

MSP 99

USER
GROUP

THE MSP 99 NEWSLETTER

POOR RICHARD'S PERIPHERAL ROUND-UP

By Dick Dunbar

WHAT'S GOING ON HERE: As promised (threatened?), this month's column is devoted primarily to a review of the FOUNDATION 128K Memory Card. However, if you persist and read on to the end, you will find some additional information about their 80 column card as well.

But before we get started, I would like to take this opportunity to request something from you. I would like to be sure that I am covering things which are of interest to you, the readers of this column and users (or potential users) of the hardware about which I write. To that end, I would like to hear from you. Drop me a line, or leave a message for me on the CBC CORKboard, or even give me a call on the telephone, and let me know what you would like to see covered in the column. I won't always be able to do reviews on everything, but I will if I can, and if you are interested.

Dick Dunbar
1229 Maywood St.
St. Paul, MN 55117
(612)488-0153 (Home)
(612)482-2000 (Work)

FOUNDATION 128K MEMORY CARD: Last month I mentioned that I hoped to finally receive the disk emulator firmware for this card. This firmware was originally supposed to be available last summer (1983). Well, I did receive the firmware, and installed it in my 128K card. It came in the form of a 2732A EPROM (Erasable Programable Read Only Memory) which replaced an existing

(continued on page 4)

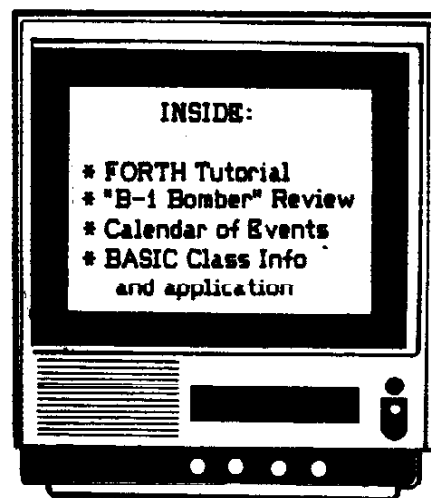
TEXAS INSTRUMENTS TI-99/4 HOME COMPUTER - ONLY \$1150!

INCLUDES 13" COLOR MONITOR!

Did I get your attention? No, I'm not trying to sell my TI home computer. We're going to take a little trip reviewing the price history of the TI-99/4.

The listing above was in a computer store advertisement appearing in the October, 1979, issue of BYTE magazine. To the very early TI-99/4 buyers, this may be an unpleasant reminder of the computer's early cost. However, to those of you who bought their 99/4A during the computer price wars, subsequent to TI's withdrawal from the home computer market, a previous pricetag of over \$1000 should make your \$50 to \$200 investment look pretty good.

(continued on page 14)



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 \$12 a year for a family, \$10 for an individual, and \$50 for a sponsor member. You're welcome to visit a meeting before you join. For more information, call or write us.

USERS GROUP MEETINGS are held the third Tuesday of each month at 7 p.m. at Dunwoody Industrial Institute, 818 Wayzata Blvd., Minneapolis, MN 55403.

MSP 99 USERS GROUP
P.O. BOX 12251
ST. PAUL, MINNESOTA 55112, U.S.A.

PRESIDENT: Joel Gerdeen 572-0148
V. PRESIDENT: Dick Dunbar 480-0153
SECRETARY: Jen Todd 920-3263
TREASURER: Brad Olson 786-1235

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 writer and not necessarily those of the MSP 99 Users Group, its officers, editor, or members. Materials accepted by the editor 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:
Bonnie Burton, 431-6064
Editor's Assistant: Jon Todd
Newsletter Committee Members:
Clarence Brockman
Marilyn McPartlin

DEADLINE FOR NEXT ISSUE: SEPT 18

COMMITTEE VOLUNTEERS

If you want to work on a committee (Education, Equipment, Program, Publicity, Software, Newsletter), or have an idea for a program, contact one of the officers.

COMMERCIAL
ADVERTISEMENT RATES

Business firms that want to communicate with our members may do so by placing an ad in the newsletter. Rates are: Full Page (7-1/2 X 10-1/2) \$40; Half Page (3-1/2 X 10-1/2) \$30; Quarter Page (3-1/2 X 5) \$22. Each ad must be camera-ready in sizes indicated and paid in advance. Inserts (printed by advertiser on 8-1/2 x 11) may be inserted in the newsletter at \$20 per sheet. Contact the editor for information or to reserve space.

CHANGE OF ADDRESS

Before you move, please mail a change of address to the group at the above address.

FROM THE EDITOR:

You could call this a help-wanted ad: newsletter editor needed to produce the MSP 99 monthly publication. The pay isn't high, but the work is rewarding.

Although I have very much enjoyed producing the club newsletter each month, my commitments this fall do not permit me to continue in this capacity. My employer has offered me a rather sudden, but wonderful educational opportunity. Working full time, having a family, and going to school several nights per week won't allow much time for anything else at this point in my life.

So MSP 99 needs a new editor.

I need to find someone interested in assuming this position. Previous experience of some sort (writing, editing, graphic arts) would be helpful, but it isn't necessary. More importantly, we need someone who is willing to commit a portion of time for the next year or so in a volunteer position. I will work with you as far as teaching you the technical aspects of the job.

If you're at all interested, if you think this is something you would like to try, give me a call and let's talk about it.

Bonnie Burton,
Editor

MSP 99 CALENDAR OF EVENTS

SEPTEMBER 18 -- (TUESDAY) 7:00-9:00 COMPUTER FAIR! Hardware and software demonstrations! Vendors selling their wares! The complete MSP 99 software library will be available so you can pick up copies of any program for \$1 each! The special attraction will be an equipment and module auction starring Dick Dunbar as auctioneer! Bring your items you want to swap or demonstrate to other members!

OCTOBER 16 -- (TUESDAY) 7:00-9:00 ADVENTURE NIGHT! Return to Pirate's Isle and find the treasures at this meeting! Host Jon Todd (The Farmer's Dilemma; Cave Maze) will demonstrate and discuss the Scott Adams series, Infocom Adventures and maybe even share his cheating 'method' for Pirate's Isle! Highlight: the youth group will present an adventure game they've written under Jon's guidance.

NOVEMBER 20 -- (TUESDAY) 7:00-9:00 GRAPHICS! Color; banners; Super Sketch; pictures; visual; fun!

GRAPHICS MADE EASY! By Amy DeMars

Until now, drawing graphics has been both difficult and expensive. SUPER SKETCH Graphics Tablet is an easy way to create graphics at a very inexpensive cost of only \$59.95.

The SUPER SKETCH is about the size of a large clipboard. By simply moving a control arm, SUPER SKETCH reproduces your drawing on your T.V. or monitor screen.

SUPER SKETCH can really bring out the creativity in anyone, while introducing them to computers. Sketch free-hand or trace from an original drawing, then fill in colors any way you wish. A starter kit enclosed in the package includes several attractive drawings that can be traced and colored on the screen. If you decide you don't like the colors you chose, you can change them with the push of a button. When your picture is complete, you can then save it to cassette tape.

The manufacturer plans to have other cartridges so you will be able to do many more things with the SUPER SKETCH.

SUPER SKETCH is available at "Specialist In". You may come in for a demonstration or call 544-6219 if you have any questions.

SUBGROUP MEETINGS:

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

BUSINESS--Second Tuesday, 7-9 p.m., Norwest Bank, Hopkins. Call Bob DeMars (544-6219) or Dick Clemetson (926-8083).

EDUCATION--At monthly MSP meetings.

YOUTH GROUP--At monthly MSP meetings. Pres.: Scott Morgan.

COMMITTEE CHAIRS:

- Education--Marilyn McPartlin, 636-5663
Equipment--We need someone.
Newsletter--Bonnie Burton, 431-6064
Program--Dick Dunbar, 488-0153
Publicity--Dave Wunderlin, 544-8266
Software--Ed Neu, 425-8744

(Poor Richard, continued)

EPROM on the card. I had no problems installing the new chip, but I was a little surprised to find that Foundation had forgotten to enclose the documentation on the new features. It was an oversight which was quickly remedied. The documentation arrived two days after I called them to inquire about it. The reason for the oversight was that I disrupted their routine by requesting them to send the EPROM to me to install myself rather than send my card in for them to install it. I couldn't afford to be without the memory long enough to send it in.

The Foundation 128K memory card comes in a fairly heavy sheet-metal (aluminum) case. It fits nicely in the Peripheral Expansion Box (no stand-alone unit is available), but does not have the positioning fins that the TI cards have, so it is best located between two other cards so that it is well supported. Foundation is currently offering to replace the sheet-metal case with a genuine TI clamshell case for those who want it badly enough to pay \$9.95 for it. I didn't feel it was necessary.

This card contains a GREEN pilot light, rather than the amber normally found on PEB cards, making it easy to tell which one it is, if you are interested.

Logically, the card contains four (4) 32K banks of memory. At any given time, one of those banks (program-selectable) is mapped into the TI-99/4A's address space, the same as TI's 32K memory expansion, and serves the same function. That leaves the other three banks to be used to emulate a disk drive, or anything that an assembly-language programmer wants to use them for. If you are adept at assembly-language programming, you may be able to make use of this extra memory in some creative ways, particularly if you have a Mini-Memory module to work with. The reason that a Mini-Memory comes in handy is that the bank-switching mechanism Foundation uses switches all of the available 32K at once, leaving no safe, unswitched place for the code that does the switching, unless you have the 4K in the Mini-Memory. It is remotely possible that a very clever programmer might be able to move some of the contents of the >8300->83FF PAD memory out to VDP memory, move his bank-switching code in, switch banks, and then restore the PAD contents from VDP memory, but this would be very tricky.

The simplest use of the extra memory is as a disk emulator. This is easily available to BASIC programmers, or even to those who use only pre-programmed applications such as TI-WRITER. I was a bit disappointed in the implementation that Foundation used for the disk emulator. It is severely limited, in that it allows only three (3) files to reside in the memory at any given time, one in each of the unmapped banks. However, it is a usable implementation which works for many cases. The three files which may reside in memory are limited in size to 32K bytes each for the files in banks 1 and 2, and 24K for the file in bank 3. There is also a way to use the entire 88K as a single file, provided you can live with a fixed length record, relative file. I can hear you asking, "What happened to the other 8K? Why only 24K in bank 3? Why only 88K overall?", and the reason is that Foundation uses the upper 8K of bank 3 as working space for the DSR (Device Service Routine) which drives the disk emulator.

File names to be written or read from the memory card are prefixed with "DSKX.", for example: DSKX.DATFILE, just as if the file was on a disk drive whose number was "X" instead of "1", "2", or "3". They are opened the same in BASIC as a disk file, and in general, anywhere you can use a disk file you can use a DSKX file:

- * You may SAVE, OLD, and RUN (in Extended BASIC) a program file.
- * You may use a DSKX file as input, output, or list output for Editor/Assembler or TI-WRITER, and as list output for MULTIPLAN. (Unfortunately, you may not put a MULTIPLAN spreadsheet on a DSKX file, because MULTIPLAN checks for a DSK1, DSK2, or DSK3 as the first four characters of the spreadsheet file name.)
- * You may use a DSKX file to hold source code for an assembly, or object code from an assembly.
- * You may use a DSKX file as a working file, for example while sorting data.
- * When you are finished with it, you may DELETE a DSKX file.

(continued on page 13)

TI FORTH: A BEGINNERS INTRODUCTION
by David Wunderlin

FORTH was developed by Charles Moore in the late 60's and early 70's. FORTH's name came about because the IBM 1130 (the first computer Moore used in the development of FORTH) only allowed 5 letter words. Moore was trying to design a fourth generation language, so it became FORTH instead of FOURTH. His first FORTH application was for the National Radio Observatory in the early 70's. In 1973 Moore formed FORTH, INC to market commercial versions of the FORTH language. He left the company in 1978 to form FIG (the FORTH Interest Group) to encourage the use of FORTH. The address for FIG can be found on SCREEN 2 on your master TI FORTH disk. Membership is \$15 a year and you receive the bi-monthly magazine "FORTH DIMENSIONS". FIG also has source listings available for many different machines and CPU's.

Why bother with a new language, you may ask! Here is a speed comparison I performed using a program called LIFE (from 99'er Vol 1, No. 4 page 58). It is a program about a living population going through successive evolving generations. The simple rules are: a living cell with 2 or 3 neighbors survives to the next generation, a living cell with 4 or more neighbors dies of overcrowding, and an empty cell with exactly 3 live neighbors is born in the next generation. This version of life gets into a repeating pattern after 114 generations. The article included a BASIC program listing and an ASSEMBLY LANGUAGE program listing, both of which I had already loaded into my machine in hope of learning the 9900 assembly language. I found that I didn't have the patience to learn the language at that time. But in learning FORTH for the August assembly language presentation I was working on, I found the FORTH version of LIFE was easy to write. Here are the speed comparisons thru 114 generations:

BASIC takes 4 Hrs 1 Min (14460 SEC)
estimated time.

FORTH takes 17 MIN 5 SEC (1025 SEC)
or approximately 14 times faster than
BASIC.

ASSEMBLY LANGUAGE takes 1 MIN 5 SEC
(65 SEC) and is approximately 16 times
faster than FORTH and 223 times faster
than BASIC.

As you can see by the comparisons, ASSEMBLY LANGUAGE still is the option for speed. And FORTH has an assembler that could be used to write assembly language subroutines for the slower parts of a FORTH program. But I doubt if I will run into many applications where I would worry about the little difference in speed.

This is a sample of how I set up my FORTH disk to simplify the editing and writing of my FORTH programs. If you follow this example you will be able to start editing as soon as the disk has completed booting, instead of having to select from a menu and wait for the editing screens to be compiled each time you boot the FORTH disk.

Disk set-up to allow auto loading
of your editor:

Make a back up copy of your FORTH disk but use selective copy to copy only two files, FORTH, and FORTHSAVE (these two files are required to get the basic FORTH library loaded). Do not copy SYS-SCRNS (a waste of time for this set-up procedure). SYS-SCRNS contains all the FORTH procedures listed in the back of the manual you received with your FORTH disk. We will be loading the ones that are needed for editing and learning TI FORTH.

After making the back-up, boot your master FORTH disk using the EDITOR ASSEMBLER cartridge. With the TI FORTH menu listed on the screen, you are now ready to load the editor of your choice (either 40 col or 64 col). For the 64 col editor, select -64SUPPORT; for the 40 col editor, select -EDITOR, and -GRAPH; then for both select -VDPMODES, -DUMP, -BSAVE, -COPY, and -CRU. Only select -ASSEMBLER, -TRACE or -FLOAT if you plan on using them.

Now before we load the final procedure to complete your editing disk, you will need to correct an error on SCREEN 72 of the master FORTH disk, if you have not already done so. Type 72 EDIT (ENTER). This will get SCREEN 72 and execute the editor you loaded. Using the FCTN-arrow keys, move down to

line 5 (line counts start with 0). Change: PAB_ADDR to PAB-ADDR. Also at this time, set line 4 to the configuration that matches your printer requirements. Don't forget the " (quote) at the end of your set-up. For example, change RS232.BA=9600" to PIO/1", then exit the edit mode with a FCTN-9. Now type UPDATE FLUSH (ENTER). Be sure the write protect TAB is removed from your master FORTH disk for this step only. This will save the corrected SCREEN 72 to your master disk. Now type -PRINT (ENTER). You now have all the SCREENS loaded in the library that you will need to learn more about TI FORTH.

Insert the editing disk you are generating in Drive One. Type: ' TASK 21 BSAVE (ENTER). This will save all the previously loaded SCREENS in binary form to your editing disk starting at SCREEN 21. " " (tick) leaves the parameter field address on the stack. The file will start loading to screen 21. BSAVE is the procedure that saves the memory image to disk.

Now type: 3 CLEAR (ENTER). This is to clear SCREEN 3 to allow you to enter a new SCREEN 3.

Type: 3 EDIT (ENTER) (This will allow you to edit SCREEN 3) SCREEN 3 is the auto boot SCREEN and is loaded at the initial booting of the FORTH disk. We need to change it to get auto booting of the binary file that we just generated.

This is my suggested SCREEN 3:

```
0 ( WELCOME SCREEN ) BASE->R
1 DECIMAL 16 SYSTEM ( CLEARS SCREEN )
2 HEX 10 83C2 C! ( DISABLES QUIT )
3 DECIMAL 21 BLOAD ( LOADS EDITOR )
4 0 0 GOTDXY ." BOOTING FORTH "
5
6 1 VDPMODE ! ( TEXT MODE )
7 0 DISK_LO ! ( ALLOWS EDIT ALL SCRIN )
8
9 : FREE SP22 HERE - . ; ( FREE WILL )
10 ( DISPLAY AVAILABLE USER MEMORY )
11
12
13
14 R->BASE
15 EMPTY-BUFFERS
```

After the above SCREEN is entered, hit FCTN-9 to get out of the edit mode. Then type: UPDATE FLUSH (ENTER). This
(continued)

(FORTH, continued)

will save your version of SCREEN 3 to your new disk. Now type MON (ENTER). This will get you back to the TI color bar screen. Then boot your new FORTH editing disk. Hopefully you now have a working disk.

Next we have to copy SCREENS 4 and 5 from your master FORTH disk. These SCREENS contain the error messages that the FORTH operating system needs. Insert your master FORTH disk then type:

```
EMPTY-BUFFERS (ENTER)
4 BLOCK DROP UPDATE (ENTER)
5 BLOCK DROP UPDATE (ENTER)
```

Insert your new FORTH editing disk type:

```
FLUSH (ENTER)
```

You now have a working disk with your editor set up in the FORTH dictionary ready for immediate use.

TI FORTH disk usage is substantially different from the format used by the other languages and programs available on the TI 99/4A. TI FORTH has 90 SCREENS on one TI disk numbered from 0 to 89. Each SCREEN has 1024 bytes. The SCREEN format of 64 characters by 16 lines is the standard FORTH format of a SCREEN. SCREENS 0 thru 6 are where the catalog of a normal TI disk resides. The rest of the disk is used for normal TI files and programs. For a TI FORTH disk, SCREENS 0 and 1 are needed for catalog space. This is to allow the Editor/Assembler cartridge to load FORTH and FORTHSAVE as TI program files. TI FORTH uses SCREEN 3 as the auto boot SCREEN. SCREENS 4 and 5 are the error message SCREENS. SCREENS 6 and 7 are not used, and can be used by the user. SCREENS 8 and 9 are where the file FORTH (6 sectors) resides and these SCREENS can not be used by you. SCREENS 9 thru 19 are where the file FORTHSAVE (39 sectors) resides. These SCREENS also can not be used by you. SCREENS 21 thru 31 are where we saved your binary editor file. SCREENS 32 thru 89 are free for your own procedures. Examine any of the SCREENS by using the EDITOR. No SCREEN is changed on the disk until you use the UPDATE and FLUSH commands.

Here are two utility procedures I

find useful. I saved them on SCREEN 6 and only load them when they are needed.

```
( CLEAR SCREENS )
( HAVE THE START AND STOP SCREEN )
( NUMBERS ON THE STACK )
( BEFORE CALLING CLEARALL )
: CLEARALL EMPTY-BUFFERS
  1 + ( ADDS ONE TO STOP )
  SWAP DO I CLEAR LOOP FLUSH ;
```

CAUTION: This procedure will clear any SCREEN! As there is no way of locking or protecting a SCREEN from inadvertent change.

Procedure 2:

```
( FULL DISK INDEX ON PRINTER )
: FULL-INDEX SWCH ( TO PRINTER )
  0 89 INDEX ( LIST INDEX )
  CR UNSWCH ; ( TURNS RS232 OFF )
```

The list printed on the printer is the equivalent of the CATALOG command on the disk manager cartridge. It lists all line 0's that contain comment lines.

The STACK is the most important and most used feature of FORTH. FORTH has two stacks: The Parameter Stack (the STACK) and the Return Stack. The Parameter Stack is the main location for data and logic operations. Each one is 16 bits, making FORTH a 16 bit language. This stack is central to FORTH's design, and most information is passed from one procedure to another through the STACK. All math and logic operations are performed on the STACK with the results left on the STACK. The Return Stack is used by FORTH to return to the calling procedure. It can be used by the user, but any data stored there needs to be removed before the end of your procedure or the computer will not be able to return to the calling procedure. Use the Return Stack with caution.

Since the STACK is so important it is necessary to understand a few of the STACK manipulation commands before being able to write even a simple FORTH program. DUP takes the top number off the STACK and leaves two identical copies on the STACK. DROP is used to remove the top number from the STACK. SWAP takes the two top numbers and reverses their order. OVER copies the second number and leaves a copy of it on top of the STACK. ROT takes the third

number on the STACK and moves it to the top.

The arithmetic and logic operations also manipulate the STACK but they can easily confuse the beginning FORTH user, especially if you are not familiar with postfix notation (also called Reverse Polish Notation). Forth math and logic operations take two operands from the STACK and leave a single value as a result. I have selected some of the more basic math and logic operations for discussion: + removes the two top numbers and leaves their sum on the STACK. - removes the two top numbers and leaves their difference on the STACK. * removes the two top numbers and leaves their product on the STACK. / removes the two top numbers and leaves their quotient on the STACK. /MOD divides using the two top numbers leaving the remainder and the quotient on the STACK. (<=) each perform comparisons of the two top STACK numbers and leaves a 0 or 1 on the STACK depending on the outcome of the test, 1 = true, 0 = false.

FORTH also uses the STACK for all memory accessing. When you use a variable name in your procedure, the address for that variable is placed on the STACK. Then you can store or retrieve data depending on which memory access WORD follows your variable. @@ replaces the word address by it's contents (2 bytes). C@@ replaces the word address by it's contents (1 byte). ! stores the second number on the STACK at the word address which must be the top of the STACK (2 bytes). C! stores the second number on the STACK at the word address (top of STACK) (1 byte). ? prints the contents of the word address on the STACK (2 bytes) to your output device.

The input and output devices in TI FORTH are specified by the user variables ALTOUT and ALTIN. For example, SWCH will set ALTOUT so all output will go to the printer and UNSWCH sets ALTOUT for outputting to the monitor. . (dot) outputs the top number on the STACK to the output device as a signed number. U. (U dot) outputs unsigned numbers. ." (dot quote) outputs a ASCII string terminated by a ' (quote). ?TERMINAL tests the input to see if the BREAK key (FCTN-4) is being

(continued on page 10)

```

*****
*
*           WARGAME REVIEW
*           by
*           John M. DiIorio
*
* GAME: B-1 NUCLEAR BOMBER
* COMP: AVALON HILL
* TYPE: MICRO-SOFTWARE
* KIND: CASSETTE
* LANG: TI 99/4A BASIC (OTHERS)
* BYTE: 16K RAM
* YEAR: 1983
* COST: $16.00
*
* This is Avalon Hills' first
* game released for the TI 99/4A.
* Avalon Hill has over 40 software
* games and almost 100 Board games
* in their 'library'.
* B-1 NUCLEAR BOMBER is a game
* of a manned B-1 Bomber nuclear
* strike mission in the Soviet
* Union.
* The display is simple with
* only basic B-1 data (NO sound/
* color). There are 12 commands
* to pilot and defend the Bomber.
* The object of the game is to
* destroy the primary target with
* your Short Range Attack Missile
* (SRAM). The primary target is
* randomly selected as is your
* 'Fail-Safe' Code. In order to
* launch your SRAM, you must KNOW
* this code.
* There are 10 targets in the
* Western portion of the Soviet
* Union. A map is provided from
* which the player uses to pilot
* your B-1. However, this is not
* always so clear. The computer
* plays the Soviet Union with 20
* Defense Complexes(DC's). These
* DC's are activated when your B-1
* comes within 750 kilometers(km),
* that's about 450 miles. Each DC
* has the ability to launch one
* MIG (jet fighter) and one SAM
* (surface-to-air missile). Each
* MIG and SAM is different. For
* example, a MIG-33 is better than
* a MIG-25 and will respond to
* your defenses accordingly.
* Your B-1 has SIX Phoenix mis-
* siles to defend with.
*
*****

```

```

*****
*
* A sample game would start
* like this: A message comes
* in 'FLASH-HOT WAR' your target
* is 'MURMANSK' (the closest) and
* your Fail-Safe Code is 'CLTSX'.
* You would take off from Thule
* Air Force Base (AFB) in Greenland
* with an Course (in degrees TRUE),
* a Speed (in km per Hour), the
* Altitude (in meters), the Fuel
* Range (in km), and the number of
* Phoenix missiles (0 to 6). This
* data is displayed when you type
* in STATUS (ST for short) command.
* The command RA does a radar
* search of enemy flying targets.
* NA gives the course and range to
* the target in question. SE does
* a search of the nearest target
* or DC. AU is the auto-pilot and
* can be pre-set for any length of
* time. EV is evasive action for
* close flying targets and EC is
* electronic counter-measures for
* any range, however the strength
* decreases with use. FH is to
* launch one Phoenix missile. AR
* arms the BOMB after the CORRECT
* Fail-Safe Code is given. BO is
* to drop the BOMB on the target.
* HE is for HELP which will list
* your command options.
* Below is a complete listing
* of Targets and DC's:
*
* MURMANSK ARKHANGELSK
* LENINGRAD SVERDLOVSK
* MOSKVA KIYEV
* VOLGOGRAD SEVASTOPOL
* ASTRAKHAN YEREVAN
*
* PECHENGA UKHTA VYBORG
* TALLINN ONEGA PSKOV
* ROSTOV PODOLSK SYKTYUKAR
* PINSK LIPETSK YARANSK
* VINNITSA KHARKOV ODESSA
* KONOSHA SARFA OCHAMCHIRA
* TBILLSI DUBOVKA
*
* Some tips on play: Use your
* Phoenix missiles to destroy the
* DC's. Stay at a low altitude.
* Use EV when MIG's and SAM's are
* close. Use EC in emergencies.
*
*****

```

PLAY-TEST
by
John M. DiIorio

This is a first in a series of play-tests of computer software games
where the program(s) are tested, analyzed, and improved. This month's
game is 'GALAXY', which was reviewed in the last 'MSP 99 NEWSLETTER'.

Human Player: John DiIorio (Planet A at 10 ships/month) vs. 'TEX'

Test: #1(4 games, 1-4) #2(8 games, 5-12) #3(7 games, 13-19)

Date: 7/23/84 8/04/84 8/11/84

Plts: 26 13 10

Table with columns: TIME, MOS, and sub-columns for each game (1-19). Values represent time and months for each game.

PLT : G1 : G2 : G3 : G4 : G5 : G6 : G7 : G8 : G9 : G10 : G11 : G12 : AVE

Main data table with columns for planets (A-Z) and average values. Rows show ship counts for each planet across different games.

KEY: PLT=Planets
MOS=Months
AVE=Average of Time, Months, & Ships/month/planet for all 19 games.
TOT=Total of Ships/month for each game, including Planet A.

(FORTH, continued)

pressed. ?KEY returns with a ASCII value or 0 if no key is pressed. KEY waits for a keystroke before returning with the ASCII value. EMIT outputs the top STACK value as an ASCII character. EXPECT reads a string of characters from the keyboard and stores them at the word address specified.

Here are some miscellaneous control words that will help when using FORTH. GOTOXY positions the cursor on the monitor screen at the x and y coordinates specified by the two top numbers on the STACK. VARIABLE defines a new word as a variable in the dictionary and allots one byte of memory. ALLOT sets aside the number of bytes of dictionary memory desired, and is needed for variables which are more than 1 byte long. The word ((parens) is not acted on by the FORTH compiler. The interpreter ignores the input following the blank and the trailing ")" character on the same line and is used to comment your FORTH procedures. FORGET is used to delete all the word definitions in the dictionary from the word you call to the end of the dictionary.

When testing my LIFE program I

found that after about 12 minutes the screen went blank (the normal screen blanking operation present in the TI 99/4A). As this was undesirable for a program that uses the screen only and takes 17 minutes to execute. I found that the console routines that do the screen refresh, increment a counter every 1/60th of a sec and when the counter times out, it blanks the screen. This counter is reset every time a character is input by the keyboard. Since I did not use or need any input routines for my program the counter was never reset. I found this information in the Editor/Assembler manual page 406, >8306 VDP screen counter. By resetting this counter periodically the screen will not blank out. As this counter is incremented every 1/60 of a second I felt that it would make a good timer for testing the time each procedure takes in my FORTH programs. (HEX 8306 is equal to DECIMAL -31786) To use it, enter: -31786 @@ (enter procedure name) -31786 @@ SWAP - . (dot). This will print out a number indicating how many screen refreshes occurred. This can be used if timing is important in your programs. It can help you find out which procedures to spend your time speeding up, and test to see if any changes you

implement are helping.

In preparation for the assembly language meeting of August 7 and for this article, I found several good sources of information about FORTH. My first purchase was "Starting FORTH" by Leo Brodie. It was most helpful because of the the list of changes in Appendix C of the TI FORTH manual. My second and more helpful purchase was "FORTH FUNDAMENTALS Volume 1" by C. Kelvin McCabe. It was easier to read, and I felt it gave me a better understanding of FORTH. Other references I used were: BYTE, Aug., 1980, (this issue was devoted to just the FORTH language); BYTE, Oct., 1980, page 274; BYTE, Mar., 1981, page 155; POPULAR COMPUTING, Sept., 1983, (this had a special report on popular newer languages); and MILLER GRAPHICS, THE SMART PROGRAMMER, Feb. and Mar., 1984.

For those people interested in the FORTH program of LIFE that I wrote for my benchmark, send a stamped, self-addressed business size envelope to:

DAVID WUNDERLIN
FORTH LIFE
9324 30th Ave N.
NEW HOPE, MN 55427

THE POWER OF FORTH By Jim Szydowski

(Reprinted from THE COMPUTER BRIDGE, St. Louis 99'ers.)

For those of you who have problems dumping your data to printer because you are using a parallel port on your RS232 card, the March issue of THE SMART PROGRAMMER solves the problem. Five changes are required:

1. Line 0 should read; (ALTERNATE I/O SUPPORT FOR PIO PNTR 12JUL82 LCT)
2. Line 2 should read; 0 0 0 FILE >PIO BASE->R HEX
3. Line 3 should read; ; SWCH >PIO PABS 10 + DUP PAB-ADDR ! 1- PAB-VBUF !
4. Line 4 should read; SET-PAB OUTPT F-0" PIO" OPN 3
5. Line 5 should read; PAB-ADDR VSBW 1 PAB-ADDR 5 + VSBW PAB-ADDR ALTOUT ! ;

The above five corrections should take care of printer problems. THE SMART PROGRAMMER has been extremely helpful to me and is well worth the \$12.50 per year. It looks like it may be as valuable as the FORTH DIMENSIONS to FORTH programmers.

Courage Center



3915 Golden Valley Road • Golden Valley, Minnesota 55422 • (612) 588-0811

August 20, 1984

Texas Instruments User Group MSP 99
P.O. Box 12351
St. Paul, MN 55112

Dear Computer Club Members:

We at Courage Center are interested in setting up a computer club for people with disabilities. Any information that you could give us about your club would be a great help. Please send it to:

Denis Meyer
Electro-Mechanical Technician
Courage Center
3915 Golden Valley Road
Golden Valley, MN 55422

or give me a call at 588-0811, ext. 192.

Thank you very much.

Sincerely,

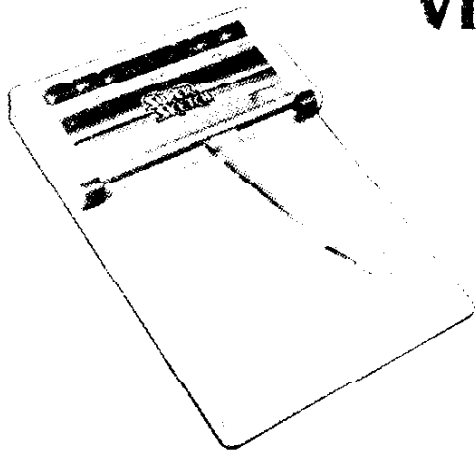
Denis Meyer
Electro-Mechanical Technician
Rehabilitation Engineering Services

DJM/lkm

(We have sent the Courage Center information regarding MSP 99 meeting times, dates, and so on. However, some of you may have experience or interest in working with people with disabilities. If so, don't be shy. Contact Denis Meyer and ask where your services might be most useful. - Editor.)



CREATE SUPER VIDEO GRAPHICS



JUST
\$59⁹⁵

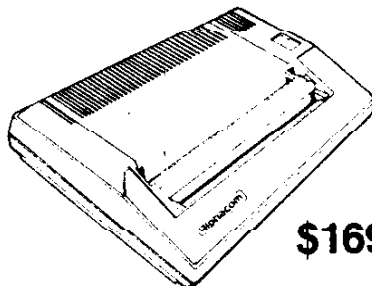
**SUPER
SKETCH™**



PERSONAL PERIPHERALS, INC.
PEP
Etc.

WE ALSO HAVE EXPANSION BOXES, RS232 CARDS, DISKS DRIVES, 32K MEMORY CARDS AND MUCH MORE!

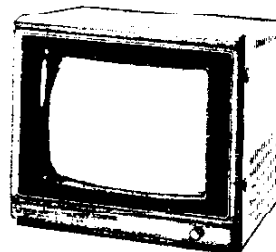
80-COLUMN PRINTER



\$169.95



MONITORS

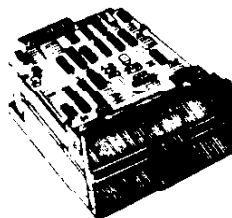


\$299



\$99

DISK DRIVES



VISIT OUR SHOWROOM FOR MANY SPECIALS - WE CARRY THE LARGEST SELECTION IN THE TWIN CITIES



SPECIALIST IN:
TEXAS INSTRUMENTS HOME COMPUTERS
(612) 544-6219



1815 FORD ROAD (Near Hwy. 12 & Co. Rd. 18 in Minnetonka)

Hours: 11am - 7pm Tue. - Sat. (Other hours by appointment)

WE ALSO SELL MAIL ORDER AND OFFER COMPETITIVE PRICES WITH ALL MAIL ORDER COMPANIES

(Poor Richard, continued)

These are just some examples; as I said before, you may use a DSKX file just about anywhere you could use a real disk file. HOWEVER -- and this is very important -- you must always remember that the data in DSKX files is volatile. What does that mean? It means that when you turn the power off, the data is lost, finally and irrevocably. Even if you only turn off power to the console, and not to the PEB, the data is usually corrupted to the point of unusability. What this means from a practical standpoint is that DSKX files can only be used as temporary files. When the time comes for permanence, you must get the data from your DSKX files to real disk (or cassette) files (and you can't use the Disk Manager). In the meantime, you save time manipulating data on a DSKX file over using a disk file. For example, this column takes 37 - 38 seconds to load or save from TI-WRITER using a disk file, but only 8 - 9 seconds using a DSKX file. Another example of DSKX use is CAT, a small program I use frequently. It is stored on many (but not all) of my diskettes. It's purpose is to find out what files are on the diskettes currently in my disk drives. I find it convenient, now that I have the disk emulator, to SAVE this program to DSKX.CAT when I start using my computer. Then I can use it whenever I need it, without worrying about whether the diskette currently in the drive contains CAT or not. An added bonus is that CAT loads much faster from DSKX than from disk. In fact, since CAT is a small program, it is loaded from DSKX almost before the disk drive would be up to speed.

In addition to the method mentioned using the "DSKX." prefix to identify variably named memory files, there are fixed file names which may be used instead. Banks 1, 2, and 3 may be referenced as MEM96A, MEM96B, and MEM96C respectively, using either sequential or relative organization, fixed or variable record lengths. The designation MEM96 references the entire card as a single file, but only allows fixed length record, relative files. Examples of OPEN statements using these designations are:

```
100 OPEN #1:"MEM96A", VARIABLE 80,
    OUTPUT
```

```
100 OPEN #1:"MEM96", FIXED 192, UPDATE,
    RELATIVE
```

Unfortunately, there appear to be some problems in the current implementation of the disk emulator. Those I have noted are as follows:

- * Very large BASIC files cannot be saved and recovered properly. The first file I tried was a 72 sector BASIC program. The SAVE seems to work correctly, but loading the file via the OLD command produces very strange results, with the upper half of the screen containing garbage characters. Smaller files seem to work OK.
- * Reading from relative files addressed as DSKX.whatever, MEM96A, MEM96B, or MEM96C does not appear to work correctly in one of my older programs (which I use frequently). However, this same program is able to use the memory card as a single 88K file, using the MEM96 designation, without difficulty. (I should point out that a small test program writing and reading a file of the same characteristics worked properly, so the exact cause of this problem is still up in the air.)

I will be contacting Foundation regarding these problems and will report back on the response and their ultimate resolution.

Summary: The Foundation 128K Memory Card has performed flawlessly as a 32K Expansion Memory for my TI-99/4A for a year and a half. The previous version of the firmware was limited to a fixed name, flawed implementation of relative files, but it also performed well, within its limitations. However, it would not work with existing programs - only with programs built specifically to use it. The current version of the firmware is much improved, allowing much wider latitude in use of the hardware with existing programs, thus increasing its usefulness substantially. The 3 file maximum implementation is a long way from what I was hoping for, but it is workable. It will be interesting to see whether Foundation will correct the remaining problems. Inquiries may be directed to:

Foundation Computing
74 Claire Way
Tiburon, CA 94920
(415) 388-3840

(continued)

(Poor Richard, continued)

80-COLUMN SCREEN REVISITED: I have received additional information regarding Foundation's 80 column card for the TI-99/4A, so I am passing it on as promised last month.

The Foundation 80 column card will function just as any other peripheral device. You will be able to OPEN it as a file and WRITE to it in BASIC, just as you do to the RS232 card. The device name will be "SCR80". It will drive a monochrome monitor (no color). Separate monitors will be required for the normal 28/32/40 column display and for the 80 column display. Existing programs will still write to the normal display and INPUT and ACCEPT statements will still show on the normal display. To get these kinds of activity to come to the 80 column display will require special assembly language or FORTH programming. It COULD be done in BASIC, but it would be slow, using CALL KEY subroutines to simulate INPUT or ACCEPT.

In addition to the 80 column version of the COMPANION word processor which I mentioned last month, Foundation also plans an 80 column telecommunications package, which apparently won't require an RS232 card, since the 80 column card has it's own RS232 interface built in.

THE OLD CLOCK ON THE WALL SAYS...: I really do have a clock on the wall here in my computer shack, and it's telling me that I've been at this long enough for this month - it's time to hang up the old TI-WRITER for now, and bid you fond adieu (or was that fondue?) until next month.

(Nostalgia, continued)

TI has had its problems with the marketing of the 99/4 since its introduction in October of 1979. An interesting article in the June 16, 1980, issue of FORTUNE magazine discusses TI's early problems and marketing strategy of the 99/4A. They reported that TI "...unlike most of its rivals...is concentrating on selling to the ordinary consumer...TI figured that consumers would be willing to pay a premium for the TI name, but the price of the 99/4 - now \$1400 - seems much too high". Amen to that! I don't know what was included in that price.

Let's jump ahead to December, 1980, and BYTE magazine again. It's reported that "...although the unit has many unique features, it is overpriced and undersupported...TI is experimenting with \$200 rebates (\$100 cash and \$100 worth of software)...the 99/4 lists at \$950, but many dealers are discounting it to as low as \$699 plus rebate". Hmmmmm - that was about \$600 for a TI-99/4 if you could find that discount house.

Back to BYTE again, this time the May, 1981, issue. Under 'EYTELINES': "...Texas Instruments is determined to make its TI-99/4 home computer a success...the new list price for the console is \$649, a reduction of \$300". Wonder what the discount house was asking then?

The TI-99/4A was introduced, I believe, in June, 1981. I don't know what the initial price was. I do know it was September 1, 1982, when I first became aware of a computer price war. A K-Mart ad caught my attention with a VIC-20 at \$178 and the TI-99/4A at \$299 with a \$100 rebate. I jumped at that bargain and bought my first home computer, the TI-99/4A. The price war continued. By February, 1983, the TI rebate was still offered, so you could get a console for \$150 with the \$100 rebate. By summer of 1983, with a TI rebate of \$50, the cost was down to \$90.

On July 26, 1983, the WALL STREET JOURNAL reported "...in June, Commodore slashed the price of its Model 64 by 45%...demands for...VIC-20 are flat, and distributors note sharp declines in sales of Atari, Texas Instruments, and Timex offerings since June...". It appeared the handwriting was on the wall as the price wars continued with TI's peripheral package deals. Then came the 'coup de grace' and on October 29, 1983, newspapers announced "TI PULLS OUT OF THE HOME COMPUTER MARKET". Rebates were off and shelves began to clear of TI-99/4A's. Usually the sales were unadvertised and at unbelievably low prices. \$50 per console was not uncommon. Merry Christmas everyone!

Exit TI, enter IBM: TI stock went up some 22 points!

From reported prices of \$1400 to \$50 is some jump in three year's time. I guess the moral to this story is don't fool around with Mother IBM or something like that. Seriously, I hope if you paid a small sum for your TI-99/4A, you realize that you do have something more valuable than a game machine and got in on a fabulous deal.

(Reprinted from the April, 1984, Rocky Mountain 99's newsletter. Author: Bob Grossart.)

**TEXAS INSTRUMENTS COMPUTER CLASS
BASIC PROGRAMMING CLASS**

Dunwoody Institute and the Texas Instruments Computer Club are offering a BASIC Programming course exclusively for members of the TI Users Club. The course curriculum includes computer familiarization, elementary flow-charting and beginning basic programming procedures.

Classes will be held in the Computer Learning Center on the IBM Personal computers. Programming differences between the IBM and TI will be discussed. Or students could arrange to bring their own TI equipment to class.

WHEN: Monday nights - 6:30-9:30 p.m.
12 weeks beginning September 24, 1984

WHERE: Dunwoody Institute
Computer Learning Center

INSTRUCTOR: To be selected by Dunwoody
and the Texas Instruments Club

HOW TO REGISTER: Just complete and mail in the application form. Time is of the essence: Enroll NOW if you are at all interested. Send your completed form and with your check to:

MSP BASIC CLASS
6240 Alden Way
Fridley, MN 55432

COST: \$68.00 plus textbooks (approx. \$17.00)
Make checks payable to MSP 99 User Group.

Continuing Education Application

Name _____
 Address _____
 City _____ State _____ Zip _____
 Home Phone _____ Work Phone _____
 Social Security Number _____
 Course No. _____ Title _____
 Start Date _____ Day _____ Time _____ Fee _____
 Course No. _____ Title _____
 Start Date _____ Day _____ Time _____ Fee _____
 TOTAL FEES ENCLOSED _____
 I have read the general instructions:
 Signature _____
 Are You A Former Dunwoody Student? _____ Yes _____ No

TIDBITS

*** Do any of you enjoy Avalon Hill games? MSP 99 member John DiIorio is an avid enthusiast. John will give a brief presentation at September's meeting, discussing the two available games and explaining his "play-test" shown on pages 8 and 9 of this issue. Looks like we may have the makings of another special interest sub-group.

*** Want to learn how to program BASIC? Sign up NOW using the special application form found on page 15. Classes begin in less than three weeks!

*** MSP member Dave Wunderlin was a guest on 'Computer Talk', August 25th. Be sure to tune in on September 22nd, when the TI computer will again be featured.

*** If, for some reason, you haven't received your new MSP 99 Software Catalog (the one issued in April, 1984), please let us know. This especially applies to new members. Drop a note to:

MSP 99 Software
6761 Deerwood Lane N.
Maple Grove, MN 55369

MSP 99 USERS GROUP
P.O. BOX 12351
ST. PAUL, MINNESOTA 55112

Address Correction Requested

WANT ADS

Members may place want ads, at no charge, by calling Bonnie Burton 431-6064 or by turning in written copy of the ad at any MSP 99 meeting. Businesses may purchase ad space at the cost of \$1 per line.

DISKS/TAPES -- Top quality blank disks and Ampex 10-min. tapes at users group rates. Box of 10, \$20; single disk, \$3. Tapes: 10 for \$5; 75 cents each. Tape boxes: 10 for \$1.50; 25 cents each. Mailing charge \$1 or pick up at meeting. Brad Olson, 786-1235.

BACK-ISSUES AVAILABLE MSP 99 NEWSLETTER

Only \$1.50 per issue. Submit your order by mail to:
7500 Germane Trail
Apple Valley, MN 55124

Enclose a check or money order made payable to the MSP 99 User Group and allow 2 - 3 weeks for delivery. (1982, 1983, 1984 issues available.)

Bulk Rate U.S. Postage PAID Permit #1285 Minneapolis, MN
--

CENTRAL ALABAMA
USERS' GROUP
551 LAKWOOD DRIVE
MONTGOMERY, AL 36117

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY