MSP 99 USER GROUP

Vol. 7 No. 7
September, 1985

THE MSP 99 NEWSLETTER

"Flower Power et al in TI Forth"

by Glenn Davis

This month I am presenting a tiny demonstration program to you. I had to find one that was small enough to publish, but still did something interesting. The one I chose is a translation of an Apple "HCM one-liner" to TI Forth. The routine is one screen long, but needs two screens of support code to allow Forth to use the sine and cosine functions without resorting to floating point. (I did try using floating point, but it ran 29 times slower: and that's using my CODE definitions of the -FLOAT words!) Richard Lauhead (another MSP member) added a few features to this routine too. Among them, that you must press clear to get back to text mode after the final "flower" is drawn. By the way, you must have the -GRAPH2 routines loaded (or -64SUPPORT) to run the one-liner.

The Apple one-liner had a few "hidden" bugs in it, which we might as well discuss here. That author louped from 0 to 359 to generate the sine and cosine values. I do the same thing here. Do you see what the difference is? In BASIC (and floating point on micros in general), the arguments are supposed to be in RADIANS. (A Radian is about 58 degrees of a circle) This made the dots skip around so the "flowers" materialized instead of being drawn. I tried doing the same thing, but found Forth was so fast that the drawings weren't visible long enough to enjoy them!

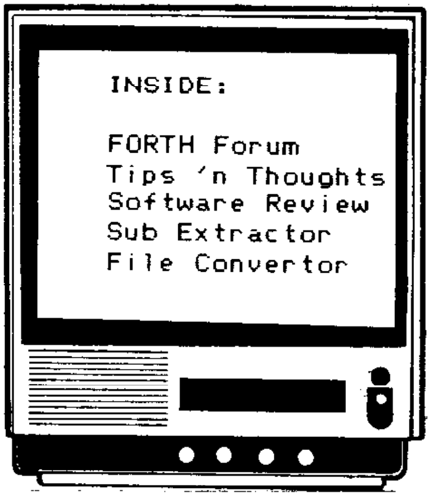
Adventuring In Familiar Surroundings

AN MSP 99 SOFTWARE REVIEW

by Gary Gese

This month, I'd like to tell you about a new addition to the MSP 99 Software Library. It's a new text adventure game that is somewhat different from the usual games of this type. It is writen by a regular contributer to these pages, MSP 99'er Mike Kabala. It operates in basic and requires 32K Memory Expansion and Disk system, and consists in 4 seperate files and uses 128 sectors on the disk.

(Continued on Page 5)



The MSP 99 USERS GROUP meets each month for discussions and presentations that enable its members to be better informed about their computers. Users group members share and exchange information. Some members have a broad range of computer expertise, others are just beginning. We are not affiliated with or sponsored by any other group or company. Membership dues are $12 a year for a family, $10 for an individual, and $50 for a sponsor member. You're welcome to visit a meeting before you join. Call or write for more information.

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

MSP 99 USERS GROUP
P.O. BOX 12951
ST. PAUL, MN 55112, U.S.A.

PRESIDENT: Dick Dunbar 488 0153
VICE PRESIDENT: Ed Neu 425 8744
SECRETARY: Gary Gese (no phone)
TREASURER: Jeff Hogden 227 2378

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

NEWSLETTER EDITORS:

Gary Gese

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

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

COMMERCIAL ADVERTISEMENT RATES: Busines firms that wish to communicate with our members may do so by placing an advertisement in the newsletter. Rates are: Full Page (7 1/2 X 10 1/2) $40; Half Page (3 1/2 X 10 1/2 or 7 1/2 X 5) $30; Quarter Page (3 1/2 X 5) $22. Each ad must be camera ready in one of the sizes indicated and paid in advance. Inserts (printed by the advertiser on 8 1/2 X 11 or 8 X 10) may be inserted in the newsletter at $20 per sheet. Contact the editors for information.

CHANGE OF ADDRESS: Before you move, please mail a change of address to the Users Group at the address listed in this issue. Please allow at least 1 month for processing.

SOFTWARE CONTEST UPDATE

Well, everything's over but the shouting. This years software contest comes to a close with an unusual ruling by our Software Chairman, Steve Gonnella.

Many fine programs were submitted in this years contest making the judging difficult. Our Teasurer, Jeff Hogden, submitted 2 programs; one a checkbook helper, the other a base number converter. Equipment Chairman George Madline submitted a program that prints Display/Variable 80 files without TI-Writer, while Mike Kabala and Glen Davis each submitted FORTH programs. Some of Mikes other entries include a graphic demo, a trip planner for bicycling, and a unique text adventure game.
(Ed. See this months software review.)

Paul Boyd submitted 5 different programs including a Football record keeper, a BASIC Memo writer, a calculator program, and a time clock. And, from Gary Gese comes a computerized Christmas card, and a BASIC and XBASIC word processor. The judging committee was hard pressed to select a winner.

As you can see, nearly all entrants had submitted more than one program in more than one category. It turned out that there were a total of 8 entrants in 9 different categories. So, in an unprecedented move, Steve declared all entrants winners, and each winner was awarded a box of disks. That's right, a full box of disks went to each person that entered a program in the 3rd Annual MSP 99 Software Contest. Now don't you wish you'd entered.

EASY RETURN ADDRESS LABELS

Here is a quick and easy way to get return address labels. Insert your name, address, etc. on Line 150. From the Eugene Users Group.

```
100 CALL CLEAR
110 OPEN #1:"PIO"
120 PRINT "READY TO PRINT LABELS."
130 INPUT "HOW MANY? ":N
140 FOR P=1 TO N
150    PRINT   #1:"NAME":"ADDRESS":
"CITY,STATE,ZIP" : : : :
160 NEXT P
170 CLOSE #1
```

MSP 99 Calendar of Events

SEPT 17: Extended BASIC -- This month we take an indepth look at this powerful extension of our computers BASIC language; how is it different, and what it's capabilities and limitations are.

OCT 15: ANNUAL Module Swap Auction -- This is it. The night many of you have been waiting for. Members are encouraged to bring down any Hardware or software that they're not using any more for the auction.

NOV 19: FORTH -- This time around, we dig into the mysteries of this extremely powerful programming language. What is FORTH? What are the system requirements? What can it do? And Can I learn to use it? Find out this month.

Subgroup Meetings

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

BUSINESS--Second Tuesday of month, 7-9 p.m., Call Bob DeMars (554-6219) for details.

EDUCATION--At monthly MSP 99 meetings.

YOUTH GROUP--At monthly MSP 99 meetings.

Committee Chairs

EQUIPMENT --
George Madline (341 3780)

NEWSLETTER --
Gary Gese (529-3989)

PUBLICITY --
Dave Wunderlin (544-8266)

SOFTWARE --
Steve Gonnella (533-8494)

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

News Flash - As of October 1985, the MSP 99 Users Group is going On-Line. That's right, the Mound TIBBS has offered us a space on its operating system. Be sure to check us out. For those of you that may have missed it last month here is the number. 472-3490

Several months ago, the Perth TI Users Group published the complete plans and wiring diagrams for adding 32K Memory directly into your console. Rather than reprint this extremely informative article in our newsletter (it would fill an entire issue) we have decided to instead offer photocopies of the original to any member for a modest $2.00 including postage. Contact the Newsletter staff to order your copy.

Beginning in September, MSP 99 members and guests attending the regular monthly meetings will be asked to sign in at the door when they arrive. There are many reasons for doing this; one is so we can keep track of the number of members attending the meetings. Another is to alert us to the presence of prospective new members. Your cooperation will be most appreciated.

DISPLAY/VARIABLE 80 to PROGRAM CONVERTER

Article written by Al Kinney

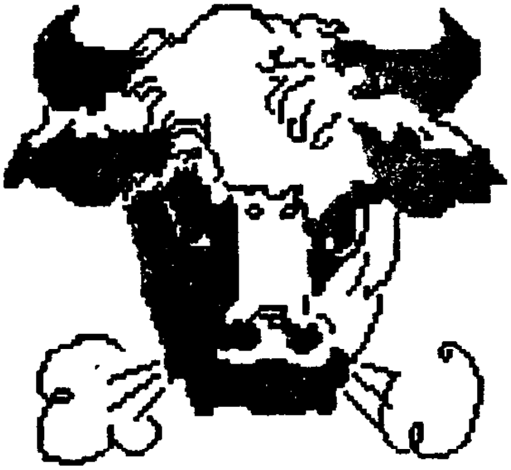
Here is a nifty program to convert text that has been stored as DIS/VAR 80 back into a runnable program.(The origins of the program are uncertain to me, but, here it is!)

Now why, you might ask, would you want to do that? Glad you asked!! And now, I'm gonna' tell ya!! If you use COMPUSERVE, there is a Special Interest Group (SIG) for TI home computers, with over 1000 members!! These wonderful folks are constantly trying to out-do each other by putting excellent Public Domain software into the Down Load area of the SIG. Think of it a Mail Box, and the files are NOT bills.

Now, the only problem you have, is that for lots of very technical reasons, the files are stored as text in the DIS/VAR 80 format. In the "OLDEN" days, whenever you downloaded a file, you had to sit and type the darn thing in, and the way I type, I created more errors than I could fix!! Now, all that drudgery is gone, and by simply running the program listed below, you can recreate the program in the MERGE (DIS/VAR 163) format! Then, by removing the exclamation (REM) marks from each line, you will have a MERGEable file.

This also give you another way to edit existing files you may already have. Let's say you have a program, and you want to make a LOT of changes to it! First, you would simply load the program into memory, as usual, by entering "OLD DSK1.NAME". When loaded, you would then enter "LIST "DSK1.NAME/1"". That would cause the program to be written to the disk in DIS/VAR 80 format. The reason for changing the name should be obvious, to prevent over-writing the original file! Now, instead of flailing through the entire program, line by line, looking for variable XYZ, you can use TI-WRITER or EDITOR/ASSEMBLER, and do "Global" search and replace's! Those features are described in the respective manuals. After you have modified the program as needed, you simply run the conversion program, which rewrites the file with a NEW name. When it is finished, type "NEW", and "MERGE "DSK1.PROGRAM/2"", and proceed as before!

```
100 !!!!!!!!!!!!!!!!!!!!!!!!
110 !!  TRANSLATES FROM    !
120 !! DIS/VAR80 TO MERGE  !
130 !!       FORMAT        !
140 !!!!!!!!!!!!!!!!!!!!!!!!
150 !
160 !Use a "Full Screen"
170 !Editor to create Ext-
180 !ended Basic programs
190 !
200 !Create a file using TI-
210 !Writer \\ Make sure you
220 !disable the word wrap
230 !mode and limit the line
240 !length to 80 characters
250 !
260 !
270 CALL CLEAR
280 DISPLAY AT(3,7)BEEP ERAS
E ALL:"!!!TRANSLATE!!!"
290 DISPLAY AT(7,5):"DIS/VAR
 80 FILENAME:"
300 ACCEPT AT(9,5)SIZE(15):I
N$
310 DISPLAY AT(12,5)BEEP:"ME
RGED OUTPUT FILENAME?:"
320 ACCEPT AT(14,5)SIZE(15):
OUT$
330 OPEN #1:IN$
340 OPEN #2:OUT$,VARIABLE 16
3
350 LINPUT #1:L$
360 S=POS(L$," ",1)
370 ON ERROR 490
380 N=VAL(SEG$(L$,1,S))
390 ON ERROR 440
400 A=INT(N/256)
410 A$=CHR$(N-A6):: PRINT L$
420 PRINT #2:CHR$(A);A$;CHR$
(131);SEG$(L$,S&1,80);CHR$(0
)
430 GOTO 350
440 PRINT #2:CHR$(255);CHR$(
255)
450 CLOSE #2
460 PRINT : :"ENTER ""NEW""
AND THEN ""MERGE""THE TRANSL
ATED FILENAME:":"          ";OU
T$: : :
470 PRINT "REMEMBER TO REMOV
E THE       LEADING ""!"" IN
EVERY
LINE
   ": : : :
480 END
490 ON ERROR 440
500 RETURN 350
```



PROGRAMMING AIDS

Subroutine Extractor

For those of you that have gotten tired of extracting subroutines that you want from long programs, either by hitting FCTN-3 (Delete) to delete lines or by typing in the subroutine from scratch, here is an answer.

This program only works in Extended BASIC. Save it in Merge format, then add it to the beginning of your program. That is the reason for the line numbers being 1-6. If your program already contains those line numbers, resequence it first. After merging, type RUN. The screen will prompt you for the starting and ending line numbers of the subroutine.

When the program ends, type "LIST" and you should only see the lines you wanted. The final step is to save the result as a merge file. This is accomplished by typing "DSKx.Filename",Merge. Do not save the result in normal fashion because the missing lines of the program are still in memory and will save. From George Steffen (printed in the LA 99ers Newsletter.)

```
1 CALL CLEAR :: CALL INIT :: INPUT
"LINE NUMBERS OF ROUTINE TO BE
SAVED: FIRST,LAST? ":L,M :: G=256
:: CALL PEEK(-31952,H,I,J,K)

2 C=INT(M/G):: D=M-C*G ::
F=(J-G)*G+K :: FOR E=(H-G)*G+I TO F
STEP 4 :: CALL PEEK(E,A,B):: IF A=C
AND B=D THEN 4

3 NEXT E :: PRINT :"LINE";0;"NOT
FOUND!" :: STOP !@P-

4 H=INT(E/G):: I=E-(G*H):: H=H+G ::
C=INT(L/G):: D=L-C*G :: FOR E=E+4
TO F STEP 4 :: CALL PEEK(E,A,B)::
IF A=C AND B=D THEN 6 !@P-

5 NEXT E :: PRINT :"LINE";N;"NOT
FOUND!" :: STOP !@P-

6 E=E+3 :: J=INT(E/G):: K=E-(G*J)
:: J=J+G :: CALL
LOAD(-31952,H,I,J,K):: STOP !@P-
```

Review...continued from Page 1.

One of the things that is different about this game is it's story line. The game starts off naturally enough in your living room. The TV set is on and an anouncer is telling you about a software contest that the ABC Personable Computer Company is sponsoring. You are told that you can win fame and fortune if you can win the contest. In the next room of course is an ABC Personable Computer.

While this game does not include any terrifying monsters to defeat, it does contain many elaborate puzzles that must be solved before the ultimate goal, winning the software contest, is achieved. Yet though the game is complex, Mike says that everything can be figured out logically.

When asked about his unique choice of story line, Mike explained that he remembered that his teachers used to caution him to write about things that are familar. I congratulate him on his choice.

Another item of note is that unlike 2 other text adventure games in our Software Library "Jackpine Savage" or "Cave Maze", Mikes adventure does not use John Todds' "Adventure Skeleton". Instead. Mike wrote the entire program from scratch. Everything in the program is his own creation including the parser which like infamous Infocom games will accept more than 2 word commands, and more than one command at a time. At one point while playing, I entered a total of 19 commands at once and while I did have to wait a little bit for the computer to do its stuff, it performed perfectly.

I recommend this game for all adventure fans, especially for those players that are new to the genre. The game is easy to play, yet complex enough to interest the veteran player. Try it, I'm sure you'll like it.

PARAMETER PASSING BETWEEN EXTENDED BASIC PROGRAMS

by Mike Kabala

One of the more advanced features available to the Extended BASIC programmer is the ability to chain programs by using the RUN statement within a program. This allows the creation of software that would not fit into the memory all at once. When you're done with one program segment, just "RUN" the next one.

Unfortunately, there is one bug. The RUN statement clears out all variables even if it is executed from within a program. Type in the following 2 programs and you will see for yourself. Be sure to save the first as "DSK1.DEMO1" and the second as "DSK1.DEMO1A" and then load and run "DSK1.DEMO1".

```
100 CALL CLEAR
110 DISPLAY AT(1,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(2,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "DSK1.DEMO1A"
```

```
100 CALL CLEAR
110 ! Variable has been
120 ! cleared by RUN
130 ! statement
140 PRINT A$
```

See what I mean? Somewhere between DEMO1 and DEMO1A, the value of A$ got lost. Now you could, if you wanted to, create a file, store A$ in it, and then read A$ from that file after entering DEMO1A, but that seems rather wasteful if you only need to pass one or two variables. In the remainder of this article, I will show you two other methods I have found to get around the problem.

The first method will work even if all you have is a console and cassette recorder (although you'll have to change all of the program names to "CS1"). That method is to use the screen as auxiliary memory. Just display the information you want before executing the RUN statement and your chained program will be able to take the data right off the screen as demonstrated below.

Load DEMO1 back into your computer and modify its RUN statement to chain in DEMO2A as shown below. Then save the first program as "DSK1.DEMO2" and the second one as "DSK1.DEMO2A". Finally, load and run "DSK1.DEMO2" and what you type in the first program should be correctly printed by the second.

```
100 CALL CLEAR
110 DISPLAY AT(1,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(2,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "DSK1.DEMO2A"
```

```
100 ! Read what last program
110 ! left on the screen.
120 ! Offet of 2 is needed
130 ! for difference between
140 ! ACCEPT and GCHAR
150 ! statements
160 A$=""
170 FOR I=3 TO 30
180 CALL GCHAR(2,I,A)
190 A$=A$&CHR$(A)
200 NEXT I
210 ! Parameter has been
220 ! passed using the
230 ! screen as auxiliary
240 ! memory
250 CALL CLEAR
260 DISPLAY AT(5,1):"YOU TYPED:"
270 DISPLAY AT(6,1):A$
```

For this second method you will need to have the 32K memory expansion installed. The trick here is to use the space reserved for an assembly language program for temporary storage. This has the advantage that you don't need to display something to pass it to the next program. Be sure to use the memory betwen 9984 and 16184 after executing CALL INIT because the rest of memory is already in use by your program and other things.

Save the first program as "DSK1.DEMO3" and the second as "DSK1.DEMO3A". Then run "DSK1. DEMO3".

```
100 CALL CLEAR
110 PRINT "TYPE SOMETHING:"
120 ACCEPT A$
130 !
140 ! Reserve space for
150 ! assembly language
160 ! program
170 !
180 CALL INIT
190 !
200 ! Store length of string
210 !
220 A=LEN(A$)
230 CALL LOAD(9984,A)
240 !
250 ! Store string
260 !
270 FOR I=1 TO A
280 CALL LOAD(I+9984,
    ASC(SEG$(A$,I,1)))
290 NEXT I
300 CALL CLEAR
310 RUN "DSK1.DEMO3A"
```

```
100 CALL CLEAR
110 PRINT "YOU TYPED:"
120 !
130 ! Get string length
140 !
150 CALL PEEK(9984,L)
160 !
170 ! Get string
180 !
190 X$=""
200 FOR I=1 TO L
210 CALL PEEK(I+9984,X)
220 X$=X$&CHR$(X)
230 NEXT I
240 !
250 ! Print string
260 !
270 PRINT X$
```

You may have noticed that I have only used string type variables in these examples. That's because the data manipulations required are a bit simpler for string variables. That doesn't mean that you can't pass numeric variables, however. Just use STR$() to convert them to string variables and VAL() to convert them back. I'm sure it wouldn't take too much effort to figure out a way to pass them directly, either. Just be careful if you use the second method that you don't try to pass a number greater than 127 or less than -128 unless you split it up into more than one byte.

Finally, if you don't have a disk drive and want to try the first method, type in the program below and save it to CS1. Then type in the DEMO2A program and save it to the same cassette WITHOUT rewinding it. Then rewind the cassette, load the first program, and run it.

After the program begins running, it will ask you to rewind the cassette. Ignore this message and just proceed to load in the second program or you will end up chaining the first program back into memory.

The reason lines 110 and 160 of the first program had to be modified is that the computer prints 14 lines to the screen when chaining the second program. If this offset had not been accounted for, the second program would have read the wrong portion of the screen.

```
100 CALL CLEAR
110 DISPLAY AT(15,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(16,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "CS1"
```

FLOWER...(Continued from page 1)

Speed always seems to be important to computer users. BASIC is always maligned for being less-than-fast. Forth is a very fast interpretive language, but sometimes it is necessary to code things in machine language. In the June issue, I presented an alternate floating point word set written in machine language.

This month I present the "double number word set" in machine language. Double numbers are 32-bit (4-byte) numbers. These words are intended to manipulate two single length numbers or one double-length number. Often when these words are used in a word at all, they are used frequently. Definitions are given in the TI Forth manual, but they are in Forth, not in machine language. Of course, those are slower. See a book on Forth or the TI Forth manual for additional descriptions of these words.

To enter these screens, type them in beginning with the initial comment line. DO NOT TYPE THE "SCR#n:" PART. THAT IS ONLY FOR YOUR REFERENCE. To use them LOAD the first screen in the series. For example, to use the flower power routine, you would load the sine and cosine functions and then the flower-power routine by typing -GRAPH2 90 LOAD 92 LOAD ONE-LINER. If you have any questions on these screens or on Forth in general, drop me a line via the ComputerLine BBS (729-0408) or c/o the MSP newsletter editor.

SCR#90:

```
( transcendental functions                          30JUN84 GED )
: TABLE   ( data type of table                         LJS 1982 )
   <BUILDS DOES> SWAP  DUP + + @ ;
BASE->R DECIMAL
TABLE (SINES)  ( Angle in degrees >=0 AND <=90 --- SINE * 10000)
 0 , 175 , 349 , 523 , 698 , 872 , 1045 , 1219 , 1392 , 1564 ,
1736 , 1908 , 2079 , 2250 , 2419 , 2588 , 2756 , 2924 , 3090 ,
3256 , 3420 , 3584 , 3746 , 3907 , 4067 , 4226 , 4384 , 4540 ,
4695 , 4848 , 5000 , 5150 , 5299 , 5446 , 5592 , 5736 , 5878 ,
6018 , 6157 , 6293 , 6428 , 6561 , 6691 , 6820 , 6947 , 7071 ,
7193 , 7313 , 7431 , 7547 , 7660 , 7771 , 7880 , 7986 , 8090 ,
8191 , 8290 , 8387 , 8480 , 8572 , 8660 , 8747 , 8829 , 8910 ,
8988 , 9063 , 9135 , 9205 , 9272 , 9336 , 9397 , 9455 , 9511 ,
9563 , 9613 , 9659 , 9703 , 9744 , 9781 , 9816 , 9848 , 9877 ,
9903 , 9926 , 9945 , 9962 , 9976 , 9986 , 9994 , 9998 , 10000 ,
R->BASE -->
```

SCR#91:

```
  ( trig functions                                 30JUN84 GED)
BASE->R DECIMAL
: SINES         ( [N<361] --- sine * 10000  )
    DUP 270 > IF 360 SWAP - (SINES) MINUS ELSE DUP 180 >
    IF 180 - (SINES) MINUS ELSE DUP 90 > IF 180 SWAP -
    ENDIF (SINES) ENDIF ENDIF ;
: <SINE         ( n --- [n<360] )
   DUP 0<  IF ( - ) 360 MOD 360 +  ELSE ( + )  360 MOD ENDIF ;
: SINE          ( n --- sine * 10000   )   <SINE SINES ;
: COSINE        ( n --- cosine * 10000 )   90 + SINE ;
: TANGENT       ( n --- tang  )   DUP  SINE SWAP  COSINE / ;
: COTANGENT     ( n --- cotan )   DUP  COSINE SWAP  SINE / ;
: SECANT        ( n --- secant)   SINE 10000 SWAP / ;
: COSECANT      ( n --- cosec )   COSINE 10000 SWAP / ;
R->BASE
```

SCR#92:

```
( a translation from HCM "ONE-LINERS" vol 5 no 3
                                              06JUL85  GED  )
BASE->R HEX
: CLEAN    ( --- )    2000 1800 0 VFILL ; DECIMAL
: SCALE    ( --- )    10000 */ 110 / ;
: ?RETURN ( --- )     BEGIN  ?TERMINAL  UNTIL TEXT ;
: UNE-LINER  ( --- )
   GRAPHICS2
11 1 DO  CLEAN
        I 2 MOD  MINUS 180 *  360 +
     0 DO    I J  * COSINE  DUP
             I COSINE  SCALE  128  +
       SWAP  I SINE  SCALE  MINUS  95 SWAP -
       DOT LOOP
    LOOP  ?RETURN ;
R->BASE
```

SCR#93:

```
( Double number word set  CODE definitions  05JUL85 GED  )
BASE->R  DECIMAL 74  R->BASE  CLOAD ;CODE
BASE->R HEX

CODE 2!      C039 , CC39 , CC39 , 045F ,
CODE 2@      C019 , C070 , C650 , 0649 , C641 , 045F ,
CODE 2DROP   8E79 , 045F ,
CODE 2DUP    0229 , -4 , CA69 , 6 , 2 , C669 , 4 , 045F ,
CODE 2OVER   0229 , -4 , CA69 , A , 2 , C669 , 8 , 045F ,
CODE 2SWAP   C029 , 2 , C059 , C669 , 4 , CA69 , 6 , 2 ,
             CA40 , 6 , CA41 , 4 , 045F ,
CODE 2ROT    C029 , A , C069 , 8 , C089 , 0222 , 6 , C892 , 4 ,
             0642 , 8242 , 16FB , CA40 , 2 , C641 , 045F ,

R->BASE -->
```

SCR#94:

```
  ( Double number word set cont' )
BASE->R DECIMAL
: 2CONSTANT  <BUILDS , , DOES> 2@ ;
: 2VARIABLE  <BUILDS 0. , , DOES> ;
: D-     DMINUS D+ ;
: D=     D- 0= SWAP 0= AND ;
: D0=    0. D= ;
: D<     D- SWAP DROP 0< ;
: DU<    ROT SWAP  2DUP  U<  IF  2DROP 2DROP
          ELSE  = IF  U<  ELSE  2DROP 0  ENDIF  ENDIF ;
: DMAX   2OVER 2OVER  D- SWAP DROP 0<
          IF  2SWAP  ENDIF 2DROP ;
: DMIN   2OVER 2OVER  2SWAP D-  SWAP DROP 0<
         IF  2SWAP  ENDIF  2DROP ;

R->BASE
```

TIPS 'N THOUGHTS

by Tom Fairbairn

I spent some time during the past few days installing Glenn Davis' character set from the February 1985 newletter into my TI-WRITER. Congratulations, Glenn, it is a good bit more readable, I think, than the upgrade that TI supplied. After some added changes (I like my capital A to have a pointy top, for example), I am very pleased with the result. The changes are easy once you have been through the operation once or twice.

Glenn, maybe you could look into the character set used by MULTIPLAN and give us the character table from the file MPCHAR for that upgrade. I, for one, would like to incorporate the same character set into MULTIPLAN, but I find the character file structure is different, and you can't use the same one as with TI-WRITER. It would also be very interesting to know if there are any other cartridges that also call a character file if one is available, and build files for them.

Now if you recall, TI-WRITER has a multilingual menu screen. As simple as 1, 2, 3 you could build up a set of character files for each of the languages that are in the menu. That is really all that is necessary to make the screen talk in the different languages. Of course, you would also need to set up the character codes to match the international character sets in your printer (dot-matrix) or on your thimbles or daisy wheels (if you are running letter quality for a printer).

If you are taking one of the languages in school, or there is someone in your family who is, what easier way to type up those reports all set up with the new characters and others that are unique to the other languages? I do not think, though, that I will try to set up a Katakana set at this time; my Japanese isn't good enough for that! Nor will the download character memory on my printer hold enough characters.

By the way, some dot-matrix printers can have home-brewed character sets downloaded to them (my Gemini 10X has this capability, for example). I could just as easily work up a Greek, Hebrew, or even Russian character set, download it to the printer, and build a matching set for the TI-WRITER. You could also do this by using the appropriate thimble or daisy wheel, and building a matching character set for the TI.

The only problem is, can anyone make me a set of keytop caps to use so I know which key is what?

And let me further note, this is yet another capability that I do not have on the quarter-million doller comprehensive electronic office/word processing system I use at work!!

I think it is worth pointing out, right here, that you do not have to be a master programmer to make use of the many capabilities of the TI computer. Yet by using very simple means, you can extend the value of the machine and its capability far beyond what you might have dreamed when you first bought the machine. I can pick several nits with my TI but I really can't seem to believe that it is useful only as a TV game, as some folks seem to think.

Much of what I have mentioned so far in this column simply makes use of capabilities that are part of the existing machine and software. Glenn's improved lettering file is easy to duplicate, or I'm sure you can find someone who would get you a copy if you don't have the Ed/Asm capability.

So what else?? Well, the TI-WRITER does have some limitations that become apparent when you need to build a document that exceeds 8 pages of single-spaced printing. This is because of the memory space that the TI can use for the printable file. But the word processor can handle this in stride. What you have to do is to create printable files called, for example, TEST1, TEST2,....TEST6.

To print these out, you need to set up another document to the sequence you want them printed. The extra file which we'll call TEST for now, contains only the page headers, page footers, and the Formatter commands needed to print the document. All the printable data comes from the other files. The page numbers will be consecutive, the headers will be on every page of the entire final printout, and the printout will look as though it all came from one source file.

Now what I am trying to put across here is that the memory limit on the TI is not a problem for TI-WRITER because of its inherent design. Since they did not have an almost unlimited memory, it is designed to be efficient and very useful within the memory that is available. That, my friends, is excellence in program design in my book! And it is carried off very elegantly.

I have had several people bemoan the fact that TI-WRITER does not allow you to "cut" a document on a particular word so that you can save parts of a line for future reference. Ah, but you can! What you have to do is use the INSERT CHARACTER key to do the "cut." This places the cut-away portion on a separate line. Now you can SaveFile between line numbers at the cutaway lines. Then use the REFORMAT key to "paste" the source document back together. The saved file contains only the desired part of your original data.

This is not the most convenient way of handling such a need, but it does work and is not that bad after you get used to it. By the way, you should realize that you can use the INSERT CHARACTER key to make cuts in more than one place at a time -- provided the subsequent cuts are made at progressively greater line numbers. If you then reformat "top-down" after saving the file, you may find that all the pasting is done in one move. The only thing is that reformatting will not work across page breaks and carriage return symbols. You can also use this method to Move multiple lines that do not begin and/or end at the start of a printed line.

Now the ability to include data from multiple files is not limited to the .IN)clude function. If you want to use short lines in your printed document over and over, you can make use of the .ML (mailing list) Formatter command.

By using this command, you can call in a specific entry from a list of them, and include it in the printout as if it was really a part of the original text. The announced purpose for this is to allow merging names and addresses into a form letter. If you think about this capability very carefully, you will realize that it could have other applications as well, such as page headings that change with the chapter or subject. In a previous column I mentioned alternate side of the page headers. I use the mailing list to accomplish this. I'll explain it better in another column. A word of caution on using .ML is necessary, though. Each of the values to be read into your document must start at absolute column 0 of the document. Do not set margins on the list document, or the .ML command will never find them.

This frustrated me for days! Then I found out that the Formatter moves the list values to the margins you have set for the main document. Do not use centering on a list entry. Use the required space symbol (SHIFT/6) to space over to where you want the item. Also be aware that the control entry eats up some line space, so the list item can't fill up an entire line to the right margin. This amounts to between 3 and 5 characters short of complete fill in most cases.

WANT ADS

FOR SALE Epson RX 80 FT cable, and much more. Best Offer.

Call Dave 944-9704

FOR SALE External TI Disk Drive w/ power supply. Excellent condition. $85.00

Call Dick 489-8427 or John (512) 834-6882

FOR SALE TI 99/4A Computer, P-Box W/ 32K Mem Exp, RS232, Dual Sided Tandon Disk Drive, TI Silent 700 Printer, Ext BASIC, + 12 disks full of software including TI Count General ledger. Asking only $600.

Call Clarke 227-4289

STILL FOR SALE One used projection TV unit. Needs a reverse video monitor. Make an offer to any of the group's officers.

SOMETHING NEW FOR MSP 99 MEMBERS

How many times has it happened to you? You sit down to begin keying in a new program and suddenly you need a certain piece of information. You check the table in the reference card that came with your computer. That refers you to another table, which refers you to a manual. By the time you've tracked down your lead, you've lost the train of your thought.

Well, we hope to be able to alleviate your problem a little. Starting this month the MSP 99 Newsletter will begin running a series of helpful reference aids. These aids will be on seperate sheets printed on both sides which you can keep handy beside the computer for quick, easy reference. They could be collected in a folder or within a 3 ring binder. You can even make paper airplanes out of them, they're your sheets, but we recommend that you keep them safe and secure and all in one place.

Look for more reference aids in future issues.

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

Address Correction Requested

U 8510
CENTRAL ALABAMA 99/4
USERS' GROUP
551 LARKWOOD DRIVE
MONTGOMERY, AL 36109

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY.