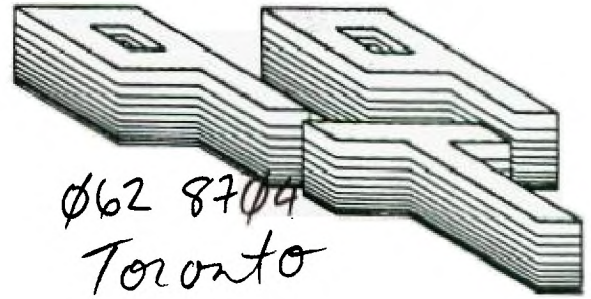


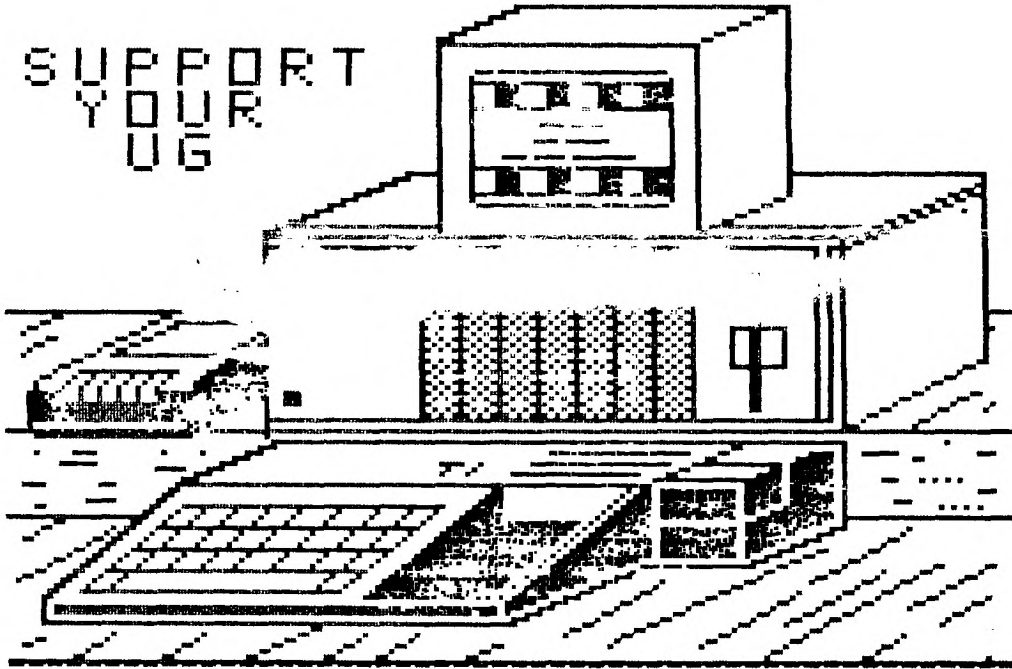
NEWSLETTER 9T9

Texas Instruments
USERS GROUP
TORONTO

FOR THE TI-99/4A COMPUTER



SUPPORT
YOUR
UG



APRIL 1987 ISSUE

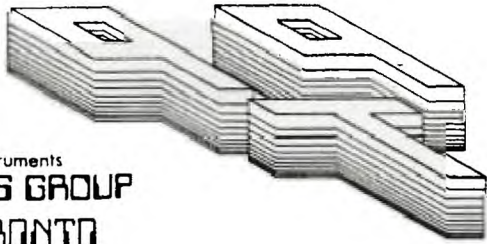
FROM:
9T9 USERS GROUP
#109-2356 GERRARD ST. EAST
TORONTO, ONTARIO, M4E-2E2
CANADA

POST
SEP 7
M4L 3T0



TO:

Edmonton UG
Box 11983
Edmonton, AL T5J 3L1



Texas Instruments
USERS GROUP
TORONTO

FOR THE TI-99/4A COMPUTER

EXECUTIVE COMMITTEE

PRESIDENT Steve Mickelson (1494)
VICE-PRESIDENT Allan Allen (8600)
TREASURER Bob Rand, Rossetto (469-3468)
SECRETARY Peter Sandford (820-7389)
CLERK AT LARGE Tom Brown (793-3761)
LIBRARY DIRECTORS Cary Bowser (960-0925)
Andy Parkinson (275-4427)
Cecil Chin (671-2052)

PAST-PRESIDENTS Lloyd Lindsay (743-3868)
Emile Verkerk (633-1451)

NEWSLETTER EDITOR

Steve Mickelson (657-1494)
Associate Editor - Blair MacLeod

MEMBERSHIP FEES

FULL MEMBERSHIP \$30.00 / year
NEWSLETTER SUBSCRIPTION \$20.00 / year

All memberships are household memberships. A newsletter subscription is only for those who do not wish to attend meetings but wish to receive our newsletter and have access to our library. You are welcome to visit one of our regular meetings before joining the group. If you wish more information contact our president in writing at the club address on the front cover or call and leave a message with an answering machine.



Get the Message.

Submitted by Peter Sandford, Treasurer

9T9 Users Treasurer's Report

As of March 30, 1987:

Expenses \$901.78

Income \$749.00

Balance \$1273.17

DRAW*DRAW*DRAW*DRAW

As noted in the Tidbits column, Bruce Ryan of Ryte Data will demonstrate Mechatronic and Atronic hardware for the TI-99/4A. We are pleased to announce that there will be a draw at the April meeting. The prize will be a full year's subscription to the R/D Computing newsletter. The draw will be made by Bruce, who will present the winner with his prize. Tickets will sell for \$0.50 each or three tickets for \$1.00. Proceeds will go to the club's treasury.



"LET ME BEGIN BY SAYING THE COMPUTER WAS OLD AND PROBABLY DUE TO BE REPLACED ANYWAY"



Get the Message.

2



H. MARTIN IN SATURDAY REVIEW

COMMERCIAL ADVERTISING

Any business wishing to reach our membership may advertise in our newsletter. The rates are as follows:
(width by height)
PAGE (9" x 6") \$1.00
PAGE (4 1/2" x 6") \$1.00
PAGE (4 1/2" x 3") \$1.00
Please have your ads camera ready and paid for in advance. For more information contact the editor.

Don't forget, that any member wishing to place ads, may do so free of charge as long as they are not involved in a commercial enterprise.

NEWSLETTER ARTICLES

Members are encouraged to contribute to the newsletter in the form of articles, mini-programs, helpful tips, jokes, cartoons and questions. Any article may be submitted in any form by mail or modem. We welcome the reprinting of any article appearing in this newsletter providing credit is given to the author and 9T9. If more information is required, call the editor. The names 9T9, Nine-T-Nine, Newsletter 9T9, 9T9 Users Group and Nine-T-Nine Users Group are Copyright, (c), 1987 by the 9T9 Users Group of Toronto, Canada, all rights reserved.

DISCLAIMER

Opinions expressed in this newsletter are those of the writers and are not necessarily those of the 9T9 USERS' GROUP. 9T9 cannot assume liability for errors or omissions in articles, programs or advertisements. Any hardware modification or project is presented for informational purposes, and the author, newsletter staff, and/or 9T9 Users Group cannot be held liable for damage to the reader's equipment. All such projects are done at your own risk!

NEXT MEETING

The meetings are held on the last Tuesday of each month. The next will be held at the York Woods Public Library in Willowdale, starting at 7:30 pm. The library is at 1705 Finch Ave. W just west of Keele St. The entrance to the library is on Finch Avenue. See meeting schedule, below

April 28
May 26
June 30
July 28



"Look at me when I'm programming you!"
HIGHEST IN BOYF LIFE

If you missed the last meeting, besides missing travelling in one of the winter's worst snowstorms, (actually folks it wasn't really that bad!), you missed a couple of very interesting demonstrations. Both were scheduled in mid-month and therefore not able to get included in Newsletter 9T9 for publication prior to the March meeting. I think they were interesting enough to be reported here.

The first demo, Bernie Roache and another member of his amateur radio club running Bernie's TI system, with Corcomp microexpansion system interfaced to a Packet Radio transceiver, (that's radio transmitter/receiver, using VHF frequencies).

The system, using Fast term terminal emulator software, was able to patch-in to a radio repeater at Uxbridge, approximately 30 miles northwest of the meeting's location. The signal was then "repeated" to a receiver in Mississauga, about 15 miles southwest of the same location. This second receiver then "patched" the signal to the phone lines. The rest was like operating via modem.

The interesting idea about this, is that you could link, through a series of repeaters, to phone patches half a continent away. As this is amateur radio, an operator is limited to non-commercial applications and a license is required. For your license no code is required; though a knowledge of packet radio theory and protocol is needed. A system runs about \$1000 Canadian, but offers an interesting alternative to Datapack, toll calls, and P.C. Pursuit.

Packet radio is a relatively new branch of amateur radio, though seems to be spreading quickly. The set-up demonstrated allowed both voice and data transmissions. The heavy steel superstructure of the library, above our basement meeting room hindered "patching-in" to a repater in Barrie, (60 miles north), but I was told that a regular external antennae would have no problem.

Bernie was involved in a Red Cross exercise, where he had two separate TI systems patched-in to one packet radio transceiver via a special splitter cable. The first system transcribed all terminal communications to disk; while the second spooled the same to a printer.

Perhaps we could ask Bernie to give us another demo at a meeting where the weather is more cooperative.

Demonstration #2, consisted of Gil Tennant patching-in the club's system to acoustic TI modem, (loaned by Bernie), to Gil's FIDO Net BBS. Initially we had trouble getting an outside line on a poorly functioning phone. Eventually, with the help of Neil we got the dial tone.

This demo was helpful to both the newly initiated^{to} the modem, but helped users with the FIDO's rather unique, (to Tiers), command set. Since Gil's board is dedicated only for TI uploads and downloads, albeit on an IBM clone, the demo was enlightening.

Thanks to both Bernie and Gil for sharing with us these interesting applications of the TI. Any member who has an application of the TI which they would like to share with others feel free to give me or any other member of the executive a call.

OHMS? Answer:

I received a call from Jiri, who told me that he calculated the answer to the 3-D resistor puzzle which appeared on page 14 of March Newsletter 9T9. He said it was just a matter getting the network into a configuration of three parallel resistor-strings. Jiri's calculations produced an answer of 5/6 of an ohm.

Corrections:

The article on adding 8K of RAM and battery back-up to Edito/Assembler or the poor man's superspace appeared in the Ottawa Users' Newsletter. The EPR(M) needed for the poor man's DD disk controller, (software downloaded from DELPHI), are available from Bruce Ryan at RYTE Data, 210 Mountain St., Haliburton, Ont., KOM-1S0, (705) 457-2774.

Speaking of Ottawa, please bring your cheques for DM-1000 Fairware/Triware, if haven't donated yet payable to the Ottawa users group, as this will be the last meeting we will have before the Ottawa TI-Fest, May 16. Please include your name and address so that you can be "registered" as a users to this software. Andy, Randy or I will be glad to pass-on your donations. Support the authors of your software!

Our club will have a booth set-up at Ottawa. We plan to pay for it through sales of back issues of Newsletter 9T9 and of the disk(s) of the month(s), from our library.

Contact Randy, if you would like to go to the TI-Fest and/or the dinner afterwards, (\$10 per head), as he has a application to be mailed out. Discounts on motel accomodations are also available.



This meeting we will have Bruce Ryan as guest. Bruce will bring some of the various hardware and software he markets for the TI. Bruce stocks various books, software utilities and hardware items. Such items as the mouse, Gramcard, one megabyte memory expansion, multifunction cards and so forth. I'm not sure which of the items he'll show us, but it should be interesting to see this hardware, hopefully in action.

We are still working on having Clint Pulley in for a demo of the Geneve.

Expect to see 9T9 Hats in the near future. The artwork was passed on to Randy and soon we will have hats with the clubs LOGO, just in time for summer.

You will see that Gary has a trick or two up his sleeve, in addition to the disk of the month. Check-out his column.

A quarterly statement of our bank balance has been submitted by our treasurer.

I hope to bring camera in-hand to Ottawa and, hopefully, a future issue of the newsletter will give those who can't make it, an idea of what's going on.

Well that's it for April. Hope to see you at the next meeting.

LIBRARY NEWS BITS

-by Gary Bowser

Well, this month we added over 10 disks to the library and some newsletters and a book. Starting this month I am publishing a 'Trick of the month'.

TRICK OF THE MONTH:

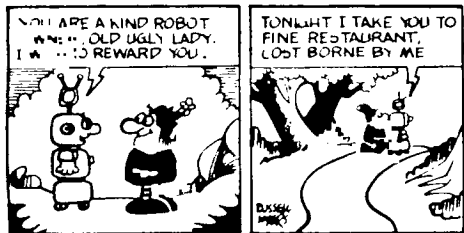
Type this line in Extended BASIC command mode, then press enter; when the cursor starts flashing press 'CLEAR' (fctn 4). The sprites will continue to move but you will still be able to load, edit, save and list a program. The trick is in the ACCEPT A statement. I will tell you why it works next month.

```
CALL MAGNIFY(4)::FOR X=1 TO 24::CALL SPRITE(#X,X*4+28,2,X*2,X*4,24-X,X)::NEXT X
::ACCEPT A
```

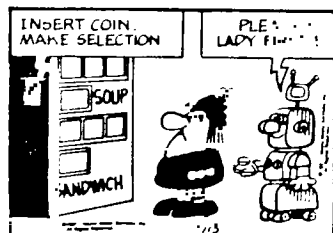
9T9 TrIware and Public Domain Software

PROGRAM	SIZE	RUNS	DISK#	DISCRIPTION
4thMUSIC	Pub - 1	EA/LDR	9	Music or Graphics Demo
APECLOCK	Pub - 1	EA	32	Clock
ARTIST01	Pub - 2	RLE	51	More TI-ARTIST pictures for MAX/RLE
AXLE	Pub - 1	EA/XB	24	Music (Beverly Hills Cop)
E/A WRITER	TrI - 1	XB	38	TI Writer Disk Version
ERAKS	TrI - 1	XB	36	Editor Assembler Disk Version
CBASIC	TrI \$20 2	EA	14	cBasic Language Compiler
CA:FNDAFS	TrI - 1	XB	48	A set of different calender programs
CH:K/BOOK	? - 1	XB	33	Cheque Book and Budget Management
CH:WACHESS	TrI - 1	XB	49	A new game
CRAPS	? - 1	XB	26	Crap Game Requires 32K
CRUNCH	TrI - 1	XB	41	Crunches Dis/Var 80 files
DIM	Pub - 1	EA	18	Disk information managemme
DIFIX	TrI \$10 1	EA	43	Disk fixer type program with docs
DM1000	TrI - 1	EA/XB	42	Disk Manager 1000 from Ottawa
DRAWPIO	Pub - 1	EA	3	Computer assisted drawing
FAST-TRANS	TrI - 2	XB	44	Checkbook Recapper/planner
FASTTERM	TrI \$15 1	EA/XB	17	Terminal Emulator program
FORTHDOC	Pub - 5	DV80	11, 12, 13	Forth Manual on disk
FRACTAL	TrI \$10 1	EA	16	Fractal explorer
FRENZY	Pub - 1	XB	24	Space game very well done
FREEWARE	Pub - 1	DV80	50	A freeware catalog
FUNLWRTR	TrI \$10 2	EA/XB	1	TI Writer and E/A systems disk
GRPHJACKET	TrI - 1	XB	40	Prints a disk jacket using a GRAPHX file
JETSPRITE	TrI \$ 5 2	XB	20	Sprite builder program
JPGGRAPHIC	Pub - 2	EA/LDR	8	Forth graphics and demo
LABELER	TrI - 1	XB	45	Print Labels with over 100 logos
MAX/RLE	Pub - 1	EA	2	Display special graphics, with some
MONA	Pub - 1	XB	23	Prints Mona-Lisa
MONOPOLY	TrI - 1	XB	46	A better monopoly on computer
MS-ADVEN	TrI \$10 1	XB	30	Adventure game
OIL INVAS	Pub - 1	EA/LDR	10	Forth Game
CMVI LDR	TrI \$10 1	EA	19	E/A program loader
PRINTFFC	Pub - 1	EA	23	Selection of printer art
FRANZ	TrI \$10 2	EA/XB	5	A good Data Base Program
RAPID	Pub - 1	EA/XB	19	Rapid scroll for DV80 file

BROOM-HILDA



4



(from DELPHI)

A PREVIEW OF 9640 ADVANCED BASIC

Copyright (c) 1987 by Walter Howe.

This article may be freely placed on bulletin boards and information services. It may be copied in its entirety in user group newsletters provided that a copy of the newsletter is mailed to: Walter Howe, 43 S. Chelmsford Rd., Westford, MA 01886. All other rights are reserved.

This is a preview, not a review. My comments are based solely on the draft manual released by MYARC ahead of release of the 9640 computer. It is quite possible that the final version will contain differences from this.)

The new MYARC 9640 brings us a new version of Extended BASIC, which MYARC is calling Advanced BASIC. It is an outgrowth of the previously released Extended BASIC II for the 4A (or perhaps XB II is an ingrowth of Advanced BASIC - not sure which really came first).

XB II gave us substantial new graphics capabilities, integer math, and a lot more speed. The speed increase was gained by writing XB II in assembly language, whereas TI's XB was written in TI's Graphics Programming Language (GPL). GPL is very economical in its demands on memory, but leads to a slow XB, because every program step had to be doubly interpreted - first into GPL, and then into machine code. By writing XB II in assembly, the XB interpreter goes directly to machine code.

Advanced BASIC gains still more speed - partly because the new 9995 processor operates more efficiently and partly because the new 9938 graphics chip runs at blinding speed as a co-processor. In addition to the increased speed, it adds more usable memory, 80-column capability, double-precision arithmetic, better program editing and debugging support, WHILE/WEND, more efficient ways to do things, and a lot more as well. Where XB II added about 20 new commands and functions, Advanced BASIC adds over 30 more commands and functions than XB II contains. Here is a quick look at the new features:

BCOLOR and BTIME are used with ACCEPT and DISPLAY to provide BLINKing text. ACCEPT and DISPLAY appear to have multi-line capability now.

BEEP is now a command as well as a modifier.

DATE, TIME, DATE\$, and TIME\$ allow you to set and read information from the new clock chip.

DEF has been expanded to include single and double precision variable types. CINT, CREAL, CDBL, and CSNG can be used to Convert variable types.

CALL FILES will print the current disk catalog.

FREESPACE will tell you how much space you have available as SIZE used to. It may also be used to force garbage collection wherever you want it. (This means that the computer gathers all no longer used memory space and makes it available again.)

HEX\$ will return a hex value for a normal base 10 number. MOD returns a modulo arithmetic value (essentially a remainder) for a number to any given base.

DELETE will now delete named lines from a program. KILL is now used where DELETE used to be used to delete a file.

OUT and INP will send and receive one or more bytes to a specified port.

LPT sets a default printer device. PPR prints the current default device. LLIST lists to the default device. LTRACE traces executed line numbers to the default device instead of the screen. LCOPY is also used, but its description was missing from the draft manual. Your guess is as good as mine.

LEFT\$, RIGHT\$, and (perhaps) MID\$ give easier partial strings than SEG\$ did.

LINE appears but its description was also missing.

LPR returns the Last Point Referenced in a graphics command.

MEMSET allows you to set all variables in an array to the same value.

MOUSE includes a series of commands providing mouse support to graphics.

SPEED lets you set 4 different speed setting for the computer while running Advanced BASIC. This is necessary for many 4A programs where the faster speeds of the 9640 would interfere with the program unless slowed down.

SWAP lets you swap the values of two variables of the same type.

WHILE/WEND allows you to set up a loop that will continue to execute as long as (while) a given logical condition exists, for example, as long as a file record is not named LAST.



Several comments are worth noting:

Although the 9938 chip is capable of 512 colors, it appears that there is no support from Advanced BASIC alone for anything but the original 16 colors of the 4A. Perhaps it can be achieved through assembly links, though, and if that is the case, someone will no doubt write routines to access other colors and make them available.

Advanced BASIC has been written to permit running older programs from both TI XB and XB II on the 9640 as much as possible. Even where commands have changed names, it will normally recognize the old commands.

With the new features of Advanced BASIC, the larger memory available (64k for programs, which does not include the additional memory used for variable space, I believe), and the much greater speed of the 9640, many more programs will be feasible to write in BASIC than before. Routines that were too slow under the earlier versions may be quite acceptable now.

Some bugs are likely to be found after release, as is the case with any new product. MYARC hasn't been able to avoid these any more than any other computer developer has. It shouldn't have as many growing pains as XB II did, partly because many of the bugs have been worked out in XB II and partly because there are not the severe memory limitations that were always hindering XB II. MYARC has always been good about standing behind their products and fixing or replacing them at little or no cost to the buyer. I don't expect MYARC's handling of the 9640 to be any different.

The following was downloaded from Compuserve:

LOGO TIMES
by Larry Spohn

Over the course of the next year, I'm going to try to write a monthly column on TI Logo. Why bother you might reasonably ask? Isn't it true, that our user group virtually never talks about LOGO? There are no LOGO programs in our library! And other user groups seem equally content to ignore the language, with little if any discussion about it in their heady newsletters.

Indeed, you might ask, whatever happened to what Texas Instruments described as "a limitless learning environment." Noting in early promotional literature that the TI-99/4A was one of only two home computers to offer LOGO, the company also described it as "a very special child-appropriate computer language." All of which probably made many of us parents wise enough (or intimidated enough) to buy LOGO for our kids, but not secure enough to venture beyond booting up the canned procedures on the sampler disk. Why? Perhaps because like Trix, LOGO is for kids. Right? Wrong, I'd like to argue. It may not generate gee-whiz utilities or spreadsheets four arms long, but LOGO is being used outside the elementary classroom, for instance, by college mathematics, physics and computer science students. Indeed, TI LOGO Guru Harold Abelson says as a basis for exploring mathematics, LOGO's turtle geometry is useful in presenting "mathematical topics ranging from elementary geometry through (Einstein's theory of) General Relativity." To say the least, LOGO is a mind stretcher. And we could all use a lil' bit of that. If there is one thing that has been painfully obvious at our club meetings, it is that most of us are not programmers or have only self-taught rudimentary programming skills. There are several notable exceptions, of course, and for their expertise we should all be grateful. Still, admit it: most of us couldn't pass a basic test in BASIC. We know enough to function, tinker and be astonished by the programming wizardry of others. If you aren't smiling, just try to recall the last time you tried to do anything really imaginative or original with any computer language.

Enter LOGO. Why again? Maybe cause its there. We ought to try to conquer it, or at least experience it. But don't be intimidated by Abelson's high-minded opportunities. General Relativity ain't everybody's cup of tea. For the less adventurous, how about because its much easier to learn LOGO than probably any other language and it seems to me to be the best foundation language for beginners. It appears likewise to be the most easily mastered, though LOGO has a way of bending back on you, taking you through turns you never knew were there. Incidentally, let me make REAL clear right now that like you, I am very far from being a master and very much the beginner. Short of reading the manual, experiencing the sampler programs and about two hours worth of tinkering during the last month, I don't know diddly about LOGO. Hardly know a turtle from a sprite. So, anybody out there with LOGO blackbelt, feel free to jump into the fray. Otherwise let's see if we can't learn together from little LOGO.

Actually, I think that's one of LOGO's greatest assets. It lends itself very well to team programming. And as such, is an ideal family challenge. It goes hand in hand with LOGO's interactive nature. My very first LOGO session was exhilarating because I was quickly doing things I thought would take weeks and endless sorting through the manual. Abelson is correct when he says LOGO gives control of the computer in a personal, "self-directed" way. In this sense LOGO is not just a computer language, but a "philosophy of education" and a powerful one at that. He says it gives people the power to "establish intimate contact with profound ideas from science, from mathematics, and from the art of intellectual model building." In a way that even the most sophisticated computer

adventure game can't match, LOGO turns you into an explorer. And let's face it. Exploring can be a riot.

That brings us to Micheal Crichton's (author of the "Terminal Man," "The Great Train Robbery," and "The Andromeda Strain") famous seven maxims about computers, all of which apply to LOGO and how we see it:

"People are more important than computers. Much of what we believe about computers is wrong. It is easy to use a computer. This is fortunate, because everybody's going to have to learn. It is not so easy to use a computer wisely. This is unfortunate, because everybody's going to have to learn. Computers can actually be a lot of fun. There are people who want to put a stop to that."

Believe me, LOGO can be a lot of fun. (This article was written for the March issue of the newsletter of the Guilford 99er Users Group, Greensboro, NC.)

[73557,3447] LOGO.TXT 04-Mar-87 6199

HORIZON RAMDISK 256K EXPANSION PROJECT
by EDWARD A. HALLETT
SOUTHWEST NINETY-NINERS

Edited by Kevin Kapus, Front Range 99ers, for use with HORIZON SOURCE CODE VER_04

The HORIZON RAMDISK is available in 90K SSSD (360 SECTOR) and 180K DSSD (720 SECTOR) sizes. This project expands the size to 256K (976 SECTORS) for an increase in storage capacity of 64K (256 SECTORS) or 35.5%.

This increase is accomplished by adding one 74LS154 (4 to 16 DECODER), one 74LS02 (NOR GATE), and eight 8k 6264LP-15 STATIC RAM chips, removing one 74LS138 (3 to 8 DECODER) chip, and modifying the DSR CODE to recognize the existence of the added memory. The original HORIZON RAMDISK CIRCUIT does not fully decode one of the five memory address lines from U9 limiting it to 180 K. By fully decoding this line we pick up eight more CHIP SELECT SIGNALS bringing us up to 256K (976 SECTORS). This utilizes the original design to its fullest potential with only a few SIMPLE MODIFICATIONS.

CAUTION: THIS MODIFICATION IS UNDERTAKEN AT YOUR OWN RISK AND MAY VOID YOUR HORIZON WARRANTY.

CAUTION: REMOVE THE NICAD BATTERIES FROM THE RAMDISK BEFORE STARTING. USE CARE WHEN HANDLING THE RAM CHIPS TO AVOID DAMAGE FROM STATIC.

1. Remove U1, the original 3 to 8 DECODER CHIP, from its socket and DISCARD.
2. Remove the EIGHT PIGGYBACKED PAIRS of 8K RAM CHIPS from their sockets U3-U6 and U12-U16.
3. Remove U2, the original 4 to 16 DECODER, from its socket.
4. Remove U10, the original NOR GATE, from its socket.
5. Install a THIRD ADDITIONAL 8K RAM CHIP PIGGYBACKED on top of the removed PIGGYBACKED PAIR of 8K RAM CHIPS connecting EACH PIN to its CORRESPONDING PIN below with the EXCEPTION of PIN 20 (CHIP SELECT). BEND PIN 20 outward like PIN 20 on the CHIP below it. Reinstall these EIGHT PIGGYBACKED TRIOS into their sockets (U3-U6 and U12-U13) and RECONNECT the ORIGINAL lines from PIN 20 of the CENTER CHIPS to their ORIGINAL POINT on the EXPANSION JACK next to U3.
6. Install the ADDITIONAL 4 to 16 DECODER CHIP (74LS154) PIGGYBACKED on top of the ORIGINAL 4 to 16 DECODER CHIP, U2. Connect PIN 12 and PINS 20 THRU 24 to their corresponding PINS below. Bend PINS 1 THRU 11 and PINS 13 THRU 19 OUTWARD. Reinstall the PIGGYBACK PAIR of 4 TO 16 DECODERS in its U2 socket. Connect lines from the UPPER CHIP PINS 1 THRU 8 as follows.

PIN 1 TO U1 SOCKET PIN 15.
 PIN 2 TO U1 SOCKET PIN 14.
 PIN 3 TO U1 SOCKET PIN 13.
 PIN 4 TO U1 SOCKET PIN 12.
 PIN 5 TO U1 SOCKET PIN 11.
 PIN 6 TO U1 SOCKET PIN 10.
 PIN 7 TO U1 SOCKET PIN 9.

7

cont



These provide the CHIP SELECT SIGNALS to the ORIGINAL (CENTER LAYER) of 8K RAM CHIPS.

Connect lines from the UPPER CHIP PINS 9 THRU 11 and 13 THRU 17 as follows.

PIN 9 TO PIN 20 U3 TOP 8K CHIP.
PIN 10 TO PIN 20 U4 TOP 8K CHIP.
PIN 11 TO PIN 20 U5 TOP 8K CHIP.
PIN 13 TO PIN 20 U6 TOP 8K CHIP.
PIN 14 TO PIN 20 U12 TOP 8K CHIP.
PIN 15 TO PIN 20 U13 TOP 8K CHIP.
PIN 16 TO PIN 20 U14 TOP 8K CHIP.
PIN 17 TO PIN 20 U15 TOP 8K CHIP.

These provide the CHIP SELECT SIGNALS to the ADDITIONAL EIGHT 8K RAM CHIPS (TOP LAYER).

7. Install a new NOR GATE (74LS02) PIGGYBACKED on top of the ORIGINAL NOR GATE, U10. Connect PINS 2, 7, and 14 to the CORRESPONDING PINS below. LABEL PINS 1, 3 THRU 6, and 8 THRU 13 outward. Reinstall the PIGGYBACKED PAIR or NCA GATES in its U10 socket. Connect LINES from the UPPER CHIP as follows.

PIN 1 TO PINS 18 AND 19 U2 UPPER CHIP.
PIN 3 TO U1 SOCKET PIN 6.

These provide the CHIP SELECT SIGNAL for U2 UPPER 4 TO 16 DECODER CHIP thus fully decoding the available MEMORY ADDRESS LINES.

PINS 4 THRU 6 and PINS 8 THRU 13 of the UPPER NOR GATE U10 are not used and are left NOT connected. They may be used in future modifications.

This completes the HARDWARE modifications to the RAMDISK CARD. Next the DSR SOFTWARE must be modified so that this ADDITIONAL MEMORY can be accessed.

Software Modifications updated to Ver_04 by Kevin Kapus.

The original DSR CODE, CALL SUBPROGRAMS, ETC. are located in RACKS 90-92 at the top of the RAMDISK MEMORY MAP. The MODIFIED RAMDISK MEMORY MAP now extends to RACK 124 and the DSR must be moved to the new top, in RACKS 122-124.

NOTE: IF THE CODE IS NOT MOVED IT WILL BE ERASED WHEN THE RAMDISK IS INITIALIZED TO MORE THAN 720 SECTORS.

The changes to the CODE consist of changing ALL REFERENCES for the three upper 2K blocks of memory to a NEW LOCATION, changing the LOADER PROGRAMS to LOAD the NEW CODE at the NEW LOCATION, changing the MAX SECTOR CALL, and MAX SECTOR CALL, and modifying the FORMAT ROUTINE of the DSR.

Luckily, this is much EASIER than it might appear since the SOURCE CODE for the HORIZON RAMDISK was provided with the KIT and is very well Documented!

The following PROGRAMS will need to be modified and then REASSEMBLED with the EDITOR-ASSEMBLER. CALL/S, CREATE/S, LOADER/S, PARTA, SVXB/S, and XB/S.

1. CALL/S
Change "C1 R2,1441" to "C1 R2,977" at LABEL MAX02.
2. LOADER/S
Change "DATA>BB00" to "DATA>FB00" at LABEL LINK.
Change "BYTE>BB" to "BYTE>FB" at LABEL MXL1.
Change "BYTE>BD" to "BYTE>FD" at LABEL MXL2.
Change "BYTE>BF" to "BYTE>FF" at LABEL MXL3.
3. PARTA
Change "DATA 720" to "DATA 976" at LABEL MAXSEC.
Change "DATA 720" to "DATA 976" at LABEL FORSEC.
Change "DATA >B100" to "DATA FB00" at LABEL LINK1.
Change "DATA >B200" to "DATA FD00" at LABEL LINK2.
Change "DATA >BF00" to "DATA FF00" at LABEL LINK3.
Add the LINE "C R8 @MAXS:" and "JEQ FLINE" after the LINE "INC R8" (fourth LINE after LABEL FMTLP1.)
Add the LINE "FFDONE MOV R8,R3" after the LINE "JNE FMTLPO" (sixth LINE after LABEL FMTLP1.)
4. SVXB/S
Change "LI R1,>BF00" to "LI R1>FF00" (FOURTH LINE after LABEL SVXB.)
5. XB/S
Change "C1 R2,1441" to "C1 R2,977" at LABEL MAX02.



"It says, 'Leave me alone, I have a headache!'"

Reassemble CALL/S, LOADER/S, SVXB/S and XB/S to create NEW OBJECT FILES. ASSEMBLE the ORIGINAL FILES "CHAR/S" and "DOWNLD/S" from the HORIZON SOURCE DISK.

Type in the following program with the Editor Assembler.

COPY "I.1.1. PARTA"
COPY "I.1.1. PARTB"
COPY "I.1.1. PARTC"
COPY "I.1.1. PARTD"
COPY "DSK1. PARTE"

8

cont



Assemble this file, use DSR256 as the object file, with the R option.

Next RUN the "LOADER" program assembled from "LOADER/S" to LOAD the following:

"DSR256" into BLOCK 1.
"CALL" from the assembled FILE "CALL/S" into BLOCK 2.
"CHAR" from the assembled FILE "CHAR/S" into BLOCK 3.
"DOWNLD" from the assembled FILE "DOWNLD/S" into BLOCK 3.

Now RUN this BASIC program.

```
100 CALL INIT
110 CALL LOAD("P:1.XB")
120 CALL LOAD("P:1.SVXB")
130 CALL LINK("P:1.XB")
140 END
```

NOTE: The RAMDISK MUST be set CRU 1000 for the SVXB program to work as it does NOT search for the HORIZON CARD CRU like the other programs do. If you have another CARD at CRU 1000 (like the MYARC 128K or 512K CARD) you can change the sixth LINE of the "SVXB/S" FILE from the "LI R12,1000" to "LI R12,(CRU of your HORIZON CARD)".

The modified INT CODE, CALL SUBPROGRAMS ETC., are now LOADED in their NEW locations in RACE: 122 THRU 124.

This completes the DSR modifications. All functions of the HORIZON RAMDISK will function as they did originally but now being able to UTILIZE 976 SECTORS (256K).

Editors Note: The VER 04 MEMTEST I have will not work. Use Disk Manager II's comprehensive test to check your upgraded ramdisk.

When formatting the 976 SECTOR, RAMDISK will show "974 SECTORS FREE" and "466 SECTORS USED". This is because the DISKMANAGER is trying to format 1440 SECTORS and reads 466 USED during SECTOR VERIFICATION. This does not affect RAMDISK OPERATION in any way, but it can be corrected to show "974 SECTORS FREE" and "2 SECTORS USED" by changing BYTES 10 and 11 of SECTOR 0 from >05A0 to >03D0. The following program is used to correct the SECTORS FORMATTED number.

```
DEF START
SECTOR DATA>03D0
START LI R12,>1000 (CRU OF YOUR CARD)
LI R1,7
SWPB R1
LDCR R1,8
MOV @SECTOR,@>580A
JZ 0
RT
END START
```



This completes the HORIZON RAMDISK 256K EXPANSION PROJECT for VER_04.

If you have questions concerning this expansion project you can send them to Kevin Kapus, 1026 Norwood Ave., Colorado Springs CO 80906, or phone (303) 576-3199 or contact Edward A. Hallett, 5600 S. Countryclub #64, Tucson AZ 85706. Phone (602) 889-6930.

PLEASE NOTE: Randy Rossetto has completed this upgrade to his Horizon Ramdisk. Randy mentioned some incorrect bin and chip designations in the upgrade procedure. This article has come from another source and appears not to have those errors. Since I don't have the RAMdisk, I cannot say for certain. I suggest that you contact Randy, if you decide to do this project and he could advise you as to how correct this information is. Remember the usual caveat applies; you do this at your own risk!- Editor.

USING TI-WRITER

As I have learned to use the TI-WRITER program, as well as learning to use my printer, I have noticed that the printer was capable of doing many useful things that are not included in TI-WRITER.

The problem was how to use the features of the printer in my documents that are written with TI-WRITER. After beating my head against the wall trying to solve this problem I decided to read the instructions, again, which is like watching your favorite movie ten times, you keep seeing something new that wasn't seen before.

After reading the instructions and looking at some examples from the printer operators manual, I found that the problem was rather easy to solve by using the TRANSLITERATE command that follows the form .TL n1:n2....nz. This assigns one or more ASCII character values to another ASCII character value. The "n1" in this command is the old value, and "n2....nz" represents the new value or values to be assigned to it.



A word of caution though, Once the printer is placed in a particular mode it stays that way until it is told to change. In other words when the code is given to print in italics it will continue to print italics until the code is given to cancel the previous code. For every function engage code there must be a function cancel code.

There cannot be more than one transliterate command per line or they will not work. They are very helpful and worth the trouble.

It's even good for the fine print like saying that this document was printed on a STAR SG-10 printer, and that you can print 132 columns in the space of 80 columns with the use of compressed print mode.

Another useful feature of TI-WRITER is the CONTROL U command in EDIT mode. With this command you can also send control commands to the printer, but which will not be printed in your document.

The CONTROL U mode is entered by pressing the control and the u keys together. Generally it works like the TRANSLITERATE command except that you don't have to redefine any characters because the codes themselves are entered into the file.

Each code MUST begin with an escape character, CTRL U FCTN R CTRL U, and may be followed by any none CTRL U letter or number and more CTRL U codes. If more than one printer code is sent at a time each code MUST be preceded by an ESCAPE code.

One minor problem though. In edit mode all those printer codes take up space on the line and although they are not printed the line gets closed up and you will have an extra space at the end of the printed line for each character in the code, so even though they are very useful they shouldn't be overdone.

Play around with these commands to find out what you can and cannot do. One more thing, don't forget that if you use compressed print it is 132 column on a 8 1/2" wide paper and your margins will have to be changed to reflect this. This cannot be done in EDIT mode as it has a max of 80 columns, so FORMAT mode is the only way to use compressed print and have it come out right.

I hope that this has been of some help.

BY STEVEN LISONBEE
323 No. 750 East
Orem, UT 84057
January 3, 1987

The following article is taken from the Ozark 99 ER, November 1985. For those of you who use TI WRITER this list of commands should be of great help.

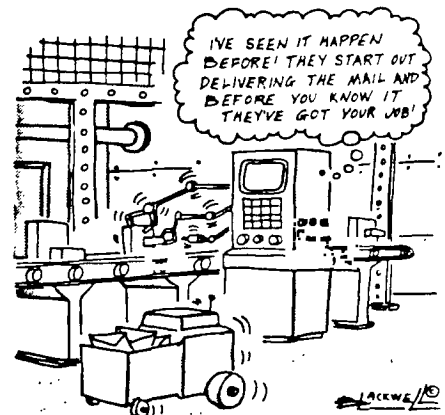
**** TRANSLITERATION COMMAND SET ****

A couple of month's ago Von Barker brought copies of a set of "Transliteration Codes" to the meeting for members who were interested. At the time, I had no idea what these codes were for and unfortunately, I'm afraid, neither did many of our members. I say "unfortunately" because a little "digging" has revealed to me just how useful this set of codes can be once you figure out how to use them.

The first place I was able to find a description of how to use these codes was in an article, written by Ron Castleton, in the February, 1985 issue of MICROpendium. Proper use of the codes allows you to access the special capabilities of your printer (such as different type styles, undefine, superscript, etc.) by inserting one or two symbols into your text while you are writing it.

Here is how it is done. First, type in the left column of table below using the Editor of TI-WRITER. Make sure each .TL command is on a separate line.

TRANSLITERATION	KEYSTROKE	PRINTER COMMAND
.TL 123:27,52	FCTN F	Start Italics
.TL 125:27,53	FCTN G	Stop Italics
.TL 91:27,83,0	FCTN R	Start Superscript
.TL 93:27,83,1	FCTN T	Start Subscript
.TL 124:27,84	FCTN A	Stop Super/Subscript
.TL 1:27,66,3	CTRL U SHIFT A	Start Condensed
.TL 17:18	CTRL U SHIFT B	Stop Condensed
.TL 2:27,87,1	CTRL U SHIFT R	Start Enlarged
.TL 18:27,87,0	CTRL U SHIFT C	Stop Enlarged
.TL 3:27,66,2	CTRL U SHIFT S	Start Elite
.TL 19:27,66,1	CTRL U SHIFT D	Stop Elite
.TL 4:27,45,1	CTRL U SHIFT T	Start Underline
.TL 20:27,45,0	CTRL U SHIFT E	Stop Underline
.TL 5:27,71	CTRL U SHIFT U	Start Doublestrike
.TL 21:27,72	CTRL U SHIFT F	Stop Doublestrike
.TL 6:27,69	CTRL U SHIFT V	Start Emphasized
.TL 22:27,70	CTRL U SHIFT V	Stop emphasized



Then, move to the line immediately below your last .TL command and press CTRL U SHIFT P. This will print a character that looks like a "funny" zero. It marks the end of your file. Next remove the carriage returns from the end of each line, and then save the file to disk using a name that's easy to remember (for example : DSK1.TRANS). Now, your file is ready for use whenever you need it.

To use the commands, include the statement .IF DSK1.TRANS (or whatever you called it) at the start of the text file you are creating. This will include your transliteration file automatically in the text file you are creating. You could merge the transliteration file into each document, but use of the Include File (.IF) is more efficient! Then, to underline a section of text (for example) enter CTRL U, SHIFT D to start underlining - leaving a space after the preceding word-and CTRL U, SHIFT T to stop the underlining. (You'll also need to press CTRL U again to return to the "normal" cursor, after you have put in each control code.) When your text file is run through the Formatter the control codes will not be printed, but rather they will tell the printer to do what you want it to do. This makes it easy to mix print styles, underline sections of text, add subscripts or superscripts, etc.

You'll probably want to print out and save the right and middle columns of the table above to use as a "reference card". The codes in the table work with my Gemini 10x printer and should work with most Epson or other Epson compatible printers. If your printer uses different control codes, you should be able to easily substitute them in the transliteration statements to achieve the same result. If you have any questions about any of this, bring them to the next meeting and we will get them answered for you.

The following article is reprinted from archived articles and I think worth re-examination considering the theme of other submissions in Newsletter 9T9 -Ed.:

TI-Writer's Special Functions -by Nicos Evdemon

Our TI-Writer certainly is a powerful word processor. Some of our experts say you can do graphics, even play games with it. Well I haven't looked in great detail into all the extra capabilities, but I have explored its SPECIAL FUNCTIONS (see Page 146 of the manual).

I have a Mannesmann Tally, Spirit-80 parallel printer and had a lot of surprises trying these functions.

By imbedding these commands before & after your letter, word or sentence; your printer will execute these commands without, of course, printing the symbols or spaces, on which, they were written.

Here is a list of the commands:

CTRL U	SHIFT N	CTRL U ->	LARGE CHARACTERS ON
CTRL U	SHIFT T	CTRL U ->	LARGE CHARACTERS OFF
CTRL U	SHIFT O	CTRL U ->	COMPRESSED CHARACTERS ON
CTRL U	SHIFT R	CTRL U ->	COMPRESSED CHARACTERS OFF
CTRL U	SHIFT NO	CTRL U ->	DOUBLE STRIKE ON
CTRL U	SHIFT RT	CTRL U ->	DOUBLE STRIKE OFF
CTRL U	im	CTRL U ->	UNDERLINE ON
CTRL U	im	CTRL U ->	UNDERLINE OFF
CTRL U	it	CTRL U ->	ITALICS ON
CTRL U	iu	CTRL U ->	ITALICS OFF
CTRL U	[CTRL U SA	->	MINI CHARACTERS ON
CTRL U	[CTRL U T	->	MINI CHARACTERS OFF
CTRL U	o[CTRL U SA	->	MICRO CHARACTER ON
CTRL U	o[CTRL U T	->	MICRO CHARACTER OFF



In your printer's manual, you can find the different control codes & the corresponding Dec. numbers. For my printer, for example, to print enlarged characters (Double-width mode), the Dec. 14 is necessary. To get that 14 with TI-Writer, you have to add 64 to it. So 14+64=78. In your TI manual or reference card, you find that 78 (ASCII) represents the letter N. Therefore, to go into SPECIAL FUNCTION mode with CTRL U. Then type SHIFT N. To exit the SPECIAL FUNCTION mode press CTRL U again. Note when in the SPECIAL FUNCTION mode the cursor changes to a flat line-Ed.). Now every thing entered will be printed enlarged, until you exit this mode. To cancel that command, the Tally needs Dec. 20. So 20+64=84 and 84 is T. Therefore, the sequence CTRL U SHIFT T CTRL U, will return to the standard font.

Another example is underlining. Tally needs ESC code which is Dec. 27, then Dec. 45 and n=1 to start. There it is again; 27+64=91 or [which you get by FCIN R. And 45+64=109 or m. Therefore, the command should be CTRL U [m CTRL U. To cancel that command, Tally needs n=0 and to get zero, Dec. 64 which is character must be entered, because TI-Writer will deduct 64 from it; 64-64=0. The command should be CTRL U [m CTRL U.

The final example is Micro-characters. Tally needs Dec. 27 (ESC code), Dec. 83 (set Superscript), Dec. 65 (Lf spacing n/72). The first control code is 27+64=91 or [. The next code is 83+64=147 and page 98 of TI-Writer manual explains that only codes 32-127 can be used with the SPECIAL FUNCTION. But that's no problem, we can use CTRL U [CTRL U SA. The ESC code is activated, (CTRL U [CTRL U). The printer now waits for the next command, which is Dec. 83 or S and 65=A. So immediately get out of the SPECIAL FUNCTION mode; we type SA and the printer knows what to do. To cancel Micro, Tally needs Dec. 84 or T. So we type the above sequence, substituting T in place of SA.

ANNOUNCEMENTS; - A NEW PARTNERSHIP

(By Don Veith)

The newsletter is free to anyone who wishes to spend the time downloading the issues. It was designed for the individual computer owner to enhance the use of their equipment. Permission is granted to all TI Users Groups to reproduce articles taken from the Compuserve Online Editions of THE NATIONAL NINETY-NINER provided the four conditions outlined below are met.

CONDITIONS ON THE USE OF THIS NEWSLETTER:

This edition of THE NATIONAL NINETY-NINER was provided for the subscribers of COMPUSERVE to increase their enjoyment of their TI 9900 based systems. This newsletter and any article contained herein may be republished in ANY LEGITIMATE TI USERS GROUP NEWSLETTER with the provision that:

- (1) PROPER CREDIT IS PROVIDED FOR ITS SOURCE (THE NATIONAL NINETY-NINER INCLUDING VOLUME AND ISSUE)
- (2) THE AUTHOR'S NAME ON ANY ARTICLE USED IS NOT DELETED
- (3) THE ARTICLE IS NOT ALTERED IN ANY MANNER
- (4) IT WOULD BE APPRECIATED IF A COPY OF THE NEWSLETTER ISSUE IN WHICH THE ARTICLE IS PRINTED WERE FORWARDED TO THE 99'ERS ASSOCIATION.

NOTE: Permission IS NOT GRANTED for this newsletter or any portion thereof TO BE UPLOADED TO BULLETIN BOARD SYSTEMS or other ELECTRONIC MEDIA. Commercial publications may NOT COPY OR REPRODUCE THIS newsletter or any portion without the written permission of THE NINETY NINERS ASSOCIATION.

USING SUBPROGRAMS ON THE TI-99/4A

-by JIM PETERSON

Many TI programmers have never used subprograms, partly because they do not understand them or think that they are difficult to use, and partly because they cannot see that they have any advantage over the GOSUB with which they are familiar.

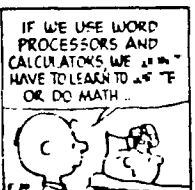
Actually, they are easy to understand and to use, and they have several advantages over the GOSUB. The one advantage, which is seldom mentioned, is that they can save many lines of programming, because the values to be used in a subprogram can be listed in the parameters of the CALL, whereas the values to be

used in a subroutine must each be defined by a separate statement.. For instance, 100 CALL DISPLAY(3,1,"This is an adaptation of the old TI Basic HCHAR print routine, to print multiple 32-column lines")...SUB DISPLAY(R,C,M\$):: FOR J=1 TO LEN(M\$):: CALL HCHAR(R,C,ASC(SEG\$(M\$,J,1))): C=C+1+(C=32) :: R=R-(C=1):: NEXT J :: SUBEND. Compare that to 100 R=3 :: C=1 :: M\$= (etc.) :: GOSUB 1000 !

The other great advantage is that the values given to variables in the subprogram do not affect the values of the same variable in the main program, unless the variable is listed in the parameter list. In the above example, if J had a value of 0 in the main program before the CALL to the subprogram, it will still have a value of 0 after the return from the subprogram, even though the J in the subprogram left the FOR-NEXT loop with a much higher value.

What is the advantage? This means that you can build up a disk library of useful routines in subprogram format, saved with the MERGE option, and merge them into any program where you need them, without worrying about duplicating variable names or interfering with their values. Just be sure to give your routines high line numbers so they may be merged into your programs without overwriting existing lines of code. Line-number them in a continuous sequence so you can merge them in without overwriting each other. If you have my Nuts Bolts disks, or think you might get them, keep your line numbers below 20000 which is where mine begin.

cont



Subprograms are usually short. On my Nuts Bolts disks, I have used every programming trick that I know to compact them into only 2 or 3 sectors, so that I could offer 100 or more of them on an SS/SD disk and so that they would take up as little program space as possible. However, they do not need to be short. Some programmers like to write their programs in modular format, in which the main program is nothing but a series of GOSUBs to several large subroutines. Those subroutines could be easily converted to subprograms, and saved as a merge format library. Custom programmers can build up a library of commonly used business routines, and quickly create new programs by merging in the required subprograms and linking them with CALLS.

Now, as to the difficulty - you are already using subprograms every time you write a program, with no difficulty at all. You start out with CALL CLEAR - that is a subprogram! So is CALL SCREEN(16). However, when you use a built-in subprogram like that, it will only do what it is supposed to do. If you write your own, you can make it do anything you want to. Try this -

```
100 CALL SCREENSUB(5,7,16)....1000 SUB SCREENSUB(S,F,B):: FOR SET=1 TO 14 ::
CALL COLOR(SET,F,B):: NEXT SET :: CALL SCREEN(S):: CALL VCHAR(1,31,1,96)::
::BEND.
```

You can even use the name of a built-in subprogram for your own subprogram, but then you can't use it for its original use. That's why I called this one SCREENSUB, so that I could still use CALL SCREEN within it.

You have been wishing that TI had given you more CALLS in Extended Basic? It has HCHAR for horizontal lines and VCHAR for vertical lines, so why not something for diagonal lines? Well, write your own! 100 CALL DCHAR(1,1,42,100) :: C=C+1+(C=32) :: NEXT J :: SUBEND.

There are two things to watch out for when building a library of subprograms. DATA statements are read from both the main program and the subprogram in their line-number sequence, regardless of whether they are in the main program or a subprogram, unless a RESTORE directs otherwise. This can be very handy, because DATA in the program can be read into a subprogram. However, it can also cause problems. Since subprograms must follow the main program, there is no danger of DATA in subprograms being read by the main program unless it erroneously calls for reading more DATA than the main program contains. However, if a merged-in subprogram contains DATA statements, it must RESTORE them before reading them, or previously unread DATA in the main program will be read instead. And then, the RESTORE will prevent any further DATA from being read in the main program, unless it is restored. It is a good idea to avoid putting DATA statements in subprograms if possible, and it is good programming practice in any case to RESTORE all DATA before reading it.

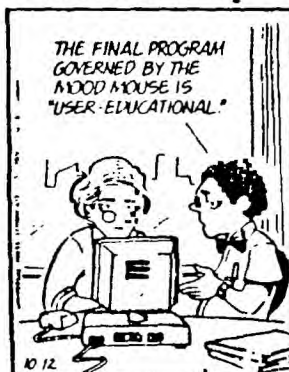
And, a running program will act on the last ON ERROR instruction it has received, whether that was in the main program or a subprogram. This can create some very puzzling situations. An error encountered in a subprogram can cause program execution to jump directly to the RETURN line in the main program, and if a subprogram contains ON ERROR a subsequent error in the main program will jump the program execution to the RETURN in the subprogram, causing a MOST BE IN SUBPROGRAM error when it reaches SUBEND.

I generally avoid using ON ERROR in my programs, except to trap I/O errors which cannot be trapped by other means, and then I place an ON ERROR STOP immediately after the program line for which the trap is intended. If it is essential to use ON ERROR in a utility subprogram, I place ON ERROR STOP before the SUBEND and each EXIT; any active ON ERROR in the main program must then be reactivated after leaving the subprogram.

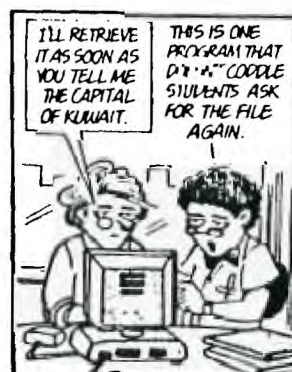
The Tigercub Software's Nuts Bolts Disk (#1) contains 100 utility subprograms in MERGE format, plus a tutorial on the use of subprograms. Nuts Bolts Disk #2 contains another 108 utility subprograms. I am working on a third disk which will contain at least 100 more. These include a very wide variety of redefined screen character sets, screen display routines, sprite routines, wipes, pauses, saving and reading, time and date, music utilities, sorts and shuffles, keyboard and joystick control, math routines and conversions, graphs, speech, editing, word processing, file handling, etc., etc. They are fully documented, with an actual example of the use of each one, and are available from Tigercub Software, 156 Collingwood Ave., Columbus OH 43213, for \$19.95 each or both for \$37, postpaid.

Doonesbury

By GARRY TRUDEAU



13



NEW FROM MONTY SCHMIDT: **GPL LINKER V1.1 Run Time Version**

now \$49.95
w/Linker \$59.95
plus Intern \$69.95
add \$3 shipping

GPL Linker is an ingenious program that places the power of Graphics Language Programming (GPL) at your command. No extra hardware is required beyond standard 32k and disk system. In short, Linker creates runnable program files from compressed (or uncompressed) GPL Assembler object files. You can then run these programs with "Option 5 Run Program Files" of the Editor Assembler Module.

Up to 24k GPL programs can be developed and run on standard 32k systems. Included in the run time version are two demonstration programs and "CONVERT," a public domain conversion program that converts MS BASIC statements to TI BASIC statements. Price: \$21.00 CDN funds \$15.00 US funds.

ENHANCED GPL Assembler V2.1

NOW with high memory loader package

UNLOCK ALL THE SECRETS! New GPL Assembler Version 2.1 available exclusively through RYTE Data.

This program provides the power to write, edit and assemble true GPL programs for the TI 99/4A. Create code that accesses console operating system routines directly. Develop programs that use the GPL Interpreter and all the features of the TI 99/4A.

This package includes the GPL Assembler disk, printed documentation, GPL tips and hints, update support service and commented GROM/ROM listings (with the book "INTERN"). An example for a command module type GPL program is included with source, object and list files on disk.

Requires: 32k memory, disk drive(s), TI Editor Assembler package. Printer/RS-232 recommended.

GPL price Reduction

R/D COMPUTING Technical Newsletter

with Bill Gronos on assembly!

We have a vision. Our vision is one of continued TI 99/4A support. We're dedicated to the power of the machine. From the novice to the experienced computer user; for management, home, education, entertainment or advanced applications our publication "R/D COMPUTING" is for you. TI never revealed all the important inner workings of the 99/4A. We bring you this vital information every month.

A major feature of R/D COMPUTING is the regular "upgrade projects." These electronic construction projects are designed to give the 99/4A owners more features and improvements. For example, it is possible to increase the speed of your computer with a very simple part and switch. Each month we present new circuits, diagrams and projects for your computer.

From the moment your new subscription arrives at your home, you will have access to critical technical information that makes your computer more valuable, powerful and versatile.

We believe that the TI 99/4A deserves new products, innovative hardware, software, information and a dedicated technical publication. This is what makes a computer "viable" in the fast paced microcomputer industry. Now that the 99/4A has been 'opened up,' all the secret information is available. You can have all these benefits and more each month. **SUBSCRIBE NOW!**



Designed for the CorComp Clock Peripheral—Triple Tech Card or Stand-alone models. This utility package provides more functions for use in your Extended Basic programs. Direct access to the clock ROM at assembly speed gives you these features: three independent timers to set and read; alarm function; two interrupt routines to display time and date on screen with CTRL T—continuously or on your command; all time and date displays are in 12 or 24 hour format using **TEXT**. This program also allows the week, date and time to be set independently rather than all together.

Program disk is not copy protected to allow you full use in your Extended Basic programs. Package includes disk and instructions. Only \$17.95 plus \$2

XBII plus

As reviewed in Micropendium October 1985. This command module gives you all the features of Extended Basic PLUS 40 new commands

Totally compatible with TI's XB, this enhanced version gives your programs more power to access your 99/4A. Commands such as MLOAD, MSAVE, VPEEK, VPOKE, GPEEK are superior to most other Basic environments. Various demo programs and new applications using high resolution graphics make this module a "must" for Extended Basic users. Comes complete with a 95 page manual. Requires console and 32k. \$75.00 (US) plus \$2 shipping

BASIC Compiler V1.1

New Basic Compiler that is finally easy to use! Supports virtually all Basic and Extended Basic commands in

existing programs. Simply load and compile programs from a menu driven directory on your screen. No extensive re-writing, variable declarations or conversions are required. Compiler produces code-list in one pass containing all variable addresses and jump list. Package includes Extended Basic Loader, Floating Point Loader, Integer Loader, Disk Menu program and DSR program for the Compiler support. This Compiler cannot unravel DEF statements and stops on the END statement—no SUB's allowed. TRACE, BREAK, ON ERROR, CALL LOAD and CALL LINK may produce execution errors. Requires 32k. disk. Price: \$20.00 plus \$2 shipping (US funds).

\$14/year - back issues 3-15 available

THANK YOU! Our business has grown 300% this year. To show our appreciation we are giving away hundreds of dollars in TI products to 99/4A owners or users groups. To enter drawing, (no purchase necessary) send your name and address on a postcard to RYTE Data. For subscribers to R/D Computing we are giving away XBII plus, 32k memories, GPL Assembler package, etc. **Enter your subscription today!**

Prices listed in U.S. funds.

New catalogue available.

Ryte Data (705) 457-2774



MILLENNIUM COMPUTERS

210 MOUNTAIN STREET,
HALIBURTON, ONTARIO K0M 1S0
TELEX 06-986766 TOR. ATTN. RYTE DA