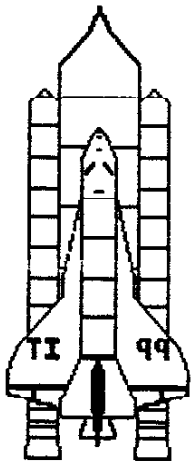


B
R
E
V
A
R
D



U
S
E
R
S

G
R
O
U
P

B. U. G.

P.O. Box 1402

Palm Bay, Florida

32906-1402

BUG VOLUME 2 NUMBER 2

JUNE 1984

PRESIDENT'S MESSAGE

Last month I visited Washington D.C., and while there, I spent some time with Jim Horne, the president of the Washington D.C. User's Group, and with Bill Whitmore - currently the editor of their newsletter and past president of the group. Unfortunately, my schedule did not permit me to attend their May meeting.

Thanks to Bill, the confusing streets of an unfamiliar town was made easier because he picked me up at the Hotel and took me to dinner the first night, and to Jim's house the second night.

For those of you that haven't been there, Washington D. C. is a beautiful town, especially this time of year. With the cherry trees and other flowers and blooms in full swing, the colors and sights are memorable. However, the highlight of my visit was the evening I spent with Jim and his wife Cheryl. Although very busy folks, they took me into their family and fed me one of the best meals on the East Coast. The rest of the evening I spent helping Jim and Bill set up and "smoke test" Jim's TI Professional Computer.

By the way, those of you using a telephone MODEM and accessing COMPUERVE, when you talk to the SYSOP (SYSTEM OPERATOR) under section PCS-27 (the TI section) - say "hello" to Jim Horne for me!

Now for a little more serious business. This month is election month. Enclosed you will find the ballot for the elections. Please complete this form as soon as possible and either mail it to the club or bring it to the June Meeting.

This month's newsletter is stocked with little tid-bits of information we have collected from other newsletters from around the country. I would also like to see more articles from our group - and some covering FORTH and PASCAL. How 'bout it? Anyone out there used these languages yet? If so, how about some simple programs in these languages that the rest of us can enter and see how these things work?

More and more information is hitting the streets concerning Assembly Language for the TI. I recommend to those of you who are struggling to learn it, pick up the book titled "Introduction to Assembly Language for the TI Home Computer" by Ralph Molesworth. You can order these through the City News Stand (don't forget the 10% discount for members.) This book is excellent for both the beginner and the advanced assembly language programmer.

This month's program will continue where it left off last month. Amos Brown will take you from DATA statements into files and file structures, how to read 'em, write 'em and the advantages/disadvantages of Sequential vs. Random files. In his discussion, he'll cover the OPEN statement and what DISPLAY and INTERNAL file formats mean to you.

In addition, I will begin a series on TI Writer. We have received many requests for a program on this subject. During this first session, I will demonstrate the software and it's capabilities and explain how to get started. If you are interested in TI Writer, you will need to bring your questions and a copy of your printer's control codes - bring an extra copy for me if you can.

EDITORIAL . . .

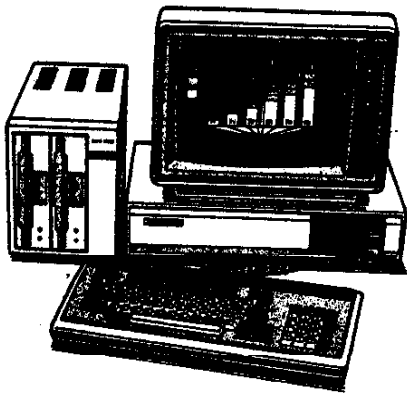
by Kathy Frye

Please note the information questionnaire on page 14. Fill it out and return it as soon as possible. This information will help us plan the newsletter articles that you would most be interested in. It will also help in planning the future programming lessons and the most desirable additions to the Library.

I will not be at the June 19th meeting because of vacation plans. If you have any newsworthy items please write them up and give them to Mike Walker.

I need some 'reporters' for the newsroom. The job would consist of assisting in typing up articles, researching other user's groups newsletter, and possibly stapling & addressing our newsletter. My time for our newsletter is going to become rather limited because I am going back to college this fall. If anyone is interested then please either call me (259-4896) or write to the newsletter address. Any help will be appreciated.

The first installment of a series of Assembly Language articles from another newsletter is included in this newsletter. Watch for next month's continuation of it. As soon as we get all the BUGs out of a FORTH article we will be publishing that and, hopefully, future FORTH articles.



BASIC NEWS:

Those who bought their TI99/4A before October 31, 1983 can send for a free "Teach Yourself Basic" cassette tape from T.I. Call 1-800-TI-CARES to make arrangements.

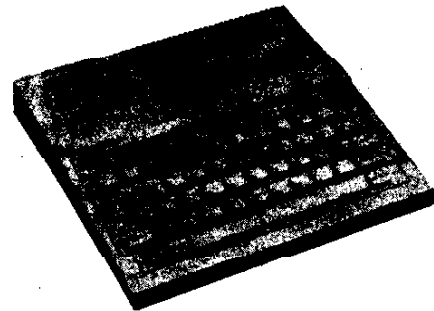
NOTES . . .

by Mike Walker

In our position as the Southeast Regional Coordinator for the 99'ers Users Group Association, we receive a monthly newsletter that we distribute to all the officially recognized TI User's Groups in Florida, N. Carolina, and S. Carolina.

Portions of the newsletter will be extracted and included in the BUG Newsletter from time to time. This newsletter is available to all members to read by contacting Kathy Frye at the meetings. Copies are available to members for \$.75, and non-members \$1.50.

Of special interest in this newsletter is a series of articles on Assembly Language by David Ramsey.



INDEX

Assembly Series	Page 6-9
Ballot	Page 15
Chuckle	Page 7
Classified	Page 3,4, 13
Disk Formats	Page 5
Editorial	Page 2
Education Schedule	Page 4
Epson Graphics	Page 12
Hall of Fame	Page 3
Kathy's Korner	Page 13
Library File	Page 14-18
Library Report/Review	Page 11, 12
Load Program	Page 4
Officer's Club	Page 3
President's Message	Page 1
Protection	Page 3
Tidbits	Page 11
TI News	Page 2, 12

PROTECTION
by Art Green

"PROTECTED". Those of us who know a little about saving data on our computers, know what "Protection" means--but do we?

I found out the hard way, that it also applies to the protection that can be put on a file by the Disk Manager. It does essentially the same job for a file that the sticker over the tab does for the disc. It stops you from overwriting the file.

Now this is something that everybody should know--but I did not. So after I modified a "Name-it" maillist file, and tried to save it, I just could not give the right commands to 'SAVE' the modified file. I tried everything I could think of and always got an error message that indicated disk was full when it wasn't, or something else was out of order.

I finally gave up and called a fellow BUG member and he said simply, "Save it under a different file name.". So I did, and the program was saved. He then advised me how to use the Disk Manager to remove the protection from the original "Name-it" program, and then substitute the modified program. I probably could have figured that part out for myself, but those blanks that are so frustrating can only be solved by hours of study, or a few words from those who know.

I was at Amos Brown's suggestion that I finally tackled modifying the file to let me get the print out that I wanted. So now I have the results I want, and a bit more knowledge than I had last month, and I figure that is worth far more than I just paid in dues. So the rest of the year with BUG is a bonus.

OFFICERS CLUB:

- President.....Mike Walker
- Vice President.....Jim Baker
- Treasurer.....Dick Taylor
- Secretary.....Linda McCormick
- Member at Large.....Rod Williams

COMMITTEE CHAIRPERSONS:

- Hardware.....Rod Williams
- Program/Education....Amos Brown
- Refreshments.....Linda McCormick
- Software Library.....Jim Hilley

NEWSLETTER PERSONNEL:

- Editor.....Kathy Frye
- *****

BUG Hall of Fame

Because of the enthusiasm at the last meeting we now have some new scores to honor. If you wish to try your luck or show your skill, come to the next meeting and bring your favorite cartridge. Due to space limitations, only the games available in cartridge form will be considered.

The score must be verified by one of the judges before the score is accepted. If you can't make the meetings to show your mastered talents then the score must be verified by a picture and submitted to the BUG. To be fair, two categories have been established: Children (15 and under) and Adult. Once listed, your score will remain on the list until another member beats it.

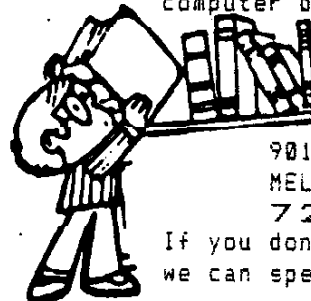
ADULT:

- Norman (Mike) Blalock- Parsec - 274,400
- Kathy Frye - Hopper - 32,420 Level 10
- Gail Hilley - Munchman - 257,420 Level 75
- Gail Hilley - Burgertime - 580,000 Level 143
- Mike Walker - Moonmine - 48,460 Level 17
- Mike Walker - Tombstone City - 290,700

CHILDREN:

- Michelle Blalock - Alpiner - 11,504
- Rick Mucklow - Moonmine - 13,000
- Rick Mucklow - Tombstone City - 79,400
- Ray Schneible - Pac Man - 28,760
- Christi Walker - Car Wars - 26,910
- Julie Walker - Attack - 29,550
- Patti Williams - Ms Pac Man - 39,200

SPRING by and see our full line of computer books and magazines at the **CITY NEWS STAND**



901 E. NEW HAVEN AVE.
MELBOURNE, FL. 32901
725-0330

If you don't see what you want we can special order it.

NEW MAGAZINES IN:

- TI 99/4A (quarterly/year) \$ 2.95
- TI PROFESSIONAL COMPUTING (monthly) \$ 3.50

LOAD PROGRAM

(program by Jim Baker, typed by Kathy Frye)

Back by popular demand is Jim's LOAD program. In case you don't remember, this program can be used for up to 3 disk drives. It does require the use of an Extended Basic cartridge and it takes up only 6 sectors on a disk. 'Nuff said, Here's the program!

```

100 @=1 :: OPTION BASE 1 :: DIM A$(40):: CALL CLEAR :: CALL
COLOR(3,13,12)
110 DISPLAY AT(5,4):"DISK DIRECTORY PROGRAM": : " * WITH
RUN OPTION *": : : "
    BY: Jim Baker"
120 DISPLAY AT(18,4):"PLEASE SELECT DISK NO 1" :: ACCEPT AT(
18,26)SIZE(-@)BEEP:B
$ :: IF B$<"1" OR B$>"3" THEN 120
130 OPEN #@:"DSK"&B$&".",INPUT ,RELATIVE,INTERNAL :: INPUT #
@:C$,A,A,B :: CALL C
OLDR(3,2,@):: DISPLAY AT(@,@)ERASE ALL:"DSK"&B$&". "&C$&" AV
ALL="";B :: C=@
140 D=@ :: E=2 :: FOR F=C TO 127 :: INPUT #@:C$,G,A,B :: IF
LEN(C$)=@ THEN 190
150 D=D+@ :: IF D>40 THEN 220 ELSE IF (INT(D/2)*2)<>D THEN E
=E+@ :: H=@ ELSE H=1
5
160 IF D<10 THEN DISPLAY AT(E,H):"0"&STR$(D)ELSE DISPLAY AT(
E,H):STR$(D)
170 DISPLAY AT(E,H+2):"-";C$ :: A$(D)=C$
180 NEXT F
190 CLOSE #@
200 DISPLAY AT(24,@):"PGM # TO RUN (99 TO END)" :: ACCEPT AT
(24,26)BEEP SIZE(2)V
VALIDATE(DIBIT):D$ :: IF LEN(D$)=@ THEN 200 ELSE IF D$="99" T
HEN END
210 IF VAL(D$)<@ OR VAL(D$)>D THEN 200 ELSE 260
220 DISPLAY AT(24,@):"PGM # TO RUN (@ FOR MORE)" :: IF LEN(D
$)=@ THEN 220 ELSE I
F D$="00" OR D$="0" THEN 240 ELSE 250
230 ACCEPT AT(24,27)BEEP SIZE(2)VALIDATE(DIBIT):D$
240 C=F :: FOR I=3 TO 24 :: DISPLAY AT(I,@):"" :: NEXT I ::
GOTO 140
250 IF VAL(D$)<@ OR VAL(D$)>D THEN 220 ELSE CLOSE #1
260 J=VAL(D$):: CALL INIT :: CALL PEEK(-31932,6,K):: CALL PE
EK(G*256+K-65534,6,K
):: N=G*256+K-65534 :: C$="DSK"&B$&". "&A$(J):: CALL LOAD(N,L
EN(C$))
270 FOR I=@ TO LEN(C$):: CALL LOAD(N+I,ASC(SEG$(C$,I,@))):
NEXT I :: CALL LOAD(
N+I,0)
280 RUN "DSKX.1234567890"

```

PROGRAMMING FOR FUN

by Amos Brown

Programming is and should be fun. No one needs to be afraid of trying to program. Approach programming as a mystery puzzle. When you begin writing your and testing your program, the computer gives you clues as error messages for helping you to solve the mystery. By applying the knowledge you have aquired, using common sense, and a little help from your reference manual you can solve most problems. Sometimes, you may need to call a friend for help. But most of all you have aquired a little more knowledge about your computer because of that mistake.

Remember that you learn the most about computers and programming by making mistakes. Futhermore, the person that makes the most mistakes of different kinds will know the most about his computer. So here's to the mistake makers and the knowledge they bring with them to the BUG-User's meetings.

EDUCATION SCHEDULE

by Amos Brown

During the JUNE meeting we will take a closer look at file structures. We will also cover the OPEN statement and the advantages of sequential and relative files, as well as explore the use of display and internal formats.

On the schedule for the July meeting is the control of program execution. We will also take a look at GOTO, ON GOTO, GOSUB, and ON GOSUB's will be discussed. Included in this class will be a look at program design techniques.

```

*****
*
*           FOR SALE
* Adventure module (Disk version) and 3
* disks (Pirate, Mission, Adventure-Land)
* $50.00 Call Jim Baker at 725-6065
*
*****

```

DISK FORMATS

by Bryan Wilcutt
ALPHA 99/4A User's Group
Modified slightly by Mike Walker

This information may not be accurate. Texas Instruments does not publish the actual format of their disk system, and consequently, this section is only an interpretation of the format used. You should find it adequate for most purposes.

SECTOR 0 - Disk Information

ADDRESS	CONTENTS
0000-0009	Disk Name - up to 10 characters
000A-000B	Total number of sectors on disk >0168=360, >02D0=720
000D-000F	'DSK' (>445348)
0010	>50=Disk backup protected, (the letter "P") >20=not protected (a space)
0012-0013	>0201=DS/SD >0101=SS/SD
0038-0064	Disk bit map side 1. See note
0066-0092	Disk bit map side 2. See note

NOTE. On >0038 to >0092: take word at a time, and split into bytes. Take byte at a time and split into bits. Reverse the order of this byte. 1=sector used, 0=not used. (eg. - >0038=>FC00. This equals 1111 1100 0000 0000. Take first byte (1111 1100), reverse the order (0011 1111). This means sector 0 and 1 are unused, sector 2 to 7 are used. This is impossible, sector 0 and 1 are always used, but you get the idea.)

SECTOR 1 - Directory Link

Each word lists the sector of the file directory (hex) in alphabetical order.

SECTOR 2 TO 21 - File Headers

NOTE. Headers for the first 19 files are placed here by default. If the disk contains more than 19 files, additional file headers are placed in the first available sector. If there are fewer than 19 files but they fill the disk (as in the case of FORTH), the sectors in this area may be used to contain the last few sectors of the last file.

ADDRESS	CONTENTS
0000-0009	File Name - up to 10 characters
000C	>00=DIS/FIX >01=Program >02=INT/FIX >00=DIS/VAR >02=INT/VAR

The protection invoked by disk manager will be shown by >8 added to the above.

000F	Number of sectors allocated to the file. (Disk manager will list one more than this number, don't forget this sector counts!)
0011	Record length of files.
001C	First sector of file.
001D-	Block cluster linkage (see note)

Note on file storage: Files are placed on the disk in first-come first-served manner. The first file written will start at sector >0022, and each subsequent file will be placed after it. If a file is deleted, the next file written will start in this hole. If the hole is not long enough for the file, the file will be fractured, and the remainder will be placed in the next available block of sectors. The block cluster link map keeps track of this fracturing. Each block cluster link is 3 bytes long. Byte 2 is divided into two nybbles (4 bits). The rightmost nybble is appended to the left end of byte 1. The leftmost nybble of byte 2 is appended to the rightmost end of byte 3. The result is two 12-bit numbers. The leftmost 12 bits represent the address of the starting sector number for this cluster. The rightmost 12 bits represent the highest logical sector number (within the file) that resides within this cluster.

SECTORS 22 THROUGH 359 (719 FOR D/S)

These sectors contain file data and file headers.

FIRST SECTOR OF THE FILE:

The first byte of the first sector of each file is where the proprietary protection invoked by Extended Basic is flagged. To unprotect this file, change >0000 to it's 2's complement, leaving the LSB=1.

(REPRINT FROM WASHINGTON DC AREA NEWSLETTER 4/84)

AN INTRODUCTION TO 9900 ASSEMBLY CODE

David L. Ramsey
 SOURCE - TI5099
 MAGUS - 4300

This is the first in a series of articles on how to use assembly language routines in your extended basic programs. In it I will cover the major utilities at the disposal of the programmer and how to interface these with TI's Extended BASIC. Finally, I will show those interested in assembly language programming how to construct routines to read joysticks and move sprites, all at speeds far faster than those available in Extended BASIC.

First, let's discuss the unusual features of the TMS 9900 microprocessor. The 9900, unlike most microprocessors used in home computers today, is a 16 bit processor. It has a 64 kilobyte direct memory address range unlike its 8 bit cousins who use various paging techniques to achieve the same memory address range.

Another important feature of the 9900 microprocessor is its lack of built in hardware registers. Only the workspace pointer, the program counter and the status register are built into the hardware. The 16 working registers are defined by the user. In addition, each subroutine can have its own set of registers thus eliminating the need to save any but the three hardware registers. The constant "pushing" and "popping" of values onto and off of the stack is not necessary with the 9900 chip.

Another feature of the 9900 instruction set is its memory to memory architecture. This allows the programmer to perform many operations on data in memory without ever moving it into a register first. For instance, I can use the instruction CB (compare bytes) and reference the two bytes of data being compared in the symbolic addressing mode. If I have previously defined the locations LABEL and CHECK, I could use the following line of code and never move the data into the workspace registers:

```
CB @LABEL,@CHECK .
```

Yet another feature of this versatile chip is its ability to extend its own instruction set by using the XOP instruction. This gives the programmer the capability to define up to 16 of his own operations and use them in his assembly language programs. Related to this is the ability of the 9900 chip to build and use "macros". If I needed to use a type of operation where a stack became necessary, I could define the two instructions PUSH and POP and then designate a specific area of memory as the stack with the BSS (Block Starting with Symbol) directive.

The final important feature of the TI 99/4A's assembly language instruction set is its large number of built in machine language subroutines. These make it much easier for the beginning programmer to develop application programs. As an example, the TI sees the screen as a memory mapped device and cannot access it directly. Instead, values must be read into certain registers and the VDP must be given an instruction on what it is to do with these values. The necessary instructions to access the VDP RAM can be written by the programmer but he need not bother since, with a simple BLWP (Branch and Load Workspace

Pointer) instruction he can access any of a number of utilities to do precisely that. To write a single byte of data to the VDP RAM you could simply write BLWP @VSBW (Video Single Byte Write) or you could write the necessary routine yourself which would be made up of from 5 to 10 separate instructions depending upon how you decided to do it.

A SIMPLE TUTORIAL

To get a feel for using the 9900 instruction set, let's put together a simple routine to read the keyboard for input and then to output that data on the screen.

```

DEF READS
REF VSBW,KSCAN

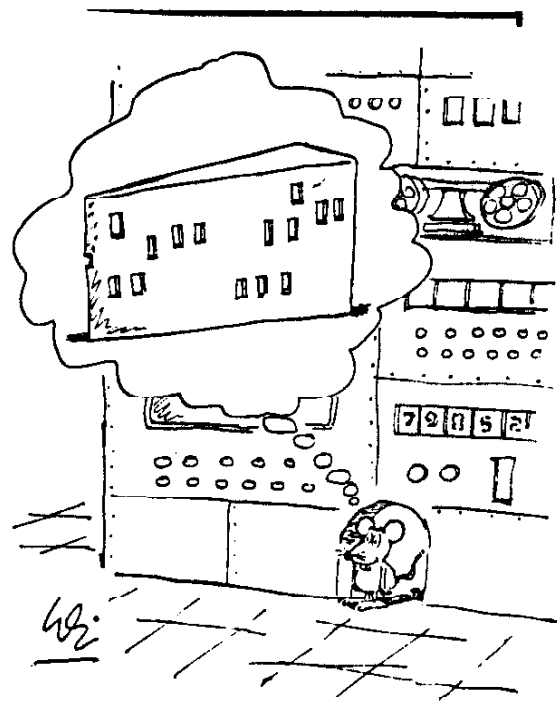
*
HEXFF BYTE >FF
*
STATUS EQU >837C
*
BUFF1 EQU >8375
*
READS NOP
*
RESET LI R5,>0000
      LI R6,>0300
*
KEYBD LI R0,>0000
      MOVB R0,@>8374
      BLWP @KSCAN
      CB @HEXFF,@BUFF1
      JNE WRITE
      JMP KEYBD

*
WRITE MOVB @BUFF1,R1
      MOV R5,R0
      BLWP @VSBW
      INC R5

*
LOOP  LI R1,0
      LI R2,6300
LOOP1 DEC R2
      C R1,R2
      JNE LOOP1
      C R5,R6
      JNE KEYBD
      JMP RESET

*
      END

```



This listing shows a simple program to scan the keyboard and display the input on the screen. It was put together on the Editor/Assembler package and uses some of the unique features of that package. Those of you who have the Mini Memory module and the Line-by-Line Assembler should note the following differences. First, the Editor/Assembler package supports 6 character label names; the MM module supports only 2 character label names. Next, the E/A package supports the DEF

directive; the MM module does not. What this means is that to use the above routine, MM module users will need to shorten label names to 2 characters and they will need to delete the DEF directive. Also, after they have completed entering the program they will need to make an entry in the REF/DEF table. This is what the DEF directive accomplishes for the E/A user. Finally, the MM module user cannot use the names of the utilities such as VSBW. Instead, they must use the address where the utility entry point is located. The BLWP @VSBW example given before becomes BLWP @>6024 with the MM module. (Please note that the > symbol indicates a hexadecimal value.)

At this point we can begin to examine the routine that is listed above.

```
RESET LI R5,>0000
LI R6,>0300
```

This portion of the program is given the name RESET. In it we simply load the values for the first position on the screen and the last position on the screen. In this way, we can compare our present screen location with register 6 and determine when it is time to return to the top of the screen. Register 5 is used by the program to indicate the first screen position to which we must write.

Next is the principal routine of the program, the keyboard scanning routine. It is listed below.

```
KEYBD CLR R0
MOVW R0,@>8374
BLWP @KSCAN
CB @HEXFF,@BUFF1
JNE WRITE
JMP KEYBD
```

The label that I gave to this portion of the program is KEYBD. First, I placed all zeros in register 0 with the clear (CLR) instruction. Before we can access the KSCAN utility, it must know what device to scan. A value of >00 placed in memory location >8374 tells the utility to scan the entire keyboard. A value of >01 tells it to scan the left side of the keyboard and joystick #1. A value of >02 sets the utility to scan the right side of the keyboard and joystick #2. Since we want to scan the entire keyboard for this tutorial we need to place a value of >00 in memory address >8374. Now that we have a value of >0000 in register 0, all we need to do is move the most significant byte in that register to >8374. We do this with the move byte (MOVW) instruction. We place the byte from register 0 (R0) at (@) memory address >8374. Now that we have set this single necessary parameter we can access the utility with the single instruction BLWP @KSCAN.

At this point we need to check if the utility actually found a key depressed during the keyboard scan. There are two ways to accomplish this. The first method is to check the status byte and see if there has been a change since before we accessed the KSCAN utility. The other, which is simpler, is to simply check the value at >8375 with a value of >FF. This value (>FF) is placed at that location by the KSCAN utility when it finds that no key was depressed during the scan.

Since we have used the compare bytes (CB) instruction to make this check, we can now use any of the conditional jump statements to transfer program control. I have chosen to use the jump if not equal command because the value I am testing against is an indicator of no

(REPRINT FROM WASHINGTON DC AREA NEWSLETTER 4/84)

input. What I want to do is jump if the two values were not equal to the WRITES routine. For this I use the JNE instruction and I tell the computer to jump if not equal to WRITES. The actual instruction looks like this:

```
JNE WRITES
```

If the value is equal to >FF then the jump will not occur so I need to cover those cases where no input is received. To do this I follow the conditional jump instruction with an unconditional jump instruction of JMP KEYBD.

This means that if input is found the computer goes to the WRITES routine; if no input is found, then the computer goes back to the KEYBD routine and scans the keyboard again.

This segment of our program is the heart of it. All of the other routines are built around it and with it in mind. It drives our program and we will always come back to it for more input.

The second most important routine is the WRITES routine that I mentioned above. It is listed below.

```
WRITE  MOV  @BUFF1,R1
      MOV  R5,R0
      BLWP @VSBW
      INC  R5
```

The WRITES routine will write the character of the ASCII value that the computer detected during the keyboard scanning routine. To do this we use the VSBW utility. There are two things that the VSBW utility must know before it can accomplish what we want. First, in register 0 we need to have the location in the video RAM that we wish to write to and secondly, we must place in register 1 the value we wish to write to the chosen location. The value we wish to write is obviously the value at BUFF1 (>8375) which is the value we detected during the keyboard scan. We can move it into the register with the move byte instruction. Next, we need a screen location. If you think back, when the program was just beginning we loaded register 5 with just such a value. So we can use the move byte instruction to move our location into register 0. Now we can access the VSBW utility. We do this with the branch and load workspace pointer instruction and the character will appear on the screen. At this point we increment the location counter by one with the INC instruction and we fall into our delay loop.

Now, the delay loop is not a mandatory part of this program but to not use it can mean that a single press of a key could fill one half to two thirds of the screen with the same character. Those of you who are unfamiliar with the speed of machine language may like to delete the lines from the label LOOP to the line with the instruction JNE LOOP1. If you assemble the program without the delay loop you will get a vivid indication of the real speed of machine language.

The delay loop is nice however because it illustrates something that every assembly and machine language programmer knows all too well. When working with assembly language, you must frequently think of ways to slow down the program rather than speed it up.

After you have it assembled, go ahead and run it with the LOAD AND RUN option on the E/A module.

SOFTWARE LIBRARY

Welcome to the Software Library report. This past month has basically been a relaxing month for me. Not very much has been done to the library except for some heavy thinking. Our library is quite extensive and covers a broad spectrum of subject areas. Now that the quantity is there it is time to concentrate on the quality. In the near future we will add a commercial grade game to our library. We received a one-time opportunity to purchase 'KING OF THE CASTLE' by Cydex Software. Normally we try to make sure that no copyrighted software enters our library, but Cydex Software is offering us the license to distribute the program to our users.

This program is written in assembly language. It requires a cassette recorder and Mini Memory or a disk drive and expanded memory system and Editor/Assembler on the expansion system and Mini Memory. I know that the above systems does not represent the entire Brevard Users Group but I'm sure there are quite a few members that can take advantage of this offer. In the future we will try to acquire more commercial software and we will try to concentrate on a larger segment of the group users. I will have some brochures on the 'KING OF THE CASTLE' at the software library table during our June meeting.

There are a number of individuals out there that have indicated an interest in helping me with the software library. While I was trying to upgrade the quantity of the library I really couldn't call on them. Now I would like to start working on upgrading the quality. This will be an ambitious project and I will need all the help I can get. I will be having a get-together at my house on 17 June for all that are interested in helping with the software library. My address and home phone will appear at the end of this article. I need a few individuals with full systems to both work at home and work during the monthly meetings. A full system is

not a must for someone to help out. There are many jobs that can be done without a full system. We will be discussing these during my get-together. Rather than being super busy during the meetings, I would like to be available to answer questions and help out with any problems that the users might have with the software library programs. I will look forward to seeing a number of you on the 17th of June. If the instructions that I have provided below is not sufficient to find my house give me a call.

NOTE: In this newsletter you will find a questionnaire that asks you to list the kind of equipment that you have. The inputs to this questionnaire is very important. We can more readily judge the kind of programs that will be of interest to the majority of you. Let us also know what kind of background you have and the uses you are putting your computer to. I'm sure there is a wealth of knowledge out there that can be extremely valuable to the group as a whole.

NOTE: THIS WILL BE THE LAST TIME THAT A FULL SOFTWARE LIBRARY LISTING WILL BE PROVIDED IN THE NEWSLETTER. AFTER THIS MONTH ONLY UPDATES WILL BE LISTED. FULL LIBRARY LISTINGS WILL BE AVAILABLE AT THE MEETINGS FOR PASSING AROUND. PERSONAL COPIES WILL BE PROVIDED UPON REQUEST FOR A FEE OF \$1.50.

Jim Hilley
146 Browning Ave N.E.
Palm Bay, Florida
725-0371 (AFTER 6 P.M. DAILY)

(Browning Ave. is located off Malabar Rd, west of I-95, and five streets east of Minton Rd.)

PROGRAM OF THE MONTH

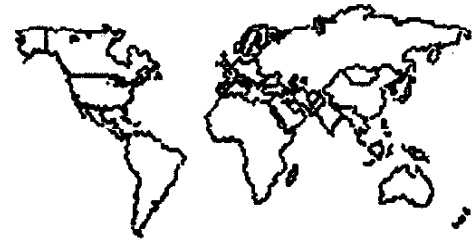
The program I have picked for this month amazed me when I first saw it demonstrated. The author of this program has to be good. In only 17 sectors of disk space he has created a world wide navigation aid. The name of this program is 'WORLDMAP', by Paul Severance of Kasson Minnesota. It requires only the basic console and Extended Basic.

The first instruction that appears on the screen asks you for the number of points that you want to plot on a world map. After that you provide the longitude and latitude of each of those points. What happens next will amaze you. The program draws a detailed map of the world, one of the best I have seen, and plots a point at each location that you provided. The possible uses of this program is endless. I can see it used as an educational tool for your older children--teaches them the concepts of latitude and longitude. It can be used as a basis to a game--lets see who can plot a point closest to a designated area of the world. With the hurricane season upon us you can use it to find out the location of the storm and keep track of its progress. Mike Walker is working on a screen dump utility that will allow you to make hard copy maps of the world directly from this program. Check this program out, seeing is believing.

Note: For the past three newsletters, I have been reporting on programs that have caught my interest. Many of them have been of educational value and I'm sure have interested some of you. There must be many more programs in the library that I'm sure I will never get to that is of interest to a lot of you, therefore I'm am challenging all of you to help with these monthly program reports. If you find a program that you like share the experience with the rest of us. Submit an article about it. Send the article to the BUG address that appears on this newsletter. Who is going to be the first person to get their name and their article in the next

newsletter. I'm sure that I speak for all the regular contributors of the newsletter when I say: 'We need more input from everyone to keep this newsletter going. Any article on any computer subject will be graciously accepted'.

JIM HILLEY



TIDBITS of TIPS

(CORNERSTONE, Jim Schuneman)

To disable the "FCTN QUIT" key only two commands are necessary:

(1) CALL INIT (2) CALL LOAD(-31806,0)

The QUIT key will be disabled until you either turn off the console or type in BYE.

(FROM MILWAUKEE 99/4 US)

When in TI-WRITER, try printing your documents under the "Formatter" to disk by typing in "DSK1.ANYTHING" for the prompt "Print Device Name". This little trick will cause the output from the Formatter to be directed to disk rather than the printer. Then by using the Text Editor facility of TI-WRITER, you can then edit your printed file to see how it will look when printed or to correct information before printing. To print your document, use the "print file" option under the text Editor. All the print control characters that were imbedded in the text will still be passed on to, and recognized by, your printer.

TI-WRITER BUGS--(from Mike Walker)

Typing past end of a line may lose characters.

Typing a @ sign will cause overstriking. Typing 2 of the @ signs will cause 1 of the @ signs to be printed. Typing 3 of the @ signs will cause the Formatter to hang up!

EPSON GRAPHICS

by Kathy Frye

After getting my Epson RX80 one of the things I wanted to do with it was make pretty pictures. Since no one else that I know of had an Epson this appeared to be one of those things easier said than done. I was told by several persons that I would just have to learn about my printer by myself. Feeling slightly overwhelmed, I stubbornly tackled the Epson printer book and the TI 99/4 printer book to learn the different control codes for graphics and text. The text portion proved to be fairly simple but the graphics still came out as so much garbage.

So I turned to Mike Walker for assistance. He explained the relationship between data and the dot wires of the Epson and how to turn them on for my graphics.

Using graph paper, you count off eight of the little squares vertically and horizontally; then you draw your picture. Next you have to calculate the binary value of each small square vertically, starting with the bottom left square. The values of each these squares, starting with the LSB (least significant bit), are as follows: 1,2,4,8,16,32,64,128. If you have turned the wire on then you add the numbers up but if the wire is not turned on then its value is 0.

For an example look at line 260 of the program on the next page. The expression 'C49\$' will be the entire 8 x 8 block definition. 'RPT\$' says repeat the following character, 'CHR\$(0)', X number of times (in this case 0 is repeated 5 times). The '&' could be substituted with a semi-colon and say the same thing. Note that each expression, such as 'C49\$', defines ONLY eight vertical lines; this will finish ONE of the large squares. To complete your drawing you must define all of the large squares in the same fashion.

Next we put these large squares together, which is what I'm doing in LINE 400. The first expression, "PRINT #1:", tells the printer to print that line. The quotes with 10 spaces in between will equate to 10 spaces prior to the beginning of the graphics. The expression 'C#' has been previously defined with printer codes to enable normal-density printing of 32 characters (n1) with 0 (n2) left over. To calculate 'n1' and 'n2' refer to the Epson manual (page 3-57). In

normal-density the maximum number of bit-image data per line is 400, this program however has only 32 characters defined (4 large squares times 8 small squares). Therefore, the maximum number of large squares that you can define is 60 per line. The next four expressions are my bit-image data to be printed on that line. Finally, the expression 'C1\$', which has been previously defined, tells my printer to start printing and advance paper one line.

The next very important line within my program is LINE 395. When I accidentally left that one out, a space appeared between every print line. The 'CHR\$(27)' calls forth the escape code. The 'CHR\$(85)' sets/cancels unidirectional printing, required with this command is a setting 'n' to equal 1 or 49 for unidirectional printing, or 'n' equals 0 or 48 for bidirectional printing (except in Bit image mode). If you leave the 'CHR\$(1)' out of the statement then you get the extra blank lines. Next is 'CHR\$(65)' which sets the line spacing to n/72", and 'CHR\$(8)' defines 'n' to equal 8, making the line spacing 8/72".

Now for the fun part. Take a look at the program, use it to draw YOUR picture on your Epson printer. Happy picture drawing I hope this helps!

TI NEWS

by Kathy Frye

(information from Aloha UG Newsletter)

As of April 12, TI will no longer sell products for the TI 99/4A. Arrangements have been concluded for all existing software to be provided by a company called TRITON Co., P.O. Box 8123, San Francisco, Ca 94128. Their toll free number is 1-800-227-6900.

(We have a TRITON catalog in with the User's Groups newsletters)

Also of interest is the establishment of a HELPLINE by Amnion Stoneware. They are offering assistance, FREE of charge, in regards to hardware and software quality and sources, genuine regional user's groups, help with programming problems and corrections, and will give completely unbiased information on all products. To obtain service you may call 1-415-753-5581 between 9-3 or send a self-addressed stamped envelope to: Helpline, 116 Carl St., San Francisco, Ca. 94117.

```

100 REM *****
110 REM * EXTENDED BASIC *
120 REM * PROGRAM FOR EPSON*
130 REM * PRINT/GRAPHIC *
140 REM * by Kathy Frye *
150 REM *****
160 CALL CLEAR : DISPLAY AT(12,3):"One Moment Please I'm
working HARD!"
170 C40$=RPT$(CHR$(0),5)&CHR$(3)&RPT$(CHR$(1),2)
180 C41$=CHR$(1)&RPT$(CHR$(0),7)
190 C42$=RPT$(CHR$(0),8)
200 C43$=RPT$(CHR$(0),4)&CHR$(1)&CHR$(3)&CHR$(7)&CHR$(1)
210 C44$=RPT$(CHR$(0),5)&CHR$(128)&CHR$(224)&CHR$(248)
220 C45$=CHR$(248)&RPT$(CHR$(252),2)&RPT$(CHR$(126),2)&CHR$(
63)&CHR$(31)&CHR$(7)
230 C46$=CHR$(1)&RPT$(CHR$(0),6)&CHR$(1)
240 C47$=CHR$(7)&CHR$(31)&CHR$(127)&RPT$(CHR$(254),2)&CHR$(2
52)&CHR$(248)&CHR$(2
24)
250 REM C48$ IS SAME AS C42$SO WE USE IT AGAIN!
260 C49$=RPT$(CHR$(0),5)&RPT$(CHR$(1),2)&CHR$(129)
270 C50$=CHR$(153)&CHR$(60)&CHR$(126)&RPT$(CHR$(255),2)&CHR$(
126)&CHR$(61)&CHR$(
152)
280 C51$=CHR$(128)&RPT$(CHR$(0),7)
290 C52$=RPT$(CHR$(0),5)&CHR$(3)&CHR$(7)&CHR$(15)
300 C53$=CHR$(31)&CHR$(63)&RPT$(CHR$(127),2)&CHR$(254)&CHR$(
252)&CHR$(248)&CHR$(
248)
310 C54$=CHR$(192)&RPT$(CHR$(0),5)&CHR$(1)&CHR$(195)
320 C55$=CHR$(60)&CHR$(8)&CHR$(28)&RPT$(CHR$(62),3)&CHR$(28)
&CHR$(8)
330 C56$=RPT$(CHR$(0),3)&CHR$(48)&RPT$(CHR$(240),2)&RPT$(CHR
$(224),2)
340 C57$=CHR$(192)&RPT$(CHR$(128),2)&RPT$(CHR$(0),5)
350 C58$=RPT$(CHR$(0),4)&CHR$(96)&CHR$(240)&CHR$(152)&CHR$(1
2)
360 C59$=CHR$(6)&CHR$(3)&CHR$(1)&RPT$(CHR$(0),5)
370 C1$=CHR$(13)&CHR$(10)
380 C$=CHR$(27)&CHR$(75)&CHR$(32)&CHR$(0)
390 OPEN #1:"P10.CR",OUTPUT,VARIABLE 255
395 PRINT #1:CHR$(27);CHR$(85)&CHR$(1)&CHR$(27)&CHR$(65)&CHR
$(8)&C1$
400 PRINT #1:" ";C$;C40$;C41$;C42$;C43$;C1$
410 PRINT #1:" ";C$;C44$;C45$;C46$;C47$;C1$
420 PRINT #1:" ";C$;C42$;C49$;C50$;C51$;C1$
430 PRINT #1:" ";C$;C52$;C53$;C54$;C55$;C1$
440 PRINT #1:" ";C$;C56$;C57$;C58$;C59$;C1$
450 CLOSE #1
455 END

```



KATHY'S KORNER

When I published the 'tip' about the Tunnels of Doom cartridge and the Adventure series, I did not do it for pirating purposes. The primary reason was for back-up purposes. I do not wish to give the idea that I sanction the pirating of software. If this idea was conveyed, then I apologize and wish to now correct any assumption of that idea.

For those persons who enjoy adventure type games, I have a question. INFOCOM supposedly is making adventure games for the TI99/4A, however I haven't been able to purchase any. If anyone has information on INFOCOM TI99/4A compatible software, please contact me, either at the meetings or by phone (259-4896). Thank you.

NUMERIC
KEYPAD
by Dave Benne

You may add a numeric keypad to your TI99/4A computer for a small amount of money and a little effort. The keypad works like the normal numeric keys. This type of hookup allows one to use the keypad with any program. There is no software required. If you are using Multi-plan or any other program that needs to have numbers keyed in, then numeric keypad can make the task easier. It is set up in the same configuration as an adding machine keypad, which makes the actual numeric keying in process somewhat less tedious. The diagram and instructions are included in the package described in the ad below.

Although bookkeepers and accountants may especially be interested in this new hardware item, you as an everyday home computer programmer can benefit from this addition to your computer also.

TI99/4A USERS:

ADD A NUMERIC KEYPAD to your TI99/4A, NO SOFTWARE required. Send \$2 for a complete schematic diagram and instructions to:

DAVES COMPUTER SERVICE 228 NE 3rd ST.
SATELLITE BEACH, FL. 32937
