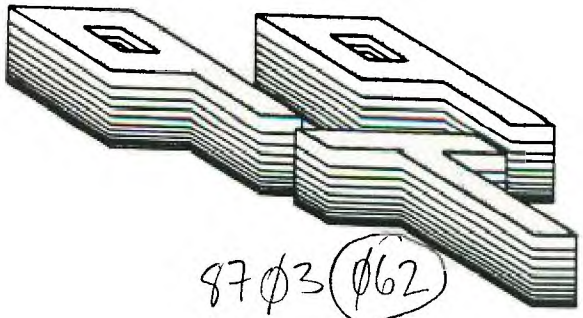
NEWSLETTER 9T9

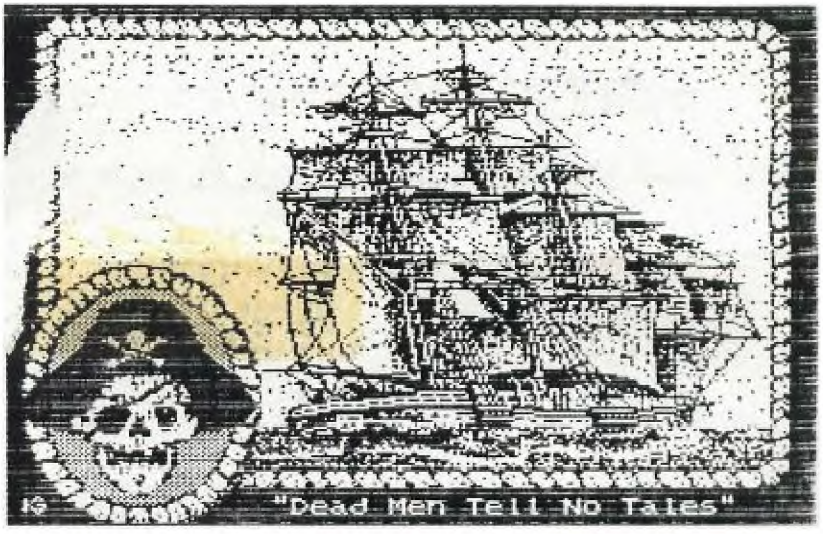
Texas Instruments
USERS GROUP
TORONTO

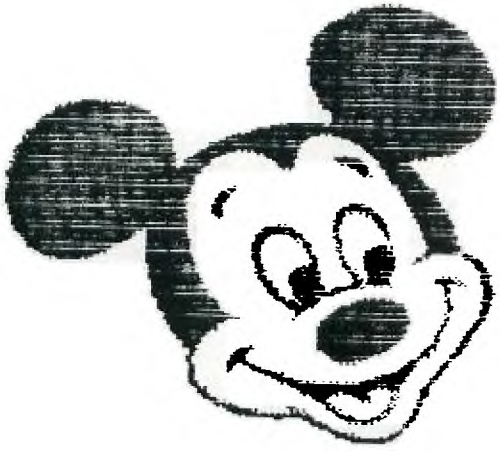
FOR THE TI-99/4A COMPUTER

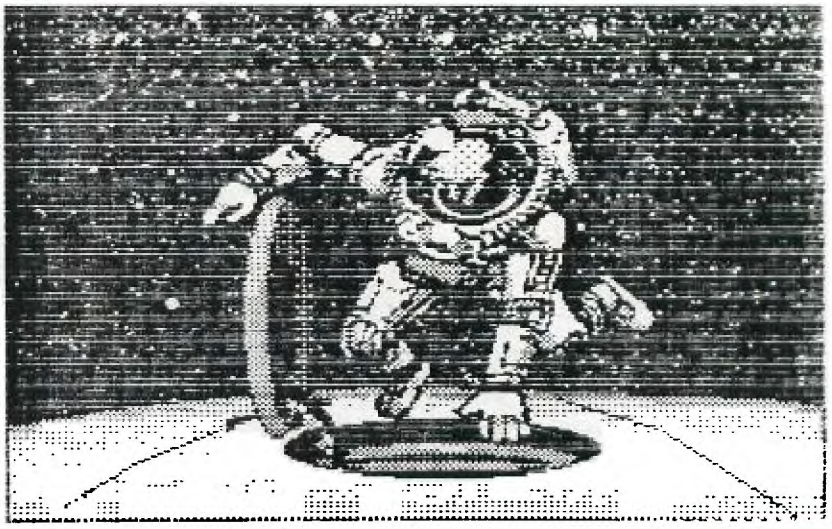


8703(062)
Toronto

MARCH 1987 ISSUE







FROM:
NEW MAILING ADDRESS
9T9 Users Group
#109-2356 Gerrard Street E.
Toronto, Ontario
M4E 2E2
CANADA

25
1987

TO:
Edmonton UG
Box 11983
Edmonton, AL T5J 3L1

Texas Instruments
USERS GROUP
TORONTO

NEWSLETTER 9T9
FOR THE TI-99/4A COMPUTER

COMMERCIAL ADVERTISING

Any business wishing to reach our membership may advertise in [illegible] newsletter. The rates are as follows. (width by height)

FULL PAGE	(9" x 6")	$[illegible] 00
HALF PAGE	(4 1/2" x 6")	$[illegible] 00
QUARTER PAGE	(4 1/2" x 3")	$[illegible] 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 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 [illegible] of any article appearing in this newsletter [illegible] is given to the author and [illegible]. 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), [illegible] by the 9T9 Users Group of Toronto, Canada, all rights reserved.

DISCLAIMER

Opinions expressed in this newsletter are those of the writers and are [illegible] 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 information purposes, and the author, newsletter [illegible], and/or 9T9 [illegible] cannot be held liable for damage to the reader's equipment. All such projects are done at your own risk!

EXECUTIVE COMMITTEE

PRESIDENT Steve Nickelson ([illegible]-1494)
VICE PRESIDENT Neil Allen [illegible]
SECRETARY MEMBERSHIPS [illegible] (469-3468)
TREASURER [illegible] (820-7389)
OFFICER AT LARGE [illegible] (793-3761)
LIBRARY DIRECTOR Gary [illegible] (960-0925)
Andy Parkinson (275-4427)
Cecil [illegible] (671-2052)

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

NEWSLETTER EDITOR

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

MEMBERSHIP FEES

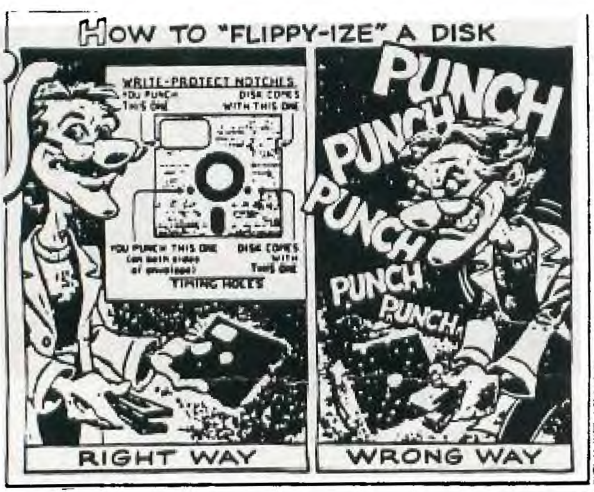
FULL MEMBERSHIP	$[illegible] 00 / year
NEWSLETTER SUBSCRIPTION	$[illegible] 00 / year

All memberships are household memberships. An newsletter subscription is only [illegible] 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 general 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 his answering machine.

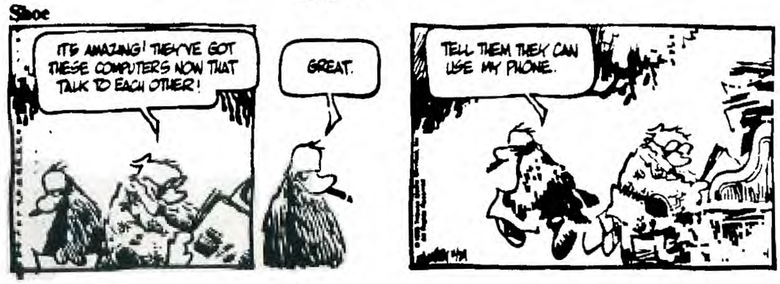
NEXT MEETING

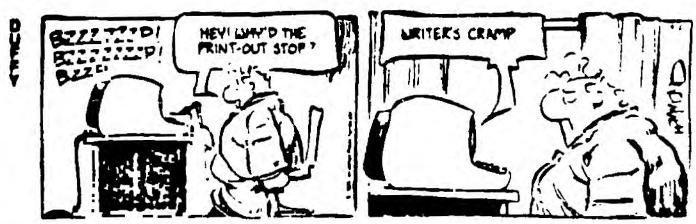
The meetings are held on the last Tuesday of each month. The next will be held at the York Woods Public Library in Downsview, starting at 7:30 pm. The library is at 1785 Finch Ave W just west of Keele St. The entrance to the library is on Finch Avenue. See meeting schedule, below:

March 31
April 28
May 26
June 30



Shoe





TI SYSTEM FOR SALE!!

HARDWARE:	*SOFTWARE:*
TI Console	*Extended Basic*
Expansion Box	*Household Budget Manager*
32K Memory Card	*Personal Record Keeping*
TI Disk Controller Card	*Touch Typing Tutor*
TI Disk Drive	*Number Magic*
RS232 Card	*TI Writer(Includes Manual)*
Cable For Printer	*Multiplan(Includes Manual)+MORE!*

TOTAL PACKAGE - $500. OR BEST OFFER!

Contact: *Rick Schmidt*
30 Troy St.
Kitchener, Ont.
N2H-1L8

Phone #: *(519)742-2916*

Steve's TId Bits #7

Corrections, corrections: Besides some of the obvious typo's was Sean Wainstein's number, which should read: 494-6176. Sorry about that one!

As was announced at the last meeting, by John Palmer, his Glakonian BBS is back on, with a vengence, after lightning zapped his old modem. The "Glank" is now up at 300/1200 BBS, 24 Hrs., 7 Days a week. His number is 283-0847.

As you may have noticed, I have joined DELPHI and dropped my SOURCE I.D.(TI6780). Part of the reason is economics, $7.20(US)/Hr. for DELPHI(at any BAUD rate upto 2400 BPS) vs $12.50/Hr.(US) at 1200 BPS for the SOURCE. This, plus the lack of activity on the SOURCE, as well as a missing $15.00 credit for signing-up my boss on the SOURCE made the decision to change easy. The absence of a minimum billing with DELPHI, also helped. I will remain with the Source for another month, to confirm that the credit,(which seems to have been straightened out), has been processed. I will miss alot of those on TISIG, including Terry Atkinson,(who appears more often since the sudden demise of TIMELINE) and LEO, in Florida. I have been with the SOURCE for over three years and wish them well!

By the way, you may have noticed in the Walt Howe's article a procdure for signing-on DELPHI. It should be noted that when you first connect, you'll see alot of garbage, as if your on the wrong BAUD rate; just hit the letter, A, and a carrage return and follow the prompts as he has outlined. Randy Rossetto is planning to sign-on GENIE, which will soon be available,locally. Perhaps we can get a review from Randy, then. As far as DELPHI is concerned, the XMODEM downloads seem faster than those from Compuserve or the SOURCE.

Also, thanks to Gil Tennant, who provided the Terry Atkinson Gram Kracker series, from the defunct TIMELINE. We were quoted in February's MICROpendium, particularly the Atkinson series. Our VALGAL language was quoted in March's Toronto Computes! Toronto Computes, also quoted from the Myarc 9640(Geneve') vs the TI-99/8. This, plus being quoted in other TI Newsletters, is a good yardstick as to how current and newsworthy Newsletter 9T9 is to both the TI and general computer communities.

You may have noticed the switch from italic to standard font, which should make the print more legible. I have downloaded a number of Freeware/Public Domain programs, which I'll pass-on to the librarians for distribution. They include V2.1 of ARCHIVER, WORDCOUNT(a TI Writer utility which counts the number of charcters,words,spaces, and more. Written in C(fast), runs E/A #5), some new RLE files, plus four different programs which let your TI sing! They have to be heard to be believed!

Please note our new mailing address. If you send correspondence to the old one, don't worry, I'll see it gets to whomever. So next time your updating your mailing address, please note the new one.

For the next meeting, we hope to have a demo of the Supercart. For you Corcomp users I'll try to bring my system with the new EPROMS from Millers Graphics, with its extra calls and other features.

The databases have been curiously quiet re: news of the Geneve', AKA the 9640. I guess they are being "wrung-out". Perhaps Lou Phillips is waiting for the TICOFF for formal introduction. If Texaments, as rumoured, did, indeed, get most of the first 500, then Texaments, being an exhibitor for TICOFF, may be calling the shots, as to where and when the "coming out" party will occur.

RE: OUR CLUB LIBRARY. THE ORPHAN CHRONICLES HAS NOT BEEN RETURNED, NOR THE DOCUMENTS FOR THE DIAGNOSTICS DISK FROM TI. PLEASE SAVE YOURSELF AND THE 9T9 OFFICERS THE EMBARRASSMENT OF PUBLISHING YOUR NAME, AND BRING-IN BOTH THESE PUBLICATIONS. THERE ARE MEMBERS WHO HAVE, YET TO HAVE A CHANCE TO CHECK EITHER OUT.

Also, several of our executive intend to go to Ottawa TI Fest, May 16. Check last month's newsletter for details. Or contact Gil's FIDO BBS, at 288-9412, (Mon.-Thurs.,10PM-10AM; and from 10PM,Fri. to 10 AM,Mon.). We are announcing that the executive is taking a collection for DM-1000,(I, too, am guilty of not paying the author, previously, so will contribute), so that a cheque, on behalf of the members of the 9T9 users can be presented to the Ottawa Users Group for this fine software. See Andy, Randy or myself, to contribute. Remember, none of Freeware or Triware is free, and as any of you who are working on the clubs programming contest know, it takes alot of time to write a good program. So if you use such software, support the writer's efforts.

That's it for now got a deadline to meet. CUL (see you later)!

This file was origi nally downloaded and edited by Steve Mickelson, CIS

MAX-RLE by Travis Watford

- USER DOCUMENTATION FILE (PG)

Objectives of this file are as follows: (1) Brief explaination of RLE files as used by CompuServe, (2) Equip ment required to load and run MAX-RLE, (3) Downloading RLE files from CompuServe, (4) Loading and running MAX-RLE.

(1) What is an RLE?
RLE is an abbreviation for Run Length Encoded. These files are located in various locations on CompuServe are files containing encoded represention s of high resolution images displayable on the screen of the users computer. RLE is mostly intended for users operating with CompServe 's "VIDTEX" Terminal Emulator. However many computers (The 99/4a included) do not have a "VIDTEX" Emulator available.

Here is where the need for an offline display program comes into being. MAX-RLE and its early ancestors and future decendants are intended to display these files, which have been saved to your disk while online, to your screen offline.
(2) What do I need to run MAX-RLE?
MAX-RLE requires the following hardware and software to run:
-A- 99/4a Console (of course)
-B- Monitor or TV (ditto)
-C- TI Expansion System (or equivalent
-D- 32k Memory Expansion (or more)
-E- RS232/PIO Interface (optional)
-F- Disk Drive(s) and Controller
-E- Editor Assembler Module (PHM 3055)
(Note Ex Basic, PHM 3026, or eqivalent and appropriate loader, Systex can be used instead of Editor Assembler -S.M.)
(3) How do I get these "RLE" files?

RLE-files are located in several areas on CompuServe among these are our own beloved TIFORUM Data Librarys, the PICSI G, ARTFORUM, and the CB simulator area.

MAX-RLE was developed with Paul Charlton's Fast-Term Terminal Emulator as the Emulator used in capturing files and is currently the emulator suggested for use in downloading RLE files. For specific information on F-T's operations see the documentation for F-T. For general information on downloading from CompuServe, examine the information here in the FORUM by typing GO TINEWS at the main function prompt while using CompuServe.

Files downloaded from the Data Library's should be done with the XMODEM protocol. Files from the CB simulator area, should be captured in your buffer and saved to disk just as a message would be. XMODEM will save to disk in Display Fixed 128 format and those with the buffer Display Variab le 80 format. The chief difference is th at XMODEM format saves the file in exactly the same format as CIS and the file may be re-uploaded at a later date, or even transmitted to a bulletin board which supports XMODEM. Buffer saves cannot be re-transmitted to CIS as they will lack certain header information identifying them as RLE. This information is stripped out by F-T on downloading.

(4) Ok, How do I use MAX-RLE?
* To load MAX-RLE perform the following steps with the system turned on and the Editor Assembler cartridge in your cartridge port.

- Place the disk containing MAX-RLE in o ne of your drives.
- Press any key to get the Consoles sele ction screen and select option 3 Load and Run.
- At the Filename Prompt type in DSKn.MAX-RLE where n is the number of the drive containing MAX-RLE and MAX-RLE is the name you have called MAX-RLE on your disk.
- Press Enter again to get the Program name prompt. Type in START
* Actually using MAX-RLE (the reason you're reading this)
- From the title screen you have 3 options.
-A- Exit to the 4/a Title screen by pressing FCTN and = at the same time
-B- Catalog a disk by typing in DSKn.
-C- View a file. MAX-RLE supports the translation of 4 types of files to your screen: DF 128 RLE's; DV 80 RLE's; GRAPHIX; TI-ARTIST V/2; to do this just type in the files name. e.g. DSK1.LOGO. TI-ARTIST files used with MAX-RLE do not require you to type in the C nor the P seen in the catalog. MAX-RLE will automaically load both if present. * Ok, Tt's on the screen now what?
With the image on the screen you have three options available.
-E- Return to the MAX-RLE title screen by pressing Enter.
-F- Print screen on your printer. Press P . You will be given the MAX-RLE default of a parallel printer ou tput for EPSON/GEMINI printers. If you are using a Serial printer press enter and then enter your printers description.
-G- Save the image on your screen in one of the 4 above mentioned formats. Press S you will be given the MAX-RLE default of GRAPHIX format to save to. If you wish to save the image in a different format then press the space-bar to get the prompts for t he other formats. And type in the device and filename you wish to save to e.g. DSK1.MYPICT

Next page

NOTES **********
MAX-RLE Supports the Horizon Ram Disk HD command, at the MAX-RLE title screen prompt. Uploading files converted to RLE format to CompuServe. Read the help information available online in the Data Library by typing H. RLE files for uploading to CIS should be saved in DF128 format and uploaded using Fast-Term. While uploading se le ct XMODEM (Modem 7) from CIS's upload options and upload in RLE format. This will automatically mark the title as RLE filetype for identification by CompuServe. In addition when the time finally comes the we TI'ers get a "VIDTEX" compatable emulator we will be able to view the images on line.

Final Note **********
In a conversation with Travis, He commented to me that although he couldn't have the first RLE-Translator for the 99 /4a on CIS (in reference to my 99RLE), he was determined to have the BEST. Well I'll agree with him on that Travis has written the BEST. Thanks from all of us,Travis.

Additional Features(Latest Version)
Keywords: XMODEM TI-RLE GRAPHICS (UPDATED RLE DECODER FOR THE 99/4A TO DISPLAY "RUN LENGTH ENCODED" GRAPHICS AS DISPLAYED ON COMPUSERVE) E/A OPT 3; PROGRAM NAME "A" OR "START". Load and run RLE graphics utility. Name is "START" or "A". Uses un-"cleaned-up" DV80 and DF128 files. While picture is on screen: "P" dumps to "PIO.CR"; "R" dumps to "RS232.BA=1200.DA=8.PA=E.CR"; "S" saves a GRAPHX compatible screen image with the same filename as the original except the last charactor is changed to "*"; "ENTER" allows next file; hex "0"-"f" alters foreground color; hex shift "0"-"f" alters background color. This version is GRAPHX correct.-Travis

From the Front Range 99ers comes the following key inputs, to change the respective foreground and background colors:

Color	Foreground	Background
Black	SHIFT 1	1
Green Med	SHIFT 2	2
Green Lgt	SHIFT 3	3
Blue Drk	SHIFT 4	4
Blue Lgt	SHIFT 5	5
Red Drk	SHIFT 6	6
Cyan	SHIFT 7	7
Red Med	SHIFT 8	8
Red Lgt	SHIFT 9	9
Yellow Drk	A	a
Yellow Lgt	B	b
Green Drk	C	c
Magenta	D	d
Gray	E	e
White	F	f

The following exerpt was downloaded from DELPHI (Editor):

THINGS MY MODEM TOLD ME

(c) copyright 1986 by Walter Howe

This column is an exerpt from my column in the December Boston Computer Society TI User Group newsletter. It may be freely used by any bulletin board or user group, provided that credit is given.

As you have heard in this column before, there are three major national information services with support for the 99/4A: CompuServe, The Source, and GENIE. A fourth is trying hard to join those three, Boston's own DELPHI. I subscribe to the first three, which helps me immeasurably in writing this column, and I used to subscribe to DELPHI, too. I joined it because Randy Holcomb - the original TI columnist in Computer Shopper - was the 99/4A sysop. He dropped out as he went over to the ATARI ST world (although we still hear from him from time to time), and the DELPHI TI Interest Group never got off the ground, except for a core of TI-Pro supporters. I dropped the service a year or so ago, because I felt it wasn't worth it. Well, DELPHI is getting better support than ever now. Jeff Guide of Disk Only Software has taken over as the main 99/4A sysop, and he has enlisted the support of an illustrious group of 99'ers to serve as its TI-99/4A Professional Council. It must be an illustrious group, because I'm one of them (I say modestly, ha ha)! Jeff asked me if I would join Ron Albright (author of the Orphan Chronicles), Paul Charlton, Chris Bobbit (ASGARD Software), and Art Byers (the NJ Faire chairman) on the new Professional Council.

So, I'm back on DELPHI again! In return for some free connect time, it is our job to help shape DELPHI into a service to compete favorably with the big three. Ron Albright is writing a bi-weekly column for it which will be available nowhere else. And in addition to the usual public domain software and fairware, it will be a source for some commercial software, too. Instead of sending to a mail-order company for software, you may be able to download it from DELPHI directly and have it charged to your credit card. You can even put your own software there and charge a fee for others to download it.

next page

If you are interested in trying it out, you can sign up for as little as $10, which includes your first hour of connect time. Charges are $7.20 per hour at any baud rate (including 2400) from 6pm to 7am. There is no monthly minimum. If you want to try it out, you can take advantage of the $10 offer by doing the following: Dial 576-0862(NOTE: In Toronto call 365-7630-Ed.). Hit enter a few times until you are prompted for a user name. Type JOINTI99 (read carefully - that is JOIN TI 99 without spaces, not JOINT 199). When asked for a password, type TELEDATA. You will then be connected and presented with sign-up options (there is a 29.95 sign-up option with three hours connect time and a user manual, too).

You should have your capture buffer on for all this, because a lot of information on the system is given out on your first visit. If you stay with it and decide to sign up, you will get your own password by phone during the next business day. To reach the TI Interest Group from the main menu, the quick way is to type GR TI, which will take you straight there. Then explore the TI files and see what there is to offer. Most commands can be executed by just typing the first few letters of the menu item and successive commands can be chained together by separating them with spaces. I'll keep you posted in months to come how this is all working out.

I haven't much news to offer this month. I could give you an update on the MYARC 9640, aka Geneve, with its completed, delivered gate arrays, but anything I write today will be obsolete by meeting time. I will guess that a production 9640 will be at the meeting! [IT WAS!] If you have ordered one, be patient unless you ordered it a long time ago. Ron Albright reports that TENEX gets 500 of the first ones produced. The smaller dealers may have to wait longer or will just get one or two for a start.

Two new operating systems for the 9640 are coming from Access Engineering Inc. along with a compiler BASIC and a compiler PASCAL. The operating systems are apparently ported over from the TI minicomputer world and offer many sophisticated features, including multi-tasking. AEI is an ASGARD offshoot, headed by Chris Bobbit, Jim Horn, and Richard Roseen. I think we're going to see a lot of things become available for the 9640 in the months to come with capabilities we never thought were possible. There is much more non-99/4A software in existence now that will run on the 9640 than most of us realize.

Rumors are that the 9640 will appear in the March issue of BYTE magazine, (not in March, perhaps a later issue. Ed.) , and it will appear in Computer Shopper. Other national computer magazines are being primed, too.

I have one new piece of software this month. I met Donn Granros, author of Old Dark Caves, in Chicago, and he was kind enough to send me a copy of it. It is a very clever, well-executed Dungeons and Dragons type game with fine graphics. It is an XBASIC/ Assembly hybrid. It was reviewed very favorably in the July 1986 MICROpendium, and I agree with the review after trying the program. A few features that MICROpendium criticized have been improved since, too. It is available from Donn R. Granros, 216 East 60th St., Minneapolis, MN 55419 (updated address from that in MICROpendium) or from Disk Only Software (TELEDATA here). It requires memory expansion and disk drive.

The worst news of the month is that Craig Miller says he has made his last GRAM KRACKER and has sold his last copy of EXPLORER. The problem with the GRAM KRACKER is NOT diminished demand, but that the increased price of components has taken all the profit out of them. He would have to raise his price from about $180 to about $250 to get back where he was before. I guess he does not feel that market can bear that price. It is very unfortunate, because the GRAM KRACKER was just coming into its own with the advent of such marvelous utilities as GRAM KRACKER UTILITIES I and Peter Hoddie's GRAM PACKER. I really hope that something can be worked out for someone else to pick up the GRAM KRACKER or its technology and keep it going in some form. The problem with EXPLORER is that Craig ran out of the large manuals. He didn't think the remaining market for the product would justify another expensive printing. If you have been meaning to get either of these, get on the phone and call dealers now. You just might be lucky enough to find one still.

And while we're talking about bad news, the late Home Computer Magazine continues to impress with their efforts to reimburse former subscribers. One subscriber reports receiving a check for something less than $1 to cover the six or so remaining issues. They claimed that the "free software" he had received with his subscription offset what they would have paid for the unfulfilled issues. What a contrast between Barry Traver, who sends bonus disks when he thinks his GENIAL TRAVelER is coming out too slowly, and Gary Kaplan, who thinks his old magazines were worth an apparent 10 or 15 cents an issue in reimbursement (with offsetting factors, of course), but his new ones are worth $25 apiece in offsetting old subscriptions. How's that for creative accounting?

LATE BREAKING NEWS Received word at the BCS meeting last night that an enhanced MYARC RAMdisk operating system will soon be available for the Horizon RAMdisk. It will include both tagged object code (E/A 3) and program image (E/A 5) assembly program loaders. And best of all, it will be on an EPROM! No more lost operating system at just the wrong time! Thank the cooperative efforts of Peter Hoddie, Lou Phillips, and Paul Charlton for this one. Not sure who is going to release it, but it looks like GENIAL Computerware with MYARC's permission.!

Peter Hoddie's MYARC 9640 was at the meeting, complete with the new gate array chip. He received it at 4pm that day and had to move chips and other components from his prototype board to the new one. He hadn't finished at meeting time, so we didn't see it in action yet.

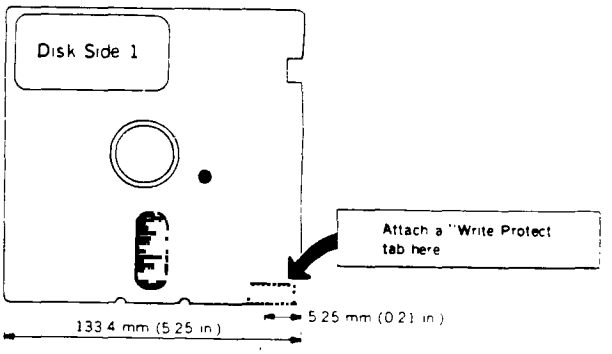
```
0  ; P O O R   M A N S   D O U B L E
   ;     D E N S I T Y   D I S K
   ;       C O N T R O L L E R
   ; or how to get 360K bytes per drive
   ; using the TI Disk Controller Card
   ; and 96 tpi double side drives.
   ; (eg TEAC 55f, TANDON TM100-4

   ;**************************************
   ;*
   ;* This code, when substituted for the
   ;* existing DSR code, converts disk #4
   ;* and disk #5 into 40/80 track inter-
   ;* lace mode and disk #9 to 40/80 track
   ;* non-interlace (single side). The
   ;* modification has been completely
   ;* compatable with all software tested
   ;* including P-System, Disk Manager,
   ;* Editor Assembler and Basic. Note
   ;* however that files can only be
   ;* exchanged with "normal" format disks
   ;* using the disk copy routines or dsk3
   ;* (disk #9).
   ;*
   ;* This code is placed into the public
   ;* domain by the author for non comm-
   ;* ercial use.
   ;*
   ;* Any questions may be directed to the
   ;* author at
   ;*                Andy Cooper
   ;*                121 Clearview Drive
   ;*                Downingtown Pa 19335
   ;*
   ;* or Compuserve 71016,1743
   ;*    Delphi     andy4820
   ;*
   ;**************************************
```

from Toshiba...

A Final Word on Colorful Diskettes and Toshiba Floppy Disk Drives

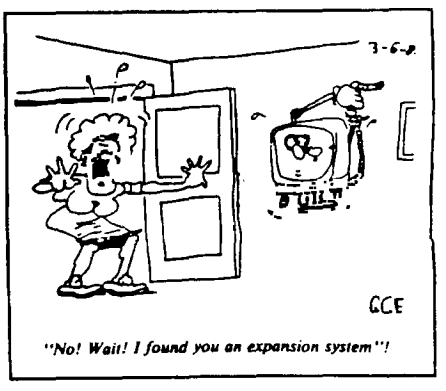
Toshiba Floppy Disk Drives have been designed to give you the optimum in troublefree performance. A part of that built in reliability is the use of electronic rather than mechanical sensors throughout the drive.

Occasionally, these sensors can be confused by the use of some colored diskettes with translucent jackets. If, upon inserting a light colored diskette into your disk drive, you should experience a failure of the drive to sense the presence of the disk; please place a "Write Protect Tab" over the area of the diskette jacket where the disk sensor "senses". Placement of the tab is shown below as a small rectangle with diagonal fill lines on the right side of the leading edge of the disk.



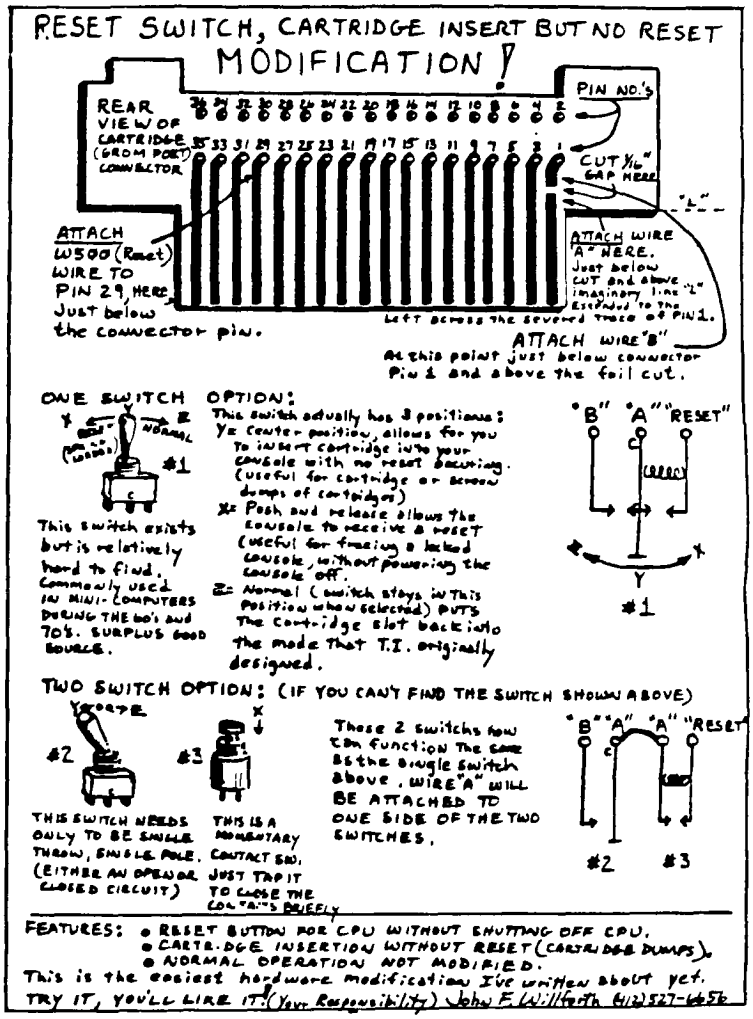

```
(.absolute
(.proc newdsk
(.org 4116h
(clr     r7                     ; existing code
(ci      r1,      1440          ; 1440 is max sector #
(jhe     $+152                  ; jump to error if greater
(ci      r1,      1             ; check for sector = 0
(jh      $5                     ; jump if not
(bl      @4524h                 ; if sector = 0 then restore
 $5      clr      r0
(cb      @004Ch(r9),@4BA6h      ; compare drive # to 3
(jl      $10                    ; jump if 1 or 2
(div     @441Eh, r0             ; if drive = 3 use 9 sectors per track
(jmp     $20
 $10     div      @547Ch, r0           ; if drive < 3 use 18 sectors per track
 $20     swpb     r0
(inv     r0                     ; R0 msb contains inverted track #
(bl      @4614h                 ; set up vdp write
(movb    r0,      @0FFFEh(r15)  ; Store new (calc) track #
(movb    r0,      @5FFEh        ; write trk # to 1771 disk controller
(ci      r1,      9             ; check for sector > 8 (side two)
(jl      $30
(ai      r1,      -9            ; If >9 subtract nine and....
(sbo     7                      ; select side two (head 2)
(li      r7,      0100h
 $30     swpb     r1
(inv     r1                     ; R1 msb contains inverted sector #
(movb    r1,      @5FFCh        ; write to 1771 sector register
(jmp     $40                    ; waste a word !!!!!!
 $40     cb       r0,      @5FF2h       ; Back to original code.
(.end
(
   ;**************************************
   ;*
   ;* To install a 2732 or 2732A prom on
   ;* the TI Disk controller card with the
   ;* above changes perform the following
   ;* steps.
   ;*
   ;* Remove the ROM at location U26.
   ;*
   ;* On the back (non component) of the
   ;* board cut the wide (+5v) etch between
   ;* U26 pins 21 and 24.
   ;*
```

7 next page



"No! Wait! I found you an expansion system"!

```
;* On the front of the board cut the
;* wide etch going to U26 pin 21.
;*
;* On the front (component side) of the
;* board cut the etch going to U26
;* pin 18.
;*
;* NOTE: to ensure that the etch is cut
;* I recommend that two cuts 1/16 inch
;* apart be made and the copper trace
;* be removed between the two cuts.
;*
;* Install a 24 pin socket at U26
;*
;* On the back (non component) side of
;* the board add the following wires:
;*
;* U26 pin 18 to U26 pin 20.
;*
;* U27 pin 18 to U26 pin 21.
;*
;* Via hole in center of chip area
;* between U26 pin 21,22 to junction
;* of C26 and U36 pin 24 (+5v).
;*
;* Install prom in socket and.....
;*
;* thats all!
;*
;*****************************************
(
;*****************************************
;*
;* The following is an Editor Assembler
;* program that may be used to convert
;* normal disks to interlace format
;* prior to changing the Disk DSR. If
;* you wish drive 3 can be used to copy
;* single side disks to drives 1 and 2
;* using the P-System filer or Disk
;* Manager. (adapted from sector RW
;* tutorial written by Tod Kaplan).

MYREG   BSS 32
(   START
(   DSRLNK,VSBW,VMBW
(REF VMBR,KSCAN
START
(LWPI MYREG
LOOP    CLR @>8374
(BLWP @KSCAN
(CB    @KKEY,@SPACE
(JNE   LOOP
(CLR   R4
(CLR   R5
(CLR   R6
LOOP1   CI    R4,360     ;720 FOR DOUBLE SIDE
(JL    LOOP2
(LIMI 0
(LIMI 2             ;WAIT FOR QUIT
(JMP   LOOP1
LOOP2   MOV  R4,@[illegible]TOR ;SECTOR TO PROCESS
(BL    @RSECT       ;READ SECTOR
(MOV   R5,[illegible]ECTOR ;SECTOR TO WRITE
(BL    @WSECT       ;WRITE SECT
(INC   R4           ;INCREMENT READ SECTOR (LINEAR)
(MOV   R4,R1
(CLR   R0
(DIV   @NINE,R0     ;CALC TRACK
(MOV   R0,R2        ;TRK TO R2
(ANDI  R2,1
(JEQ   EVEN         ;JMP IF EVEN TRACK
(CLR   R2
(MOV   R0,R3        ;TRK TO R3
(DIV   @TWO,R2      ;DIVIDE IT BY 2
(CLR   R3
(MPY   @NINE,R2
(LI    R2,711
(S     R3,R2
(A     R1,R2
(MOV   R2,R5        ;INTERLACE (SIDE 2) TRK/SECTOR
(JMP   LOOP1
EVEN
(CLR   R2
(MOV   R0,R3        ;TRACK TO R3
(DIV   @TWO,R2      ;DIVIDE TRACK BY 2
(CLR   R3
(MPY   @NINE,R2
(A     R1,R3
```



8 next page

```
(MOV  R3,R5       ;SIDE 1 TRACK
(JMP  LOOP1
(
 ************************************
 * READ SECTOR ROUTINE
 *
 RSECT
(MOV  @DUMPAB,R0
(LI   R1,DUMDAT
(LI   R2,2
(BLWP @VMBW
(MOV  @KD1RD,@DRVFLG ;READ FROM DRIVE 1
(MOV  @DUMBUF,@BUFADD
(MOV  @DUMPAB,@PABADD
(BLWP @DSRLNK
      DATA 10
(RT
 ************************************
 * WRITE SECTOR ROUTINE
 *
 WSECT
(MOV  @DUMPAB,R0
(LI   R1,DUMDAT
(LI   R2,2
(BLWP @VMBW
(MOV  @KD2WR,@DRVFLG  ;WRITE TO DRIVE 2
(MOV  @DUMBUF,@BUFADD
(MOV  @DUMPAB,@PABADD
(BLWP @DSRLNK
(DATA 10
(RT
 ************************************
 * DEFINES AND DATA
 *
 NINE     DATA 9
 TWO      DATA 2
 KKEY     EQU  >8375
 PABADD   EQU  >8356
 BUFADD   EQU  >834E
 SECTOR   EQU  >8350
 DRVFLG   EQU  >834C
 SPACE    DATA >2020
 DUMPAB   DATA >1500
 KD1RD    DATA >0101
 KD1WR    DATA >0100
 KD2WR    DATA >0200
 KD3WR    DATA >0300
 DUMDAT   DATA >0110
 DUMBUF   DATA >1000

(END
```

(Do this project at your own risk:)

TI WRITER MNEMONIC (MEMORY) TRICKS

CTRL	MNEMONIC	FUNCTION	ALTERNATE
A	ADVANCE DOWN	ROLL DOWN	F4
B	BACK UP	ROLL UP	F6
C	COMMAND MODE	COMMAND MODE	F9
D	(RIGHT ARROW)	RIGHT ARROW	FD
E	(UP ARROW)	UP ARROW	FE
F	FLYAWAY CHARACTER	DELETE CHARACTER	F1
G	GET A HOLE FOR CHAR	INSERT CHARACTER	F2
H	HOP BACK TO LAST	LAST PARAGRAPH	C6
I	INDENT	TAB	F7
J	JUMP TO NEXT	NEXT PARAGRAPH	C4
K	KILL TO END OF LINE	DELETE TO END OF LINE	--
L	LEAP HOME	HOME CURSOR	--
M	MAKE NEW PARAGRAPH	NEW PARAGRAPH	C8
N	NO MORE LINE	DELETE LINE	F3
O	OPEN BLANK LINE	INSERT BLANK LINE	F8
P	PAGE BEGINNING	NEW PAGE	C9
R	REFORMAT	REFORMAT	C2
S	(LEFT ARROW)	LEFT ARROW	FS
T	TAB BACK	BACK TAB	--
U	(USED FOR SPECIAL CHARACTER MODE)		
V	VEER TO LEFT	CURSOR TO LINE START	--
W	WORD TAB	WORD TAB	C7
X	(DOWN ARROW)	DOWN ARROW	FX
Y	YANK MARGIN CONTROL	LEFT MARGIN RELEASE	--
Z	ZIP BACK	OOPS!	C1
-	----------	SCREEN COLOR	C3
-	----------	DUPE LINE	C5
-	----------	NEXT WINDOW --)	F5
-	----------	WORD WRAP	C0

NOT ONLY DO MOST WORD PROCESSING PROGRAMS RECOGNIZE THE FUNCTION OR CONTROL KEYS USED WITH NUMBERS BUT ALSO MANY HAVE CONTROL/LETTER COMBINATIONS.

REPRINT FROM ROM - USERS GROUP OF ORANGE COUNTY, CA.

Gram Kracker Tuitorial Parts III And IV

-by Terry Atkinson(Parts I / II appeared last Sept. / Oct.,Respectively -Ed.)

I guess the best way to do this is to kind of "Dive right in". We will start by moving the CLSALL and CLKOFF subprograms. Like I said before, these are both short 6 byte routines. The CLSALL subprogram starts at the address D870 and CLKOFF starts at D880. This leaves 10 bytes inbetween the two subprograms and since 10 bytes isn't that much, we're not going to worry about them. So in order to save us a little work we will move both subprograms in one shot.

Now before you do the move, go back to the address BD98. Look fimiliar??? That's right!! That's the location of the subprogram headers that you moved in the last fix. There should still be several bytes of unused memory following the headers. In fact, there should be enough to fit not only CLSALL and CLKOFF but also the CLOCK subprogram. If the area is free, you can go ahead and do the move. Enter a Start address of D870 and a Finish address D885 and use BDD2 as the Destination address. When your finished with that, turn the W/P switch off and than press FCTN 2 for the move. Now use Fill to erase the old location. Use the same Start and Finish addresses and make sure you press FCTN 3.

Now that you have moved the subprograms, you will need to make some changes to the headers. Switch the display back to ASCII and find the CLKOFF header. Place the cursor on the byte just after CLKOFF you should be on byte BDBD. Change the display back to Hex. This is the address of the CLKOFF subprog. You need to change it to the new adderss, so replace D880 with the new address which is BDE2. The same kind of change must be done to the CLSALL header. Place the cursor on the byte following CLSALL and enter BDD2. Last of all turn the W/P switch back on so you don't overwrite anything.

next page

That's it for moving CLKOFF and CLSALL. Before you continue you might want to go to ExBasic and try the subprograms to make sure that they work. If everything checks good it might not be a bad idea to save the new module to disk CLOCK is the biggest piece of code that we will move in this first set of GK fixs. Don't let its size scare you. It's really not that tough. The main thing to remember is that when moving any piece of code, you must locate and change all bytes that contain an address that points to something within that code. For example, if you look at the code in CLOCK you will see two strings. Near the beginning of the code you will notice the string CALL LINK("SETCLK","HHMMSS"). This string has nothing to do with the actual Clock routine. It's that little reminder that appears on your screen when you enter CALL CLOCK. Its one and only purpose is to instruct you on how to set the clock. So what's really happening here is that the string is being written to VDP memory. For those of you that aren't fimiliar with the process, when ever you place a character code in certain bytes of VDP, those characters will appear on the screen. So what does that have to do with the moving CLOCK?? Well to start with the CLOCK subprogram contains a GPL move routine that will move the string from gram to VDP. The objective is to move 28 characters (The number of characters in the string. Go ahead, countem' if ya don't believe me!!) to VDP. The VDP address that the string will be moved to and the number of character to be moved can remain the same. The byte that contains the start address of the string is another story. After you move the CLOCK subprogram, this byte will still be pointing to the previous location. To correct this, change the start address of the string to its new location.

The second string that needs to be moved is at the end of the CLOCK sub program. This string is SETCLK. It isn't moved to VDP like the previous string. Instead, this string is moved to the REF/DEF table in CPU memory. This is so that when you execute the CALL LINK("SETCLK","HHMMSS") there is something there for your computer to find. The two strings are moved in the same way. The only difference is the location of memory to which they are moved. So the point that I'm so longwindedly trying to get across is that you're going to have to change two bytes in the CLOCK code.

Soooo.....FOR THOSE IN A HURRY TO GET IT DONE THE ABOVE TWO PARAGRAPHS ARE OPTIONAL READING MATERIAL ONLY, THEY ARE THERE ONLY FOR YOUR EDUCATIONAL ENJOYMENT. THE REAL STUFF IS DOWN HERE!!!

Put your GK in the Edit mode and go to the address B55E. There should still be plenty of memory following the CLKOFF subprogram to squeeze CLOCK in. The CLOCK subprogram starts at D900 and ends at DA55. Use these addresses for the Start and Finish addresses of the move and use B55E for the Dest address then press FCTN 2. Now, like we've done so many times before, Fill the old location of the subprogram with 00s to clear the memory. Now we need to make the changes to the bytes that contain the starting addresses for the two strings. The first bytes that need to be changed are at address B59E. Place the cursor on this address and change the bytes to read B5B7. This is the new start location for the string CALL LINK("SETCLK","HHMMSS"). The Second set of bytes that require changing are at the address B5A6. Place the cursor on this address and change these bytes to read B6AB. This is the new location of the string SETCLK, and is the last change you will need to do in the actual subprogram. Not that bad considering the length of the routine!

One last thing before we're done. Like before we need to go back to the subprogram headers and tell them the new location of the CLOCK routine.

First of all, go to the subprogram headers at BD98. Find the header CLOCK and place the cursor over the byte directly after the "K" in CLOCK and change the bytes that read D900 to read B55E which is the new location for the CLOCK routine. When done, make sure you turn the W/P switch back on and then save the new module to disk.

Well that's it. You're all done. You now should have free memory from D0f4 to DBFF and another large section between DE3D and FFFF. Now all we need to do is to try to find something to do with it. I've already thought of several things I'd like to do and I'm sure you all have a favorite program or module that you would like to try to squeeze in. But before I get to involved in that I think I'll make one more attempt in moving the CAT subprogram. I really think it's just a matter of finding which bytes need to be changed. Maybe with some of your help we can get this mess straightened out once and for all. I do hope that some of you did get a little more out of this than just a couple of bytes of free memory. I also hope this will help to get the ball rolling. Let's face it, the amount of info on the GK is still a little scarce. I doubt that everyone that bought one has got it sitting on the closet shelf. So let us know what your doing. I'm sure someone out there will want to know!!

Before I go I'd like to know what any of you would like to see in the future. (nothing too complicated. Remember I'm still a novice!!) I know that some of you may still be having problems loading more than 1 or 2 modules into the GK, so I could write an article on how to do that, Novice to Novice instead of Pro to Novice as Craig Miller did. I also have some fixes for my fixes that for one include how to get rid of that last foreign language option of the TI-Writer, compliments of Steve Shackerfield.

next page

FIX FOR THE FIXES

Well I blew it!! What can I say?? I guess I need to slow down a bit and take a look at what I'm doing before I pass the info on to all of you. Nothing to worry about. All of the changes you have made so far still work with no problems (I hope!). It was, however, brought to my attention that it is possible to eliminate the last foreign language option from the menu screen. Oh, I knew it was possible I just didn't spend enough time at it. Anyway I'd like to thank Steve Shackelfield and the rest of you for keeping me in line. My GK keeps getting better all the time. But that doesn't mean that it's good enough. So I Dare ya!! Go ahead and show me up, catch me off guard, and show me what you know. I don't mind looking like a jerk, especially if I learn something..... Enough of that it's time to get started.

Unfortunately, because you might have TI-Writer stored in any of the several Grams in the Gram Kracker, this could get a little confusing. I used Bank 1 while doing the changes but they can be done just as easily in Grams 1 or 2 or Bank 2. It is best if you make the changes to TI-Writer in the area of memory where you will have it stored. The reason for this is because when you use the TIW-MOVER program after the changes, the program will rewrite some of the addresses that we will be deleting. It would still work, but why go through all of the trouble to make the changes, just to have a lot of the stuff put back.

If you plan to make the changes in an area of memory other than Bank 1, you can still use the addresses I have supplied, but you must first calculate the offset. For example, Bank 1 has a start address of 6000. If you wish to make the changes in Gram 1 which has a start address of 2000 your offset will be -4000. So, if the byte I said to change is 6006, then 6006 - 4000 which is 2006 is the byte you want to change. If you will be making the changes in Grams 1 or 2, you will need to turn the loader off after entering the GK Editor. This is because the GK loader is stored in this area when the Loader switch is in the "ON" position. LET'S GO... To start off our changes, we first need to clear Banks 1 and 2 by using item 3 of the GK menu (Init Module Space). Now that that is done, you need to load TI-Writer into the GK (the info on how to do all this stuff is in the GK Operators Manual, so I won't cover it here). Go to the GK Editor and press "G" to display Gram memory. Make sure you are in Gram because there isn't anything in CPU or VDP that we're concerned with, at least right now. Once in Gram, use the Search function to find the word "PROCESADOR", using a Start address of 6000 and a Finish address of 6100. This is all "old hat", since we've already been through it in the original GK Fixes (Sorry!!).

When you've found "PROCESADOR", use FCTN 9 to place the cursor on the edit screen. Then use the arrow keys to place the cursor over the "P" in PROCESADOR". Now, using the left arrow key, backspace 14 (fourteen) bytes. This is the second byte of the pointer that points to the first menu option. the complete pointer. If you're using the same version of TI-Writer that I have, the number here will read 6010. These are the bytes we are going to change. Write down the location of these bytes (6006) because we're going to pack our bags and do some more exploring for a while.

Press FCTN 4 to go to the next screen and search for the string "TI". Put the cursor on the "T" in "TI". Now back- space 5 (five) bytes (60CB). The byte at this location, along with the next byte, is the pointer for the next menu item, and since "TI-WRITER" is the last menu item, these bytes should contain zeros. Shouldn't they??? Let's switch to Hex display and see. "Yep, I was right!!! You can switch back to ASCII now." Write down the location of these bytes (60CB). This is the address that you want in the starting pointer for the menu.

NOTE For those of you that are still scratching their heads, saying, "TI-Writer isn't the the last menu item it's the first. I know, because every time I use it, I press 1 not 7. What is this guy trying to pull!!" Take it easy guys, and take another look. Figure it out yet?? That's right, their in reverse order from the menu screen!! TI-Writer prints the menu screen from the bottom up. "What, Where, Why, When???" I don't know why!!!

Now that we have all the information we need. We have to to go back to the location of the Starting Pointer for the Menu (the first number you wrote down. 6006) and change the bytes to the start of the "TI-WRITER" menu option (the second number you wrote down, 60CB). Remember, if you don't turn the W/P off, nothing will change.

It's as simple as that. If you like you can exit the GK and check what you've done. There should only be one TI-Writer selection on the menu. If not, or if isn't the right one, you screwed up, and will need to start over. It might be because the version of TI- Writer that you are using is different than mine. It's best if you don't use the addresses that I supply. Instead, use the locations where you find the menu items.

Well, the first thing that comes to my mind is that, "Since we're just jumping over these menu items, why are we going to keep them??" Well, were not!! But for now, leave them as is. Because we will need them for future reference. End of file.

ADDING 8K OF RAM AND BATTERY BACK-UP TO YOUR EDITOR/ASSEMBLER CARTRIDGE

By Richard J. Bailey, NH99ER USER GROUP

John Clulow and Ron Gries came up with an excellent modification for the E/A cartridge. They wrote an article in the June, 1985 LEHIGH newsletter on how to add 8K (8Kx8) of RAM to the E/A cartridge. This opens all sorts of possibilities, some of which they mention in their article.

I have taken this modification to the next logical step (that John and Ron suggested) and that is to add battery back-up to the 8K of RAM. The circuit I've used is a modification of the back-up circuit TI used in the Mini Memory cartridge and the caution on page 6 of the Mini Memory manual applies to this modification as well. When removing or inserting the modified E/A cartridge with a program stored in the added 8K, the console should be switched OFF to prevent the possibility of any data being lost or altered. I have inserted and removed the modified cartridge with a stored program when the computer was ON and most of the time the program was not distroyed. Any program should be backed-up on disk, just in case.

I will describe the entire modification of adding the 8K and battery back-up here because I changed the way the 8K RAM was added to the board. You should read both their article and mine to decide which way you want to add the 8K. If you want to know how the back-up circuit works, read the following explanation and refer to the schematic.

When the cartridge is powered by the console the +5 volts will reverse bias the two diodes connected to the battery, effectively disconnecting the battery from the circuit. When the power is turned off, the two diodes from the battery are forward biased allowing the battery to supply power to pin 28 and bringing pin 20 high for the low power standby mode. The 2N3904 is used as an inverter to allow the battery to bring pin 20 high during standby but bringing the junction of the 1K and 3.3K resistors low during line operation. During back-up the two diodes connected to +5 volts are reverse biased and the 2N3904 is off. The total drain on the battery is less than 100 microamps with the 6264PL-15 and less than 1 microamp with the 5564PL-15. Either RAM will give a battery life of serveral years. If you don't use the low power version of the 6264, battery life could be decreased by a factor of 20.

WARNING!!! Proceed at your own risk. I have made this modification to my own cartridge, and it worked. Neither John, Ron, or I are responsible if your cartridge is destroyed.

First get a Munchman, Parsec, or other module that has one 24 pin chip and space for up to 4 to 16 pin GROMs. These cartridges have plated "pins" on both sides of the edge connector, and more importantly, most of the runs needed to add the 8K of RAM. Desolder and remove all chips on this board. Add one 16 pin socket in the location nearest the 24 pin chip. This is for the E/A GROM and makes it easier if you want to undo the modification in the future. Also add a 16 pin socket to the original E/A board and save it for the same reason.

The added 8K chip is a Hitachi 6264LP-15 or Toshiba 5564PL-15. These are Low Power chips with the 5564PL-15 being the superior of the two with 1/100th of the current drain of the 6264LP-15 in the stand-by mode. The 6264LP-15 is much more readily available and the battery will supply either chip for years. If you are going to use battery back-up, make sure that you get the low power version of these chips.

Put the 28 pin chip in the 24 pin location with the index toward the top of the board and the "extra" pins toward the top of the board. Mark where you must drill four #60 holes for these pins. After the holes are drilled, use a 1/4 inch drill bit and remove the copper ground plane on both the front and back side of the board. Do this countersinking by hand otherwise you may end up with 1/4 inch holes through your board. Note that the run near the old pin 24 on the upper right hand corner of the old 24 pin chip will be cut almost through by the countersinking operation and you will have to cut and remove the section of this run for 1/4th of an inch on either side of where it passes near the pins and replace this section of run

with a piece of #26 solid teflon insulated wire soldered in place. All pin references for now on will be for the 28 pin chip.

next page

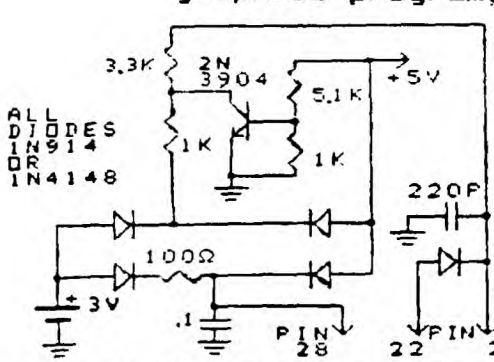
Modify the top side of the board first as shown in the pictorial. Cut the foil between the two pads near pins 22-23 and move the capacitor to the right of holes. Insert a 1K resistor between the bottom left pad and the ground run. Insert a 220 picofarad capasitor between pin 20 and the ground run. Solder the two leads in the ground run pad. Add wire from pin 20 to the logo pad that you will be using for a tie-point. Add wire from pin 27 to edge connector pin 3 and pin 2 to edge connector pin 7 making sure they don't cross and pinch under the 6264LP-15 chip. Add a .1afd capacitor from pin 28 to the ground plane feed-through hole. Also into pin 28 push the cathode end of a 1N914 or 1N4148 diode. Push a large sewing needle in these holes to open them slightly if the chip and parts don't fit. Solder a 100 ohm resister to the anode end and sleeve these components and leads so they can't short out. Add the other components (less battery) using existing feed-through holes in the 5 volt run and ground plane. Note that some components just have their leads soldered together and are free-standing. Make sure that none of these leads short out.

Now modify the back of the board. Cut the runs near pins 20 and 23. insert the 6264LP-15 with the index notch towards the top of the board. Solder pins 5 and 17 to hold the chip in place. Add a bare wire jumper from pin 22 to the pad immediately to its left. The ground plane near the top of the board has been severed by the countersinking operation so a jumper must reconnect it as shown. Pin 23 must be jumped to the run 20 and pad near pin 22. Add diode between pin 28 and run going to pin 26. This diode is shown positioned stightly different in the pictorial for clarity so mount it in the most direct way on the actual board.

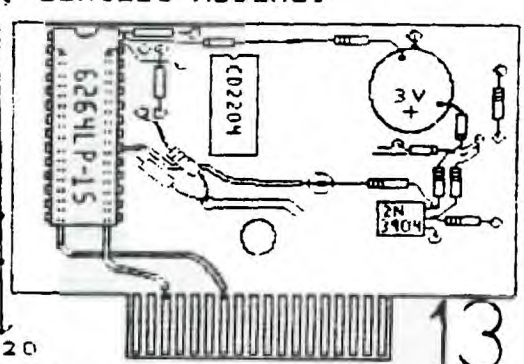
All components, except battery, should now be on the board. Double check all components for location and polarity, especially where component leads come through pin holes, and any diodes going to the battery. Now consult the article by me for replacing the mini Memory battery for instructions for soldering to the battery and the warning with it. Solder a piece of #26 wire to the negative side of the battery with insulation extending 1/16 inch beyond the edge of the battery. Now use one layer of double-back tape over this terminal. Cut a slit in the tape so the solder and wire are not covered by tape. Now put another layer of double-back tape over the first, covering the wire. Push the wire through a cleared ground plane feed-through hole, and stick the battery to the circuit board. The tape should extend well beyound the edge of the battery so there is no chance of it shorting out. Now carefully, following the battery soldering isntructions in the other article, solder the two diodes to the positive terminal (case) of the battery as shown. Measure the voltage on pins 20 and 28 with a high impedance voltmeter. Both pins should read about 0.2 volt lower than the battery's 3 volts. Reassemble the cartridge making sure the two wires soldered onto the edge connector pins don't interfere with the spring-loaded protector if you have the black case cartridge; or if you have the white case, cut groves in the case where the wires interfere with its closure.

You are now ready to try the modified cartridge. If you have the CARSTBUSTER disk, this is a good check. Take one of your ASTARISOFT or similir carstridges and store it on disk. Read it back to the RAM in the E/A cartridge. Quit then press enter to see that you now have 3 choices, the last being the name of the cartridge that you saved. Turn the system off or remove the cartridge and reinsert it to see if the 3rd choice is still there. If it is the modification is complete and you now have an added 8K of battery backed-up RAM.

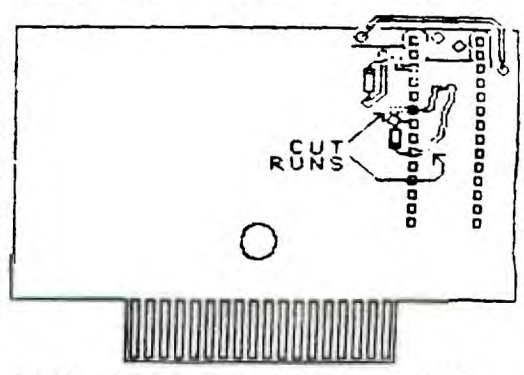
As with my other articles, all drawings and schemetics were done with GRAPHX. If you need more information on this excellent graphics program, contact Helene.



SCHEMATIC OF ADDED PARTS



TOP-SIDE OF MODIFIED BOARD



BACK-SIDE OF CIRCUIT BOARD

The following is the latest from the club library. For your copy or more information, contact Andy, Cecil or Gary. (Note the new RLE files, Star Trek and Disney series have been added to the library, as well as more extensive RLE documents, which are printed in this newsletter -Editor)

9T9 DISK OF THE MONTH Vol 87-02

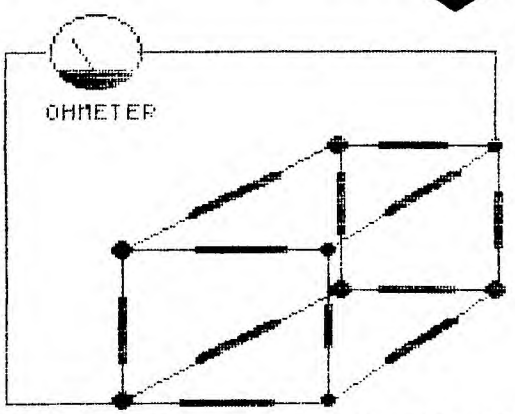
- Most Programs require XB, 32K and a PIO Printer -

File	Size	type	Runs	Discription
9T9LOGO	B 13	Prg	XB	9T9 Logo for letterhead use with Assembly Dump
ASSDUMP	B 3	Prg	XB	Part of AssDump, also requires DSRLNK DUMPIO Run a Basic or XB program when you see a screen you want to print break the program and insert a line RUN "DSK1.ASSDUMP" at the breakpoint. Run the program again insuring this disk is in DSK1.
BOMB	F 14	Prg	EA/LDR	Assem game, catch the bomb in a pail of water.
BREAKTHRU	F 13	Prg	EA/LDR	Assem game of Breakout on the TI. Fairwair
BUGSBUNNY	B 41	File		MAXRLE File
CANTINA	F 8	Prg	XB	Graphics for the STARWARS Cantina Band Song
CANTINA1	F 41	Prg	XB	Music for above
CARTOON	B 21	File		MAXRLE File
DIR	F 2	Prg	XB	A very small program to show DSK1. contents.
DSRLNK	B 7	File		Part of Assembly Dump
DUMPIO	B 10	File		Part of Assembly Dump
ET	B 8	Prg	XB	A demo program for Assembly Dump
FTPWCK_P	B 25	File		MAXRLE or TI Artist File
GHOST	F 41	Prg	XB	Music Ghost Busters theme.
GODFATHER	F 37	Prg	XB	Music Godfather movie theme.
GORILLA_P	B 25	File		MAXRLE or TI Artist File
GRAPHPAPER	F 10	Prg	XB	Utility to print graphpaper Fairware.
JOYST*TEST	F 11	Prg	XB	A small utility to test joysticks
LOAD	F 12	Prg	XB	Loader with built in assem. routines to allow loading standard program image ass. programs
MAXRLE	F 25	Prg	EA/LDR	A very useful utility to load, manipulate, and display graphic files (RLE files can be DV80, DF128, Graphix, or TI-Artist) See Doc file on Back. We have many more files in TrIware.
NASAMAP	B 50	File		MAXRLE File
PIRATE_P	B 25	File		MAXRLE or TI-Artist File
RLE/DOCS	B 5	File		MAXRLE docs.
SAFARI	F 21	Prg	XB	Generate wordsearch puzzles to printer or screen
SCHEDULE	F 70	Prg	XB	Generates a monthly schedule calendar. Your printer must be online.
SCROOGE	B 39	File		MAXRLE file.
TINY*CAL	F 9	Prg	XB	Prints a tiny calendar. We have many more calendar programs in TrIware.
TISORT	B 33	Prg	EA	A DV80 soft utility for sorting lists. Also runs from TI-Writer. Docs on back
TISORT/DOC	B 32	File		Docs for above.
WORLD	F 40	Prg	XB	World Map lets you plot cities and check distances. Several cities are already plotted (Try some capitals e.g. London, New York etc.)

The following was recently downloaded from the Glankonian BBS.

OHMS? P Sent by: KEITH HEFFER
Description:
PRINT THE PICTURE OUT ON TI/ARTIST AND SEE IF YOU CAN FIGURE OUT THE ANSWER. I DON'T KNOW THE ANSWER. PLEASE PUT IT ON THIS BOARD IN THE MESSAGE SECTION IF YOU CAN SOLVE IT. K HEFFER.

-see below-



12 RESISTORS ARE IN THE CIRCUIT.
EACH RESISTOR IS 1 OHM. IN VALUE.
WHAT WILL THE OHMETER READ ?.