

SUPER 99 MONTHLY

TI-FAIRE A MAJOR SUCCESS MYARC AND MILLERS GRAPHICS DRAW A CROWD

SHOW NEWS.....	1
MULTIPLAN™.....	4
c99.....	4
FORTH.....	5
EXTENDED BASIC.....	8
LETTERS TO THE EDITOR..	10
99 POTPOURRI.....	11

SHOW NEWS

TI-FAIRE Draws Thousands to Chicago

While chilly winds whipped into most of the Windy City of Chicago, the atmosphere was warm on the campus of Triton College, where over 2000 99'ers attended the third annual TI-FAIRE on November 2. 99'ers from throughout North America packed the large facility within moments after the show opened. The show's sponsor, the Chicago 99/4(A) User's Group, is certainly to be commended for pulling off a seemingly impossible task after months of planning.

The event drawing the most attention was a talk by Lou Phillips, President of MYARC. Though MYARC had planned to demonstrate their new computer and Extended BASIC II, a blown CPU in the computer and a missing disk from the Extended BASIC backup of MYARC's hard disk dampened those plans. Maintaining his composure, Phillips officially announced the new computer and offered an outstanding discussion of the

architecture of the as yet unnamed computer and the structure of Extended BASIC II. The breadboard and console case and keyboard for the computer were displayed to the delight of those present. Phillips pointed out MYARC's history of being a leader in producing 99'er products, with the firm having always delivered quality products. Our investigations have disclosed that the firm continues to concern itself with properly tested items that can be upgraded to suit market needs.

Details of the computer announcement included release in the first quarter of 1986 at an estimated price of \$499. The computer will be much as we described it last month and Phillips disclosed many additional details. The processor will be a TMS9995 running at an incredible 10.7 Mhz., about 2.3 times as fast as the 99/4A's TMS9900. The computer will come with 256K of CPU memory, 64K of VDP, 64K of ROM, parallel output, RS-232 I/O and a Mouse port (true Mouse support). The ROM is defined with 48K of library routines, 8K of GPL to the interpreter and 8K of Mouse support. The VDP will be a 9938, which is upwards compatible with the 99/4A's 9918A. The 9938 supports a variety of graphics modes, including an 80 column display. Phillips indicated that the 9938 supports a few features that MYARC's 64K implementation will not accommodate and added a typical manufacturer's disclaimer for those hardware hackers

-->

intent on installing the full 192K that the 9938 supports. Other features available on the 9938 include both composite and RGB support, 10 horizontal sprites (as opposed to only 4 on the 9918A) and 32 planes (as on the 9918A). Phillips also indicated that, outside of obvious structural incompatibility, the new computer will be 99% compatible with existing 99/4A software. Linkage to the 99/4A's Peripheral Expansion Box will be via a narrow cable connected to an interface card that will include a real time clock, with the entire linkage being priced at approximately \$80.

Phillips also confirmed rumored IBM compatibility projects at MYARC, pointing out that the 84-key keyboard on display was obviously very similar to PC keyboards. The confirmation drew a very positive response from the audience. In conversations with this publication, Phillips has indicated that several popular PC software titles have already been targeted.

The most exciting portion of Phillips' presentation came when he promised open architecture for the new computer, with full documentation being available to programmers. The lack of open architecture was a major problem for TI in marketing the 99/4A, so it is a very positive note that MYARC will encourage the efforts of software authors.

Phillips also described MYARC's Extended BASIC II. The new Extended BASIC will function from 128K and will offer expanded memory allocations for the various segments of Extended BASIC. Extended BASIC II will remove operations from VDP that blocked access to fancy video modes and will include routines for Graphics mode (the only mode fully supported by TI Extended BASIC), Text mode and Bit-map mode. As we previously disclosed, Extended BASIC II will support TI Extended BASIC statements, many GW BASIC statements (such as DRAW and FILL) and will run 3 to 10 times faster than TI Extended BASIC by not

making use of GROM (GROM is programmed in the byte-efficient, but relatively slow GPL language). We will have extensive information on Extended BASIC II in upcoming issues.

Unquestionably the best new product shown at the TI-FAIRE was Millers Graphics' GRAM Kracker™, a hardware device with unprecedented capabilities. GRAM Kracker™ contains its own module port for easy saving of modules to disk, cassette, RAM Disk or hard disk, thereby providing a backup of the module. Utilizing 56K of lithium battery backed up RAM and GRAM (the READ/WRITE equivalents of the READ ONLY ROM and GROM in modules) expandable to 80K plus 8K of system ROM/GROM, the device plugs into the module port and is easy to operate. Production will be in accordance with strict guidelines, with the module interface, computer connection, control switches, logic to memory board and optional GRAM sockets being gold plated for high reliability and long life. Measuring 4.25 X 6.5 X 1.125 inches and weighing 2 lbs., GRAM Kracker™ is finished in classy black textured epoxy coating. GRAM Kracker™ also comes with a disk that includes a utility for writing BASIC programs that can be stored and executed in the module space, a utility to add new CALLs to Extended BASIC, a utility to allow moving the Editor/Assembler and TI-Writer to different GRAMs for multiple module storage, a utility for setting up a GRAM Disk for loading the Editor/Assembler and a utility for a GRAM Disk for loading TI-Writer. With 3 optional GRAM chips, overriding the operating system and BASIC's space will be possible for customizing your system. Craig Miller, the mastermind head of Millers Graphics, demonstrated GRAM Kracker™ in Chicago. The demonstration was, as the saying goes, totally awesome. Miller had the audience's full attention throughout, with the really unbelievable point coming when Miller showed his amended and often humorous Extended BASIC error messages, which brought roars of both approval and laughter. In early 1986, Millers Graphics will release a

-->

GPL Assembler (with a GPL Programmer's Guide) and a GPL Disassembler. Gram Kracker™ is available for \$174.95 plus \$4.00 shipping and handling (outside the U.S. shipping and handling is \$20, all funds quoted in U.S. dollars) from Millers Graphics, 1475 W. Cypress Ave., San Dimas, CA 91773.

Millers Graphics also announced their soon to be released book, "The Orphan Chronicles", by Dr. Ron Albright. The book is a frank look at the major figures of the 99/4A world. The author spent many hours conducting interviews for the book, often tracing the whereabouts of individuals for weeks.

Also present at the Millers Graphics booth were MG office coordinator Sue Miller, MG Explorer programmer Doug Warren and Tom Freeman. Freeman is rumored to be in the final stages of a program to be released by MG (probably by the end of the year).

Our selection for the best new software at the TI-FAIRE is TI-Artist Version 2.0, a great drawing program. TI-Artist features icon-driven selections for draw, point, line, k-line, rays, fill, frame, box, circle, disc, clear image, horizontal or vertical line, swap, invert, alpha/numeric, clear color, store, zoom, mirror, hard copy, 8 brush styles and color selections. Enhancement options include move and copy with and without color, slides and Monogram font character input. Printer output is available in a variety of densities and sizes. We'll have more coverage of the capabilities of this outstanding graphics package in future issues. Version 1.0 owners can exchange their disk for only \$5. The TI-Artist address is Insebot Inc., P.O. Box 260, Arnold, MD 21012.

Asgard Software displayed a full line of quality software at the show. The firm offers specialized databases, games, graphics enhancement packages and more. Making its debut at the TI-FAIRE was GRAPHX Companion II, a

program for use with GRAPHX. GC II has 14 character fonts available as clipart, 8 collections of clipart and 6 full pictures. Asgard's address is POB 10306, Rockville, MD 20850.

The list goes on and on, but we recommend you contact Thompson Software (3507 Murl, Muskegon, MI 49442), Dragonslayer Software (2606 Ponderosa Drive, Omaha, NE 68123), *Benial Traveler* diskazine (835 Green Valley Drive, Philadelphia, PA 19128) and RAM Disk Project Development Team (David R. Romer, Box 554, Walbridge, OH 43465). Romer is working with John Clulow and Ron Gries on a RAM Disk that is said to work exactly like a floppy disk while allowing use of almost any other peripheral device!

Covering the TI-FAIRE live on CIS were TI FORUM Sysops Jonathan Zittrain and Jim Horn, two of the friendliest men you'd ever want to meet. Also attending were Sysop Barry A. Traver and former Sysop Mack McCormick (who left CIS due to pending military duty in West Germany).

Above all, we salute the many people who attended the show, without whom such an event is not possible. Vendors reported that sales were brisk and people friendly. Both users and vendors indicated their eagerness for attending another such event.

Finally, we want to report on the many things that happen at a show that are often overlooked. There are social activities, sight-seeing tours and many blossoming friendships and business relationships that are all an integral part of a show. There's computer talk, pizza, disk swapping, pizza, jokes in the hotel room, pizza, plane rides and, yes, more pizza. It all adds up to a fantastic experience.

The TI-FAIRE, which, by the way, happened just over two years after "Black Friday" (TI's withdrawal from the Home Computer market, when we were all "supposed" to disappear), was an overwhelming success!

SYLK Builder 2.1

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

In our August issue, we revised SYLK Builder to directly overwrite the File Descriptor Record (FDR) of our SYLK file, thereby avoiding the use of a sector editor. Though the program worked, it used a crude sequential search to locate the FDR. This month, we have modified the program to search the File Descriptor Index Record located at Sector 1. This revision accomplishes two things. First, it greatly reduces the search time for a file that is near the end of a full or nearly full disk. Second, it allows you to select a SYLK filename that is alphabetically first or near first of the filenames on the disk to greatly improve the access time in searching for the proper FDR. To refresh your memory, byte 12 of the FDR must be modified to create a SYLK file.

While we were at it, we turned off most of the pre-scan, speeding up the start of the program by about 2 seconds. These changes are in lines 90, 1000, 21020 and 21465. You may want to make the changes to lines 21020 and 21465 by changing your R_A_W MERGE format file, then MERGE it back into your program.

As a reminder, your file to be converted must be in DIS/FIX 80 format or the program will generate an error.

Another interesting aspect of the SYLK format is that it does not accept ASCII 13, the carriage return. We are looking into possible methods of simulating a carriage return and will report on that as soon as we find a method that works on most printers. Using a carriage return will generate a non-fatal error, with the corresponding cell subsequently showing "#VALUE!". Using a carriage return followed by line feed (ASCII 10) will be read by the spreadsheet as a command that is out of place and will therefore create a fatal error (a fatal error is one that completely

prevents use of a program or file).

The changes that follow are additional lines or changes to existing lines. You should not delete any previously existing lines.

```

90 GOTO 100 :: A$,A1$,B$,B1$
,C,D,F1$,F2$,I,J,R$,S,S1,T$,
T1$ :: CALL WRITE :: CALL RA
W :: CALL LINK :: !@P-
390 S=1
401 CALL LINK("READ",D,1,A1$
,B1$)
402 IF S>64 THEN A1$=B1$
403 S1=256*VAL(STR$(ASC(SEG$
(A1$,S*2-1,1))))+VAL(STR$(AS
C(SEG$(A1$,S*2,1))))
410 DISPLAY AT(24,1):"READIN
G SECTOR ";S1 :: CALL LINK("
READ",D,S1,A$,B$)
420 IF (SEG$(A$,1,MAX(0,POS(
A$," ",1)-1))<>SEG$(F2$,3,10
))* (SEG$(A$,1,MAX(0,POS(A$,C
HR$(0),1)-1))<>SEG$(F2$,3,10
)) THEN S=S+1 :: GOTO 402
440 CALL LINK("WRITE",D,S1,A
$,B$)
1000 !@P+
21020 CALL LOAD(16360,87,82,
73,84,69,32,39,138):: !@P-
21465 !@P+

```

c99A 99/4A Compiler!STANDARD: 1A 2EA 3B 4B 5A 6B 7B 9B
10B

c99 is an implementation of a subset of the popular programming language C. Developed for the 99/4A, c99 is based on small-c, as published by Ron Cain in *Dr. Dobb's Journal* No. 45, May 1980. It is made available for only \$20 in the form of FAIRWARE (which we formerly called FREEWARE) by Clint Pulley, 38 Townsend Avenue, Burlington, Ontario, Canada L7T 1Y6. Send an initialized SS/SD disk, mailer and return postage to Pulley to receive c99. If paid response is adequate, further implementations will be made available as updates. The disk includes sample programs,

-->

including three versions of one program, in BASIC, c99 and the compiled version. The compiler produces 9900 Assembly source code with an option for c99 comments. By assembling the source code, assembly object code is produced.

Many 99'ers will certainly find c99 to be sufficiently different from other 99/4A languages to make it rather cumbersome at first. However, it is easier than Assembly Language and it does allow insertion of Assembly code within the c99 code, which is very useful in light of the fact that c99 does not support all of the C language. Though documentation is provided on the disk, the basics of the language are not, of course, explained and the author directs users to several useful books on C that are widely available.

As c99 uses a single-pass compile process, the code it produces is often not as efficient as would be produced by an experienced Assembly programmer. Even so, it runs the *Byte Magazine* prime number sieve benchmark in 68 seconds, compared to 175 seconds in TI FORTH and almost a day (4,000 seconds, we exaggerated a little) in console BASIC.

Users will likely ask if there is anything of immediate use when the c99 disk is first received, and indeed there is! A program called FCOPY on the disk copies D/V 80 files. What makes this so great is that it is very fast for printing DV/80 files, such as communications files, to your printer (including to the MYARC Print Spooler)! Output can also go to the screen (it scrolls very quickly, but pressing a key pauses the screen), enabling a user to spot whether the file called is indeed one to be printed! And, unlike what is available from the E/A or TI-Writer Editors, accessing the file can be terminated at any point (the Assembly program reads a record from the disk, outputs that record, checks keystrokes, then proceeds to the next record). Because the text is not buffered into memory, there are no restrictions on the size

of the file!

Clint Pulley spent months in developing c99 and by offering it as FAIRWARE is trusting that you will show your appreciation by sending money. If you use this software, do send \$20! The 99'er community needs programmers like Clint Pulley!

FORTH

Utilizing that Extra Hidden Memory

By

Howie Rosenberg

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

TI FORTH, when booted without any options at all, leaves about 16K of space available for both User Dictionary and the Parameter Stack. Ordinarily, this amount of memory is quite sufficient for almost any conceivable application. FORTH's virtual disk system can also be utilized in a variety of ways as additional memory both as a file structure for text or data or as a buffer area.

The program below is illustrative of another way to obtain additional memory, namely the use of the 8K of video memory available starting at >1400 in GRAPHICS or TEXT mode. The illustration shows how to move data to that area on boot, then FORGET the code that moved the data to that area of memory, and finally how to retrieve the data.

The extra 8K of space can be used to hold character sets as in the illustration, tables, Graphics sets, and any other kind of data, where you require faster access than afforded by disk transfer. The example used stores several additional character sets which can then be called for a quick change of screen characters from normal to italics, and several sets of large characters. The normal set is also stored in this area, as restoring the normal character set by a call to a GPLNK routine does not do nice things to some areas of CPU used by FORTH.

-->

In this example I have separated lower from upper case sets because of requirements for use in my applications. The words to locate the sets in video memory appear on screen 80 along with the words required to move sets into the character definition area of video memory. The remainder of the code on screens through the middle of screen 87 move sets of characters to our new storage space and then FORGET the compiler code and characters in the dictionary space. At the end of this process all definitions are in video memory and all that remains in the users dictionary is the code at the beginning of screen 80 and the executable words which illustrates the example (second half of screen 87).

The process of compilation is in this case slow. To speed up booting the application we will use the BSAVE and BLOAD utilities. First load any utilities that you require for your application. -SYNONYMS and -BSAVE are needed for this code to operate. Then proceed to LOAD screens 80 through 87. After these screens are compiled, the character sets are in video memory. Now type in the following or load from another screen.

```
: MARK ; * TASK HEX 0 VARIABLE TEMPBUF
820 ALLOT 1400 TEMPBUF 820 VMBR * TASK
16 BSAVE
```

This code saves the application including an array of the chardefs on screen >16 (arbitrary, any screen can be used). One final step and we're all set.

Add the following code to the boot screen (screen 3).

```
HEX 16 BLOAD TEMPBUF 1400 820 VMBW
FORGET MARK DECIMAL
```

The array is moved to video memory on boot, the original array forgotten and all executable words are ready for use.

NOTE: the character sets I used were originally part of several EXTENDED BASIC programs in which the author's name did not appear.

```
SCREEN 80
( CHARACTER SETS IN VDP MEM)
HEX 0 VARIABLE CHARBUF DO ALLOT 1400 CONSTANT CHARLOC ( UNUSED)
800 41 8 * + CONSTANT CAPLOC ( START OF CAP "A")
800 61 8 * + CONSTANT SMALLLOC ( START OF SMALL "a" )
800 30 8 * + CONSTANT NUMLOC ( START OF NUMBERS )
: SETLOC ( GROUP 0,1,2 ETC--STARTLOC IN VDP)
DO * CHARLOC + ;
CAPLOC CHARBUF DO VMBR CHARBUF 0 SETLOC DO VMBW ( SAVE CAPS )
SMALLLOC CHARBUF DO VMBR CHARBUF 1 SETLOC DO VMBW ( SAVE L.C.)
: MOVCAPS ( setnumber- ) SETLOC CHARBUF DO VMBR CHARBUF CAPLOC
DO VMBW ;
: MOVLOW ( setnumber- ) SETLOC CHARBUF DO VMBR CHARBUF
SMALLLOC DO VMBW ;
: MOVNUMS ( set#- ) SETLOC CHARBUF 50 VMBR CHARBUF NUMLOC
50 VMBW ; ( move 10 chardefs from set# to 0-9 defs)
NUMLOC CHARBUF 50 VMBR CHARBUF 6 SETLOC 50 VMBW ( SAVE NUM) -->
```

SCREEN 81

(CHARACTER SETS SCREEN 2- BIG CAPS IN SET 2)

3C24 VARIABLE TEMPBUF

247E , 6262 , 6200 , 7C44 , 447E , 6262 , 7E00 , 7E42 ,
 4060 , 6062 , 7E00 , 7E42 , 4262 , 6262 , 7E00 , 7C40 ,
 407C , 6060 , 7C00 , 7C40 , 407C , 6060 , 6000 , 7E42 ,
 4066 , 6262 , 7E00 , 4242 , 427E , 6262 , 6200 , 1010 ,
 1018 , 1818 , 1800 , 0202 , 0206 , 0646 , 7E00 , 4444 ,
 4478 , 6464 , 6400 , 4040 , 4060 , 6060 , 7E00 , 7F49 ,
 4969 , 6161 , 6100 , 7E42 , 4262 , 6262 , 6200 , 7E46 ,
 4242 , 4242 , 7E00 , 7C44 , 447C , 6060 , 6000 , 7E46 ,
 4242 , 4A46 , 7F00 , 7C44 , 447E , 6262 , 6200 , 7E42 ,
 407E , 0646 , 7E00 , FE10 , 1018 , 1818 , 1800 , 4242 ,
 4262 , 6262 , 7E00 , 4242 , 4262 , 6224 , 1800 , 4141 ,
 4169 , 6969 , 7F00 , 4444 , 3864 , 6464 , 6400 , 4242 ,
 427E , 1818 , 1800 , 7E42 , 0418 , 2042 , 7E00 ,

TEMPBUF 2 SETLOC DO VMBW FORGET TEMPBUF -->

SCREEN 82

(CHAR SETS SCREEN 3- LITTLE LOWER CASE)

0000 VARIABLE TEMPBUF

0030 , 0838 , 483C , 0000 , 2020 , 3824 , 2438 , 0000 , 0018 ,
 2420 , 2418 , 0000 , 0808 , 3848 , 4838 , 0000 , 0018 , 243C ,
 201C , 0030 , 2820 , 7020 , 2020 , 0000 , 1C24 , 241C , 0418 ,
 0000 , 2020 , 3824 , 2424 , 0008 , 0018 , 0808 , 081C , 0008 ,
 0008 , 0808 , 2818 , 0000 , 2024 , 2830 , 2824 , 0000 , 1808 ,
 0808 , 081C , 0000 , 001C , 2A2A , 2A22 , 0000 , 0038 , 2424 ,
 2424 , 0000 , 0018 , 2424 , 2418 , 0000 , 0038 , 2438 , 2020 ,
 0000 , 001C , 241C , 0404 , 0000 , 0028 , 3420 , 2020 , 0000 ,
 0018 , 2010 , 0830 , 0000 , 1038 , 1010 , 1408 , 0000 , 0024 ,
 2424 , 2418 , 0000 , 0022 , 2222 , 1408 , 0000 , 0022 , 2A2A ,
 2A14 , 0000 , 0022 , 1408 , 1422 , 0000 , 0022 , 1408 , 0808 ,
 0000 , 0038 , 0810 , 2038 ,

TEMPBUF 3 SETLOC DO VMBW FORGET TEMPBUF

-->

SCREEN 83

(CHAR SETS SCREEN 4 ITALICS CAPS)

001F VARIABLE TEMPBUF

1122 , 3E22 , 4444 , 001E , 1121 , 3E22 , 427C , 001F , 1020 ,
 2020 , 407C , 001E , 1121 , 2122 , 427C , 001F , 1020 , 3E20 ,
 407C , 001F , 1020 , 3E20 , 4040 , 001F , 1020 , 2E22 , 447C ,
 0011 , 1122 , 3E22 , 4444 , 001F , 0408 , 0808 , 107C , 0001 ,
 0102 , 0222 , 447C , 0011 , 1122 , 3C28 , 4444 , 0010 , 2020 ,
 2040 , 407C , 0037 , 294A , 5242 , 8484 , 0011 , 112A , 2A2A ,
 4444 , 001F , 1122 , 2222 , 447C , 001F , 1122 , 3E20 , 4040 ,
 001F , 1122 , 222A , 447A , 001F , 1122 , 3E28 , 4444 , 001F ,
 2020 , 3E02 , 027C , 001F , 0408 , 0808 , 1010 , 0011 , 1122 ,
 2222 , 447C , 0011 , 1112 , 2224 , 2438 , 0021 , 2142 , 4A52 ,
 94EC , 0011 , 110A , 0C14 , 2222 , 0011 , 1122 , 3E08 , 1010 ,
 001F , 0204 , 1820 , 407C ,

TEMPBUF 4 SETLOC DO VMBW FORGET TEMPBUF

-->

SCREEN 84

(CHAR SETS SCREEN 5-- NUMBERS)

(ITALICS SET 5) 001F VARIABLE TEMPBUF

1122 , 2222 , 447C , 0001 , 0102 , 0202 , 0404 , 001F , 0102 ,
 3E20 , 407C , 001F , 0102 , 3E02 , 047C , 0011 , 1122 , 3E02 ,
 0404 , 001F , 1020 , 3E02 , 047C , 001F , 1020 , 3E22 , 447C ,
 001F , 0102 , 0202 , 0404 , 001F , 1122 , 3E22 , 447C , 003F ,
 2142 , 7E02 , 047C ,

TEMPBUF 5 SETLOC 50 VMBW FORGET TEMPBUF

(BIG NOS. SET 6) 7E42 VARIABLE TEMPBUF

4242 , 4646 , 7E00 , 0808 , 0818 , 1818 , 1800 , 3E02 , 027E ,
 6060 , 7E00 , 3E02 , 021E , 0606 , 7E00 , 4042 , 427E , 0C0C ,
 0C00 , 7C40 , 407E , 0606 , 7E00 , 7C40 , 407E , 6262 , 7E00 ,
 7E02 , 0206 , 0606 , 0600 , 7E42 , 427E , 6262 , 7E00 , 7E42 ,
 427E , 0606 , 0600 ,

TEMPBUF 7 SETLOC 50 VMBW FORGET TEMPBUF

-->

SCREEN 85

(CHAR SETS SCREEN 6- 2ND BIG CAPS)

7C82 VARIABLE TEMPBUF

```
8282 , FE82 , 8282 , FC42 , 427C , 4242 , 42FC , 7C82 , 8080 ,
8080 , 827C , FC42 , 4242 , 4242 , 42FC , FE80 , 80F8 , 8080 ,
80FE , FE80 , 80F8 , 8080 , 8080 , 7C82 , 8080 , 9E82 , 827C ,
8282 , 82FE , 8282 , 8282 , FE10 , 1010 , 1010 , 10FE , 0202 ,
0202 , 0282 , 827C , 8890 , A0C0 , A090 , 8884 , 8080 , 8080 ,
8080 , 80FE , 82C6 , AA92 , 8282 , 8282 , 82C2 , A292 , 8AA6 ,
8282 , FE82 , 8282 , 8282 , 82FE , FC82 , 8282 , FC80 , 8080 ,
7C82 , 8282 , 829A , 847A , FC82 , 8282 , FC88 , 8482 , 7C82 ,
807C , 0202 , 827C , FE10 , 1010 , 1010 , 1010 , 8282 , 8282 ,
8282 , 827C , 8282 , 4444 , 2828 , 1010 , 8282 , 8282 , 92AA ,
C682 , 82C6 , 2810 , 1028 , C682 , 8282 , 4428 , 1010 , 1010 ,
FF02 , 0408 , 1020 , 40FF ,
```

TEMPBUF 8 SETLOC DO VMBW FORGET TEMPBUF

-->

SCREEN 86

(CHAR SET SCREEN 8- LARGE LCASE)

0000 VARIABLE TEMPBUF

```
3C42 , 0E12 , 261A , 4040 , 407C , 4242 , 427C , 0000 , 3C42 ,
4040 , 423C , 0202 , 023E , 4242 , 423E , 0000 , 3C42 , 427C ,
403C , 1C22 , 2070 , 2020 , 2020 , 0000 , 3242 , 423E , 027C ,
4040 , 407C , 4242 , 4242 , 0010 , 0010 , 1010 , 1010 , 0002 ,
0002 , 0202 , 423C , 4040 , 4850 , 6050 , 4844 , 1010 , 1010 ,
1010 , 1010 , 0000 , EC92 , 9292 , 9292 , 0000 , 7C42 , 4242 ,
4242 , 0000 , 7E42 , 4242 , 427E , 0000 , 7C42 , 427C , 4040 ,
0000 , 3C44 , 443C , 0406 , 0000 , 7C42 , 4040 , 4040 , 0000 ,
3C42 , 300C , 423C , 2020 , 7020 , 2020 , 221C , 0000 , 4242 ,
4242 , 423E , 0000 , 4444 , 2828 , 1010 , 0000 , 4242 , 5A5A ,
6642 , 0000 , 4224 , 1818 , 2442 , 0042 , 2212 , 0E02 , 023C ,
0000 , 7E04 , 0810 , 207E ,
```

TEMPBUF 9 SETLOC DO VMBW FORGET TEMPBUF

-->

SCREEN 87

(CHARSETS SCREEN 8- NUMBERS FOR BIG CHARS AND SOME END WORDS)

3C42 VARIABLE TEMPBUF

```
4242 , 4242 , 423C , 1030 , 7010 , 1010 , 107C , 7C82 , 0408 ,
1020 , 40FE , 7C82 , 023C , 0202 , 827C , 060A , 1222 , 42FE ,
0202 , FC80 , 80FC , 0202 , 827C , 7C80 , 80FC , 8282 , 827C ,
FE02 , 0408 , 1020 , 4040 , 7C82 , 827C , 8282 , 827C , 7C82 ,
8282 , 7E02 , 027C ,
```

TEMPBUF A SETLOC 50 VMBW FORGET TEMPBUF

```
: SWITCHCASES 1 MOVCAPS 0 MOVLOW ;
: NORMALSET 0 MOVCAPS 1 MOVLOW 6 MOVNUMS ;
: ITALICS-UC 4 MOVCAPS 5 MOVNUMS ;
: ITALICS-LC 4 MOVLOW 5 MOVNUMS ;
: SHADEDCAPS 2 MOVCAPS 7 MOVNUMS ;
: LARGESET 8 MOVCAPS 9 MOVLOW A MOVNUMS ;
: SMALL-LOW 3 MOVLOW ;
```

EXTENDED BASIC

TRANSLATING STRING COMMANDS FROM OTHER BASICS

by Barry A. Traver

STANDARD: 1A 2XB 9B

Although few books of BASIC programs for the TI-99/4A have been published in the last six months (except for some worthwhile titles from *COMPUTE!* Publications), there is a wealth of good material available that was written for other BASICS or "generic" BASICS. The good news is that many of these

-->

programs can be translated fairly easily to run in Extended BASIC on our machine. The bad news (and it's not so bad) is that -- although most commands require no change -- the same cannot be said for string commands, which must be handled in a different way on the TI-99/4A than in most other BASICs.

The purpose of this article is to provide a brief tutorial on translating string functions -- INST, LEFT\$, RIGHT\$, and especially MID\$ -- from other BASICs to their equivalents on the TI-99/4A. We can thus make use of good books of programs or subroutines written in other BASICs. Such books include, for example, John P. Grillo and J. D. Robertson, "SUBROUTINE SANDWICH" (New York: John Wiley & Sons, Inc., 1983) and its companion "MORE SUBROUTINE SANDWICH", which contain many useful subroutines in the areas of word processing, cryptography, mathematics, table management, statistics, table lookup, business, plotting, and other utilities. These two books provide a good example of material that was not written for the TI but can be very useful for the TI.

Provided that one exercises care in finding appropriate equivalents for string functions, all of these subroutines can be converted with little difficulty to run on the TI-99/4A as Extended BASIC subroutines (and almost all of them can be converted to run as Extended BASIC subprograms, which is even better!). The first thing that must be understood, however, is that there are two different meanings of MID\$, depending upon which side of the equals sign it appears in a program statement.

When the MID\$ function appears on the right side of the equals sign (the usual situation), it is an extraction function, whereas if it appears on the left side of the equals sign, it is a replacement function. Let's consider the former situation first. The statement $X\$ = \text{MID}\$(Y\$, A, B)$ extracts from the string Y\$ the sub-string which begins at position A and is B characters long. The result is called X\$. Here the MID\$ works exactly like the SEG\$ function on the TI-99/4A, so that the TI equivalent would be $X\$ = \text{SEG}\$(Y\$, A, B)$.

But what about the situation where MID\$ appears on the left side of the equals sign? You may see examples of this in Radio Shack, IBM, or Commodore programs. Here, as noted earlier, the MID\$ is a replacement function. The statement $\text{MID}\$(X\$, A, B) = Y\$$ takes the sub-string in X\$ which begins at position A and is B characters long, and replaces it with the first B characters of Y\$, thus changing the inner contents of X\$. The parameter B is optional and is assumed to be the length of Y\$ if it is omitted (a frequent situation).

The TI-99/4A equivalent for $\text{MID}\$(X\$, A) = Y\$$ (the simpler case) is CALL MID1(X\$, A, Y\$) where the subprogram MID1 is defined as follows:

```
30000 SUB MID1(X$, A, Y$) :: H=LEN(Y$)+A ::  
X$=SEG$(X$, 1, A-1)&Y$&SEG$(X$, H, LEN(X$)-H+1):: SUBEND
```

The TI-99/4A equivalent for $\text{MID}\$(X\$, A, B) = Y\$$ (the more complex case) is CALL MID2(X\$, A, B, Y\$) where the subprogram MID2 is defined as follows:

```
30010 SUB MID2(X$, A, B, Y$) :: H$=SEG$(Y$&RPT$(" ", B), 1, B) ::  
H=LEN(H$)+A :: X$=SEG$(X$, 1, A-1)&H$&SEG$(X$, H, LEN(X$)-H+1)::  
SUBEND
```

The latter version I have used rarely (if at all), whereas the former version I have found to be very useful, e.g., in reading the contents of the first half of a disk sector into a string, changing a particular byte in the string, and writing it back to disk.

Now, the TI-99/4A lacks the LEFT\$ and RIGHT\$ functions found on many other computers. The TI-99/4A equivalent for $\text{LEFT}\$(Y\$, A)$ is fairly simple: $\text{SEG}\$(Y\$, 1, A)$. The TI-99/4A equivalent for $\text{RIGHT}\$(Y\$, A)$ is slightly more complicated: $\text{SEG}\$(Y\$, \text{LEN}(Y$)-A+1, A)$.

A few computers define RIGHT\$ in a different way, so that rarely you may find the working equivalent of their RIGHT\$ to be $\text{SEG}\$(Y\$, A, \text{LEN}(Y$)-A+1)$. This is because they define $\text{RIGHT}\$(Y\$, A)$ not to mean the last A characters of Y\$ but

-->

to mean a string created by starting at the character at position A and continuing to the end of Y\$. To provide a specific example, on most computers RIGHT\$("SUBROUTINE",7) means "ROUTINE" while in a few others it means "TINE".

Finally, the TI equivalent of the INSTR function is very simple. P=INST(A,X\$,Y\$) on other computers is equivalent to P=POS(X\$,Y\$,A) on the TI-99/4A. With these TI-99/4A equivalents for LEFT\$, RIGHT\$, MID\$, and INSTR, you should find converting other BASIC programs to run on your TI to be a much easier task!

LETTERS TO THE EDITOR

Bob Boone
25 Ottawa St.,
Arprior, Ontario,
CANADA K7S 1W7
October 26, 1985

To the Editor,

Thank you very much for your response (July issue) to my last letter. Our TI National Users Association of Canada is growing daily and response to both of our freeware packages (DM-1000 and 2D-Graphics) is almost overwhelming. We'd appreciate your help with the following problem.

Our Ottawa TI-99/4A User Group feels its been grievously wronged by Tex-Comp in that they have used our freeware package (namely DM-1000) as a promotional item for another of their products. To add insult to injury, they altered our LOAD program to indicate it was rightfully distributed by them and at the same time deleted our 'freeware' notice! Also included in the Tex-Comp package (which one of our members ordered) was a notice that the Source code for DM-1000 could be purchased from them for the sum of \$7.50 U.S.

We would like your readers to know that our group purchased DM-1000 and its source code from Bruce Caron in May of this year and released it as freeware in June. We will, in fact, be releasing the first update to the program at the Chicago TI Faire on Nov. 2nd. Source and object code for it are available as 'freeware' to anyone contacting me through our User Group or directly at the address given above. Donations for the program can be made by cheque to The Treasurer (Ottawa TI User Group). We are not requesting a set amount for it as we feel users themselves should be the judge of its worth.

Our hope is that Tex-Comp will make reparation for this serious error in judgement on their part. If they don't, we hope you and the other TI publications will do whatever you can to try to make it right. Response from Tiers in England, Canada, Australia and the U.S. has indicated that the Freeware concept can work. I hope Tex-Comp will see the light too!

Sincerely,

(signature)

Bob Boone,
Past President
Ottawa TI-99/4A U.G.,
Box 2144 Station D
Ottawa, Ontario
Canada K1P 5W3

Ed.: Also enclosed was a letter to Tex-Comp. Bob, those are some pretty serious allegations. Unfortunately, we have been unable to obtain any comment from Tex-Comp, as the only telephone listing we have found is their order line. Equally unfortunate is that yours is not the first complaint we've heard about the firm.

99 POTPOURRI

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

CORRECTIONS:

SEPTEMBER: On page 9, our reference to disabling Call Waiting should have read "#70".

On page 9, there is a better way of finding the end of a TI-Writer file. Barry Traver suggests using "9999" as the terminating reference will read the file until the buffer is full.

On page 10, the SYLK format does not support the eTERNAL COPY procedure. Use LOAD to fill a portion of the sheet, then LOAD a SYLK file into a blank portion of the sheet. Or, SAVE the SYLK file in NORMAL format and then access it with the eTERNAL COPY command.

On page 11, the description of Mack McCormick's MASTER CATALOG should have read "a 100% Assembly language disk catalog program".

Mark your calendar and make your plans! LA 99'ers is planning TI-Fest West '86, a show to be held March 1 and 2 at the Shrine Auditorium in Los Angeles. The show will be held in conjunction with the commercial Sell-A-Thon show, with the Sell-A-Thon being on the main floor of the Shrine and 99'ers participating on the mezzanine level, thereby giving those attending the opportunity to view and purchase both 99/4A products and other computer items, such as disk drives, printers, modems and much more. LA 99'ers would like to coordinate discounts for hotel rooms and tours of Disneyland and the movie studios, so contact the group early if you want to participate in such programs. The address is LA 99'ers, P.O. Box 3547, Gardena, CA 90247-7247.

[] Multiplan is a registered trademark of Microsoft Corp. []
[] GRAM Kracker is a registered trademark of Millers Graphics. []

Though this issue is very late due to our coverage of and participation at the TI-FAIRE, we now have enough material for at least two more issues and will release the November issue by the end of November, followed by the December issue by mid-December.

I WISH I HAD:

2W. A program to make it simple to build 1 to several Assembly language screens for access from Extended BASIC. John Singleton, Westlake, LA.

John, we've seen some similar programs, but none exactly like what you describe. We'll either locate one or write one for you.

COMING NEXT MONTH:

FORTH and Assembly! The requests have been overwhelming. You asked, we'll deliver!

Have you been looking for a fantastic deal on bulk diskettes? NCC has 100% guaranteed (and they do stand behind them) DS/DD diskettes for only \$.66 in 100 lots or \$.56 in 1000 lots plus UPS and C.O.D., which comes to about \$71/100. The disks come in bags of 25 with reinforced hubs, jackets and write protect tabs. The address is NCC, 3375 Scott Blvd. - Suite 422, Santa Clara, CA 95054. We ordered some and also checked with others who have ordered. Experience has been that less than 1/2 % fail to initialize and all that have initialized have held data properly.

SUPER 99 MONTHLY ORDER FORM

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)

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	Module	XB	Extended BASIC
		MP	Multiplan (tm)
		EA	Editor/Assembler
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 (128K)
9	Monitor or TV	B	TI Color Monitor
10	Printer	B	Gemini 15-X

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.
 RUSH -- TIME DATED MATERIAL.