VOLUME 2 ISSUE 1 $1.50 SEPTEMBER 1985

# SUPER 99 MONTHLY

## ANNIVERSARY ISSUE

---

It certainly looks like our second year will be an exciting one! Among our topics to start our new year are important news stories, a great article by Barry A. Traver and the announcement of a new staff member.

We are very pleased to announce the affiliation of Dr. Steven J. Szymkiewicz as a Corresponding Staff Writer. Dr. Szymkiewicz has contributed two excellent hardware articles in previous issues and plans to report on several hardware projects on which he is currently working.

Myarc has now confirmed that there will be a new computer and it will be shown at the TI-FAIRE in Chicago on November 2 (see related story on page 7). An official announcement has been subject to minor delays due to factors such as the late arrival of keyboards (most computer firms don't make their own keyboards).

We have traced rumors to such an extent as to be confident in disclosing several facts. The computer has 256K of CPU memory on board and 64K of Video memory (a chip that is compatible with the 99/4A's 9918A VDP and similar to the 64K chip TI dropped). Carrying an 84 key keyboard (compared to 48 on the 99/4A), the computer will be 90% compatible with the 99/4A (very similar to the 99/8 prototypes that TI dropped) in 99/4A mode. In the native machine mode (a CPU that is upwards compatible with the 99/4A's 9900 CPU will be present), the computer will be capable of addressing 2 megabytes of memory (this memory count would include the 512K available on the MEXP-1 Memory Expansion Card). And, it will be compatible with Millers Graphics' GRAM Kracker™. Cards that follow TI standards will be compatible (Myarc cards are recommended). The price has not been set. The story is just beginning (more next month)!

In other Myarc news, the new Extended BASIC has not yet been released, but should be available very soon. Also, an EPROM upgrade for the MEXP-1 card that corrects null string and APPEND file problems is now available.

It now appears very likely that after years of waiting, a compatible computer is just around the corner!

---

-->

# HARDWARE

## LOAD Interrupt Switch

STANDARD: 1A 5A 7B 12A

In July, we promised an August article on LOAD interrupt switches. To be quite honest, we had not carefully examined the ramifications of such a commitment, for which we apologize. In the past, most such switches were implemented inside the speech synthesizer and were only built one way. Today, speech synthesizers are being moved out of sight by use of a ribbon cable or by placing them inside the Peripheral Expansion Box or even in the closet (not in use). The switches themselves are sometimes more complex. This month, we'll cover the old standard switch implementation. In future issues, we'll cover the more modern implementations and offer still more software possibilities and other discussions. Thanks to those of you who maintained your wit and sense of humor in reminding us of our procrastination!

First, let's discuss what a LOAD interrupt is. Interrupts are a means of interrupting the programmed sequence of a computer to access some other routine. The TMS9900 that is in our 99/4A's allows up to 16 maskable and 2 non-maskable interrupts. We are presently only concerned with one of the non-maskable interrupts, LOAD. The other, RESET, is the interrupt that is activated by pressing the button on the Navarone Cartridge Expander™. As you can now understand, we're discussing a button somewhat similar to the one on the Navarone unit (RESET and LOAD are, however, unrelated in operation).

Interrupts cause a branch to an address known as the interrupt vector. The LOAD interrupt vector begins at address >FFFC. The word at >FFFC is the workspace area of the routine and the second word, at >FFFE, is the entry point of the routine. A BLWP is performed on the addresses in the interrupt vector when a LOAD interrupt occurs. A LOAD interrupt can then be used to activate a screen dump, debugger, baud selection, color change or almost any routine you might need.

Generating a LOAD interrupt is fairly simple on the 99/4A. At the expansion port, on the right side of the computer, is a 44 pin connector, one of the pins of which is the LOAD pin, pin 13. This line is conditioned to +5V, so all that is necessary to create a LOAD interrupt is to pull the line to ground. There are 4 ground lines on the I/O bus (expansion port), 21, 23, 25 and 27 (the four lines are actually connected). By placing a switch between pin 13 and pin 21 (or any of the 4 grounds), the LOAD interrupt can be controlled!

WARNING: If you are not experienced with electronics and accomplished at soldering, do not proceed! It is very possible that improper soldering will ruin your speech synthesizer (and they don't give those away these days!). For those who are particularly inept in electronics, still further damage might ensue. If you are not absolutely comfortable in approaching this project, find somebody who is!

Begin by opening your speech synthesizer, by removing the 2 screws on the underside. Remove the "guts", as we'll work with the top of the case first. Under the cover (what was originally intended to be a speech cartridge port), drill 2 tiny holes, through which 2 wires will pass. This will tuck the switch away where it will not be easy to accidentally trip it or have young children play with it (console lock-ups are likely to ensue without a usable vector in place). Use a small single-pole single-throw (SPST) switch (such as Radio Shack switches 275-1547, 275-1571 or 275-1549) and pass wires connected to the switch through the holes. More holes may be needed for mounting the switch solidly in place and the switch should be secured with a glue product. Remove the screws from the metal cover of the printed circuit board and slide the PCB out of the cover. Attach one of the wires to pin 13 and the other to pin 21 (solder

-->

in place, of course). Pin 1 is at the lower front of the console, pin 2 is above pin 1, pin 44 is at the upper rear and pin 43 is below pin 44. Test for shorts. Re-assemble everything and test for shorts again. If you are experienced with electronics, this should be one of the easiest projects you've ever undertaken.

Though the implementation that we have described generally works, it does not adequately handle switch contact debounce, which sometimes results in an improper return linkage to the calling program. This situation can be partially resolved through software, but also using hardware solutions (a flip-flop or Schmitt trigger with a capacitor) is a more thorough solution. We'll cover contact debounce, installations not utilizing the speech synthesizer and other aspects of LOAD interrupt switches in future articles.

Credit is due to those who have written outstanding articles on LOAD interrupts, making this article possible, including Bill Gronos and Edgar Dohmann, to name but two. Those who have contributed to this article through consultations include Charles Robertson, Mack McCormick and Dr. Steven Szymkiewicz. Thanks to all of you!

---

# EXTENDED BASIC

## DISPLAY AT IN TEXT MODE FROM EXTENDED BASIC

STANDARD: 1A 2XB EA 4B 5A 6B 7B 9A

by Barry A. Traver

This is the first in a series of articles intended to supply useful assembly language subroutines that can be accessed from Extended BASIC. Such an A/L-XB combination is ideal, in that it allows one to combine the simplicity of programming in Extended BASIC with the power and speed of assembly language routines when required.

Just as built-in Extended BASIC subprograms are accessed by a CALL (e.g., CALL CLEAR), so also these A/L subprograms are accessed by a simple CALL (in this case, a CALL LINK, specifying the program name). True, at times various parameters must also be specified, but such is also the case with built-in Extended BASIC subprograms (e.g., CALL SCREEN, where the desired screen color must be explicitly indicated).

Although you will not be required to know any assembly language to use the A/L routines, you will need to know how to use the Editor/Assembler to assemble the source code contained in these articles. (For information, see pp. 33-35 of the E/A manual). When asked about "OPTIONS?", be sure to respond with an R (and never with a C) to create object code that can be accessed from Extended BASIC.

As we proceed, those who never thought they could learn assembly language may find that they're learning much in spite of themselves. Each month comments that are not needed to use the program will be included -- both in the source code and in the accompanying article -- so that those who are interested in learning more about A/L can do so from an inductive approach. (If you have questions, you are invited to write me, so that I may have ideas as to what to include in future programs and articles.)

This month's article provides assembly language code to enable the equivalent of DISPLAY AT in text (40-column) mode from Extended BASIC. There are actually three main subroutines involved: (1) CALL LINK("MODE40") (to enter text mode), (2) CALL LINK("DISPLA",R,C,M$) (to display a message M$ at row R and column C on the screen), and (3) CALL LINK("MODE28") (to return to normal graphics mode). Warning: you must take care not to access certain BASIC statements (e.g., PRINT or INPUT) while in text mode, or a system lockup may take place. Apart from an appropriate CALL LINK("DISPLA"...), it

-->

is suggested that you confine yourself at the present time to delay loops or CALL KEY routines while in text mode.

It is generally a good idea to load in your A/L routines at the very beginning of your Extended BASIC program, even though the routines may not be actually CALLed until much later. This is done with a CALL INIT and then a CALL LOAD("DSK1.FILENAME") for each object code file that is to be utilized. Later in your program, you can access the various subprograms with the appropriate CALL LINKs as needed. (Once loaded in, they're there for you to use, just as if they had been built-in.)

It is suggested that you call the object code file 40D28/O for this month's program. This is the name which is expected by the accompanying sample Extended BASIC program, which should provide sufficient information for you to be able to use this A/L material with Extended BASIC programs that you may write yourself, so that you can utilize DISPLAY AT in text mode from Extended BASIC for your own purposes.

Again, the comments in the source code are not required reading, but you should be able to learn a lot from them. The inductive approach means that you should not expect to understand everything that's there, but -- like life -- it means that you "learn-as-you-go", picking up such information as you are ready for at the present time, knowing that (hopefully) there is a tomorrow (and a next month's issue of <u>SUPER 99 MONTHLY</u>). Happy computing!

```
*                         ASSEMBLY LANGUAGE SOURCE CODE
*                      FOR DISPLAY AT IN TEXT MODE FOR XB

* 40D28 BY B. TRAVER, 835 GREEN VALLEY DR., PHILA. PA 19128

       DEF  MODE40,DISPLA,MODE28

* EXTENDED BASIC EQUATES (SEE E/A MANUAL, PP. 415-416)
FAC    EQU  >834A  * FLOATING POINT ACCUMULATOR
GPLWS  EQU  >83E0  * GPL WORKSPACE
NUMREF EQU  >200C  * GET NUMERIC PARAMETER (ROM UTILITY)
STATUS EQU  >837C  * STATUS BYTE
STRREF EQU  >2014  * GET STRING PARAMETER (ROM UTILITY)
VMBR   EQU  >202C  * VDP RAM MULTIPLE BYTE READ
VMBW   EQU  >2024  * VDP RAM MULTIPLE BYTE WRITE
VSBR   EQU  >2028  * VDP RAM SINGLE BYTE READ
VSBW   EQU  >2020  * VDP RAM SINGLE BYTE WRITE
VWTR   EQU  >2030  * VDP RAM WRITE REGISTER
XMLLNK EQU  >2018  * ROM LINK UTILITY

VDPINF BSS  192    * VDP INFORMATION STORAGE SPACE
STRING BSS  256    * STRING STORAGE SPACE
WS     BSS  32     * WORKSPACE OF 16 (2-BYTE) REGISTERS

MODE40 LWPI WS        * LOAD WORKSPACE REGISTERS

       LI   R0,768    * SAVE VDP INFORMATION IN BUFFER
       LI   R1,VDPINF *  (NEEDED LATER FOR RETURN FROM
       LI   R2,192    *  TEXT MODE TO GRAPHICS MODE)
       BLWP @VMBR     *

       BL   @CLRTXT   * CLEAR THE SCREEN (TEXT MODE)

       LI   R0,>07F4  * CHANGE TO WHITE (>F) ON DARK BLUE
       BLWP @VWTR     *  (>4)

       LI   R0,>01F0  * CHANGE TO TEXT MODE
       BLWP @VWTR      *
       SWPB R0         *
       MOVB R0,@>83D4  *

       B    @RETURN    * RETURN TO EXTENDED BASIC

DISPLA LWPI WS                * LOAD WORKSPACE REGISTERS

* GET FIRST PARAMETER FROM EXTENDED BASIC
       CLR  R0                * (NOT AN ARRAY)
       LI   R1,1              * USE THE 1ST VARIABLE
       BLWP @NUMREF           * GET ROW
       BLWP @XMLLNK           * EXECUTE ROM ROUTINE
       DATA >12B8             * CONVERT FLOATING POINT TO INTEGER
       MOV  @FAC,R5           * PLACE ROW IN REGISTER 5

* GET SECOND PARAMETER FROM EXTENDED BASIC
       CLR  R0                * (NOT AN ARRAY)
       LI   R1,2              * USE THE 2ND VARIABLE
       BLWP @NUMREF           * GET COLUMN
       BLWP @XMLLNK           * EXECUTE ROM ROUTINE
       DATA >12B8             * CONVERT FLOATING POINT TO INTEGER
       MOV  @FAC,R6           * PLACE COLUMN IN REGISTER 6

* CALCULATE SCREEN POSITION
       DEC  R5                * ROW-1
       DEC  R6                * COL-1
       LI   R7,40             * 40 CHARACTERS IN A ROW
       MPY  R5,R7             * (ROW-1)*40
       A    R6,R8             * SCREEN POSITION=(ROW-1)*40+(COL-1)

* GET THIRD PARAMETER FROM EXTENDED BASIC
       LI   R1,3              * USE THE 3RD VARIABLE
       LI   R2,STRING         * CPU RAM AREA TO PLACE STRING
       LI   R5,>FF00          * MAXIMUM LENGTH OF STRING = 255
       MOVB R5,@STRING        *
       BLWP @STRREF           * READ STRING FROM BASIC

* WRITE STRING AT SCREEN LOCATION
       MOV  R8,R0             * SCREEN POSITION
       CLR  R2                *
       LI   R7,STRING         *
       MOVB *R7+,R2           * (LENGTH OF STRING)
       SWPB R2                * NUMBER OF BYTES TO WRITE
       MOV  R7,R1             * LOCATION OF STRING IN CPU RAM
       BL   @SMBW             * MULTIPLE BYTE WRITE TO SCREEN

       B    @RETURN           * RETURN TO EXTENDED BASIC

MODE28 LWPI WS         * LOAD WORKSPACE REGISTERS

       BL   @CLRTXT    * CLEAR VDP RAM FOR SCREEN (TEXT MODE)

       LI   R0,>01E0   * CHANGE TO GRAPHICS MODE
       BLWP @VWTR      *
       SWPB R0         *
       MOVB R0,@>83D4  *

       LI   R0,>0717   * CHANGE TO BLACK (>1) ON LIGHT BLUE
       BLWP @VWTR      *  (>7)

       LI   R0,768     * RESTORE VDP INFORMATION FOR GRAPHICS
       LI   R1,VDPINF  *  MODE
```

-->

```
        LI    R2,192      *
        BLWP @VMBW        *

        B     @RETURN     * RETURN TO EXTENDED BASIC

RETURN  LWPI GPLWS        * LOAD GPL WORKSPACE REGISTERS
        CLR   @STATUS     * CLEAR STATUS BYTE
        B     @>0070      * RETURN TO EXTENDED BASIC

SMBW    MOV   R2,R3         * MY OWN SUBROUTINE FOR SCREEN
        MOV   R1,R2         *  MULTIPLE BYTE WRITE (REGISTERS ARE
SMBWR   MOVB *R2+,R1        *  USED IN SAME WAY AS VMBW)
        AI    R1,>6000      *
        BLWP @VSBW          *
        INC   R0            *
        DEC   R3            *
        JNE   SMBWR         *
        RT                  *

CLRTXT  CLR   R0          * SUBROUTINE TO CLEAR VDP RAM FOR THE SCREEN
        LI    R1,>8000    *  (TEXT MODE)
CT      BLWP @VSBW        *  >80 = >20 (SPACE) + >60 (OFFSET)
        INC   R0          *
        CI    R0,960      *
        JLT   CT          *
        RT

        END
```

Here is the Extended BASIC program (be sure you have the assembly language program above assembled and on a disk in drive one before attempting to run the test program):

```
100 ! 40D28 DEMO BY B. TRAVE
R, 835 GREEN VALLEY DR., PHI
LA., PA 19128
110 CALL INIT :: CALL LOAD("
DSK1.40D28/O")
120 CALL LINK("MODE40")
130 CALL LINK("DISPLA",1,10,
"TEXT MODE DEMONSTRATION")
140 CALL LINK("DISPLA",8,1,"
1234567890123456789012345678
901234567890")
150 CALL LINK("DISPLA",15,2,
"This screen is displayed in
 text mode.")
160 CALL LINK("DISPLA",24,1,
"Press a key to return to Ex
tended BASIC.")
170 CALL KEY(0,K,S):: IF S=0
 THEN 170
180 CALL LINK("MODE28")
190 ON WARNING NEXT
200 DISPLAY AT(11,1):"Now _y
ou_ try it!"
210 DISPLAY AT(15,1):"Row? (
1-24):" :: ACCEPT AT(15,14)S
IZE(2):R
220 DISPLAY AT(19,1):"Column
? (1-40):" :: ACCEPT AT(19,1
7)SIZE(2):C
230 DISPLAY AT(23,1):"Messag
e?"
240 LINPUT M$
250 CALL LINK("MODE40")
260 CALL LINK("DISPLA",R,C,M
$)
270 R=25-R :: CALL LINK("DIS
PLA",R,1,"Press a key to ret
urn to Extended BASIC.")
280 CALL KEY(0,K,S):: IF S=0
 THEN 280
290 CALL LINK("MODE28")
300 GOTO 200
310 END
```

-->

# SHOW NEWS

## The TI-FAIRE

The Chicago TI User's Group annually sponsors The TI-FAIRE, which features vendors from around the U.S., new products, lectures, contests, demo's, guest speakers and more. This year's event will be held Saturday, November 2, 1985, from 10 AM to 5 PM in the Ironwood Room at Triton College, River Grove, Illinois, a suburb of Chicago. This third annual TI-FAIRE has in the past attracted about 1500 99'ers. The group is going all out for this year's event and over 3000 users from throughout the U.S. are expected (we've heard from users from at least 12 states coast-to-coast who plan to attend).

A vendor list posted on the TI Forum (CIS) listed 17 of the 21 vendors who will participate:

Asgard Software
The TI Forum (CIS)
Hunter Electronics
Myarc
Computer Micro Products
J/D Limited
C & J Drives
Corporate Disk Company
Tomputer
Data Systems
Micro Format
Bytemaster Computer Services
Competition Computer Products
DataBioTics
Great Lakes Software
Millers Graphics
Thompson Software

User groups will also participate.

This will be the really big 99'er event of the year, with the list of those expected to attend reading like a Who's Who of the 99'er world! Most notable is that Myarc will be there with their latest products.

We hope to see you in Chicago! And, of course, we'll bring you a full report for those unable to attend.

# TI-WRITER

## New Default Colors for TK-Writer

STANDARD: 1A 2XB EA 3B 4B 5A 6B 7B 9A

Many users have expressed an interest in changing the default colors of TI-Writer. The default colors are in the TI-Writer cartridge and therefore cannot be changed (yes, the default for color change options can be changed in the first sector of EDITA1, but not the color default upon entering TI-Writer).

Mark Lopez has a nice solution. If you have the TK-Writer loader source code (available from DL3 of the TI FORUM), use the FIND function of the Editor to FIND >07F5 and change the F5 to your favorite colors. The F is the foreground color in Hex, add one to the Decimal equivalent to get a BASIC equivalent, and the 5 is the background color. Reassemble and you have it!

## Using the Myarc MEXP-1 RAMDISK and Print Spooler

STANDARD: 1A 2TW 3B 4B 5A 6B 7B 9A

For those of you who use TI-Writer extensively and have felt that the slow process of switching between Editor and Formatter stifled your creativity and/or challenged your patience, the Myarc MEXP-1 card is for you! The following are highly recommended or required items:

1) TI Peripheral Expansion Box
2) TI or Myarc RS-232 card
3) TI Disk Manager II or Myarc Disk Manager III
4) 2 or more disk drives
5) No other memory cards installed

Using the MEXP-1 is quite simple and we will describe herein how to set up for quick and easy TI-Writer sessions.

The following commands will initialize the MEXP-1 with 48K

-->

PARTitioned for RAMDISK and 48K PARTitioned for Print Spooler (any combination totalling 96K can be used, 480K if you add additional RAM chips), catalog the RAMDISK and establish the RAMDISK as Drive 1 (any drive up to 5 could be assigned, the EMulate Disk command is used) from BASIC (RDDIR is RamDisk DIRectory):

```
CALL PART(48,48)
CALL RDDIR
CALL EMDK(1)
```

Next, use the Disk Manager to copy your TI-Writer files onto drive 1, which is now the RAMDISK. If you have Include File's or utility programs you commonly use, copy those onto the RAMDISK also. Note that you should be copying from drive 2 (or higher) onto drive 1. You're ready to proceed as normal, except much faster!

Accessing the Editor will now take 1 second! Create your file as you normally would. If you want a permanent copy of the file, save it to drive 2, just as you probably normally do. Whoa, don't exit the Editor yet! Copy your text to drive 1, which again will take about 1 second!

Exit the Editor and boot up the Formatter. Incredible! Only 1 second again! Use the file saved to RAMDISK (drive 1) as the Input File, thereby avoiding floppy disk access entirely! The output device should be "SPPIO.LF" or "SPPIO.CR", the Print Spooler designations. According to the MEXP-1 manual, a serial output designation could be used, but we have tested only the parallel output.

Within an incredibly short time the TI-Writer menu will reappear and you'll be ready to return to the Editor! If you don't recall the name of the last file you were editing, you can usually access it again by simply using Recover Edit (just press <R> <E>), which is really useful in processing formatted documents, such as letters, so that you simply follow the previous form! If you do any disk accesses while the Print Spooler is operating, the Print Spooler will temporarily be rendered non-functional until the disk access is completed. While we're on the topic of the Spooler pausing, this will also occur when you QUIT to the main menu screen, so don't panic, just continue and in most cases the Spooler will start again. You will find some actions that do affect the printer output. For instance, turning the power off on the printer will result in the Spooler continuing to dump its contents as if the printer were still on (useful if you find you're wasting paper in printing a document). It is also noteworthy that the Print Spooler can be set up to be larger than the TI-Writer buffer, enabling you to Spool any single file created in TI-Writer very quickly! Spooling more than one large file may result in a delay in regaining control of the console, but certainly not as long a delay as you would encounter without the Print Spooler.

It is also important to note that the designation "RD" (RAMDISK) can be used throughout TI-Writer, which is especially useful if one does not wish to copy the TI-Writer files to the RAMDISK, but would still like to place files onto the RAMDISK.

Our experience has been that the MEXP-1 not only saves a great deal of time in multiple document processing, but is also very conducive to creativity, enabling the user to hold thoughts for a shorter period of time. We highly recommend the Myarc MEXP-1 card for use with TI-Writer (and elsewhere, too!).

---

## COMMUNICATIONS

### Solving the Lengthy File Problem

STANDARD: 1A 2TW TE 3B 4B 5A 6B 7B 9A 13A

Many of the terminal emulator programs in use today create files that are longer than the buffer space in TI-Writer. Many users use TI-Writer read their communications

-->

files, so that accessing the lines beyond the buffer space is often a difficult, hit or miss, proposition. The following Extended BASIC program will read the records of the file, counting the number of lines that exist:

```
100 DISPLAY AT(1,1)ERASE ALL
:"LINE COUNTER"::"FILENAME:
"
110 ACCEPT AT(4,1)BEEP SIZE(
14):F$
120 OPEN #1:F$,DISPLAY ,VARI
ABLE 80,INPUT
130 INPUT #1:A$ :: A=A+1 ::
IF EOF(1)=0 THEN 130
140 CLOSE #1
150 PRINT A
```

So, if there are 1205 lines to a file, you could use the following Load File commands:

| | | |
|---|---|---|
| 0 | 400 | DSK1.MYFILE |
| 401 | 800 | DSK1.MYFILE |
| 801 | 1200 | DSK1.MYFILE |
| 1201 | 1205 | DSK1.MYFILE |

While the "E" designator can sometimes be used with the Load File command, you should be particularly cautious in using it with lengthy files, as "E" designates the End of the memory file buffer, not the End of the file on disk or the End of the potential TI-Writer buffer space. In other words, if you have 10 lines in memory, E=10 for all operations.

We hope this clears up the problems so many of you have encountered in accessing large files.

---

## Communications News

STANDARD: 1A 2EA 3B 4B 5A 6B 7B 9A
13A

Version 1.14 of Paul Charlton's FAST-TERM will soon be available. In recent weeks, Paul has updated FAST-TERM to load the main program along with both the TE2 and XMODEM protocol programs in a single load (yielding access to both protocols instead of the single protocol previously available). To receive a copy of FAST-TERM, send at least 720 sectors of disks (SS/SD or DS/SD, we have not yet heard the sector count or content of Version 1.14) along with a disk mailer, labels and return postage. A $10 payment is customary for the FREEWARE program. Updates are available for $3.50. Write to Paul Charlton, 1110 Pinehurst Ct., Charlottesville, VA 22901.

XMODEM protocol has become the most common protocol of uploads to the TI FORUM on Compuserve™, as users have recognized that the speed and error-checking of XMODEM makes it far superior to TE2 or ASCII for most files. In addition to FAST-TERM, another XMODEM terminal emulator program is 4A/TALK, $19.95 plus $3.00 shipping and handling, from DataBioTics, P.O. Box 1194, Palos Verdes, CA 90274. 4A/TALK and FAST-TERM are widely regarded as by far the two best 99/4A terminal emulators.

In case you recently experienced difficulties in accessing the TI FORUM, the problem you likely had has now been resolved. CIS changed the "Page" of the FORUM to TEX-200 and did not immediately allow previous titles, such as TEXAS, TIFORUM or PCS-27. Though it is now recommended that you GO TEX-200, using GO with any of the names should now work. This problem was not the fault of the friendly and helpful Coordinators (Sysops) on the TI FORUM!

Many readers have asked about how to defeat their phone's Call Waiting feature because a second call coming in can abort a communications session. In many areas, disabling Call Waiting is still difficult. However, phone services are switching to a new system which provides, among other things, pressing "#70" to disable Call Waiting. Some major metropolitan areas, such as Los Angeles, already have such a service. Ask you local phone company if the service is available in your area.

---

-->

# MULTIPLAN™

## Using the eXTERNAL COPY Command

STANDARD: 1A 2MP 3B 4B 5A 6B 7B 9A

A key to progressing into advanced spreadsheeting is using two or more related sheets with one another. The eXTERNAL COPY command is used to copy all or a portion of a sheet into the current sheet.

There are two primary pre-requisites. First, the cells to be copied must have a NAME. Second, the cells into which the copy is to go must be BLANK. It is important to note that the NAMEd cells can occupy any rectangular space, including more than one Row and more than one Column (the multiple Column aspect seems to be a little known fact). Of course, you'll need the same dimensioned space to copy into as the rectangle from which you are copying.

The eXTERNAL COPY can be linked or unlinked. If the sheets are to be used together more than once, you'll likely want to use the LINK (YES) option. However, it should be noted that once a LINK is established, it is very difficult to eliminate. While the tendancy may be to think in terms of LINKing the current sheet to the eXTERNAL sheet, the reverse logic will likely make using LINK clearer. In most cases, the current sheet will be a sheet for formulae, with the eXTERNAL sheet containing data. You are linking the data to the formulae, not the formulae to the data, which may seem simply a matter of semantics, but thinking in such terms seems to help. Once the formulae are established and SAVEd, the data can be altered as frequently as one requires. Then, LOADing the formulae sheet will copy in your latest data. In other words, the BLANK stipulation mentioned above applies only to setting up the LINK, not to subsequent copying of the cells.

Using eTERNAL COPY certainly offers many possibilities. One notable consideration is that the sheets that you establish with our SYLK Builder 2.0 program can be copied into a formulae sheet. To do so, place your calculations, including any required MID and VALUE statements, in cells beginning in Column 6. Then you can eXTERNAL COPY the data into Columns 1 through 5. Such a procedure completes the cycle of writing data in TI-Writer, converting it to SYLK, converting the SYLK file to NORMAL, NAMEing the cells, then doing an eXTERNAL COPY. You may find the eXTERNAL USE command to be useful in such situtations. The eXTERANL USE substitutes a name for a supporting sheet, which would allow you to have several data sheets under different names that could still be eXTERNAL COPYd under a single filename.

The eXTERNAL LIST command is available to list all supporting and depending sheets related to the active sheet.

We hope this article clarifies how to use "data only" sheets with a sheet of calculations. If you have questions, let us know, as that is why we're here!

-------------------------------------

# GRAPHICS

## A GRAPHX Trick

STANDARD: 1A 2XB 4B 5A 6B 7B 9A

Many readers seem to haved missed seeing in Appendix 2 of the GRAPHX manual that an Assembly source file is on the disk (disk 2 for Extended BASIC version users) that loads a screen from outside the GRAPHX environment. The program is called READIN. Little Assembly knowledge is required! Change line 12 to the length of the filename (in hex). Change line 13 to the filename of your screen. Assemble and run from the LOAD AND RUN option of the Editor/Assembler. You'll no longer have to wait for GRAPHX to load to show off your artwork! This offers many possibilities for Assembly buffs!

-------------------------------------

-->

# 99 POTPOURRI

News, Corrections, Updates, Editorials, Kudos, and Come-what-may

The LA 99'ers Computer Group has had a very lively section in their monthly newsletter called "I Wish I Had:". The group's President, Teresa Masters, recently suggested that we try using such a section, responding with comments and fulfillments from our own staff and from other readers. This seems like a great idea, so we'll get it started this month with our own Wish:

W/1. A TI-Writer modification for accessing all of the Myarc MEXP-1's memory. Richard Mitchell, Editor.

If there are programs, tricks or items that you are seeking, let us know by writing to Super 99 Monthly, "I Wish I Had Department", 171 Mustang Street, Sulphur, LA 70663. Also let us know if you have any Comments or Fulfillments of someone's Wish. We'll print as many each month as possible.

---

Congratulations to Barry A. Traver for being selected as the newest Coordinator on the TI FORUM. By the way, Barry's diskazine has now been titled TRAVelER, available for $30 for 6 issues, 835 Green Valley Drive, Philadelphia, PA 19128.

---

Due to timing factors in our news coverage, this issue is late. We anticipate that our next issue will also be late, so that we can offer coverage of the TI-FAIRE. We hope to return to a more timely basis with our November issue, but will continue to place our primary emphasis on quality information.

---

## FREEWARE

To receive FREEWARE, send a blank disk (sometimes 2 are required), mailer, postage and address labels to the author. A contribution of $10.00 is expected if you like the program.

Mack McCormick, 215 A Yorktown, Ft. Lee, VA 23801, has MASTER CATALOG, a 100 Assembly language disk catalog program that is super fast. It handles up to 2000 different disk files. Order today, Mack will soon be moving overseas!

COMPACTOR, UNCOMPACTOR and TECHIE BBS are available from Monte Schmidt, 121 N. Blair, Madison, WI 53703. TECHIE BBS is obviusly a BBS program. COMPACTOR is an Assembly Language program that takes an uncompressed D/F 80 A/L program and compresses it to about 2/3 the disk space, saving disk space and yielding faster load time. UNCOMPACTOR reverses the compaction performed by COMPACTOR.

Tom Freeman, 515 Alma Real Dr., Pacific Palisades, CA 90272, has EASYSPRITE, an extremely fast Extended BASIC program with Assembly routines to create graphics and sprites with easy cursor control and saving for program insertion.

PILOT 99, a CAI (computer-assisted instruction) language, is available from Tom Weithofer, 1000 Harbury Drive, Cincinnati, OH 45220. Based around FORTH, this one is great for educators.

If you have written a useful program that you wish to have distributed under the FREEWARE concept, let us know about it! We'll see to it that your program is added to the many lists circulating!

```
[][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][]
[] Compuserve is a registered trademark of Compuserve Information Services. []
[] Multiplan is a registered trademark of Microsoft Corp.                   []
[] GRAM Kracker is a registered trademark of Millers Graphics.              []
[][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][]
```

-->

SUBSCRIPTIONS (PER YEAR):

| | |
|---|---|
| U.S. AND POSSESSIONS | |
| FIRST CLASS | $16.00 |
| THIRD CLASS | $12.00 |
| OTHER COUNTRIES | |
| AIR MAIL | $26.50 |
| SURFACE MAIL | $16.00 |

INDIVIDUAL COPIES:

| | |
|---|---|
| U.S. SUBSCRIBERS | |
| FIRST CLASS | $ 1.35 |
| THIRD CLASS | $ 1.00 |
| CANADA SUBSCRIBERS | $ 1.35 |
| OTHER | $ 1.50 |

Check or Money Order in U.S. funds.
(all issues available at press time)

## SUPER 99 MONTHLY ORDER FORM

NAME______________________________

ADDRESS___________________________

CITY____________________STATE_______

ZIP_________COUNTRY________________

For back issues, specify which:

__________________________________

READER FEEDBACK: (Attach comments)

Super 99 Monthly is published monthly by Bytemaster Computer Services, 171 Mustang Street, Sulphur, LA 70663. All correspondence received will be considered unconditionally assigned for publication and copyright and subject to editing and comments by the editors of Super 99 Monthly. Each contribution to this issue and the issue as a whole Copyright 1985 by Bytemaster Computer Services. All rights reserved. Copying done for other than personal archival or internal reference use without the permission of Bytemaster Computer Services is prohibited. Bytemaster Computer Services assumes no liability for errors in articles.

EDITOR
Richard M. Mitchell (CIS 70337,1011)

CORRESPONDING STAFF WRITERS

Barry A. Traver
Charles M. Robertson
Steven J. Szymkiewicz, MD

## STANDARD KEY

| | | | |
|---|---|---|---|
| 1 | Computer | A | TI-99/4A |
| 2 | Cartridge | XB | Extended BASIC |
| | | MP | Multiplan (tm) |
| | | EA | Editor/Assembler |
| | | TW | TI-Writer |
| 3 | RS-232 | B | TI |
| 4 | Disk Drive | B | TEAC 55B |
| 5 | Expansion Box | A | TI |
| 6 | Disk Controller | B | CorComp |
| 7 | Memory Card | B | Myarc MEXP-1 |
| 9 | Monitor or TV | A | TV & RF Modulator |
| 10 | Printer | A | Gemini 15-X PC |
| 12 | Speech Synthesizer | A | TI |
| 13 | Modem | A | Volksmodem |

Bytemaster Computer Services
171 Mustang Street
Sulphur, LA 70663

Bulk Rate
U.S. Postage
PAID
Sulphur, LA 70663
Permit No. 141

POSTMASTER: ADDRESS CORRECTION REQUESTED.