C65
8709

The Ottawa T.I.99/4A Users' Group

NEWSLETTER

VOLUME 6 NUMBER Ø6...SEPTEMBER 1987

Announcing the

AUCTION

of
hardware
&
software

Time: at our Fall meetings,
starting in September
Conditions: immediate $ pmt
Preview: before each meeting

P.O. BOX 2144, STATION D, OTTAWA
*** ONTARIO, CANADA K1P 5W3 ***

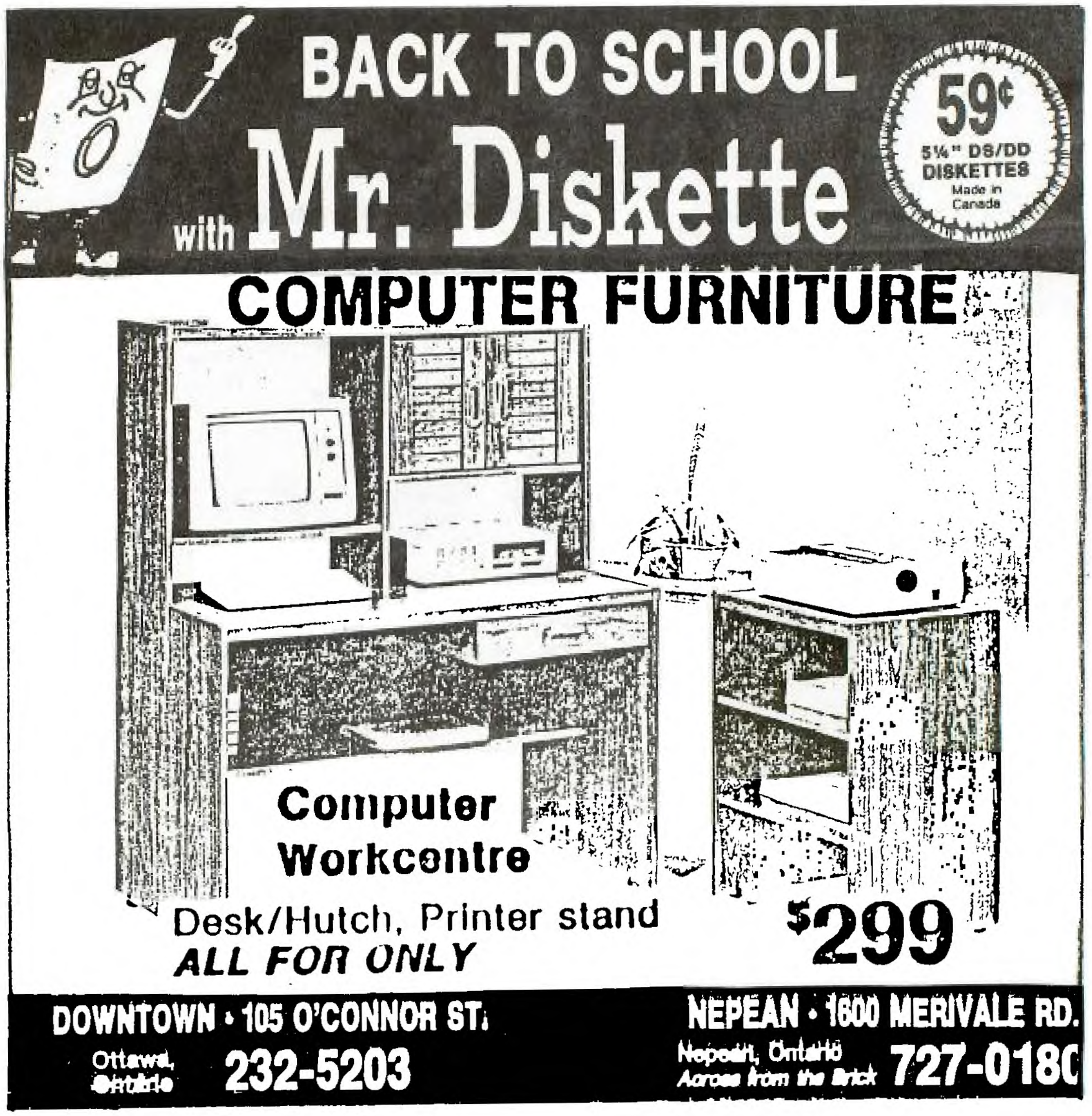
EDITORS NOTES

by **Marg O'Connor**

This month I would like to dedicate the Newsletter to your late chairman Berry Minuk who passed away on June 28.

Berry did a great deal for our club and was always there with needed encouragement and help. When I ran into trouble with the Newsletter, he was as close as the phone, as I am sure he was with all who asked him for help. We shall miss him greatly and our thoughts go out to his family.

September is almost here again and the startup of the meetings for the comming season. Lloyd Galenzoski our vice chairman is carrying on until the annual meeting. He has given me the dates of meetings for the comming season and I will include them here.



MEETING DATES

The following is a complete listing of the meeting dates, until next adjournment for the summer of 1988.

In all cases, the meeting will be held in the cafeteria of the Merivale High School. The space has been booked from 7 PM to 10 PM.

September 8, 1987

October 6, 1987

November 3. 1987

December 8. 1987

January 5. 1988

February 2, 1988

March 1, 1988

April 5, 1988

May 3, 1988

June 7. 1988

Attending meetings can be a great way of solving problems and making new acquaintenances. Coming early, or joining the gang for coffee, afterards, provides an opportunity to get caught up on the latest developments in the TI world.

The Nominating Commity for new executive for the comming year is as follows:

Bill Sponchia	523-0878
Jane Laflamme	837-1719
Marg. O'Connor	833-2626

If you would like to serve on the Executive or know of someone who would like to, please get in touch with one of us by phone or see us at the next meeting.

I would also like to turn over the paper to someone else. I find that I have less and less time to put into it and from the comments I have heard think that there are others that could do a better job. So it is time to step down.

the next meeting is Sept. 8 and is an auction. I will be away but hope you all turn out and get some goodies.

NOTES FROM THE VC

by **Lloyd Galenzoski**

I'm sorry to report that our Chairman, Berry Minuk, passed away, suddenly, on June 28th. We're over the shock now, but it still doesn't seem real.

Berry's two sisters (one from Toronto and the other from Vancouver) offered to leave all of the TI inventory with us, on the condition that the proceeds of the items sold be forwarded to them. I couldn't resist taking them up on it.

At future meetings we plan to auction off all of Berry's TI inventory....there's loads of it and it's all in my basement--software, hardware, books, manuals, etc. So, if you're looking to expand your system, [DON'T MISS THE MEETINGS].

Election of the Executive takes place in October and this is probably as good a time as any to state that I will not be standing for office. I can't tell you how much I've enjoyed my duties, firstly as Membership Chairman and this past year as Vice-Chairman.

Bill Sponchia has agreed to head up a nomination committee. So, if you've been waiting for a chance to do something for the club...here it is! Bill may be approaching you, so please consider the possibility of being nominated. Serving on the Executive can be a very rewarding and good learning experience.

Our newsletter is often taken for granted, as are the articles that appear in it. Our Editor, Margaret O'Connor, supported by her husband, John, puts a great deal of time and effort into making sure that all of us receive a quality product, on time. They need your articles, so don't be shy, sit down and write yours, now.

The BBS program seems to be working quite well, even when our capable Sysop goes away. Other groups and individuals have expressed interest in acquiring it and we may soon be in a position to make it available, at a price. If you haven't tried it, don't be shy. The number is (613)738-0617.

Dick Piche (our hardware technician) has been making steady progress with assembling pieces of hardware to be used for demos at meetings. An little electronic gadget called a video sender should greatly facilitate seeing what is being demonstrated. When being used, anyone in the room with a TV can watch what's going on. It's a miniature transmitter, so you don't have to be connected to the computer.

The club library has been well-organized under the capable leadership of Steve Bridgett. The disk of the month must seem like child's play compared to the cataloging and sorting of files, on hundreds of diskettes.

Lucie Dorais has done a super job of signing up members. She also maintains the newsletter exchange list. If your address needs updating, please let her know. She's the source of the labels for the newsletter.

Henri Monat has been busy getting the archives in shape. He's trying to collect all of the newsletters ever issued by our club and I think he's missing one or two of the oldies. If you happen to have some old ones, I'm sure he'd be glad to hear from you.

Cassette software has regained some popularity with the introducton of the 32K expansion in the console. Jack McAllister is the cassette librarian and I'm sure he would be happy to talk to you.

What about advertising? Tony Hopkins has put a great deal of effort into finding advertisers willing to place ads in our newsletter. The income has helped to keep our club quite healthy, from a financial point of view.

Oh yes, we mustn't forget about the custodian of our funds, Peter Arpin. He seems be able to keep enough money around to meet all commitments and even has a little to spare. This is one job that has something happening every month. There always seems to be somebody who wants to get paid.

Our meetings will again take place on the first Tuesday of each month, in the cafeteria of the Merivale High School. There are a couple of exceptions and our first meeting is a prime example. It will be held on Septemeber 8th (the second Tuesday). A full schedule of the meeting dates will be found elsewhere in this newsletter.

See you on September 8th!

BROWSING THE LIBRARY

with **STEPHEN BRIDGETT]**

Your library has grown and is now neatly arranged by volumes in large binders. Look for it on display at the monthly meetings. Come on out, flip through the disks and find that rare piece of software that your very TI-EXISTENCE depends upon.

This year the library will really get rolling. Last year was a long uphill battle to organize and present it all at the same time. Some things to look for this year include:

1. A new more flexible price scheme. If approved this will allow for a credit system based on documentation that you supply as well

as trades, software you contribute to the library.

2. The beginning of a well organized easily accessable data base of software documentation supplied, as the disks are used, by you the user.

3. A new release of the library catalogue. This version will be 'USER FRIENDLY', simpler and easier to use with a new version of the fareware cataloger by Marty Kroll.

4. A return of the popular 'DISK OF THE MONTH'.

5. A demonstration on the use of the catalogue and library. This will include searching for and ordering of disks. Also cataloging your own disks in a similar manner to the club's so that you can quickly compare your disk collection to that of the club's.

6. A telephone survey of all local members, to hear your views and answer questions that you may have on various aspects of the club.

7. A repeat of the free copy session held last year. This will be an orderly opportunity to flip through the catalogue and obtain a quick free copy of a particular disk.

> Note: due to time constraints, certain limitations must be imposed. Follow your newsletters to check for events.

8. The library has progressed to date with the help of your library committee. This includes:

DOCUMENTATION	BOB LANDY
FAREWARE	STEVEN McWATTY
CORRESPONDENCE	BERNARD VANDEN AND KEN McKENZIE
CASSETTE LIBRARY	JACK McALLISTER
SALES AND PROMOTIONS	STU BRUCE

Look for others in the months to come as the library really comes of age.

Your library has grown! Come on out and be a part of it.

See you on September 8th.

WRITER-WROUTES

by **Jane Laflamme**

I would like to express my deep sorrow at the passing of our Chairman, Mr. Berry Minuk and dedicate this article to him. Berry was one of my first teachers within the club and did all he could to help me with my wonderful computer. I shall dearly miss him.

GENEVE is here! I am writing this article on "her". And the wait seems years away now.

The TI-Writer equivalent on the Myarc GENEVE 964Ø is called MY-Word. The current version is VØ.4 dated May 1Øth of this year and was upgraded by Peter Hoddie of Lexington, Maine.

Unfortunately, this version, at the moment has a few bugs in the Formatter, but none that cannot be overcome. (Or none that I have not been able to overcome so far!) It manages to miss some CR>s which does not start a new line with indent, for instance. I have fixed that by setting my Editor Tabs to the same as the Formatting commands, and using .NF (No Fill) when she gets finicky. And I have put her through her paces. Since arriving at my office, I have had to do a multi-print of tabbed figures, several letters, and a Mail Listing. She has performed outstandingly. One bug that I haven't been able to overcome now that I think about it, is the text is somewhat jumpy on the screen, whereas no other program has this trouble.

The first thing I found was an impressive first screen announcing that MYARC proudly presents MY-Word. And well they should be;

proud of what has been done within the confines of the TI-Writer program, itself.

After entering the text Editor, option 1 from the menu, the first thing you realize is that you now have word processing in 80 columns; a dream. You can now set up your document the way you would like it printed, all on one screen. The second item that catches your attention is the ruler at the bottom of the screen; there, at a glance, are your margins and tabs which enable you easily to align items within your document. Under the ruler is displayed the filename, the current version of MY-Word, and the clock!

While using the Editor, it accepts keystrokes as in TI-Writer, so therefore does not feel strange. But there are several enhancements. From the command mode, you have the added feature of going straight to the FOrmatter with a FO command. (Both Editor and Formatter reside in memory at the same time) By typing in V, it will display the Version and author, ME will take you back to the MEnu for selection of options without losing your current document, a TO will Turn Off the tabs, or ruler, at the bottom. In the SH or SearcH option, you now have a wild card and can change it to whatever character you wish with CW. As there is no documentation as yet (it is in process) I don't know what I am changing it from! It will also find any combination of your string in both upper and lower case, wild card or no. Another item that is quite impressive is when one wishes to move text from one screen to another. If you enter command mode for the move command, you can type in the line as seen on the screen, scroll to another screen with the one keystroke, Pg Up or Pg Dn, and find the insert line all within command mode. No need for a good memory, or a pencil and paper to write down the numbers.

But the option I feel most useful, is VF or View File. Many a time I have had to verify some information in a related document and this option allows you to view the file without losing the current document in memory!

Within text, there is an added formatter command also. You can reassign the predefined symbols such as &, to another character.

Option 2 of the menu is for all you Assembly programmers. It will force capitals before a C that is added to the Tab "ruler" (which you can change if you wish to use it for some other purpose).

As for the Formatter, Option 3 of the menu, pure joy! You can specify the input from BUFFER instead of disk and with the output to SCREEN you see your formatted document complete, including your .TL'd characters as they will be printed. (To stop the document for checking, you just press the spacebar) No more wasted paper if you make a mistake! Just zap back into the Editor and fix it; remember all is in memory!

Option 3 of the menu, takes you straight into the Formatter; Option 4 is the Utility Loader; 5 is Help; 6 is Directory, 7 is Save Defaults (which will save one of five options of colours, two of which I don't think are options on my colour monitor! Perhaps they are better on a monochrome), and 8 is Quit (which is not enabled as yet). All but option 4 and 8 are accessible from the command line in the Editor.

Summing up how I feel about this word processor - MY WORD!! (Sorry.....)

I would like to thank all of you for the letters I have received regarding the column and suggestions for future articles. Unfortunately, some of the letters have only found their way to me me in the last month or so. I will try and reply to them personally in the near future and pass on some of YOUR tips; I really appreciate them.

FAST EXTENDED BASIC!

by **Lucie Dorais**

Everybody knows that TI BASIC is slow... but what about its big brother, Extented Basic? Do we use it to its full capacity? After four years of using it, I am still finding ways to make it work faster, in some cases almost as fast as assembly: the time lost in running the program is more than compensated for by the time saved in programming!

Over those years, two of which were spent with a cassette system, I also developed quite a few tricks to make my programs more compact, so that they would load faster from CS1. Many people have asked me to share some of my knowledge of XB, so here is the first instalIment. But I don't want to write a whole manual. Since I suppose that people dabbling in XB already have a good knowledge of TI BASIC, this will be a practical course: you write (and RUN) the program, and I will explain its main function, but other XB goodies will be thrown in as well.

(Before I go on, I to wish confess that I always resisted acquiring Jim Peterson's Tigercub Tips and Nuts & Bolts simply because I love to solve problems by myself... It may happen that my tips have been already published by him, after all I cannot reinvent the wheel, much less Extended Basic!)

So plug in your XB module and get your hands on Tex to type the following program, which demonstrates five ways to erase part of a screen (2Ø lines) almost as fast as a CALL CLEAR or an assembly routine, from the slowest to the fastest.

```
100 REM ** ERASE DEMO **
110 REM
120 GOTO 130 :: X,Y,K,S,END$,ER$,LI$,TXT$ :: CALL CLEAR ::
    CALL HCHAR :: CALL KEY :: CALLMSG :: !@P-
130 TXT$=RPT$("*",252):: ER$=RPT$(" ",252):: LI$=RPT$("-",28)
140 DISPLAY AT(1,9)ERASE ALL:"ERASING DEMO":LI$ :: DISPLAY
    AT(23,1):LI$
150 GOSUB 290 :: CALL MSG("CALL HCHARs")
160 FOR X=3 TO 22 :: FOR Y=3 TO 30 :: CALL HCHAR(X,Y,32):: NEXT Y
    :: NEXT X
170 GOSUB 290 :: CALL MSG("ONE CALL HCHAR")
180 CALL HCHAR(3,1,32,640)
190 GOSUB 290 :: CALL MSG("28 DISPLAY AT")
200 FOR X=3 TO 22 :: DISPLAY AT(X,1):"" :: NEXT X
210 GOSUB 290 :: CALL MSG("RPT$(' ',252)")
220 DISPLAY AT(3,1):RPT$(" ",252):: DISPLAY AT(12,1):RPT$(" ",252)
    DISPLAY AT(21,1):RPT$(" ",56)
230 GOSUB 290 :: CALL MSG("FAST ERASE")
240 DISPLAY AT(3,1):ER$:ER$:SEG$(ER$,1,56)
250 DISPLAY AT(24,1)BEEP:"  (1) RUN AGAIN   (2) END"
260 CALL KEY(0,K,S):: IF S=0 OR K49 OR K>50 THEN 260 ELSE IF K=49
    THEN 150
270 END$=RPT$("END ",63):: DISPLAY AT(3,1):END$:END$:SEG$(END$,1,56)
280 DISPLAY AT(24,1)BEEP:"YOU CAN USE IT FOR TEXT TOO!" :: END
290 DISPLAY AT(3,1):TXT$:TXT$:SEG$(TXT$,1,56):: RETURN
300 !@P+
310 SUB MSG(TXT$)
320 DISPLAY AT(24,1)BEEP:"Pr. a key for "&TXT$
330 CALL KEY(0,K,S):: IF S=0 THEN 330
340 SUBEND
```

TECHNIQUE #1 (line 16Ø) fills all the screen positions with a space, row by row, column by column... Pretty slow! TECHNIQUE #2 (line 18Ø) is much simpler and faster, provided you don't use borders: a unique CALL HCHAR that puts 28 spaces all at once: much faster! Highly recommended to TI BASIC users! TECHNIQUE #3 (line 2ØØ), unique to XB, is almost same speed, and I used it for years: display 2Ø empty lines. The DISPLAY AT could be replaced by CALL HCHAR(X,3,32,28), you would get the same speed.

But why not use Extended Basic DISPLAY AT function to its fullest, since it displays a line instantaneously, whatever its length? Unfortunately, it has a limit of 255 characters, but that is more than nine lines erased all at once. For exactly nine lines, you need 252 characters, and a full screen would need at the most three strings. To erase less than a full screen, you use the long string N times 9 lines, and you pad the rest with a shorter string.

To build this string, we use XB's RPT$ statement, intended just for that. In our example, wedisplay a 252 long string twice (2 * 9 lines = 18 lines), then pad with 2 more lines, i.e. 56 characters (see TECHNIQUE #4, line 220). Now, our three strings are displayed very fast, but... Tex takes its time to build each string, using a loop. As every manual on any programming language will tell you, the loop is the computer's biggest slowing factor... and all this stop and go is not very agreeable to watch.

The solution of course is TECHNIQUE #5: you build the string only once, at the beginning of the program, and you give it the maximum length (line 130). Each time you need it, you DISPLAY it completely, or just a segment of it. To make your program short and sweet, use the colon separator to display the lines, since they are consecutive (line 240).

Now, who said Extended Basic was slow???

P.S.: Did you appreciate my goodies? They are: PRESCANNING (lines 120 and 300), and a USER-DEFINED SUBROUTINE (lines 310-340). If you don't know these macro-tricks, just study the listing and try to figure them out until I write about them in later columns. The Corcomp addendum below gives some clues on User-def. subs.

P.S. 2: Since that gives me only two more topics, I would appreciate any input from the readers; please write me c/o the Ottawa Users' Group, or leave a message on Texlink (613) 738-0617, our wonderful BBS that works well and fast because... it is written totally in Assembly!

SPECIAL CORCOMP

Many TIers have purchased a Corcomp Disk Controller, and have got the "Tool Shed" as a bonus; the tools are wonderful, but not so easy to understand... It took me two years to finally start to use them, and my friend Jane, of TI-WRITER fame, has asked me to drop in a few lines on the subject.

To start, voici a SCREEN DUMP SUBROUTINE that you save in a merge format, to be MERGEd to any program that needs a quick sceen dump. Unlike the Assembly routine "DUMP", it does not change the line spacing, but it does not read the pixels either, so it is best suited for text screens. To access it, use a CALL DUMP statement.

```
30000 SUB DUMP
30010 CALL INIT :: DELETE "LD-CMDS"
30020 L$=RPT$("*",28) :: OPEN #3:"PIO"
30030 FOR X=2 TO 738 STEP 32 :: CALL LINK("VPEEK")(X,160,L$)
      PRINT #3:L$ :: NEXT X
30040 CLOSE #3 :: SUBEND
```

This is a user-defined subroutine (another one!); you can use it with confidence, since its variable are local to the sub and will not conflict with your program. Line 30010 loads the tools into Low Memory (if you need to load something else there, put this line at the beginning of your program, then load the other utilities; the first free address after loading the tools is 9568). Line 30020 defines a string of 28 char. that VPEEK will need, then it opens the printer. Line 30030 reads each line into L$ and prints it; the X variable is the address of the first screen location of the line to be read into L$, and 160 is the XB bias (please refer to the Corcomp manual for details). Line 30040 closes the printer and ends the subroutine.

BASIC TI INPUTS AND OUTPUTS PART 4

A COMPLETE LOOK AT THE KEYBOARD BUS

by **David Caron**

The table you see follows the order of the keyboard layout starting from the top leftmost character and horizontally following the keys until you end up at the lower rightmost character. All the characters in the table can be entered directly from the keyboard through its 15-pin bus to the mother board.

TI-99/4A KEYBOARD BUS TABLE

Keyboard pins	Upper case		Lower case		Function case		Control case	
	ASCII	CHAR	ASCII	CHAR	ASCII	CHAR	ASCII	CHAR
8,9	33	!	49	1	3	DEL	177	---
3,9	64	@	50	2	4	INS	178	---
2,9	35	#	51	3	7	ERASE	179	---
1,9	36	$	52	4	2	Cı EAR	180	---
7,9	37	%	53	5	14	BEGIN	181	---
7,14	94	^	54	6	12	PROD'D	182	---
1,14	38	&	55	7	1	AID	183	---
2,14	42	*	56	8	6	REDO	158	userdef
3,14	40	(	57	9	15	BACK	159	userdef
8,14	41	)	41	0	176	---	176	---
6,11	43	+	61	=	5	QUIT	157	userdef
4,8	81	Q	113	q	197	---	145	userdef
3,6	87	W	119	w	126	~	151	userdef
2,4	69	E	101	e	11	UP arrow	133	userdef
1,4	82	R	114	r	91	[	146	userdef
4,7	84	T	116	t	93	]	148	userdef
7,15	89	Y	121	y	198	---	153	userdef
1,15	85	U	117	u	95	_	149	userdef
2,15	73	I	105	i	63	?	137	userdef
3,15	79	O	111	o	39	'	143	userdef
8,15	80	P	112	p	34	"	144	userdef
8,11	45	-	47	/	186	---	187	---
8,13	65	A	97	a	124	¦	129	userdef
3,13	83	S	115	s	8	LEFT arrow	147	userdef
2,13	68	D	100	d	9	RIGHT arrow	132	userdef
1,13	70	F	102	f	123	{	134	userdef
7,13	71	G	103	g	125	}	135	userdef
7,12	72	H	104	h	191	---	136	userdef
1,12	74	J	106	j	192	---	138	userdef
2,12	75	K	107	k	193	---	139	userdef
3.12	76	L	108	l	194	---	140	userdef
8,12	58	:	59	;	189	---	156	userdef
4,15	13	ENTER	13	ENTER	13	ENTER	13	ENTER
4,13	///	SHIFT	///	SHIFT	///	FCTN	///	SHIFT
5.8	90	Z	122	z	92	\	154	userdef
3,5	88	X	120	x	10	DOWN arrow	152	userdef
2,5	67	C	99	c	96	`	131	userdef
1,5	86	V	118	v	127	DEL(char)	150	userdef
5,7	66	B	98	b	190	---	130	userdef
7,11	78	N	110	n	196	---	142	userdef
1,11	77	M	109	m	195	---	141	userdef
2.11	60	<	44	,	184	---	128	userdef
3,11	62	>	46	.	185	---	155	userdef
9,10	///	alpha lk	///	alpha lk	///	alpha lock	///	alpha lk
4,6	///	CTRL	///	CTRL	///	***	///	CTRL
4,12	32	space	32	space	32	space	32	space
4,9	///	FCTN	///	FCTN	///	FCTN	///	***

--- not apart of TI BASIC or EXTENDED BASIC ASCII table.

/// no ASCII code returned by TI BASIC or EXTENDED BASIC.

*** a combination of two command keys which causes TI BASIC and EXTENDED BASIC to return values which relate to neither case.

NOTE: The above table was produced by the results of CALL KEY when the 15-pin keyboard bus was manually shorted between two of its 15-pins.

FORTH TO YOU, TOO! SESSION 7

This is in answer to questions regarding the use of sprites and other graphics capabilities of the 99/4A with Forth. As we all know, we got a pretty good deal because TI built a fair amount of graphics into this little machine. Anyone who has seen PARSEC etc. can vouch for that. And all of them can be utilized in TI-Forth, with commands (statements) which are very similar to the ones employed in BASIC and XBASIC.
However, there are a couple of things which must be done in order to use any of the available VDP (graphics) modes. First of all, the appropriate LOAD OPTION must be booted, i.e. -GRAPH and -VDPMODES. If you are using the 64 column editor you only need to boot -SPLIT2, the rest of them are already booted. Another thing you will have to do is to fix a bug in line 10 of screen number 58. It should read as shown below:

```
VDPMDE  4  IF SMTN 80 VFILL 300 ' SATR ! ENDIF
```

(in other words the ! after 300 should be a ' (tick)).And while you are at it, fix line 1 of page 10 in Chapter 6 of the manual to read: HEX 3800 ' SATR ! Also, on screens 53, 54 and 55, last word of line 1 should be SETVDP2, not VDPSET2. For some strange reason it is correctly shown on screens 51 and 52. Finally, on screen 59, line 9, change the 00FF to 00FE.

Now let's take a look at what is available in Forth. I have drawn up the following chart for easy reference:

VDP MODE	ASCII CHARS	CHAR DEF?	FG7 BG	HCHAR VCHAR	SPRI- TES	SPR. MOT.	DISPLAY SIZE
TEXT	YES	YES	NO	YES	NO	NO	40 X 24
GRAPHICS	YES	YES	YES	YES	YES	YES	32 X 24
MULTI	NO	NO	NO	NO	YES	YES	64 X 48
BITMAP	NO	NO	NO	NO	YES	NO	256 X 192
SPLIT	same as BIT-MAP with 8 lines text on bottom						
SPLIT2	same as BIT-MAP with 4 lines text on top						

There are essentially four display modes, but one of them, bitmap, comes in three versions. When you boot FORTH, you are put in TEXT mode, 24 lines of 40 characters each. This mode is also used by the 40-column editor. New characters can be defined, but all characters have the same FG/BG color. In other words, you can not control individual character sets. The easiest way to set text and background color in the text mode is nn 7 VWTR (see tutorial 1). If -TEXT is booted, you can enter text mode by invoking TEXT.

The screen display of the GRAPHICS mode is identical to that of XBASIC: 24 lines of 32 characters. And, as the chart shows, its features are also the same. But beware! If you want to change the FG/BG colors of a character set, they start with set 0 at ASCII 0. That's ZERO. The first set to contain printable characters (from space thru ') is number 4 in Forth. Appendix A (ASCII KEYCODES) does not refer to character sets, but if you divide each column into 4 sets and number them starting from 0, you'll avoid confusion. If -VDPMODES is booted you enter this mode by invoking GRAPHICS.

Then there is the MULTICOLOR mode. I've seen those 'crazy quilt' demos but what else it might be used for, I don't know. A character consists of 4 pixels, thus there are 48 lines of 64 'blocks'. Each of them can be set to a different color. Use MULTI to enter.

Highest resolution is obtained with the bitmap mode. Each of the 256 pixels in the 192 lines can be controlled individually. There is only on drawback in that there is no automotion for sprites. But because there normally aren't any ASCII characters either, TI has provided SPLIT and SPLIT2, which enable bitmap and text combined. This is extremely helpful when in interactive mode: you can see what you are entering and the resulting display. I used

SPLIT to revise John Volk's Diamond Draw program. It was originally all bitmap and I could not remember which keys to press for the various options, so I used SPLIT to display them on the bottom of the screen. The words to activate the bitmap modes are GRAPHICS2 (for full screen bitmap), SPLIT (text on bottom) and SPLIT2 (text on top).

Study chapter 6 thoroughly. You will notice that: a) all you need to know is there, b) you are already familiar with most commands, c) if you have absorbed enough Forth, it's not hard to see that the real difference is that the parameters (in typical Forth style) go on the stack first before the operative word is invoked, and d) for sprites a little additional work is required.

If you want to use sprites, pay particular attention to the section on Sprite Initialization beginning on page 8 (Ch.6). Here is a short routine that illustrates how a sprite is set up:

```
 Ø ( SPRITE SAMPLE )
 1
 2 BASE->R  HEX
 3 : SETUP
 4        GRAPHICS ( enter standard graphics mode )
 5        2ØØØ SSDT ( move sprite desriptor table )
 6        1Ø28 4482 4428 1ØØØ 2A SPCHAR ( define the character )
 7        3F 2F ØF 2A 1 SPRITE ( sprite loc.,color, char, no. )
 8        15 Ø 1 MOTION ( motion parameters )
 9        2 #MOTION ; ( set it in motion )
1Ø
11 DECIMAL  R->BASE
12
13 SETUP
14
15
```

This example is for the standard graphics mode. If you wanted to use bit map, line 4 would have to read GRAPHICS2, or SPLIT or SPLIT2, line 5 would have to include 38ØØ ' SATR !, and lines 8 and 9 would be superfluous. If you load this screen, SETUP (line 13) displays a sprite. Once it is on display you can enter TEXT and return to text mode. Entering SETUP will again produce the sprite.

Play around with the various graphics modes. Remember that practice is the only way to become proficient, and before long you will be able to use Forth's graphics capabilities to their full extent.

TI BASIC continued from May

by Steven Shaw

Before developing some useful routines, a word about editing and debugging : no matter how good a typist you are, even entering a short program will need the use of these facilities.
EDITING is what you do when you change a program line - perhaps only one character in the line.
A number of computers use 'screen editing', where you move the cursor around the screen until it is placed where you want to make your alteration.
The TI99/4A uses a 'line editor' for programs (some modules also use a screen editor,eg TI-WRITER). To use a line editor, you first select the line number you wish to amend, place it on the bottom of the screen, and move the cursor along the LINE until you hit the place to be amended.
To bring the line you want on screen, key in the line number, then hold FCTN down and press key E or X. Your line will appear with the cursor at the beginning of the line.
Use FCTN and keys S and D to move the cursor over the line without deleting or altering anything.
If you wish merely to alter the line, typing over it may be sufficient, but the 99/4A also allows you to delete and insert characters.
To DELETE a character,place the cursor over it (FCTN plus S or D) and then press FCTN and key 1. This will delete the character the cursor is on, and everything following will move one space to the left.

To INSERT text, place the cursor after the last character you wish to leave untouched, using FCTN and S D, then press FCTN and key 2. Now anything you type will force everything after the insert point one character to the right,and your inserted text will appear. The cursor position also moves to the right as you type. You leave INSERT mode by pressing FCTN and S or D, or pressing ENTER.

When your line is correct, press ENTER to enter the new line into memory, and leave EDIT mode. However, if you wish to amend also the line before or after the line just finished, you may move directly to that by pressing FCTN and E or X.

The maximum line length in TI Basic is four screen lines, but the computer is often capable of taking a longer line, as the absolute restriction is on the length of the line in BYTES of memory used, not the number of characters on the screen.

You may use the edit function to insert 'overlong' lines as follows:

Type in the last part of the required line first. Press ENTER. With the line on screen, press FCTN and key 2, then key in the first part of your line,from the beginning. The part you first entered is pushed to the right.

Now return the line to the screen using the edit mode: type in the line number and then press FCTN and key X.

It is possible to overfill a line this way, and you will receive an error message if you do. However you will usually be able to go to an extra half screen line, and in some cases you may be able to squeeze two extra lines in!

The advantage of putting as much as possible in a line is that by using less program lines you save a little memory. In general, lines with a lot of numbers in will be difficult to expand in this fashion, but lines with a lot of text or commands can usually be considerably extended.

When the line is fully entered,press ENTER. Then LIST the line just to make sure it is all right.

You do not have to INSERT at the beginning of the line, but you may find it easier to do so. Give the computer time to move all the characters to the right when using INSERT. Keep your eyes on the screen.

A brief word of warning: in TI BASIC, your program uses the same area of memory as the values of variables, separated by a 'marker' in memory. As a program runs, the free memory is continually filling. Unwanted values are only purged when the memory is full, resulting in short pauses in program operation.

If the variable area of memory is almost full when you stop to edit a program, inserting extra material MAY result in the permanent loss of the marker which marks the limit of the actual program. This will cause irreversible damage to your program and may prevent the LIST function from operating correctly, or cause a system lockout when you try to run the program.

Therefore try to avoid running the program and then editing it! Before you edit, take a copy of the program to tape - this has the added benefit of apparently clearing the garbage, and not only gives you a security copy but actually prevents the problem occurring!

This problem will not be apparent if you use extended basic plus the 32k ram expansion, as variables then occupy a different sector of memory.

EDEBUGGING

Having entered your program, you type in RUN, and instead of the program running, you receive an error message. DEBUGGING is required.

A BUG is quite simply an error in the program, either a mistype or an error in your use of Basic.

The computer does check the lines you enter, but will only spot such things as using only a single bracket (or quotation mark ".

When you RUN your program, the computer first goes through your program and sets aside memory for each variable and sub program that you have used. During this 'prescan' further errors may be spotted and an error message printed on the screen. Typical errors spotted at this time are incorrect use of arrays (trying to use DIM after the variable has been used) or a mismatched number of FORs and NEXTs. Most errors will only produce an error message as the computer finds them when your program is actually running - for instance trying to GOTO a non existant line number, or trying to RETURN when there is no outstanding GOSUB. The error messages generated by the TI99/4A are well described in the manual, and will usually indicate a line number in which the computer has met

something it cannot cope with.
Unfortunately, the actual error may not be in the line stated in the error message.
For instance, BAD VALUE IN 100 may refer to:

```
100 CALL HCHAR(ROW,COL,42)
```

The program line is correct. The error message has appeared because one or both of the variables ROW and COL have a value which is out of range.
(Using CALL HCHAR the ROW and COLUMN values passed to the sub program must be in the ranges 1 to 24 and 1 to 32 respectively. If you go outside this range the computer will halt with an error message.)

To see what the values are, when the error message appears, just type in PRINT ROW;COL then press ENTER.
The two values on screen will be the current values of ROW and COL, which have caused the problem.
The job is then to review the program to see how the variables obtained that value, and see what changes need to be made.

This ability to check variable values after an error message is very valuable. Note that once you amend a program line all variables will be reset to zero.

Frequent causes of problem bugs are DATA statements, with a comma too many or too few.The ability to check variable values is useful here:

An array is being filled from a DATA statement, eg

```
    FOR I=1 TO 5
    READ A
    VAR(I)=A
    NEXT I
    DATA 23,54,8,A,5
```

An error message will be generated by the above, as 'A' is not a number. When the error message appears it is possible to enter:

```
PRINT I;VAR(I)
```

This will provide the clues needed to lead to the erroneous DATA line - which may be several hundred program lines away!

In such cases you need to have a good idea of what values should be found in connection with each variable, and you may need to spend some time working through the program.

If you cannot fathom why a program is not working as it should merely by reading the LIST, the TI99/4A also has a TRACE option.
Key in TRACE and then RUN. The line numbers will be listed on screen as the program progresses, and you can watch for an unexpected line transfer as the computer moves from one line to the next. Using TRACE will disrupt any screen display. Switch TRACE off by keying in UNTRACE.

To find out what the variable values are at a particular point in the program, you can insert PRINT statements in the program, or instruct the program to BREAK by adding a line in the appropriate place with the instruction: BREAK.
Then when the program stops you may enter PRINT VAR etc.
To continue the program enter CON (and press ENTER)
Remember to remove the BREAK line when you no longer need it.

As your programs become longer, so it becomes more difficult to spot the errors, but finding and removing errors is a very good (if time consuming) way of learning how to use your TI99/4A. With time and experience you will learn to quickly spot the easy bugs and to tackle the harder ones in a logical fashion. Read your manual as often as necessary,especially the section on ERROR MESSAGES.Often the answer to a difficult bug is there just waiting for you to read it.

ESystem Lock Outs

A lock out has occurred if your console no longer responds to the keyboard (especially QUIT) and ceases to function normally. Unusual sound and graphic effects may occur.
It is quite normal for all computers (and word processors) to lock out from time to time. The cause is an error in the instructions passed to the processor, which it cannot deal with. There are a number of causes:

OOPS!	REFORMAT	SCREEN COLOR	NEXT PARAGRAPH	DUPE LINE	LAST PARAGRAPH	WORD TAB	NEW PARAGRAPH	NEW PAGE	WORD WRAP		● TI-WRITER
DEL CHAR	INS CHAR	DEL LINE	ROLL DOWN ↓	NEXT WINDOW →	ROLL UP ↑	TAB	INS LINE	COMMAND/ ESCAPE	LINE #s	QUIT	

A PRINTER COM	B BLOCK	C CENTER	H TAB	J REQ SPACE	L NEW LINE	N NEW PAGE	M MID LINE	P PARAGRAPH	< MOVE > COPY DELETE	UNDERLINE)	● COMP
DELETE & CANCEL PRT	INSERT	ERASE & HIGHLIGHT	↓	BEGIN	↑	MENU	END	EDITOR COMMANDS!	V COLOR	QUIT	

HOME	TAB	NXT UNL CELL	FORWARD CHAR	FORWARD WORD	CHANGE WINDOW	REL/ABS REF				CANCEL	● MULTI PLAN
LOWER RIGHT			BACK CHAR	BACK WORD		HELP	RECALC	BACK SPACE	DELETE FORWARD		

DELETE CHAR	INSERT CHAR	DELETE LINE	ROLL UP	NEXT SCREEN	ROLL DOWN	TAB	INSERT LINE	ESCAPE		QUIT	● ES/AS
DELETE	INSERT	ERASE	CLEAR	BEGIN	PROCEED	AID	REDO	BACK		QUIT	

							INSERT BLANK LINE				● FORTH
DELETE	INSERT	ERASE	NEXT SCREEN	NEXT WINDOW	LAST SCREEN	ERASE TO END OF LINE	COPY LINE	EXIT EDITOR			

			G	R	A	P	H	X			●
SLOWER	FASTER	DRAW	ERASE	NOHELP	ZOOM	COLORS	LINES	CIRCLES	COPY	MENU	

MODEM BAUD RATE	SPOOLER ON/OFF	MODEM PARITY	MODEM PORT	PRINTER PARITY	PRINTER PORT	PRINTER BAUD RATE			TOGGLE		● FAST TERM
N FILE ' SEND	B DUMP TO DISK	Y CLEAR LOG	CANCEL	WINDOW	K-TIMER	FOREGROUND	BACK-GROUND	LOG ON/OFF	FREEZE ON/OFF	QUIT	

SPEAK	OUTPUT	CANCEL	TRANS	WRAP	CASE	PAGE			EXIT		● TE-II
										QUIT	

A STATIC discharge is a frequent cause of problems. Although computers no longer need the carefully controlled environment of the mainframe, they remain sensitive to static. The problem is most acute in warm dry weather, or if you wear clothing made of artificial fibers (acrylics are particularly bad). Nylon carpeting can also be a problem. The TI99/4A can handle static quite well, but on occaison you may meet the problem.

Use of cotton clothing, a humidifier, and an anti static spray on the carpet may be called for in especially hostile environments. A conductive carpet is also sold by some computer suppliers.

Poor communication with modules or peripherals may also be a problem: the contacts are essentially self cleaning, but it may be necessary to disconnect/connect a few times to make good contact. Contacts are silver plated and are subject to tarnishing, and may require this treatment if a module or peripheral is not used for some time. Tar can be deposited on the contacts if there are smokers in the room. In extremely severe cases of pollution, an isopropyl alcohol solvent may be used but great care is required to prevent damage.

Some modules contain insufficient error traps and permit you to pass confusing instructions to the processor. Extended Basic in particular allows a number of lock out producing errors. If your syntax is correct,this will not happen.

Loss of the stack/program marker can cause problems in TI BASIC. This occurs when you run a program (filling the stack) and then add to the program. In some cases the computer will add the stack to the program with sometimes colourful results, but permanently destroying the program (if a lock out does not occur).After running a program, it is wise to save it before editing: this appears to clear the stack.

When the computer ceases to function,and possibly makes a piercing sound, do not panic! The only way out of a lock out is to switch off and (after a few seconds) restart. Even the MOST expensive systems sometimes lock out.

Canada 55

FROM

P.O. BOX 2144,STATION D,OTTAWA
*** ONTARIO,CANADA K1P 5W3 ***