system files. Now run LOADER again and load CALL into block 2. Repeat the process loading CHAR into block 3, and DOWNLD also into block 3. Now run the following BASIC program (you MUST use Extended Basic):

```
100 CALL INIT
110 CALL LOAD("DSK1.XB")
120 CALL LOAD("DSK1.SVXB")
130 CALL LINK("SVXB")
```

Your new DSR and all the other routines have now been loaded. Test the disk to make sure everything works. If so, you can

(Continued on Page 16)

Poor Richard's Peripheral Roundup

By Dick Dunbar

MODELING FLOPPY DRIVES: Since we have been dwelling on hard disk drives recently, not a whole lot of attention has been paid to the floppy disk drive situation. In particular, it has been a long time since I gave you an update on the brands and models that will function correctly with the TI99. You should know that the TI99/4A uses disk drives that are said to be "IBM" compatible. Once upon a time, that was just about all that needed to be said, but with the advent of the IBM AT and it's 1.2 Megabyte flopply drives, that is no longer the case. What you must look for now when you go shopping about for disk drives are the IBM compatible 360K variety. Disk drives have become so standard a commodity that many times the brand names are not even given in ads, much less the model numbers. But many dealers still show the model numbers in their ads, and you may run across good deals on disk drives where you need to know model numbers, such as in a surplus or second hand store. So, as an aid to help you in selecting a disk drive for your system, I have put together a list of brand names and model numbers of drives that I have been able to identify as useable with the TI99/4A.

All the drives that fall in the "SSDD" (Single Sided Single Density) and "DSDD" (Double Sided Double Density) columns in the following table are usable with the TI99 with no qualifications. If you have the old standalone controller, which does not handle double sided disks, you can still use double sided drives, but only using one side. However, if you don't ever plan to upgrade, single sided drives can be had at very low prices.

I have also included some drives in the "DSQD" (Double Sided Quad Density) and "Othr" columns that are NOT usable with the 4A, so that you may be able to determine if a particular drive you see advertised fits that category.

The ones in the "DSQD" column are 80 track drives. Actually, they will work in the TI, but unless you have the 80 track Eprom set you will only be able to use 40 of the 80 tracks, you won't be able to read disks written on 40 track drives, and 40 track drives won't be able to read disks written on the 80 track drives. If you do have the 80 track Eprom set, you MAY be able to read 40 track disks on the 80 track drives, but 40 track drives WON'T be able to read your 80 track disks, and very likely won't be able to read 40 track disks written on the 80 track drives either. So, if you decide to go for 80 track drives, be sure to keep one 40 track drive around for reading commercial program disks or disks you get from friends, users' groups and other outside sources, and for writing disks you want to give to friends.

Those drives that fall in the "Othr" column will NOT work with the TI (except for the MPI 91S, which would fall in the "DSQD" column except that it is only single sided). These drives either run at a different RPM or are otherwise unsuitable for TI99 use.

The table is organized alphabetically by primary brand name (some drives are sold under various brand names) and model number. Portions of the model number enclosed within parentheses are frequently omitted in ads. They may reflect slightly different models which are equivalent as regards TI usability.

Brand/Mdl	Hgt	SSDD	DSDD	DSQD	Othr
Canon					
MDD210	2/3		DSDD		
CDC					
(MPI)					
9409	F		DSDD		
9409T	F			DSQD	
9428	H		DSDD		
(B)51	F	SSDD			
(B)52	F		DSDD		
(B)52S	F		DSDD		
(B)52SA	F		DSDD		
(B)91S	F				SSQD
(B)92SA	F			DSQD	
Chinon					
502L	H		DSDD		
506				DSQD	
Fujitsu					
(M)2551A	H		DSDD		
(M)2553A	H				1.2M
(M)2557A	H				1.2M

Mitsubishi					
(M)4851	H		DSDD		
(M)4853	H			DSQD	
Panasonic (Matsushita) (National)					
(JA)551-2N	H		DSDD		
(JU)455	H		DSDD		
(JU)475	H			DSQD	
Qume					
142	H		DSDD		
542	F		DSDD		
REMEX					
FDD211	2/3		DSDD		
Shugart					
SA400(L)	F	SSDD			
SA405(L)	F		DSDD		
SA455(2)	H		DSDD		
SA460	F			DSQD	
Tandon					
TM50-1	H	SSDD			
TM50-2	H		DSDD		
TM55-1	H	SSDD			
TM55-2	H		DSDD		
TM55-4	H			DSQD	
TM65-2(L)	H		DSDD		
TM100-1	F	SSDD			
TM100-1(A)	F	SSDD			
TM100-2	F		DSDD		
TM100-2(A)	F		DSDD		
TM100-4	F			DSQS	
TEAC					
FD55A	H	SSDD			
FD55B(V)	H		DSDD		
FD55E	H		DSDD		
FD55F(V)	H			DSQD	
FD55G(F)(V)	H				1.2M
Toshiba					
FD5445	H		DSDD		
(ND)04D	H		DSDD		
(ND)04DE(G)	H		DSDD		
(ND)08DE(G)	H			DSQD	
WS (Western Storage ?)					
FDD211-5	2/3		DSDD		

You may see these drives referred to in slightly different ways, and you may find others that are usable - there are many more than I have listed here. I wasn't able to determine model numbers for a number of brands. The important things to remember are that drives suitable for use with the TI99 will be designated in ads as 40 track or 48 TPI (Tracks Per Inch) drives, either single or double sided, compatible with IBM PC or XT 360K use (NOT 1.2M and not 720K except as noted for 80 track drives). They are easy to find advertised in various computer magazines. I highly recommend either the Computer Shopper or Byte magazine for advertisement shopping. In fact, the majority of the model numbers listed in this table can be found in one form or another in ads in the past two issues of these two publications.

A REAL FORTRAN COMPILER: I could not believe my eyes. As I was browsing through the latest TENEX catalog, I ran across an ad for a FORTRAN Development package. To someone who has programmed in the mainframe environment for as long as I have, FORTRAN is a familiar old friend, one that I had not expected to see anywhere around the TI99/4A.

The price was $49.95, and the company name was not one with which I was familiar. While $49.95 is cheap in the IBM world, it is toward the high end of the TI software price scale, and I was a bit reluctant to take the chance - but (as you may have guessed by now) I finally ordered it anyway. My better half contends that is what usually happens - she may be right. In any event, it has now arrived and I have had an opportunity to give it a cursory trial, although not an exhaustive workout. What follows, therefore, is more like a first impression rather than a thorough review. After a little more experience with the product, I'll return with a more complete review.

The product in question is 99 Fortran, put out by LGMA Products, and it is a complete system consisting of a text editor, compiler, linker, loader, debugger and object library.

The system comes complete with a large manual - one that explains how the 99 Fortran system works, and defines terms well enough so that a person with some familiarity with SOME language should be able to figure out how to put together, compile and execute a program written in Fortran. Unlike most manuals that you get these days, it is NOT full of spelling, grammatical and typographical errors. It does have one consistent spelling error - the word "separate" is consistently misspelled "seperate" - but that's not too bad.

The language is a subset of ANSI 1977 Fortran (NOT Fortran IV as the Tenex catalog states - Fortran IV is based on an earlier ANSI standard). It lacks several of the features that are found on large mainframe computer versions, but plenty is included - more than enough to make you feel like you are using the genuine article.

The most notable lack is character variables, but this is not a major annoyance for old time Fortran users, since the pre-'77 versions never had them anyway. Even for newcomers to Fortran, the small word size coupled with the logical functions available in this implementation make character handling easy even without special character data types. (For those of you not familiar with Fortran, character data is handled using integer variables and arrays when character data types are not provided.)

Just about anything you want to do with your TI99/4A can be done with this system. Some of the features that are supported are integer, real (floating point), double precision and logical data types, IF/THEN/ELSE/ELSEIF/ENDIF as well as DOWHILE/ENDDO structured programming statements (in addition to the normal Fortran IF and DO statements), full multi-dimensional array handling, extensions to the language to simplify screen handling, extensions to the Fortran object library to support all the CALL statements for graphics, sprites, sound, etc. that are supported by Extended Basic, plus many things that are not (40 column screen mode, for example), and the list goes on. I am very impressed with the range of features supported by this compiler.

Now that we have determined that it SOUNDS like a good product, let's look at how well it delivers.

EDITOR: The editor does what it does very well, with no apparent bugs. However, what it does is not that much. All it supports are the basic editing functions of inserting and deleting characters and lines, saving and loading files, purging the current contents of the editor, and a display of statistics, such as how much space is left in the memory buffer, an estimate of how many lines of code that will accomodate, etc. It appears to have one fixed tab at column 7, with no provision for changing tab settings, and does not support such things as moving blocks of lines from one place to another or locating and/or replacing strings. These omissions, while not fatal, are serious. One of the very first things I had to do to compile and run one of the sample programs (a simple spreadsheet program) was to change a number of statements in the source file to reference a different disk file name than the one used in the source code. With a replace string operation this is simple. Faced with having to replace numerous occurrences of the same string one by one, I opted to use the Editor/Assembler editor rather than the 99 Fortran editor.

COMPILER: I found no major bugs in the compiler, although I question some of the ways certain things were implemented. For example, the program listing file is a D/V120 file rather than the normal D/V80, for no reason that I can see. This is not a problem if you always put your program listings directly to the printer, but I normally put my listing on a disk file, then print the disk file later. But most utilities to print files look for D/V80 or D/F80 files, not D/V120.

Error detection and diagnostics need a little work. On a program I put together myself, all the errors that I made were detected, but the error messages were not always on target or easily interpreted. In one instance, I used the number of a CONTINUE statement in a formatted WRITE instead of the number of the FORMAT statement that I had intended. I was given an error indication at compile time, but no clue as to where the error was. I had to try executing the resulting program before I got the diagnostic, and even then it complained about the structure of the FORMAT statement, not that it wasn't one at all. In all fairness, I have seen mainframe compilers that did no better.

One strange thing I noticed while compiling the sample program supplied with the compiler was that whenever the compiler was directed to INCLUDE a section of code from a secondary file, the activity lights on each of my floppy disk drives would light up one after the other in order, indicating to me that a file search was in progress. The file in question was located on my hard disk, and was in fact found and included each time. The reason I found this odd is that when a file search is done in the normal sequence, the hard disk is searched first, so the floppy drives are never referenced. Obviously the 99 Fortran compiler does not use a standard DSRLNK routine. I intend to ask LGMA Products about this as well as the other things I have run across.

LINKER: The linker did as advertised, loading the object code produced by the compiler, and searching the object library for those routines which were called by the program and loading the required routines. The object library is organized simply and effectively, and can be expanded with your own object library routines if you so desire. It seems to be limited only by the size of your disk.

OTHER: The programs compiled ran and executed well. Preliminary indications are that calculations are very fast, while I/O is not outstanding, but I have not really had a chance to make an evaluation. There are provisions for making what amounts to the equivalent of an absolute program file, but it seems that these files must be loaded using the Fortran system rather than, for example, Editor/Assembler option 5. Again, I haven't had a chance to check this out to be sure I am interpreting things correctly.

I also did not get a chance to try out the debugger yet, but if it works as advertised, and based on what I have seen so far I have no cause to doubt it, it should be very useful. You can set breakpoints (using symbols such as Fortran statement label, variable name or line number), inspect and alter memory, inspect and alter variables (by name), inspect and alter workspace registers, do hexadecimal arithmetic, and so forth - a lot like the Editor/Assembler debugger, but altered for the Fortran environment.

SUMMARY: My preliminary assessment is that 99 Fortran is a good buy for those interested in having and using a Fortran compiler. It works, is well documented, does not

(Continued on Page 16)

DSR's.............cont. from Page 1

anyone ever got it right the fisrt time?). While testing the XB program, we found that all of a sudden, at unpredictable times, the screen image table and/or the pattern descriptor table and/or the color table and/or who-knows-what-else, decided to go out to lunch! The program seemed to execute normally after bombing out those tables, but we just had a hard time reading what was on the screen.

I brought up this problem at our February U.G. meeting to Dick Lauhead and Dick Dunbar. Both of them had heard of this problem before, and Dick Dunbar had run into the problem personally when he was working on his load interrupt screen dump program. He suggested that I save the entire 256 bytes of scratch pad RAM before the DSR call and restore it before returning to XB. Apparently something in the PAD area was causing the trouble.

He also mentioned that he remembered seeing a reference to this very problem in a file he downloaded from GENIE. He graciously offered to dig the file out and send me a copy. While I was anxiously awaiting it's arrival, I decided to try out his suggestion of saving and rstoring the PAD. Well, I had absolutely no luck with that method. In fact, my problems got worse. I then remembered that there was a utility to allocate string space in VDP. I dug out my E/A manual and started digging. The manual said that the routine was not only useful for reserving string space, but was also good for reserving space for PABs and buffers. SUPER! The solution seemed to be right in my fingertips. I added the necessary code and tested it out. I was once again very frustrated to find the same problem - only now it was happening sooner and with more frequency than before (Grrrr...)!

I was about to give up when Dick Dunbar's letter arrived. As I read the information he sent, I began to "see the light". Rather than paraphrase what Dick sent me, I decided to type it up for you. I wish I could give credit to the author, but my copy has no reference to his identity (can you help me out again Dick?). What follows is the article I received. I hope the article and the routines provided will save some of you the aggravation and frustration I went through.

****

The following are lists of two methods of implementing the DSR, GPL, and CIF (Convert Integer to Floating-point) routines in Extended Basic. These routines are not included in the package of routines loaded into low memory by the CALL INIT command. First is an Assembly program list that implements the routines. The second is a list of X-Basic CALL LOAD pokes that will load the routines into low memory.

EQU statements needed for BLWP to these routines are:

```
        DSRLNK EQU  >24F4
        GPLLNK EQU  >24F8
        CIFLNK EQU  >24FC
```

The DSR and GPL routines work as documented in the E/A manual. The CIF routine provided, like the routine loaded by the E/A cartridge and called through XMLLNK, takes the one-word integer value at the FAC (Floating point ACcumulator) at >834A and returns the converted result at FAC. Unlike the XMLLNK-provided routine, this routine should NOT have a "DATA >2300" statement following the "BLWP CIFLNK" statement.

The DSR and CIF routines were both created from dis-assembly listing of the routines provided by the E/A cartridge. The GPL routine was picked up many months ago on a bulletin board (I can't remember which one), and appears to work - although the specifics of HOW it works is a mystery to me.

WARNING!! Handling your own DSR PABs (Peripheral Access Blocks) and buffer areas appears to be VERY dangerous in the XB environment. You can very easily destroy data in VDP memory currently in use by the XB interpreter. One way of handling this is to OPEN the file in XBasic and use the value at >833C to find the PAB and buffer area allocated by the file (See page 302 of the E/A manual). Or, save VDP memory you're going to use to RAM and then restore it to the original value after the DSRLNK.

Also, be advised that that the Get String Space routine provided via the GPLLNK appears to not work in the XB environment. Eventually (usually after the first "garbage collection" is forced), VDP memory in use by the XB interpreter will get corrupted and (as we frequently say), "results may be unpredictable".

The following code may be used as a stand-alone code file, or, you may include the Assembler source statements in your own Assembly program. If you choose the second option, strip out the DEF and AORG statements at the top, and the FIXFFA routine at the bottom (last five lines of the listing).

```
        DEF  FIXFFA
        AORG >24F4
GPLWS   EQU  >83E0
UTILWS  EQU  >2038
VSBR    EQU  >2028
FAC     EQU  >834A
GRMRA   EQU  >9802
GRMWA   EQU  >9C02
PABADD  EQU  >8356
PAD     EQU  >8300
SUBSTK  EQU  >8373

DSRLNK  DATA UTILWS
        DATA DSRBGN

GPLLNK  DATA UTILWS
        DATA GPLBGN

CIFLNK  DATA UTILWS
        DATA CIFBGN

FNAME   TEXT '----------'
VAL1    BYTE >20
VAL2    BYTE >AA
VAL3    BYTE >2E

DSRBGN  MOV  *R14+,R5
        SZCB @VAL1,R15
        MOV  @PABADD,R0
        MOV  R0,R9
        AI   R9,-8
        BLWP @VSBR
        MOV  R1,R3
        SRL  R3,8
        SETO R4
        LI   R2,FNAME
DSR1    INC  R0
        INC  R4
        C    R4,R3
        JEQ  DSR2
        BLWP @VSBR
        MOVB R1,*R2+
        CB   R1,@VAL3
        JNE  DSR1
DSR2    MOV  R4,R4
        JEQ  DSR10
        CI   R4,>0007
        JGT  DSR10
        CLR  @>83D0
        MOV  R4,@>8354
        INC  R4
        A     R4,@PABADD
        LWPI  GPLWS
        CLR   R1
        LI    R12,>0F00
DSR3    MOV   R12,R12
        JEQ   DSR4
        SBZ   >0000
DSR4    AI    R12,>0100
        CLR   @>83D0
        CI    R12,>2000
        JEQ   DSR9
        MOV   R12,@>83D0
        SBO   >0000
        LI    R2,>4000
        CB    *R2,@VAL2
        JNE   DSR3
        A     @UTILWS+10,R2
        JMP   DSR6
DSR5    MOV   @>83D2,R2
        SBO   >0000
DSR6    MOV   *R2,R2
        JEQ   DSR3
        MOV   R2,@>83D2
        INCT  R2
        MOV   *R2+,R9
        MOVB  @>8355,R5
        JEQ   DSR8
        CB    R5,*R2+
        JNE   DSR5
        SRL   R5,8
        LI    R6,FNAME
DSR7    CB    *R6+,*R2+
        JNE   DSR5
        DEC   R5
        JNE   DSR7
DSR8    INC   R1
        BL    *R9
        JMP   DSR5
        SBZ   >0000
        LWPI  UTILWS
        MOV   R9,R0
        BLWP  @VSBR
        SRL   R1,13
        JNE   DSR11
        RTWP
DSR9    LWPI  UTILWS
DSR10   CLR   R1
DSR11   SWPB  R1
        MOVB  R1,*R13
        SOCB  @VAL1,R15
        RTWP


GPLBGN  MOVB  @GRMRA,R1
        SWPB  R1
        MOVB  @GRMRA,R1
        SWPB  R1
        AI    R1,-3
        MOVB  @SUBSTK,R2
        SRL   R2,8
        AI    R2,PAD
        INCT  R2
        MOVB  R1,*R2
        SWPB  R1
        MOVB  R1,@1(R2)
        SWPB  R2
        MOVB  R2,@SUBSTK
        MOVB  *R14+,@GRMWA
        MOVB  *R14+,@GRMWA
        MOV   @>2000,R4
```

```
        LWPI  GPLWS
        RT
GPL1    LWPI  UTILWS
        MOV   R4,@>2000
        RTWP

RADIX   DATA  100
CIFBGN  LI    R4,FAC
        MOV   *R4,R0
        MOV   R4,R6
        CLR   *R6+
        CLR   *R6+
        MOV   R0,R5
        JEQ   CIFRTN
        ABS   R0
        LI    R3,>0040
        CLR   *R6+
        CLR   *R6
        CI    R0,100
        JL    CIF2
        CI    R0,10000
        JL    CIF1
        INC   R3
        MOV   R0,R1
        CLR   R0
        DIV   @RADIX,R0
        MOVB  @UTILWS+3,3(R4)
CIF1    INC   R3
        MOV   R0,R1
        CLR   R0
        DIV   @RADIX,R0
        MOVB  @UTILWS+3,2(R4)
        MOVB  @UTILWS+1,1(R4)
        MOVB  @UTILWS+7,*R4
        INV   R5
        JLT   CIFRTN
        NEG   *R4
CIFRTN  RTWP


FIXFFA  LI    R0,EOF+2  FIX-UP FIRST
                        FREE ADDRESS
                        VALUE
        MOV   R0,@>2002 AFTER LOAD,
                        DO A CALL
                     LINK("FIXFFA")
        CLR   R0
EOF     RT

        END
```

I have left out the XB CALL LOAD listing in the interest of conserving space (and my eye-sight!). If anyone would like a copy of that listing, just let me know.

Ed Johnson
399 S Lexington Pkwy
St Paul, Mn 55105
(612) 690-3442

Editors Note: For those of you that didn't know, Ed is collaborating with Donn Granros on a new graphic adventure game. Watch for it.

TIPS 'n Thoughts

by Tom Fairbairn

In others of these columns I have referred to the use of control documents. I used one as an example in the column on making "programmable function ID strips."

I really believe that if you are using the TI-Writer program for any work more involved than one-page letters, and you are not using the FORMATTER print capability, you have thrown away your investment in TI-Writer. The Formatter commands are a very powerful tool that can make your work much simpler.

The Formatter commands, along with preset command documents, can help out your work in these ways.

- The use of the commands in preset control documents allows using the Formatter commands without reentering them every time.

- All documents prepared under any given control document will have the same general appearance (all margins, tabs, etc. will be uniform from one production document to another).

This last means, for example, that all my manuals have the same family appearance, and all my columns are formatted the same way.

- I don't have to worry about special programs that I must run under BASIC or X-BASIC to set the printer up for a printout. When I print under the Formatter, that is all taken care of for me. Using the Formatter allows me even to change printer line spacings and type face styles on the fly if I wish.

What I have done is to create a set of control documents that reside on my TI-Writer working master. When I start to write a document, I first load in the appropriate control document, which I call a "ruler document." The ruler document contains all the formatter controls I will normally use when I am entering a new document.

To illustrate this, let's look at the ruler document I use for most of my needs. I carry it on the TI-Writer diskette as RULER1.

```
.CO RULER1; 54 PRT LINES;TAB=5
.CO STANDARD DOCUMENT
.LM 10;RM 75;PL 64;NF
.CO SET PTR RIGHT MARGIN
.TL 92:27,81,76
\
.CO SET {=EXP PRINT:
.TL 123:27,87,1
.CO SET }=NORMAL PRINT:
.TL 125:27,87,0
.CO SET [=ALT CHAR MODE:
.TL 91:27 62
.CO SET ]=NRM CHAR MODE:
.TL 93:27,35
.CO ENBL HIGHLITE PRT:
.TL 92:27,69
\
.CO SET _=START UNDERLINE:
.TL 95:27,45,1
.CO SET `=END UNDERLINE:
.TL 96:27,45,0
.HE
.FO         /--/      %
```

Let's see what the controls do for us.

The first two lines of the ruler document identify which one it is. I sometimes slip a cog and call in the wrong one. Once I have gotten the right ruler into the system, I use this space to identify the new document, and add a date and revision number. This makes it easy to keep track of updates to manuals and that sort of business.

The next line sets the left and right margins in the formatter, tells the formatter how many lines to use before page ejecting, and turns on the FILL or word wrap feature. However, keep in mind that these commands affect the FORMATTER only, and do not affect settings that are controlled within the logic of the printer. The printer commands must be handled separately using other means.

I have found that if I do not tell my printer what the right margin setting should be, it sometimes causes the right margin to be messed up, and the hyphenations are upset. So I make sure the printer margins match the right margin setting of my document ruler. The comment shows where this is done in the ruler document. The command:

```
 .TL 92:27,87,1
```

causes the system to watch for the ASCII character 92 (\), and when it sees this character to replace it with the ASCII string 27,81,75. The corresponding characters are ESC, Q, and the column number of the right margin setting. The next line "prints" the information out to the printer right away. Doing this means I can reuse the character code for other purposes later, as you will see.

The next four lines cause the left and right brackets to start and end expanded mode printing, used when I want to write headings or chapter names or to emphasize something.

The next four lines enable the use of braces to start and end the Alternate Character mode. This mode permits me to embed telecommunication control characters into the text. This can be useful under some conditions when I am formatting a print file to the disk that I will later transmit using the terminal emulator.

To enable the highlight print mode I once again call on the character \. Because it will already have done what I needed earlier, I will reassign it to a new meaning. For most documents I create with this ruler, I use either 10 characters per inch pitch (normal) or 5 characters per inch (expanded). The highlight, or emphasized, format causes the printer to run at half speed and strike each dot twice. This fills in the spaces between the dots horizontally, and yields a much more attractive printout that copies extremely well in reproducing a manual or pamphlet. Once again, I "print" the character immediately, to set up the printer and leave the character available for other purposes.

Then I use another pair of .TL commands to enable the full underline capability of the Gemini. If you do a normal underscore, you have to type the underscores, and then do a backspace over the whole character group and retype the characters you want underlined. TI-Writer won't allow multiple characters in any single character cell (few word processing systems that I am aware of will permit this), so you have to tell the printer that this letter, word, or group of words is to be underlined. I transliterate the underscore character (which won't print under these conditions) to cause the Gemini to create a full, continuous underline until I tell it to stop with the transliteration of the ` character. This gives a nice, unbroken underline, by the way, unlike the normal broken line underscore that you see on dot matrix machines.

The .HE command enables the header insertion function. If I make an entry in this line after the command, it will be printed on the top of every page. If I leave it blank, the command has no visible effect. You have to be careful with this command, because the first column of a header is in the column 0 position of printout, regardless of your tab ruler setting for left margin. You have to space over the width of your margin before entering the first header character; this means the width of the header on your page appears to be reduced because you can't go past the right margin. This is where my third row on the TI-Writer strip saves the day: it reminds me that holding the CTRL key and pressing Y allows the cursor to be moved all the way left to column 0 despite the left margin setting (left margin release), and permits the heading to be printed just about full width.

The final line in the ruler document, the .FO, is set up in the original ruler to place the page number centered horizontally (or almost so, at least). The percent sign used in either a header or footer tells the Formatter to insert the current running page number at that location. By using the footer, all my documents automatically have consecutive page numbers put on the bottom center. If I do not for any reason want the numbers, I can delete the % sign to turn them off and there is no other visible effect.

I have several other ruler documents that I use. Whichever one I need, I simply read it in before I start entering my new document, and everything is set up for printer controls, tab settings, and margins in order to complete the printing of the document just as I want it.

Now, I'd like to have you try something. Set up a ruler document with a tab ruler showing the left margin at column 0 and the right margin at column 63. Set up formatter margin commands using printer commands to print at compressed pitch (17 characters per inch) with a right side printer margin of 128. Enter any text that you wish to try, and watch to see if your printer actually will print

128 columns!! You can use the same method to print full width on a Gemini 15-X or any other 14 5/8 printer with the capability. So TI-Writer comes through again, since it is not restricted to 80 columns of printout!! While the screen can not be convinced to go that wide, you can get a pretty good idea where you are if your margin and tab ruler is set to 1/2 of your expected line width. With this setup, each two screen lines becomes one print line, and you can even hyphenate properly for the wide document.

While it would be a long job, it is possible that you could take a two-page Multiplan printout, and by alternating the reading of a line at a time from the two pages, create a single wide document to print out your spreadsheet. Some of you who enjoy it might like to come up with a machine language or X-BASIC program to perform this surgery much less painfully.

As a matter of fact, I believe it is possible to set up the printer and formatter to work with as many as 255 characters per line (wide paper, condensed print) providing you turn on the control in the printer that will allow it to treat all eight bits of the character byte as data. My printer can't do quite enough to handle this, so I have not been able to try it.

Notice that in the sample ruler I have used here, the backslash (\) character is still available to be redefined for any other purpose I may need. At any time in the work, I can give a new definition to this character. As soon as I transmit it, it is once again available for use, or I can give it a standing definition for something I need to do repeatedly.

Why do I call it a ruler document? In truth, the term "ruler" is a carryover from my word processor at the office. However, it seems very fitting. How else would you explain about something that gives absolute directions to everything else? It is just like an absolute monarch of all it surveys. Hence, ruler.

I think now you can appreciate the power of the Formatter in this word processor package. There are some other mighty neat things you can do with it also, but I'm wearing out my welcome for this issue.

RAMBLINGS AND STUFF

by Steve Gonnella

From the past underwhelming response I've recieved about my comments (both good & bad) on Navarones Data Base Manager, I believe I'm the only one on this planet that uses it. Well, if you use the Personal Record Keeping module (yech), this is better. If you use PR Base (so-so), this is better. If you use something else this is better or I want to see it.

DBM is a disk & cartrige data base with all of the things missing from PRK and is just as easy to use. The entry screen can be designed your way with up to 25 data fields. Each field has its own "aid" screen available for your notes if you want to let someone else enter data. Searches for a certain record take a second or so. Sorts are as fast as PRB with up to 6 sort keys allowed. The report generator is where this program shines. Reports are designed with TI Writer then the data fields can be placed or moved anywhere on the page. Any printer commands recognized by the TIW Editor can be included in the report.

The size of the data base is only limited by disk size using one sector per record. Thats 1440 records on a DSDD disk or tons of records on a hard disk. Even 320 records on a SSSD disk is larger than most home data bases ever get. The manual says 32000 max.

Invoicing, sales and expense records, your friend or business rolodex with comments, inventory for home or business, tax records..... the uses are only limited by your imagination. If you use a data base and want to compare notes or want to learn how they're used, come to the demo at the next meeting or I'll crawl back in my hole and never mention it again.

Help is needed updating the software catalog. If you really want to learn how to use TI Writer, boy do I have a project for you. Unknowing souls please apply in person.

We keep trying to include things that interest you in this newsletter and at the meetings but computer groups are just like computer terminals, they both need input. Your turn. *SG*

XBASIC REQUIRED

From the Jacksonville U.G. in Arizona comes this tip from Ralph Devine. Many of us use BASIC and XBASIC in a truly unstructured way. We build the program as we go, using RESequence to open up lines, etc. When finished, our program LISTs in line number order and "looks" structured. However, it seems that inside our 99/4A marvel our jumbled line order is preserved. When the program is RUN, time is lost looking for the next line to execute.

To prevent this, SAVE the file using the XBASIC MERGE option. It may be a little slower, but it is saving the program in sequential line order in MEMORY IMAGE format. When it is done, type NEW and ENTER, then reload the program using MERGE and reSAVE it as a regular program. This can make an incredible difference in execution time.

****

EASTER CALCULATION

By Dick Lauhead

On what date will Easter fall in 1992? Easter is unique among our common holidays. It is not easily calculated like most other holidays. Easter is the first Sunday following the first full moon on or following the March equinox (which is assumed to ALWAYS be March 21). Furthermore, if the full moon itself falls on a Sunday, then Easter is the following Sunday. Also, the full moon date is calculated by an ancient Ecclesiastical computation which is not the same as the real full moon date. Remember, Easter is a religious holiday which had its calculations set many centuries ago.

Faster may occur as early as March 22 if the full moon occurs on Saturday March 21, or as late as April 25 if the March full moon occurs March 20, and 29 days later (the next full moon) is a Sunday. Easter falls on those dates only once every century or two. It is generally (although not universally) conceded that April 19 is the most common date for Easter. If you are interested in reading more on this subject, refer to "Sky & Telescope" magazine, March 1986 page 294, and to "Celestial Basic", a book of astronomy programs written in BASIC.

The program presented here is a TI99/4A adaptation of Easter calculation programs found in both of the above mentioned publications.

This program asks you to enter a year (1583-2099) for which you wish to calculate the Easter date, and then prints the date. Although it is written in Extended Basic, it will also run in BASIC if you break apart the multiple statement lines. To test the program, try 1987. Easter this year is April 19. When is Easter 1992? Why April 26 of course! I hope you enjoy this program.

```
100 CALL CLEAR
110 INPUT "ENTER YEAR ":Y
120 IF Y<1583 THEN PRINT "YE
AR MUST BE AFTER 1582"::GOTO
 110
130 QC=.05::QX=Y/.8::B1=19::
B2=30::B3=7::Y2=Y/100
140 L=INT((Y/B1-INT(Y/B1))*B
1+QC)*SGN(Y/B1)
150 A=(11*(L+4)+INT(.32*(1+I
NT(Y2))+.2)-INT(.75*(1-INT(Y
2))))
160 E=INT((A/B2-INT(A/B2))*B
2+QC)*SGN(A/B2)
170 IF E<=L/10 THEN E=E+1
180 F=INT(QX)-E+3-INT(.75*(1
-INT(YZ)))
190 F2=INT((F/B3-INT(F/B3))*
B3-QC)*SGN(F/B3)
200 D=26-E-F2
210 IF D<=0 THEN D=D+31::M$=
"MARCH"::GOTO 230
220 M$="APRIL"
230 CALL CLEAR::PRINT "EASTE
R IN ";Y;"IS ";M$;D
240 PRINT :"WANT ANOTHER YEA
R? (Y/N)"
250 CALL KEY(3,K,S)::IF S=0
THEN 250
260 IF K=89 THEN 100
```

****

Here's a little routine that will change the shape of the cursor.

```
1 ! TEX-CURSOR
2 CALL CLEAR :: CALL INIT
3 CALL LOAD(8196,63,248)
4 CALL LOAD(16376,67,85,82,83,79,
  82,48,8)
5 CALL LOAD(12288,48,48,63,255,
  254,124,24,12)
6 CALL LOAD(12296,2,0,3,240,2,1,
  48,0,2,2,0,8,4,32,32,36,4,91)
7 CALL LINK("CURSOR"):: END
```

Random Access.....cont. from Page 5

now create new PARTA_04, and PARTB_04 files by running the program called CREATE with the E/A option 3. This program auto starts and requires no entries. It requires a disk named ROS to be in any drive. After running CREATE, you can now load the entire RAM disk memory by simply running the UTIL1 program at any time. UTIL1, PARTA_04, and PARTB_04 should all be on a disk named ROS. Note, CREATE and UTIL1 must be modified for use with 256K cards. That is beyond the scope of this article. If you want those files send me a disk. Incidentally, while testing all these changes, I discovered that CALL LINK("NF",n) and CALL LINK("DM") do not work with Extended Basic. They do not work on the original system disk supplied by Horizon either. Fixes for those problems are also incorporated in these changes.

-RWL-

Poor Richard......cont. form Page 9

appear to have any fatal flaws, and is priced fairly. It came as a pleasant surprise, and makes me wonder if there are any more things coming to surprise us from LGMA.

LGMA Products
Box 210, RD4
Apple-Butter Hill Rd.
Coopersburg, PA. 18036

TIME AND SPACE: Well, I seem to have come to the end of time and space. So I shall enter my TARDIS and jump off to another continuum until the next issue of this newsletter.

MSP99

```
1 ! ONE LINER UNIVERSAL CALENDAR PROGRAM FOR THE DAY OF THE WEEK FOR ANY YEAR
SINCE 1905 - BY DENNIS HODGSON IN THE SIDNEY NEWS DIGEST.
2 ! INPUT DAY, MONTH, YEAR: FOR INSTANCE- 30,4,1986
100 A=1 :: INPUT D,M,Y :: FOR T=A TO M-A :: H=H+29+VAL(SEG$("20212122121",
T,A)) :: NEXT T :: J=H+(Y/4<>INT(Y/4)AND M>2)+INT((Y-A)*365.25)+D :: PRINT
SEG$("SASUMOTUWETHFR",(J-INT(J/7)*7)*2+A,2):: RUN
```

MSP 99 Users Group
P.O.Box 12351
St.Paul, MN 55112

1-0999 8706

DON SPARISH
1712 BARCLAY ST.
MAPLEWOOD, MN 55109

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY