

HOCUS

8406 (021)
Milwaukee

Home Computer Users Spotlight

A monthly publication of the Milwaukee Area 99/4 Users Group

JUNE, 1984

SST TO DEMO NEW EXPANDED BASIC COMPILER AT OUR JUNE MEETING

SST Software of Cedarburg has recently announced a new addition to their software line for the TI 99/4A. If all goes as planned, SST will be at the June meeting to demonstrate it's features.

In a product announcement received by the newsletter staff, SST proclaimed that "the SST EXPANDED BASIC COMPILER contains all of the features of the SST BASIC COMPILER, plus most of the features of EXTENDED BASIC." SST claims that some compiled commands are up to 160 times faster than interpreted ones. Output from the compiler, is in the form of TMS9900 machine language which can be stored permanently on disk. To utilize the compiler, the following system peripherals are required: memory expansion, a disk drive, and either Editor/Assembler or Mini-Memory.

SST has significantly improved their old BASIC compiler by adding many new capabilities. It features a full compliment of graphics capabilities including sprites and access to Bit Map mode graphics. It provides the ability to access a disk drive and the RS232 interface. In addition, it provides the ability to link several programs together into one large system.

Other features provide for the following:

1. The ability to write and compile up to 470 lines of BASIC code.
2. The ability to call up to 7 compiled programs from a TI BASIC program.
3. The ability to dimension up to 1800 element floating point arrays or up to 12000 element integer arrays.
4. The ability to do integer and floating point arithmetic.
5. The use of strings for input, output and character definition.
6. The ability to write and debug a BASIC program using the TI console and interpreter, and then compile it without the need for

retyping.

7. The ability to create and add up to 6 of your own commands.
8. The ability to use the TI Loader to quickly load compiled programs.

The SST EXPANDED BASIC COMPILER comes with a 60 page manual and goes for a price of \$95.00. SST Software Inc. is located at, Box 26, Cedarburg, WI 53012. If you have need to call them, they can be reached at, 414/771-8415.

ZAP, CRACKLE, AND POP

There are several different sources of electrical interference that can plague the home computer user and his semiconductor based equipment. The problems produced by line interference range from flickering or shrinkage of your screen image to crashed programs or erased ROM's. Obviously the severity of damage depends on both the type and strength of interference as well as the kind of equipment you are using. Modern microprocessor controlled devices are notoriously finicky when it comes to the quality and quantity of power they are fed.

Powerline problems basically come in three varieties: blackouts or brownouts, voltage transients, or noise. Unfortunately, only the last two of these can be economically controlled by the home user.

Blackouts are total power outages which can cause loss of data both within a running program and on diskettes while being accessed. Blackouts may be the result of problems with the utility, i.e. mechanical failure or not paying one's bill, or household problems, i.e. blown fuses or children pulling out plugs. Brownouts are similar though not as extreme. Usually they are caused during the summer when demand exceeds capacity and the utility reduces the supplied voltage by 5 to 15%.

Voltage transients are by definition short term increases or decreases in the supplied voltage. Transients are of two types: spikes, high energy and very short duration, and surges or sags, longer duration increases or decreases in voltage. Sags and surges are commonly encountered

(continued on page 3)

Home Computer Users Spotlight

HOCUS is published monthly by the Milwaukee Area 99/4A Users Group, 2007 N. 71st Street, Wauwatosa, WI 53213. The Milwaukee Area 99/4A Users Group is an association of individuals with a common interest in using and programming Texas Instruments 99/4A Home Computers. The Milwaukee Area 99/4A Users Group is not affiliated with Texas Instruments Inc., nor any other commercial organizations.

HOCUS is published for the members of the Milwaukee Area 99/4A Users Group and is composed of articles written and donated by user group members. Opinions expressed by the authors do not necessarily represent those of HOCUS. Any article appearing in this publication may be reproduced providing credit is given to the author and to HOCUS.

MEMBERSHIP INFORMATION

Membership is open to individuals and families who are interested in using and programming the Texas Instruments 99/4A Home Computer. The membership includes access to both this newsletter and to the user group library. Annual dues are: Individual, \$8.00; Families, \$12.00. To join, see the Treasurer at any of our monthly meetings.

MEETING INFORMATION

The Milwaukee Area 99/4A Users Group meets on the LAST SATURDAY of each month in the lower level of 7500 W. State Street in Wauwatosa. MEETING TIME IS 1:00 TO 4:00 P.M..

SPECIAL NOTE: Due to a scheduling conflict during 1984, the MAY and DECEMBER meetings will be held on the third Saturday of the month. Meeting time for May 1984, will be from 4:30 to 7:00 p.m..

USERS GROUP OFFICERS:

President

Jim Vincent
782-9353

Vice-president

Milton Giessen
251-2864

Treasurer

Karan Chole
242-5238

Corresponding Secretary

Gene Hitz
453-0499

Recording Secretary

Judy Brown
677-2894

USER GROUP LIBRARY:

Librarian

Steve Sanders
546-1821

NEWS LETTER COMMITTEE:

Managing Editors

Tom Kruse
475-1159

Mike Milde
784-0479

Contributing Editors

Jim Kunding
541-1999

Steve Tjensvold
962-4924

TINFO...TINFO...TINFO...TINFO...TINFO...TI
INFO...TINFO...TINFO...TINFO...TINFO...TIN
NFO...TINFO...TINFO...TINFO...TINFO...TINF
FO...TINFO...TINFO...TINFO...TINFO...TINFO
O...TINFO...TINFO...TINFO...TINFO...TINFO.

Are you looking for information about your computer? Well so are your newsletter editors and WE NEED YOUR HELP. This month we are starting what we hope to develop into a semi-regular feature article. We will be reviewing various sources of information pertaining to the TI Home Computer. After reading this month's column, it is our hope that many of our members will contact us and let us know about sources of information which they have found useful.

HOME COMPUTER MAGAZINE, published monthly by Emerald Valley Publishing Co., P.O. Box 5537, Eugene OR 97405. Recent subscription price is \$25 for 12 issues. It is probably the largest magazine devoted to the Texas Instruments product line. Although it no longer focuses exclusively on TI, it has a great deal of useful information pertaining to the 99/4A. About 200 pages in length, this magazine also represents the users of the APPLE, IBM, and COMMODORE computers. It provides reviews of new products and software for each machine. It also has articles on user groups, LOGO, and technical aspects of the four computers. HCM provides several printed programs ready to type in and run using both BASIC and Extended BASIC languages.

CREATIVE COMPUTING, published monthly by Ahl Computing, Inc., P.O. Box 5214, Boulder, CO 80321. Recent subscription price \$19.97 for 12 issues. Approximately 300 pages of articles, hardware and software reviews, applications, and modifications. It carries regular feature departments covering APPLE, ATARI, COMMODORE, IBM, LOGO, and TRS-80. Usually contains at least one article pertaining to TI products. Also has a classified section that is filled with ads for TI 99/4A software. Usually contains several general purpose articles that can prove useful to the owners of the TI Home Computer.

Know of any other publications of interest to TI owners? If you do, please provide your input to Jim Kunding on the Newsletter Committee at any one of our meetings.

**** WANTED ** BUY ** SELL ** TRADE ****

** F O R S A L E ** PRINTER - GORILLA BANANA (PIO).
ASKING \$140.00. FRANK V. HOME 251-5232. WORK 355-3601

WANT TO BE FAMOUS? YOU TOO CAN BE AN AUTHOR JUST LIKE THE REST OF US. HOCUS WILL ACCEPT AND PUBLISH YOUR ORIGINAL PROGRAMS, REVIEWS, EDITORIALS, OR ARTICLES OF GENERAL INTEREST. TYPEWRITTEN COPY IS A MUST BUT TAPE OR DISKETTES WOULD BE GREATLY WELCOMED. PLEASE CONSULT TOM KRUSE OR MIKE MILDE IF YOU WISH TO KNOW MORE ABOUT PROVIDING SOMETHING FOR OUR NEWSLETTER. PLEASE NOTE THAT THE EDITORS RESERVE THE RIGHT TO EDIT EVERYTHING.

USERS GROUP MEMBERS CAN PLACE THREE LINE ADS LIKE THIS ONE FOR FREE. JUST CONTACT ANYONE ON THE NEWSLETTER COMMITTEE TO TAKE ADVANTAGE OF THIS FREE BENEFIT.

INTERESTED IN PLACING AN AD LARGER THAN THREE LINES? OUR RATES START AT \$10 FOR 1/4 PAGE. IF YOU ARE AFTER TI OWNERS, WE CAN REACH THEM. ASK US ABOUT HAVING YOUR AD MAILED TO OUR USERS. CONTACT EITHER OF THE MANAGING EDITORS LISTED ABOVE FOR MORE INFORMATION.

ZAP, CRACKLE, AND POP continued

by home users and are produced when large loads are brought on line such as air conditioners, refrigerators, or other electric motors. Disk drives and monitors are most adversely affected by sags and surges as manifested by lost data and shrinking and blooming displays.

Voltage spikes are short duration events of less than 100-200 microseconds but of up to several thousand volts. Spikes are typically produced by turning off transformers or motors, short circuits or lightning. The CPU, RAMs, and ROMs in your computer or peripherals are easily disrupted by spikes.

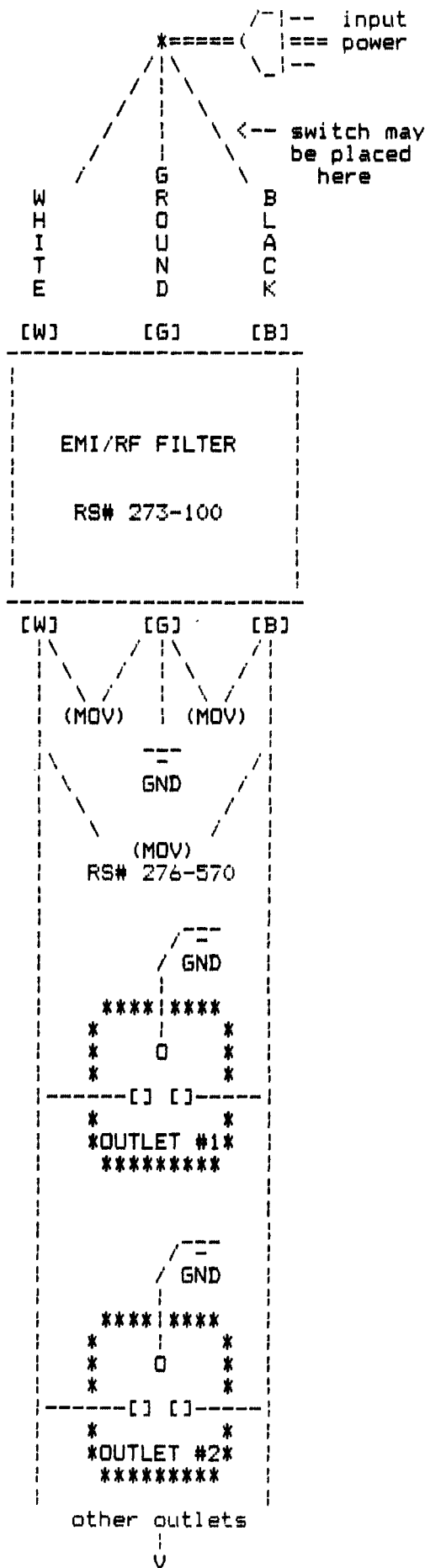
Electromagnetic interference (EMI) and Radio-frequency interference (RF) are the types of powerline noise most commonly encountered by the home user. Powerline noise of this type is usually in the 10kHz or greater range. Problems caused by this type of interference are similar to that produced by spikes.

Fortunately it is possible to fairly effectively deal with powerline problems. The easiest, and most expensive, is to purchase a commercial filter/spike protector. The other way is to build your own.

You can start either with a purchased unprotected power strip, the kind with 4 to 6 outlets, or build the entire box from scratch. Either way you will need 3 MOV's (metal-oxide varistors) available from Radio Shack at \$1.79/pair, (obviously since you'll need 3) and an EMI/RF filter also from RS at about \$11.95. Wiring all this together is really simple requiring only a soldering gun to do it right.

The filter is wired into the circuit first with connections on each side of the device for the black (hot) lead, common (white) lead and ground (bare or green) lead. The MOV's are two-legged affairs which are connected as follows: the first is connected one leg to the hot lead and one to the common lead, the next is connected with one leg to the hot and one to the ground, the last is connected between the common and ground.

The accompanying diagram should make this all crystal clear. If not, you'll have a great excuse to buy a new computer. MM



FORTH ARRAYS cont'd
by J.W.Vincent

This month we'll continue our discussion of arrays by investigating their compile and run time behavior. Before we get to that though, there is some old business to cover. I am sure I have caused a great deal of confusion with my last article due to a silly oversight. I forgot that the FORTH word @ (fetch) is a special control code to TI Writer, thus it is missing from the definitions of the FORTH words covered last time. The final versions of C_ARRAY, W_ARRAY, and the fetch commands are listed below. Sorry for the inconvenience.

I have also received some questions on the word DOUBLE-FORTH in my first article. It does not need to be saved! Its purpose is to modify your disk to be compatible with the Disk Manager II Module, and once it's been executed exact copies of your double sided FORTH disk may be made using Disk Mgr II. Another item from the first article, the challenge ... I received no responses. I'm not surprised though, it takes a lot of digging to figure out how simple double sided formatting is, and none of us have Cor-Comp's double density card yet! Anyway to get FORTH to format a double sided disk, edit screen 33 and change the FORMAT-DISK word to:

```
: FORMAT-DISK 1+ 258 33616 ! 18 SYSTEM ;
```

Now to the subject at hand, FORTH ARRAYS. Last time we defined the following words and operators for two dimensional character and word arrays:

```
: C_ARRAY <BUILDS DUP C, # ALLOT DOES> ROT 1 - OVER @ # + + ;  
r c C_ARRAY name ( defines a char array, r rows by c columns)  
r c name C@      ( fetch char from array row r column c)  
r c name C!      ( store char to array row r and column c)  
  
: W_ARRAY <BUILDS DUP , 2 # # ALLOT DOES> ROT 2 - OVER @ 2 # # + + ;  
r c W_ARRAY name ( define a word array of r rows by c columns)  
r c name @       ( fetch a word from array row r and column c)  
r c name !       ( store a word to array row r and column c)
```

Now let's look at what one of these array words does at compile time. Since both are quite similar we'll just look at the character array. Assuming we want to use a 10 row by 15 column array named WINDOW we would code:

```
10 15 C_ARRAY WINDOW
```

As we discuss C_ARRAY's operation, follow along thru each word in it's definition above. First 10 then 15 are pushed onto the stack. Next C_ARRAY builds a dictionary header for the word WINDOW. Then the top stack value (15) is duped, one of the 15's is stored in the next dictionary byte and the other is multiplied by 10 to alot the next 150 bytes of dictionary space for the array. Finally, a pointer to what WINDOW does is put in WINDOW's dictionary header (the code following DOES only resides in the dictionary entry for C_ARRAY).

Now let's store an 'A' in the fourth character of the fifth row in array WINDOW.

```
65 5 4 WINDOW C!
```

The interpreter places the three values and the address of the array on the stack, then it executes the code field for WINDOW. First the 5 (row #) is rotated to the top of the stack and decremented by one. Next the array address is copied over the top of the stack and the zeroth character is fetched from the array. This zeroth character contains the width (# of columns) of the array, which gets multiplied by the decremented row number (in this case $15 \times 4 = 60$). This value and the column number (4) get added to the array's base address to give us the address of the byte in which to store the value of the character 'A'. Finally the standard FORTH word C! (character store) is executed.

From this example you should be able to understand how arrays work in FORTH as well as being able to use them. More importantly, I hope this example has opened your eyes to the power and elegance FORTH provides with it's ability to define new defining words.

FORTH DEMO PROGRAMS
by CIN-DAY members
Ed York and Rick Mirus

The following two FORTH screens were published in a recent CIN-DAY newsletter. They have kindly granted us permission to reprint these clever examples of the power of FORTH and the 99/4A. The first draws a pattern of lines in the bit-map mode. To run it, first load -VDPMODES and -GRAPH then SCR #31 (or wherever you saved it) and enter A9. The second screen is an interrupt driven clock which will display on your FORTH screen even while you execute other instructions or screens. A warning, since the clock is interrupt driven and some tasks shut off the interrupts (notably I/O tasks) the clock cannot be reliably used in an application program.

SCR #31

```
( GRAPHICS EXAMPLE 1 )
( ED YORK )
( CIN-DAY USER GROUP )
: A0 GRAPHICS2 0 DMODE ! ;
: A1 17 0 DO I 8 # 191 128 191 I 8 # - LINE LOOP ;
: A2 17 0 DO 255 I 8 # - 191 128 191 I 8 # - LINE LOOP ;
: A3 17 0 DO 128 I 8 # + 191 255 191 I 8 # - LINE LOOP ;
: A4 17 0 DO 128 I 8 # - 191 0 191 I 8 # - LINE LOOP ;
: A5 17 0 DO 0 I 8 # 0 128 0 I 8 # + LINE LOOP ;
: A6 17 0 DO 255 I 8 # - 0 128 0 I 8 # + LINE LOOP ;
: A7 17 0 DO 128 I 8 # + 0 255 0 I 8 # + LINE LOOP ;
: A8 17 0 DO 128 I 8 # - 0 0 0 I 8 # + LINE LOOP ;
: A9 A0 A1 A2 A3 A4 A5 A6 A7 A8 :
```

SCR #33

```
0 ( Clock - to start enter hour 1 - 24 and minute and TIME )
1 0 VARIABLE XX 8 ALLOT 58 XX 2 + C! 58 XX 5 + C! 0 VARIABLE TT
2 : #UPDATE 1 TT +! TT @ 59 > IF 0 TT !
3 XX 7 + 1 OVER C@ + DUP 58 < IF SWAP C! ELSE DROP 48 SWAP C!
4 XX 6 + 1 OVER C@ + DUP 54 < IF SWAP C! ELSE DROP 48 SWAP C!
5 XX 4 + 1 OVER C@ + DUP 58 < IF SWAP C! ELSE DROP 48 SWAP C!
6 47 TT !
7 XX 3 + 1 OVER C@ + DUP 54 < IF SWAP C! ELSE DROP 48 SWAP C!
8 XX 1 + 1 OVER C@ + DUP 58 < IF SWAP C! ELSE DROP 48 SWAP C!
9 XX DUP C@ 1 + SWAP C! ENDIF XX C@ 50 = XX 1 + C@ 52 = + 2 = IF
10 48 48 XX C! XX 1+ C! ENDIF ENDIF ENDIF ENDIF ENDIF
11 XX 22 8 VMBW ENDIF ;
12 : TIME 10 /MOD 48 + XX 3 + C! 48 XX 6 + C! 48 XX 7 + C!
13 48 + XX 4 + C! 10 /MOD 48 + XX C! 48 + XX 1+ C!
14 INTLNK @ ' #UPDATE CFA ISR ! -31804 ! ;
15 : STOPCLOCK 0 -31804 ' ;
```

This is a clock program written in TI FORTH. To set the clock enter the hours and minutes then enter TIME. The word STOPCLOCK will stop it. Here is a line by line description of how it works:

LINE 0: Remark. Not required.

LINE 1: Variable TT will be used to count tenths of a second. Variable XX will be 8 bytes long. 2 bytes for hour, a colon, 2 bytes for minute, a colon, and 2 bytes for second. The time is stored in ASCII so it won't have to be converted before writing to the screen. Char 58 is a colon.

LINE 2: The word #UPDATE is the main portion and is responsible for keeping track of time and displaying it on the screen. The word #UPDATE will be executed 60 times per second. Each 1/60th of a second 1 is added to TT. If TT is greater than 59 then 1 second has elapsed and clock is updated by lines 3 to 10.

LINE 3: XX + 7 (units seconds) is incremented by 1. if greater than 9 (ASCII 58) then it is made equal to zero (ASCII 48) and tens seconds is updated.

LINE 4: Update XX + 6 (tens seconds)

LINE 5: Update XX + 4 (units minutes)

LINE 6: Correction factor. Adds 470 milliseconds to clock every ten minutes to make up for inaccuracy in clock. This can be changed from 0 to 59 to slow or speed up the clock.

LINE 7: Update XX + 3 (tens minutes)

LINE 8: Update XX + 1 (units hours)

LINE 9: Update XX (tens hours). If the maximum time has been reached then reset to all zeros. To change from a 24 hour clock to a 12 hour clock change 50 to 59 and 52 to 50.

LINE 10: Change time to all zeros.

LINE 11: Display time on screen. XX is memory location of time data. 22 is the screen location to start writing. Changing this value will change where the clock appears on the screen. 8 is the number of characters to be displayed. This could be changed to 5 to display hours and minutes only.

LINE 12: Create word TIME which breaks up starting time into tens and units and stores the ASCII value (add 48) in variable XX.

LINE 13: Continuation of line 12

LINE 14: Set up ISR so that the word #UPDATE will be executed 60 times per second. Chapter 10 page 3 of FORTH book explains this.

LINE 15: Word STOPCLOCK stops ISR by putting a zero into memory location -31804 (hex 83C4). Use STOPCLOCK before using COLD or bit map modes.

TI-TEAM BOARD PASSES 1 MONTH MARK

A new TIBBS(tm) bulletin board for TI 99/4A owners owners, TI-TEAM, has now been in operation for over one month. To date, over 850 calls have been logged. TI-TEAM is run by a local hardware/software dealer called COMP 'U' SERV. In addition to a variety of user messages, the board contains the latest COMP 'U' SERV price lists and a section describing our users group and meeting schedule. Future plans for the board include providing a section that contains portions of our newsletter.

The idea of TIBBS (TI Bulletin Board Service) is not unique to Milwaukee. Other cities also have TIBBS boards available. Below is a listing of these boards which was extracted from the TI-TEAM board using the Terminal Emulator II cartridge.

TIBBS(tm) SYSTEMS NATIONWIDE BY AREA CODE
(current as of May 17, 1984)

City	NOTES	Area Code	Phone Number
Tom's River New Jersey	Jeanete Shader	201	929-8161
Portland Maine	Mark Rideout	207	797-5690
Reading Penn.	Gene Deisher	215	929-5348
Philadelphia Penn.		215	T.B.A.
Washington D.C.	SYSOP/PHIL	301	434-0117
Newark Delaware		302	T.B.A.
Denver Colorado		303	T.B.A.
West Palm Florida	Dave Sholdar	305	793-8050
Clawson Michigan	Craig Barton	313	288-0530
Detroit Michigan	Computer Corner	313	544-7788
Ann Arbor Michigan		313	T.B.A.
Westbury New York		315	T.B.A.
Wichita Kansas	Jerry McClusky	316	681-3167
Atlanta #1 Georgia	"THE ORIG. TIBBS" Ralph Fowler	404	425-5254

Atlanta #2 Georgia	Info. Assoc.	404	928-4278
Ringgold Georgia	Robert Stone	404	935-4054
Pittsburgh Penn.	Mid-Bam Su-Th mid-noon Fr,Sa	412	854-5575
Appleton Wisconsin	Marc Schmidt	414	739-5380
Freedom Wisconsin	F.C.U.G. BBS Randy Gonnering	414	788-9730
Milwaukee Wisconsin	TI-TEAM DAN GUNIA	414	649-TEAM
San Fran. California	6pm-Midnite Mark Stephens	415	355-3092
Gresham Oregon	Mike Werstlein	503	661-0408
Knoxville Iowa	Keith Jamison	515	842-2104
Albany New York	Dick Ferrigan	518	765-4993
San Diego California		619	T.B.A.
Fontana California	Peter Covert	714	350-8583
Tonawanda New York	Peter Testa	716	837-6635
N Tonawanda New York	Mo-Fr 3-9pm 9am Sa-9am Mo	716	692-7289
Va. Beach Virginia	HAS USERS FEE Jos. Randall	804	486-1484
St. Pete Florida	Frank Watts	813	525-7998
St. Pete Florida	Ken Hunt	813	526-1265
Kansas City Missouri	John Drago	816	444-4163
Gainesville Florida	Joe Wnuk	904	T.B.A.
St Simons Georgia		912	T.B.A.
Raleigh N Carolina		919	T.B.A.

All systems 24 Hours except as noted.
T.B.A. - To Be Announced.

For information about TIBBS, write to:
TIBBS; PO BOX 383; KENNESAW, GA 30144

HELPFUL HINTS FROM DIGITHEAD!

ESCAPE FROM OLD OR SAVE CS1

Several of the out-of-town User Group newsletters we have received mention pressing "E" (or SHIFT E, if Alpha Lock is up) to escape from an OLD CS1 or SAVE CS1 command and retain the program in memory, after the first prompt is shown. In the case of the SAVE CS1 sequence, this procedure always works, with the message, " * I/O ERROR 66 * " being displayed. However, it will not be successful with OLD CS1 if one is using Extended BASIC, unless a "full" TI system is attached!

Note that the Extended BASIC Owner's Manual states, on page 129, that the OLD command "closes any open files and removes the program currently in memory before loading." Therefore, on a console with Extended BASIC only, when one would press "E", the message " * PRESS CASSETTE STOP THEN PRESS ENTER * " appears, then after ENTER is pressed, only the message " * I/O ERROR 56 * " is shown, and the program is gone! (TI told you so!)

In nominal BASIC, just the opposite is true. Page III-12 of its manual, in the lower right-hand section, has a statement saying that the "OLD command does not clear program memory unless the loading operation is successful." In fact, the procedure mentioned above will result in TWO error messages: " * WARNING CHECK PROGRAM IN MEMORY * " and " * I/O ERROR 56 * ". The program is still there, since apparently Texas Instruments uses the same message for this situation as well as for the case of an interrupted cassette program/data load.

Furthermore, with Extended BASIC, disk drive, memory expansion and RS232 card, the procedure *does* escape the operation and preserve the program, but a *different* error message results upon use of ENTER: " * I/O ERROR 03 * ". "Digithead" can't explain that case yet, but at least one won't be embarrassed by the friend with only Extended BASIC who asks why this tip fails!

-By Abdallah Clark

Software Review By Peter Radike
DATABASE 300
International Users Group

I must admit that I had been searching far and wide for a quality database program for my 99/4A.

After looking at all of the commercial grade software for other computers, I was quite disappointed at the products for the TI 99/4A....Until now.

Database 300/500 is a medium to high quality database program designed exclusively for the TI 99/4a by the T.I. International Users Group in Oklahoma. What made this program particularly appealing to me was the price...\$14.95!!!

I had seen the 3 other commercial software packages available for the TI at an average price of over \$200.00. While I wanted a good program, I didn't want it enough to pay that kind of money (I have been known to be tight).

An article in the February, 1984 issue of *Family Computing Magazine* gave an excellent account of what you need to know about databases:

1. How easy is it to use?
2. What features should one have? (such as sorting capabilities; how many files can it retain).
3. What machine will it run on, and what is the cost of the program.

Database 300/500 is a program which is offered in two Extended BASIC formats, one for single-sided and another for double-sided disks.

The single-sided version (Database 300) holds 300 files, and I'm sure you can guess how many the double-sided (Database 500) version can hold.

The program is fully menu-driven. In addition, it is easy to use if you make sure you read the massive twelve page instruction manual first.

You can operate Database 300 or 500 with only one disk drive, although two of them speeds up entries considerably. The main entry screen consists of 8 lines for whatever information you want to record and in whatever format you need.

Each line consists of TI's usual 28 character display. The first line, as well as a secondary line, can be used to "key" your files for sorting to your hearts desire - and it isn't too slow at doing it either (remember, this is only a \$14.95 program!!).

For the money, as well as for the TI owner who wants a database package of respectable quality, without spending twice your console's cost, then database 300/500 is for you.

On another note, this article was originally composed on the Extended BASIC word processing program found in the December, 1983 issue of *COMPUTE!* Magazine. I will probably have a review of this for the July users group meeting. -PR

PROGRAMMING CONTEST GRAB-BAG

This is one of several programs entered in our recent programming contest. It creates a block letter banner lengthwise on printer paper.

```
100 ! *****
120 ! * B A N N E R *
140 ! *****
150 ! BY:
160 ! JAMES STEINHART
170 ! MILW. AREA 99/4 U.G.
190 ! REQUIRES:
200 ! EXTENDED BASIC AND
210 ! PRINTER
220 !
230 CALL CLEAR :: A$="012345
6789ABCDEF" :: DIM B$(8),C$(
255):: CALL SCREEN(9):: CALL
CHAR(128,"FFFFFFF"):: FOR
A=1 TO 12 :: CALL COLOR(A,2,
16):: NEXT A
240 CALL COLOR(0,9,9,13,9,15
):: CALL VCHAR(1,1,31,48)::
CALL VCHAR(1,31,31,48):: CAL
L HCHAR(3,3,128,28):: DISPLA
Y AT(2,5):"--> B A N N E R <
--"
250 DISPLAY AT(5,5):"PRINTER
```

```
? FIO" :: ACCEPT AT(5
,21)SIZE(-7):D$ :: OPEN #1:D
$ :: DISPLAY AT(9,5):"MESSAG
E?" :: ACCEPT AT(13,1):E$ ::
PRINT #1:CHR$(14)&"BANNER -
"&E$ :: PRINT #1:CHR$(12)
260 FOR A=1 TO LEN(E$):: CAL
L CHARPAT(ASC(SEG$(E$,A,1)),
C$(A)):: NEXT A :: FOR B=1 T
O LEN(E$):: FOR B=0 TO 7 ::
F$,G$="" :: C=0 :: G$=SEG$(C
$(A),B*2+1,2)
270 C=(POS(A$,SEG$(G$,1,1),1
)-1)*16+POS(A$,SEG$(G$,2,1),
1)-1
280 C=C/2 :: F$=STR$(-(C-INT
(C)<>0))&F$ :: C=INT(C):: IF
C THEN 280
290 F$=SEG$(RPT$("00",8),1,8
-LEN(F$))&F$ :: H$(B+1)=F$ :
: NEXT B :: FOR B=1 TO 8 ::
I$="" :: FOR D=8 TO 1 STEP -
1 :: IF SEG$(H$(D),B,1)="1"
THEN I$=I$&RPT$(SEG$(E$,A,1
),B)ELSE I$=I$&RPT$(" ",B)
300 NEXT D :: I$=RPT$(" ",10
)&I$&" " :: FOR E=1 TO 5 ::
PRINT #1:I$ :: NEXT E :: NEX
T B :: NEXT A :: PRINT #1:CH
R$(12):: CLOSE #1
```

MILWAUKEE AREA USER GROUP - AT THE LIBRARY

