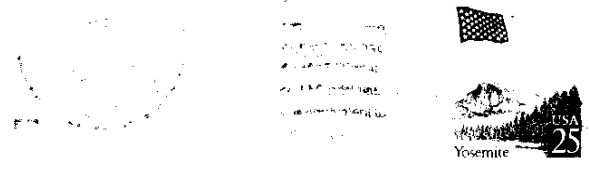
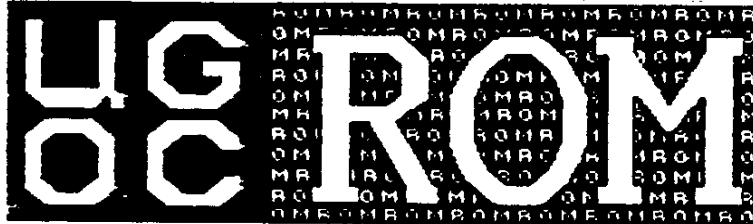


**THE R O M NEWSLETTER
 USERS GROUP OF ORANGE COUNTY
 17161 EDWARDS STREET
 HUNTINGTON BEACH, CA 92647**



01/91
 Dallas TI Computer Group (DTIHCG)
 PO Box 29863
 Dallas,
 TX 75229



MAR 1991

SERVING THE TI-99/4A HOME COMPUTER COMMUNITY

WE MEET AT FIDELITY FEDERAL

THIS IS AN ALL NEW LOCATION. SEE INSIDE STORY

TIME AND PLACE OF MEETING

The SECOND Monday of each month at
 Fidelity Federal Savings
 7:30 PM

North of Westminster Ave. at the corner of
 Seal Beach Blvd and St. Andrews at 13920
 Seal Beach Blvd. Parking is available
 west of the building off St. Andrews with
 additional parking across the street.
 All are welcome.

U.G.O.C. OFFICERS

PRESIDENT . . . BEN HATHEWAY 662-2957
 VICE-PRES . . EARL RAGUSE 347-5875
 SECRETARY . . EUGENE SMITH 347-3961
 TREASURER . . JERRY RASH 631-8579
 PAST-PRES . . JIM SWEDLOW 897-9289

COMMITTEE CHAIRMEN

BILL NELSON MEMBERSHIP . 758-6425
 NEWT ARMSTRONG . . NEWSLETTER . 633-7317
 BEN HATHEWAY . . BULLETIN BOARD 751-4032
 BILL NELSON HALL OF FAME . 758-6425

SOFTWARE LIBRARY

KNUTE ERSLAND 842-8859

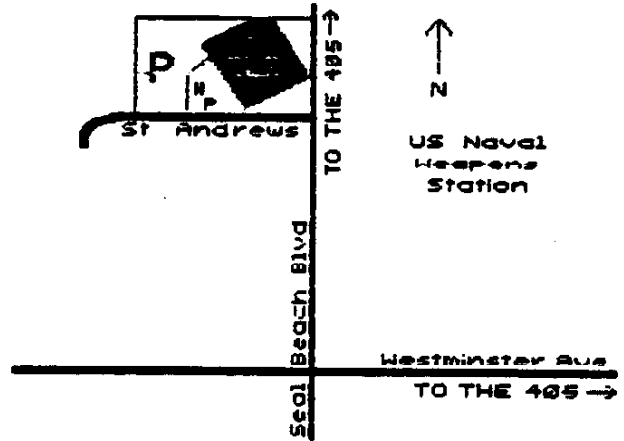
NEWSLETTER LIBRARY

EARL RAGUSE (PHONE FOR TIME) 347-5875

PUBLICATION NOTICE

The ROM is published by the Users Group of Orange County, a non-profit organization. Permission is granted to similar groups to reproduce articles published in the ROM provided author and source credits are included. Opinions expressed in this publication are those of the author and do not necessarily represent those of the ROM, it's Editor, or the UGOC, it's officers or members.

We solicit letters and articles of interest to the TI-99/4A user community. Material accepted may be edited for fit and format. No payment is offered nor intended (other than your byline).



SEE PAGE TWO FOR FULL PAGE MAP

NEWSLETTER CONTRIBUTORS

EARL RAGUSE TI FORTH
 ADRIAN ROBINSON ASSEMBLY
 EARL RAGUSE CIRCULATION
 JIM SWEDLOW AT LARGE
 BILL NELSON GRAPHICS
 SILES BAZERMAN HARDWARE

TI CLUB ACTIVITIES

CLUB	ACTION	DATE	INFO
BUG	GENERAL MEETING		871-3485
UGOC	GENERAL MEETING	CALL	662-2957
UGOC	BOARD MEETING		662-2957
UGOC	NEWSLETTER LIBRARY	CALL	347-5875
UGOC	GRAPHICS SIG	NONE	758-6425
UGOC	LIBRARY, FTNULY	NONE	842-8859

NOTE: NO GRAPHICS SIG

THE PRESIDENT SPEAKS

Those of you who missed Fest West 91 really missed something. A good time was had by all, and those of us who attended came away with the belief that the TI does have life left in it and the Geneve does have a future ahead.

The Fest West '91 committee did a super-human job coordinating everything and coming up with ideas to help further the Fest (including, but not limited to Fest West ball point pens). They are to be congratulated. Erwin Metz told me that he had attended professionally run conventions that were not as well done as the Fest.

I'm sure that there will be plenty of stories to share from all who attended and I hope that all of you had as good a time as I had.

See you at the meeting. RFN

TI BITS * Number 34

by Jim Swedlow

FEST WEST 91 WRAP UP REPORT

Fest West 91 can be added to the record book as another in a series of highly successful Fest Wests. Attendance exceeded expectations at over 250. Every one seemed to have a great time. Most major TI software and hardware dealers and authors were represented along with TI owners from across the country.

The Fest, which was hosted by the User Group of Orange County (UGOC) and Pomona User Group, was held at the Ramada Main-gate, just across the street from Disney land. Included in the Fest Guide was a map of all the attractions, restaurants and other facilities that were within walking distance. This was very helpful to visitors from out-of-town.

The Fest honored the tenth anniversary of the TI with banners, balloons and a special retrospect by Bill Gaskill.

Software and Hardware Dealers

There were representatives from the 9640 News, Asgard, Bill Gaskill, Comrodine, Bud Mills Services, Genial Computerware, JP Software, Ken Hamai Hardware, Notung Software, MS Express, LA Marketplace, TJ Software, Pomona User Group, Rave 99, TI Tax, Regina, Southwest 99ers, TAPE, Tex-Comp, and UGOC.

There was a wealth of items to purchase and many happy 4A owners walked out with new merchandise or with something from the overflowing consignment table.

Major Winners

There were three types of drawings at Fest West 91. Hourly drawings included items kindly contributed by the dealers present. The winners were too numerous to name. Two major winners, however, deserve special mention:

- o Ted Whomsley won the free night at the Ramada Maingate.
- o Mary Phillips from the Ozarks User Group in Missouri won a fully assembled and tested Horizon RAMdisk, that the Fest West committee purchased from Bud Mills Services. Mary was overjoyed at her good fortune.
- o H. R. Jeffery won the door prize; an Asgard Mouse.

The Best of TI

To honor the tenth anniversary of the TI, everyone who came was asked to vote for the "Best of TI". Ballots were collected on noon Sunday and the winners announced at the Fest. They were:

- o **HARDWARE:** Bud Mills Services and RAVE 99 tied as the best sources of hardware.
- o **PUBLICATION:** Far and away, the clear winner of the best TI publication was MICROpendium.
- o **WRITER:** Regina was picked as the best TI writer of all time. Honorable mention went to Barry Traver and Beery Miller.
- o **SOFTWARE:** There was no winner in the software category because so many fine items were mentioned.

During the Fest, Club 99 of Covina, California presented Jerry Price of Tex-Comp with a plaque to recognize his service to the TI community over the years. Jerry was surprised and touched.

Speakers

Many fine luminaries in the TI community spoke at the Fest. They included:

- o Ken Hamai on Disk Drives
- o Berry Miller on 9640 Programming
- o Ken Gilliland on new items from Notung Software
- o Bud Mills on RAMdisks and new offerings from Bud Mills Services and OPA
- o Regina on Programming in BASIC
- o Bill Gaskill on TI Base

BASIC MISCELLANY #1

By Earl Raguse

One of the things I learned while reworking TIPS 1.6, to 1.6/ER and writing TIPSLABEL was that TI didn't tell us in the XB Manual, or the later addendums, all we should know about XB. At least in what I could find. You may recall in my article on TIPS 1.6, that I found that after I had converted TIPS to using CALL KEY(3,K,S), DISPLAY AT, and ACCEPT AT, I could not enter lower case in ACCEPT AT.

I had some recollection that I had done it once upon a time, but I was not sure about it. Then I remembered that XB does not have a command to restore the lower case character set once they have been redefined. CHARSET does not do it, it only restores the uppercase set. That presumably was because early XB did not have a lowercase set. I then reasoned that since that was true, it made sense that ACCEPT AT would only take uppercase.

I had plans for writing an assembly routine to LINK that would do it. I had once written an assembly program to take keyboard text input, and further I knew that Adrian Robinson had written in the ROM, a very detailed ACCEPT AT routine in assembly. My problem was that I didn't know how to get into Irwin Hott's LOADER program for TIPS. That is where there assembly routines are hidden, submerged below the XB.

How wrong I was! I did not know until I got a call from Adrian (Robbie) Robinson, that the problem was not with ACCEPT AT, but the fact that I had used CALL KEY(3,K,S) to insure that all entries to CALL KEY would be upper case, instead of running them all through Ron Wolcott's assembly routine for converting inputs to upper case. I

didn't recall where I learned that CALL KEY(3) did that, surely not in the XB manual, but I knew it. It turns out, it was the Users Reference Guide.

What I didn't know was that once you do a CALL KEY(3,x,y) all, and I mean ALL, keyboard input thereafter, for CALL KEY, INPUT, LINPUT, ACCEPT AT, ect, etc, is restricted to upper case. I had used that fact for CALL KEY in my DIRECTORY programs. I didn't know that it stayed that way until you returned to the Title Screen. Or that you must do a CALL KEY(5,x,y), to restore normal upper and lower case, before any statement that calls for keyboard input. It matters not what x & y are so long as they are legal numeric variables. Lower case character redefinition has nothing to do with this. That is another story, later alligator, where again Robbie used his assembly knowledge to help me out of an of an XB problem with CHARSET.

After that phone call, I searched everything I had on XB, to no avail, I could find nothing to tell me this. The best source on the keyboard is the User's Reference Guide (the "Green Book"), but it does not even imply that it works that way. About two days later, I got a letter from Australia, from the Hunter Valley Assembly Guru, Ross Mudie, telling me the same thing Robbie did. I then got suspicious, why are the only people who know this the Assembly guys. I then scoured the TI Editor Assembler Manual. Firstly, I found a reference to the User's Reference Guide. There was however, a discussion, see page 250, about the fact that the keyboard "device" was selected by placing a number, they discuss only numbers 0-3, into >8374. (Hex numbers are indicated by preceding with > as in >8374). Now this discussion makes no reference to CALL KEY, it is generic, and therefore refers to all keyboard

input. Also, once a number is loaded into location >8374, it stays there until changed. I can now assume that the XB CALL KEY does among other things, a CALL LOAD of the key number into >8374 which requires a new CALL KEY or CALL LOAD statement to change >8374 to a new number. I have tried to test this theory in XB, but to no avail, Robbie says it works, but it won't for me. If I were working in assembly this would be rather understandable, but to the average XB Manual reader, TI left it quite totally unexplained. I wrote a very interesting program called LOAD/PEEK to test CALL LOAD and CALL PEEK. Next time I may publish it, you could use it to learn a lot about how this TI computer works.

So what does this all mean? If you wish XB to return upper case only, do a CALL KEY(3,x,y); to restore lower case, do a CALL KEY(5,x,y); and to keep the previous state (ie don't disturb the keyboard device previously selected) use CALL KEY(0,x,y). I note that most XB programmers use CALL KEY(0,x,y) almost exclusively. They are then not taking advantage of the computer's (and XB's) capabilities. I hope after this you will.

In the following months I plan to write about some of the other things I have learned about XB in recent days. Also including the above program LOAD/PEEK. One thing I have looked into is error trapping, and some things I have thus learned are not documented in the XB Manual. I have also learned some other helpful things which I will get around to talking about. I will have something to say about the power of ACCEPT AT, that even my High Priced Spread computer does not have. Until next time, may your 99/4A's never do a hang-up in 1991.

TRAGIC MISCELLANY #2

By Earl Raguse

My son Richard had been staying at my house temporarily while his family is getting settled in Oregon. He had decided to move up there, but here is where he made his money. He is an IBM clone user, (mine while he was here) and is a fan of Microsoft Quick Basic. It is very much like (actually compatible) IBM Basica or GW Basic, except it can be compiled into machine language when you are finished. That, of course, makes it very fast. It can be run as an interpretive language via its editor while you are writing and testing it. Very slick, and I would like it except I am addicted to the 99/4A.

I have done some programming in both Basica and GW Basic. also Radio Shack Basic, and UGH! Comodore Basic, and each have something one can say for them in some way, line and circle drawing, for instance. However, TI XB has it all over them, and any other Basic I have ever used or heard of, when it comes to text handling. We have DISPLAY AT and ACCEPT AT that allows us to display and/or accept, or edit with a full editor, what we display on the screen at a specified location. Very powerful, and very friendly.

The best Quick Basic can do is use LOCATE, and PRINT to put a message on the screen at a specific location. Then it is possible to read it, using some kind of a loop, with X=SCREEN(R,C,Flag), which is the equivalent of TI XB's GCHAR(R,C,X), except that 'f Flag is true, SCREEN reads the character's color instead of its ASCII value.

Now the difficulty here is that there is no way to edit the text you read from the screen, short of writing a very complex program, I am not even sure how I would go about it. What programmers

normally do is make you completely re-enter text that needs to be edited. Now, I know you would just love it, if FW were to put up a default file name, and all you wanted was to change the last character, or add a 1 or something, and you had to completely re-type it all over. You would if you were using some of the bigger and BETTER? computers.

OK, down off my soap box. Admittedly, one of the things I always did like about Basica, or GW Basic, was that you didn't have to write all commands in uppercase. IBM understands in lower case also. When I was typing lower case text with DISPLAY AT statements, I usually forgot to go back to UPPER case before typing the next DISPLAY AT. I was always looking at the keyboard, not the screen, I never learned touch typing. When I discovered what I was doing, I would say a few swear words and

Page 8

type it over, in upper case. I used to think about how nice it was that it didn't matter in IBM Basica.

Then I found out that it doesn't matter on the TI 99/4A either. Did you know that? Somehow I didn't. I knew the the 99/4A understood certain commands in either case, like RUN and LIST and RES.

I have now found out that unless you use the REM (!) or quotes (") in front of statements, or if it is a legitimate variable name, XB will convert all lower case to upper case. If you enter a statment in lower case, and then PCTN 8, REDO, it will still be lower case, but if you enter the line number and PCTN X or LIST it you will find that it is changed to upper case. Very nice, it saves me a lot of trouble if I just stay in lower case, and use the Shift key when I want a capital letters in text, or a variable name.

One of the other things that tends to

make programming in Basica easy is the fact that the function keys are, or can be, programmed to give you commands, like PRINT by just pressing one function key. Very Nice, but the 99/4A does even better, there is a key stroke (actually a two key stroke) for most the XB commands. This is because everything is stored as tokens, one for each key stroke code. If you press (CTRL ;) after a line number, while in the programming mode, you will not see anything but a space, but if you list the line, you will see PRINT, big as life. I will give you a few of the codes, just to titillate you, but the complete list has been printed in the newsletters several times. Just come to the Library, and I will find it for you.

TOKEN CODE SAMPLE

Command	Key Stroke(w/CTRL)
PRINT	; (Semi-easy)
IF	D (Dead)
THEN	O (On)
ELSE	A (Arrival)
GOTO	F (F)
INPUT	R r
RANDOMIZE	U u
DEF	I i
RESTORE	T t?)

I was told there there was only one catch. you must do CALL INIT first before this works. Well, I have found that even that is not required. I checked, I did a simple program, without CALL INIT, like this. I first did NUM 100 to get automatic line numbers, then did ! A CTRL A, after the 100 etc, on through the alphabet, numbers and punctuation. The Shift key doesn't count. Then I listed it, what do you suppose I got? The whole token list. I will print it next time. Its been fun, see you next time.

- o Bill Chavanne on Multiplan and TI-Tax
- o Barry Traver on Programming
- o John McDevitt on new items from RAVE 99
- o Rodger Merritt on graphics and new items from Comprodine

The speaker sessions were well attended and received.

Many Thank You's

There are so many people and organizations that helped make Fest West 91 successful that the list could go on and on. A few, however, deserve special mention:

- o The Riverside User Group (RUG), Southern California Computer Group (SCCG) and the Pomona User Group, all of California, helped some TI notables attend by partially defraying their expenses.
- o All of the dealers who kindly contributed merchandise and discounts for the drawings.
- o MICROpendium sent 100 catalogs for distribution to those present.
- o Southwest 99ers ran the registration process.
- o Southern California Computer Group provided a major assistance in running the consignment table.
- o Cris Van Allen created the giant Fest poster, made the vendor banners, designed the official Fest West 91 Tee Shirt, was our official photographer and much more.
- o TM Direct, the new TI vendor, sent us catalogs and items for the drawings.
- o Special thanks also to Gloria Anders, Stan Corbin, George Dearmin, Eugene Gibson, Daniel Hatheway, Steve Luest, Howard McDonald, Erwin Metz, Bill Mooney, Earl Raguse, Janice Shafer, Shirley Swedlow, and everyone else who helped out.

FEST WEST 91 Committee

A recap of Fest West 91 cannot be complete without mention of the Organizing Committee who spent a year bringing this event from concept to reality.

- o Siles Bazerman coordinated the Speakers and the Friday night get together.

- o Gene Bohot took care of promos, printing and keeping us on track.
- o Bill Harms was in charge of user group relations, the tenth anniversary celebration and running the front table.
- o Bill Nelson hosted uncounted meetings coordinated with the hotel, and made the outstanding graphics.
- o Jerry Rash served as treasurer and round volunteers and loaner systems.
- o Jim Swedlow coordinated with vendors, served as secretary and announced all of the drawings.

Truly, Fest West 91 lived up to its slogan as the HAPPIEST FEST ON EARTH!

Enjoy

BASICALLY BASIC*

Double sized letters can be defined in TI, plain vanilla, console basic --- but it takes a lot of hard work. You have to find how to make the letters, (read pages 11-76, -79 in the User's Guide for starters) and redefine each one into four characters. (Space requirements expand geometrically.) For instance:

PI* (pattern identifier) for B is "0078242438242478". To represent the letter, pixels in the 64-dot character matrix are turned on (H) and off (.) as shown below.

To make the letter double size, the 8x8 matrix is divided into quadrants.

And each quadrant is redefined into a full sized character. Notice that for every pixel that was turned on in the quadrant, four are turned on in the character.

Then, the characters are placed in the proper order on the screen with a PRINT, CALL HCHAR, or CALL VCHAR statement.

And you have to do this for each letter you want to use; a lot of pencil work.

But in XB, using sprites with CALL MAGNIFY(2) and the proper ASCII code, you can get double sized letters -- no problem, unless you want to use five or more on the same line. Then the higher numbered sprites disappear. Hence, back to the designing table.

However, XB also has the subprogram CHARPAT which returns the pattern identifier as a string variable. And