

(044)
8802
Vast



from:

The Vast
Users Group

Happy
Valentines
Day

NEWSLETTER

VOL. 4

FEBRUARY 13, 1988

NO. 2

CONTENTS

VAST INFO	2
SECRETARY'S SLATE	3
EDITOR'S DESK	3
COMPUTER TUTOR	4
ODDS AND ENDS	6
TIPS FROM TIGERCUB	7

VAST BBS 437-4335

VAST INFORMATION

The VAST USERS' GROUP is a support group for Home Computer users. We primarily support the TI-99 Home computer and compatibles, but all computer users are welcome. Our regular meetings are on the second Saturday of the month. We meet in downtown Phoenix at the Park Inn International Hotel at 401 N. First Street in the Phoenician Room. The meetings start at 10:00 AM and continue until 11:00 AM with socializing starting at 9:00 AM. The yearly membership fee is \$6.00.

All meetings are open and anyone may attend. Only dues paying members may vote in elections and obtain programs from the Users' Group library.

THE CURRENT OFFICERS ARE:

- President
Bob Nixon.....897-6204
- Vice-President
Ike Van Kampen.....934-5164
- Secretary
Stu Olson.....846-7624
- Treasurer
Earl Bonneau.....269-3802
- User Group Librarian
Earl Bonneau.....269-3802
- Newsletter Editor/BBS SysOp
Jim Ely.....921-3375

Newsletter subscriptions are available for \$12.00/year, First Class Mail. If a date appears in the upper right hand corner of your mailing label, this is your subscription expiration date. Address renewals to the return address on address page.

The Users Group's BBS is in operation 24 hours a day. Contact it at (602) 437-4335. There are 3 message bases, a TI specific download section, and a lot of interesting

* VALley of the Sun T Users Group *

* --- * *

conversation and information available here so why not give it a try.

Deadline for submission of articles or advertising for the Newsletter is the last Saturday of every month. Articles may be submitted in any form, however, the preferred method is by phone transfer directly to the Editor.

Advertising rates are as follows:

Commercial:

- Full Page \$10.00
- Half Page \$ 7.00
- Quarter Page \$4.00

Personal:

- Four lines,
30 Characters/line
\$1.00
\$.20 per line
over four.

All rates are for ONE issue only!

Programs are available from the USERS' GROUP LIBRARY at the following rates:

- SS/SD Disk \$2.00
- DS/SD Disk \$4.00

If copying of documentation is required, it will be at the rate of \$.10 per page. If the User Group supplies the disk, please add \$1.00 to the above charges. An exchange program for free programs is also in effect. Please contact the librarian for further information. A complete list of what is in the library is available on 2 disks free of charge if you supply the disks or for \$1.00 per disk if the User Group supplies the disks.

* VALley of the Sun T Users Group *

* --- * *

-----VOL. 4 NO. 2-----FEBRUARY 13, 1988-----

SECRETARY'S SLATE

From the Editor's Desk

M I N U T E S

January 9, 1988
VAST User Group

The meeting was called to order by President Bob Nixon at 10:05 a.m. No old business was discussed.

The program for the February meeting was confirmed, that being a demo on the Amiga computer, provided by Computer Discount.

Mike Marfisi gave a short discussion about FunnelWriter 4.0, explaining the various function and features it can provide for the 99/4A user. It was quite obvious that FW has come a long way since its introduction several years ago. Those interested in obtaining a copy can do so through the club's software library.

Dan Shell announced that the Myarc hard drive controller is still under development. He is hoping for a release date in the near future, but had no exact time available.

No other general business was discussed.

The main topic for the January meeting was telecommunications, or more commonly known as "How to use your modem". Some of the members have recently obtained modems and, like those before them, had questions on how to access a BBS', etc. Bob Nixon demonstrated the various applications of modem useage.

The meeting ended at 11:10 a.m.

Stu Olson
Secretary,
VAST User Group

HAPPY VALENTINES DAY!

IN THIS ISSUE...

We start on page 4 with Computer Tutor by Dave Howell, this month discussing the MEMORY organization of the TI-99/4A. Do you know what a BUZZWORD is or how the first ones came about? The article on page 6 answers these questions. Tips from the Tigercub is on page 7. Jim Peterson has written several tutorials on Extended BASIC, TI-Writer and the like and as time goes on, we will present them. And that is this issue.

UNFINISHED BUSINESS...

Last month, a number of letters were published, prompted by an "OPEN LETTER" to our group by Harry T. Brashear, President of the WNY 99'ers. Well, it seems Harry just doesn't want to let things rest. Here is Harry's latest "attack" taken from the January, 1988 issue of the WNY 99'ers Newsletter, the "INTERFACE"...

GREAT REMORSE FOR REJECTED TI'ERS AROUND THE COUNTRY

Something horrible has happened... a certain user group that shall remain nameless, has converted to a dual-computer group. This group was headed up by turncoats and the leadership went to hell, because instead of fading into the woodwork when they bought clones, they insisted on dragging the rest of the membership with them. They claim that the IBM'ers would be around to show the TI'ers

CONTINUED PAGE 9 >

NEXT MEETING . . .
MARCH 13, 1988

The Elements of BASIC

by Dave Howell

(Courtesy Erie 99'er User Group)

Part 7

About the MEMORY

Many people who bought the TI-99/4a wanted to know at least something about what goes on inside the mysterious little box. Even today, there is still widespread belief that this knowledge is so scientifically obscure as to go over the heads of most mortals. Don't believe a word of it! This is nothing more than propaganda put out by "experts" who confuse everyone by using jargon words they don't explain.

What do those magic words -- memory, RAM, ROM, 16k -- mean?

A computer's "memory" is the type that comes on little silicon chips inside the TI-99/4A. There is another kind of memory sometimes called mass memory or external storage that resides outside the computer -- on cassette tape or disk.

A "K" is a measurement of the amount of memory in a computer. Mathematically, K stands for 1000 but in measuring the size of a computer's memory, the number is actually 1024 which is 2 raised to the tenth power or 2x2x2x2x2x2x2x2x2x2. Computers perform all functions in powers of 2 since electricity is either on (1) or off (0). Using 1000 as K is a number close enough to 1024 and 1000 is a number that people can get a handle on pretty easily.

Actually, 1024 (or more precisely, 1024) refers to 1024 bytes of memory and 1 byte of

memory equals 1 character on the keyboard. Thus, 1K of memory holds a little more than 1000 characters (letters, digits and punctuation marks) or somewhere around 40 lines of average "basic" lines.

Obviously, the more K's a computer has in its RAM, the larger and more complex programs can be handled. RAM is an acronym for "Random Access Memory." RAM is used to hold any program and data temporarily. The computer user has total control over the contents of RAM. Any item in RAM can be changed at any time while the computer is on. If the computer loses power or the user accidentally hits the FCTN-Plus keys, the program and/or data in RAM vanishes. That is why smart users save their "serious" work on cassettes or floppy disks every 15 minutes or so.

The TI-99/4A has RAM capacity of 16K (16,384 characters to be exact). The TI Memory Expansion card adds another 32K of RAM for a total of 48K.

The TI-99/4A as with all home/personal computers, also contains another kind of memory - ROM (Read Only Memory). As the name applies, ROM cannot be written to or changed. It already has a program permanently stored in it. The 26K of ROM in the TI-99/4A holds, mostly, the TI BASIC interpreter and a thing called the operating system. To make a long story short, the ROM converts the keyboard strokes to the binary (or machine) language and carries out the program commands.

As noted earlier, it takes 1 byte of memory to store 1 character or keyboard stroke. To

SEE PAGE 5 >>>



ODDS AND ENDS

HOW TO WIN AT WORDSMANSHIP

rewritten by S.T. Peddy-Coart (Author unknown)

Now that we all have computers and word processors, some of us still want an edge over the swarming masses when it comes to writing a very effective letter or report. We want to be noticed or get action. We want our work to sound important, right? Well, finally it's here - the "Buzz Phrase Generator"! And this is how the story goes...

After years of hacking through etymological thickets at the U.S. Public Health Service, a 63 year-old official named Philip Broughton hit upon a sure-fire method for converting frustration into fullment (jargonwise). Euphemistically called the systematic Buzz Phrase Generator, Broughton's system employs a lexicon of 30 carefully chosen Buzzwords listed below.

<u>COLUMN_1</u>	<u>COLUMN_2</u>	<u>COLUMN_3</u>
0. integrated	0. management	0. options
1. total	1. organizational	1. flexibility
2. systematized	2. monitored	2. capability
3. parallel	3. reciprocal	3. mobility
4. functional	4. digital	4. programming
5. responsive	5. logistical	5. concept
6. optional	6. transitional	6. time-phase
7. synchronized	7. incremental	7. projection
8. compatible	8. third-generation	8. hardware
9. balanced	9. policy	9. contingency

The procedure is simple. Think of any three-digit number, then select the corresponding buzzword from each column. For instance, the number 257 produces "systematized logistical projection", a phrase that can be dropped into virtually any report with that ring of decisive, knowledgeable authority. "No one will have the remotest idea of what you're talking about" says Broughton, "but the important thing is they're not about to admit it"!

TIPS FROM THE TIGERCUB

DEBUGGING

by Jim Peterson

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often.

So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps you to debug better than some that cost ten times as much.

There are really three types of bugs. The first type will prevent the program from running at all - it will crash with an error message.

The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a Basic program, which uses character sets 15 and 16, in Extended Basic.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand - such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance, that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and

give you an error message - probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong, and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark).

```
100 PRINT "Program must be saved in:"merge format."
```

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program: that line may actually be correct but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that - Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended Basic manual list almost all the error codes, and almost all the causes of each one - it will pay you to consult these pages rather than guessing what is wrong.

You may create some really bad bugs when you try to modify a program

CONTINUED PAGE 8 >

-----VOL. 4 NO. 2-----FEBRUARY 13, 1988-----

TIGERCUB CONTINUES

that was written by someone else - especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray - you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-looking command !@P- up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the program to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE IN 680, and you bring line 680 to the screen and find it reads:

```
CALL HCHAR(R,C,CH)
```

just type PRINT R:C:CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended Basic, you can even enter and run a multi-statement line in immediate mode (that is, without a line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by:

```
FOR J=1 TO 100::PRINT A(J)::
: NEXT J
```

or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped but A will have a value of 100.

You can temporarily stop a program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and re-start it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as 1 BREAK 960,970,980. Note that in this case the program breaks just BEFORE those listed line numbers.

You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement, which will cause each line number to print to the screen as it is executed. If used as a command, it will

CONTINUED PAGE 9 >

-----VOL. 4 NO. 2-----FEBRUARY 13, 1988-----

TIGERCUB CONTINUES

trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multi-statement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE() with the line number between the parentheses, and a subprogram after everything else:

```
30000 SUB TRACE(X)::DISPLAY
AT(24,1):X :: SUBEND
```

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen - it may have invisible characters at one or both ends. Try PRINT LEN(M\$) to see if it contains more characters than are showing; or PRINT "*"&M\$&*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything - and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program. (Believe me, I speak from experience!)

J.P.

FROM THE EDITOR'S
DESK CONTINUES...

everything they used to know. In the same newsletter, they told of one of their most famous ex-TI'ers brought in heaps of IBM software to demonstrate. Some help!

We used to trade newsletters with this group but we won't anymore after this month. However, in sympathy, should any of their rejected membership get a chance to read this, we will be happy to supply that member, on written request, with a free subscription to the "Interface" for the remainder of the 87/88 season. 'nuff said."

This was the editorial on page 2 of the "Interface." The front page showed a very sad and dejected Opus sitting on a rock under the headline "GREAT REMORSE FOR REJECTED TI'ers."

Well, it seems that Harry just doesn't quite have all of his facts straight. Or, he just reads what he wants to and ignores all the rest. If he would read the September, 1987 issue of the "VAST Newsletter", he will find a copy of our revised constitution. It says in ARTICLE II, Number 1, that we "encourage all efforts and activities in connection with the use of ALL types of computers with a special emphasis on the TI-99 and its compatible models." How one could derive from that that we are a "dual computer group" is beyond me. It might interest Harry to know, (then again it probably won't interest him) a number of the "clone" users, including the "famous

CONTINUED PAGE 10

---VOL. 4 NO. 2-----

-----FEBRUARY 13, 1988-----

**EDITOR'S DESK
CONTINUES . . .**

ex-TI'er", have lost interest in their "clones" and have gone on to other computers, especially the AMIGA. The program for last month was on communications using the TI (see minutes). This month, the program is a demonstration of the AMIGA. Next month, a proposed topic for the program is a demonstration of the latest version of FUNNELWEB for the TI. Seems to me, we are right on target with ARTICLE II, Number 1, of our constitution. Oh well, some people are never happy. Let's move on to more important things.

NEWS . . .

(From the January, 1988 MICROpendium)

Myarc has scheduled for release in the first quarter of 1988, its Floppy/hard disk controller card. However, no REAL firm date has been set. Also scheduled for the 9640 is the release of MDOS version 1.1, version 1.0 of the GPL interpreter, Myarc Advanced BASIC and Pascal Runtime.

Other things to look for from Myarc this year include a 1.5 megabyte memory expansion card (up to 1 megabyte can be partitioned as a battery-backed RAMdisk), a multi-tasking environment called GEME and a Computer Aided Design (CAD) program.

DataBioTics has announced it is shipping its long awaited GRAND RAM card. They were shipped in "limited quantities" to dealers in January and will be shipped to everyone in early February.

Several months ago, the User Group Library received a couple of disks of software from Innovative Programming, a new company in California. Now, I.P. has announced it will no longer be supporting the TI. It seems, however, that several people have reported difficulty in receiving orders or refunds and at one point, their phone was disconnected. Their phone

appears to be connected now but they haven't been answering. If you are having problems with I.P., the latest address I have is P.O. Box 2737, Rohnert Park, CA 94928 and their phone is (707) 585-3922.

An equivalent for the MG Gram-Kracker has been designed by Mark Van Coppenele of Haverhill, Massachusetts and a proto-type demonstrated to Massachusetts User Groups. No production models as yet, as one would require an investment of about \$1000. Mark is interested in hearing from users who consider purchasing the product, or who want further information. You can reach him at 52 Audobon Rd., Haverhill, MA 01830 or phone (617) 372-0336.

Stay tuned for further updates....

YOUR BBS

Do you realize we have passed the 8300 caller mark on the board? WOW! I think we should offer a small prize to caller number 10000. What do you folks say? At the rate we are presently going, that caller should be sometime in early to mid June which also is the 3rd. anniversary of the BBS.

ELECTIONS???

If memory serves me correctly, elections are in 2 months (April) which means nominations should be made next month. We need to set up a nomination committee sometime soon. If YOU would like to serve on this committee, PLEASE contact one of the current board members listed on page 2 of this newsletter.

ONE FINAL ITEM . . .

Sorry I missed last month's meeting, but when you're sick, you're sick. Last month's newsletter and others are also available this month. 'til next month.....

Jim Ely, Editor

-----JANUARY, 1988-----

Editor's Desk Continues. . .

industry have not stood still. Equipment only dreamed of when the TI was still manufactured is now available at affordable prices. Items such as 2400 baud modems, CD-ROM disks and drives as well as Quad-density disk drives, to name just a few, are now no longer a pipe dream. If one has attended any computer exhibits, where other than just TI equipment was shown or demonstrated, one cannot help but notice the great strides the computer industry has made during the last few years.

4. Not to notice such progress or pretending not to be interested in them just does not make much sense, because these are not the attitudes which got us our computers to begin with. They are also not the attitudes which resulted in these giant leaps ahead in the development, innovations and use of computers.
5. The Newsletter article you are referring to was written for two purposes:
 - a. To give all our members and especially our winter visitor members a rundown of what

exactly had transpired within the Group during the summer months.

- b. To make the TI owners see that their becoming more involved in the Group, is of the utmost importance.

If you personally were offended by the article, I cannot help it. If, however it inspired you to write us and offer the TI owners assistance, by making available ten free subscriptions to your Newsletter, that would be just dandy! Maybe your action in turn will inspire our TI owners to take a more active interest in their computer and the Group.

Thank you for your time and efforts in our behalf.

Ike Van Kampen,
Vice-President,
VAST Users Group
P.O. Box 1442
Glendale, AZ 85311

P.S. I am still the proud owner of a 100% TI-99/4A computer, which I still use on occasion. This letter was prepared on it with TI-Writer. I also own a Leading Edge (model M). So maybe I too know what I'm talking about.

NEXT MEETING
FEBRUARY 13, 1988

VAST USERS' GROUP NEWSLETTER
c/o 1425 E. Del Rio Dr.
Tempe, AZ 85282



FIRST CLASS MAIL

TO:

EDMONTON, ALBERTA
EDMONTON, ALBERTA

FIRST CLASS MAIL
February, 1988