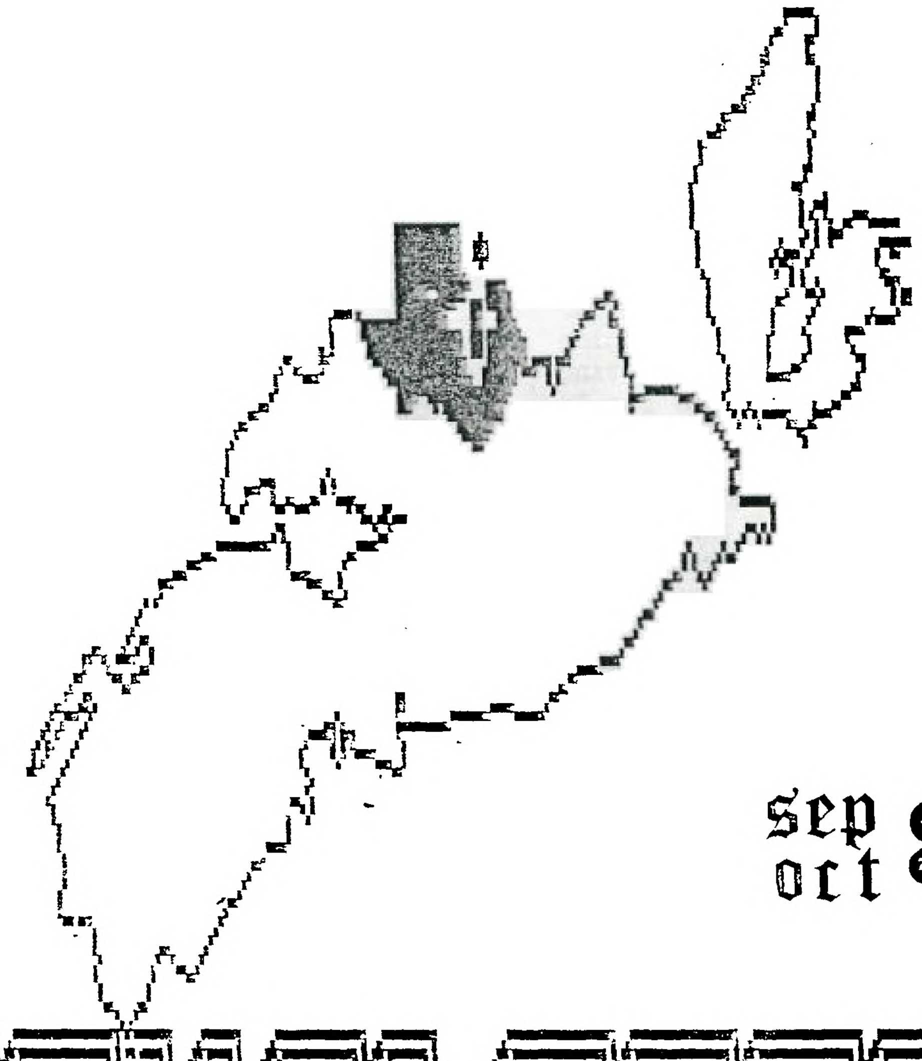
9609/8610 TINS

Rec'd. 087
87·1·25

TINS



sep oct 86

NEWSLETTER

TINS Newsletter

TINS Newsletter is published on a monthly (or bi-monthly basis if there isn't enough to print),as the means of communicating ideas obtained from solicited sources to the general membership. Material appearing in this newsletter is copyrighted by compliance with federal regulations for basic copyright protection. The effective date of copyright is the 1st day of the month appearing on the edition.

Articles and programmes appearing in this newsletter are, to the best of our knowledge, original works except where indicated by the inclusion of the source. Arrangements have been made with other newsletter staffs and magazines for permission to reprint certain material. These articles are accepted on trust and the newsletter accepts no responsibility for searching the right-to-print of the originating periodical. Errors or omissions should be communicated to the editor as soon as possible.

Groups wishing to include material from these Newsletters in their club newsletters may do so providing the author and this source are mentioned. Other use of the material is subject to release by the editorial staff.

Space for advertising is available to merchants wishing to display their wares. Full page $25, lesser sections at appropriately reduced rates (payable to TINS). All commercial ads must reach the editor in pre-prepared, ready to print form, not later than the 1st of the month in which the ad is to appear. The Newsletter is on sale to members for $1.00 per copy, non-members $1.50. The price of each issue is solely to defray publication costs.

Back issues of the Newsletter are available on written request from the editor (Jan-Dec 86, 13 issues=$15) (individual issues at $1.50 each). All queries and newsletters should be forwarded to the address below, other correspondence should be directed to the Club at PO Box 3381, Dartmouth East, N.S. B2W 5G5.

Editor TINS Newsletter
321 Iony Hill
Lr. Sackville, NS
B4E 1M6

Testing the transliterate, underline, overstrike, required spaces, and alternate input functions of TI/Writer

by Frank Geitzler

The following sample file was used to test the effect of alternate input on transliteration and underlining. The character '&' and the ascii code 38 were input for variables 1 and 2, and the character '\' and the ascii code 92 were input for variables 3 and 4. Tests one, two, three, four, nine and ten produced underlined characters. Tests five and six printed one and two backslashes, while tests nine and ten printed two and four '&'-signs. These tests determined that the following rules aply when using alternate input and transliteration:

1. Text is included one variable at a time and printed, and the character '&' input as a single variable will cause underlining even if the variable is repeated. A similar test using the character '@' produced similar results for overstriking.
2. The sequence of substitution appears to be the inclusion of alternate input, selection of underscore (or overstrike), and then transliteration. A single '&' cannot be successfully transliterated to a '&&'.
3. In another test, it was found that the character '^' can be transliterated to itself. That is, '.TL 94:94' will print '^' as the character '^', rather than ' '. Thus, for the required space character '^', transliteration occurs prior to checking for double characters, whereas for the '&' and '@' symbols,, the reverse is true.

Similar tests, in which actual values were substituted in the file in place of the alternate input values shown below, produced slightly different results, but those results confirmed the above findings. In those similar tests, tests one, three, and nine produced underlined characters, while tests two, and five produced a single '&' character, tests four, six, and seven produced a pair of '&' characters, and test eight produced four '&' characters. Test ten produced a single backslash character, which confirms rule two above: the pair of '&' characters was converted to a single '&' character (selection of overstrike) and then transliterated to a backslash.

(CONTINUED INSIDE BACK COVER)

SEATTLE TI CONVENTION
Randall Ainsworth

Before I begin and tell about what happened, who was there, and what was on display, I should note that I had reservations about the success of this event. I didn't know if they could pull it off with such a short time to plan. I was unable to attend the President's banquet on Friday night, but various User Group Presidents gave short speeches as well as honored guests such as C Regena, Lou Phillips, Craig Miller, and others.

Well, comparisons to the LA Faire were inevitable and this TI event seemed to attract at least as large of a crowd. From a vendor's standpoint, it was clearly more successful than LA. In LA, the crowd seemed to be more "lookers", but in Seattle they were definitely buying...everything from inexpensive console overlay strips to $200+ peripheral cards.

There were 9 vendors and numerous user groups from the Northwest represented. It was great to see some familiar faces of those I had met in LA as well as meet those who had called my BBS (but I had never met). The facilities at the Sea-Tac Holiday Inn were well suited to such an event, but I would recommend a larger place for next year.

I couldn't begin to list all of the User Groups in attendance, but the commercial vendors were C Regena (of Compute fame), Queen Anne Computer Shop, Bits Chips, Millers Graphics, Compuserve, GEnie, Asgard Software, T.A.P.E., Myarc, DISConsulting to name the ones I can remember. User groups included the LA 99'ers, Western Washington UG, TICTAC, Puget Sound 99'ers, and PUNN to name a few.

The PUNN UG from Portland, Oregon had two TI's hardwired and they demo'ed their BBS software at 9600 baud. In walking around, I saw four 99/4's and a complete TI stand-alone system.

One of the Canadian UG's had an 8 1/2x11 plastic sheet with console overlay strips they were selling for only one dollar. They also had copies of DM1000 Ver. 3.3.

There were demos, and product announcements by C Regena, Millers Graphics, T.A.P.E., Myarc, and Asgard software.

OK, let's get down to the particulars...where do I start?

Jim Horne had a bit of a problem connecting with CIS, but once online he was able to demonstrate the various features of CIS. He also represented Disk Only Software with many of Asgard's packages, back issues of MICROpendium and the Smart Programmer, and he also was selling the MBP clock cards. I understand that DOS has taken over the sales of the clock card and that should make support for that piece of hardware lots better.

The T.A.P.E. booth seemed to be packed with people throughout the day so it was difficult for me to talk with Franz Wagenbach. T.A.P.E. is the American importer for the Mechatronic products and their version of Extended Basic was there. Their mouse hardware was being demonstrated along with the GRAM-KARTE. Franz had the 80 column unit at the show, but was unable to show it due to lack of an RGB monitor.

The 80 column unit plugs into the side of the console and is about the same size as their 128K stand-alone (which was also there). Franz tells me that they are trying to get enough orders to make some FORTH cartridges, so if that sounds like something you're interested in, let T.A.P.E. know. Price of the 80 column box is a little over $200. He took the top off the case and the board looked pretty clean and well thought out in design.

The head sysop of the TI Roundtable on GEnie, Scott Darling, was online all day demonstrating the GEnie system. He handed out lots of brochures explaining the benefits of the electronic service.

Chris Bobbitt of Asgard presented his full line of software. He also had a ten-key on his demo machine which was hardwired into the console. He said that he had been making them on a custom basis for a few months, but it was not economical to mass produce them. If you're really interested, you may want to contact Chris.

What about Myarc? Well, they had a large space with a demo of XB II running. The demo was done by Chris Faherty and was quite impressive. I understand that the remaining bugs have been worked out of the XB II software and it looked pretty hot.

There were a couple of Geneves at the show. One was in the possession of Chris Bobbitt. His was one of the older models. (For those of you who don't know, the initial runs of Geneve circuit boards were made with the bus backwards. A couple of people beta testing them ruined their systems this way and a card was added to the bottom to reverse the wiring.)

Lou Phillips had a Geneve running TI-Writer. It was real strange to see the program in 80 columns. The P-Box was open and all you could see was the Geneve card and the controller. The keyboard looked IBM-ish and felt OK to the touch. I didn't see any other software running on it, but TI-Writer looked good. The commands from command mode were MS-DOS-ish in nature.

Lou says that the units will be ready to ship in a month, but it will probably be more like two. They are waiting for a big custom chip from Mitsubishi which will replace a cluster of chips presently being used. Since they are having a small run, the customers needing larger quantities of custom chips will get their orders first.

The 9640 seems to perform as advertised, but I am told that the DOS is still not fully completed. The final price is still not firm, but expect it to be in the neighborhood of $450-495.

In private, Lou said that the machine would be available in time for the Christmas season.

The big news of the day came from Craig Miller. Before I get into that, the PROM set for the CorComp card should be available soon. The hangup is that they don't work with the CorComp RS232. That should be worked out by the time this is published. Craig had a set in his demo system and the title screen says MG DOS above the top color bar.

Sometime before the end of the year, Millers Graphics will release some software that is geared toward a more general market than previous software efforts. Also, there will be an unannounced software package for the IBM out in December with an accounting package and a home budget program following in January (also for the IBM).

Here's the big news.....Craig could not be more specific and was only allowed to release certain information on this subject, but this is as much as can be told at this time.

Sometime in January, a large manufacturer will announce a peripheral for the TI which will allow us to run MS-DOS software. In addition, you will be able to use any IBM card on the market as well as add an 8087 (math co-processor).

The price cannot yet be released, but Craig assured the crowd that it would be affordable. The manufacturer is gearing up for production now so that it will not be a "custom" unit like the GRAM KRACKER. It will be available shortly after the company's announcement. The processor is an 8088 and does not require a P-Box. An optional keyboard will be available. The unit will come "ready-to-use". The specific interface to the TI is still a guarded secret at this time.

That means we can run Lotus, Flight Simulator, Wordstar, and any of the IBM programs out there.

While this opens up loads of possibilities, there are several concerns that need to be addressed. First, who will be stimulated to develop TI software once this unit is available? Second, once you've run Lotus, who is going to go back to TI programs?

My feeling is that this is a natural progression for the TI community. The 99/4A is not going to continue on forever and Genève will not make a very large dent (in my opinion). The ability to co-exist with the PC market not only expands the capability of our machines, but puts us in touch with the real computing world.

During the past few weeks, I have been seriously considering leaving the TI for the PC world. Prices of clones are way down and it is tempting to leave, but I have been reluctant to sell out. I like the TI and would miss it. With this development, I will probably stick around and have the best of both worlds.

In all, it was a very well attended event and due to it's success I know the organizers are planning one for next year.

****
And these comments from Rita Glickman:

As with the convention in L.A., Craig Miller spoke early in the afternoon and Lou Phillips spoke later in the afternoon.

Craig Miller:

Briefly, Miller announced a merger with an unnamed, well-established firm. The firm will officially announce the merger in January. Along with the merger will come the release shortly thereafter of a piece of hardware which will make the according to Miller, the unit will make the TI a PC-XT.

Some further comments from Miller:

* New hardware announced is already designed and will support all existing TI hardware
* Not Geneve compatible
* Comes with 256K expandable to 640K and up to 8 meg.
* Alternate keyboard will be available if desired.
* "In effect, [the hardware will allow] a separate machine"
* "The unit will come in such a fashion that you will not need to add anything" * Three new pieces of software will be available shortly before Christmas

* One new piece of software available shortly after the first of the year

Later that afternoon, Phillips discussed the Geneve card. Notes from that session:

*"Internally we are trying to look as much like MS-DOS as we can"
* Availability "in a good month"
* Waiting on gate array in order to release
* "Right now we're just finishing the software"
* As they develop new hardware or software enhancements they will be available to the public for a handling charge to add the upgrade to their system.
* "Next thing we do is a Lotus 1-2-3 lookalike"
* Hard disk capability release in November.

Asgard Software:

Chris Bobbitt announced the formation of a new company which is developing pieces of software for the Geneve computer.

CHICAGO TI FAIRE NEWS

The Chicago-Area TI-99/4A User's Group is proud to announce its fourth annual TI-99/4A Computer Faire. Come to our Faire and see the latest software, (Games, Educational and Business Programs) and hardware now available for the TI-99/4A, including a new version of the TI99 computer that has been developed by MYARC, Inc. and the new Powerhouse X-10 home controller from CorComp. Come and be a part of the on-going learning experience by attending the free seminars, workshops, and product demonstrations to be held during the Faire.

Last year this event drew 31 different TI vendors and over 1,700 TI owners. The following is a partial list of vendors that have so far acquired booth space at this year's Faire:

Asgard Software, Rockville, MD
C 6 Drives, Northbrook, IL
Competition Computer, Milwaukee, WI
CorComp, Inc.(Will send a vendor rep.)
Data Systems, Oxnard, CA
Databiotics, Greenfield, WI
Databionics, DiamondBar, CA
Great Lakes Software, Howell, MI
Horizon Computer Ltd., Wallbridge, OH
Hunter Electronics, Elmhurst, IL
L.L. Conner Ent., Lafayette, IN
Micro Format, Prospect Hts, IL
Myarc, Inc., Basking Ridge, NJ
PC Pursuit
Tenex, South Bend, IN
TI Forum, Compuserve
Tigercub Software, Whitehall, OH
Ryte Data, Haliburton, ONT, Canada

The Chicago-Area TI-99/4A User's Group invites you to attend this year's Faire being held on Saturday, November 1, 1986, between 9:00 AM and 6:00 PM at Triton College, 2200 N. 5th Ave, River Grove, IL. There will be a $2.00 admission charge.

If you are unable to attend this year's Faire, you may still participate in this resurgence of interest in the TI-99/4A computer by joining the Chicago Area TI-99/4A User's Group. For further info contact: The Chicago Area TI-99/4A User's Group, Attn: Membership Chairman, P.O. Box 578341, Chicago, IL 60657.

MENU SUPPORT FOR FORTH
by
Dennis Misener

This is the beginning of what I hope is a regular TINS FORTH column. As is the usual custom, I hereby solicit (a polite word for *beg*) your input: for without it, this column will surely fade into oblivion.

With pleading out of the way, we can now proceed to the sermon. How many of you have not yet tried FORTH? Hmmm; just as I thought... TOO MANY! What seems to be the problem? I know, you've probably heard:

- FORTH is WIERD
- uses a reverse polish notation (a kind way to say it does everything backwards)
- is completely unreadable
- has no floating point
- but does have a cult type following

Did you also hear that FORTH:

- is inexpensive (a version of TI-FORTH is available for under $3.00, actually $2.99 for a limited time only from the TINS library: see Ron Weagle for more details)
- is extensible (you can change the parts you don't like/understand)
- is fast (often 20 times faster that good old BASIC)
- generates compact code (much smaller than other languages)
- is very transportable (with an implementation on even the most obscure micro)
- is structured with IF THEN ELSE, BEGIN REPEAT, CASE ENDCASE, DO LOOP statements.
- has a simple ASSEMBLY interface for the *true hacker*
- is supported by a local users group (more about that later).

Whoops! There I go again. I better leave it at that or we'll never get back to this months sampling of actual FORTH code.

A few disclaimers are in order, first the following code was written using WYCOVE FORTH, so some modification will be required for TI-FORTH. Secondly, this code while complete, lacks any bells and whistles. The reason for this is simple: it was intended only to illustrate the power of FORTH (actually I ran out of time but...).

** WARNING **

The code that follows is not meant as a FORTH tutorial. For that I suggest you beg, borrow or steal a copy of the FORTH bible: Brodie's *Starting Forth* .

With the disclaimers aside... on with the actual code.

A primitive menu support can be implemented with the following construct:

```
BEGIN_MENU menu name
   menu word 1
   menu word 2
   ...
END_MENU
```

To call up the menu just type *menu name* and presto!!

To make our work easier, a few handy support routines will be useful. Some of these I have borrowed from the F83 standard (with so many users contributing to the FORTH's public domain, there's no end to such *goodies* up for grabs).

(See how we can give nice mnemonic names to constants. .This makes programs more readable and easier to change... *by the way text like this which is found between parentheses is just commentary*)

```
  -1 CONSTANT TRUE
   0 CONSTANT FALSE
```

(Let's define a few words to turn on and off flag type variables)

```
: OFF ( a - )
  FALSE SWAP ! ;

: ON  ( a - )
  TRUE  SWAP ! ;
```

(BOUND is a useful in confining a variable between a low and high value. We will use it to keep our menu selection within a valid range)

```
: BOUND ( n\low\high - n )
  ROT MIN MAX ;
```

(BETWEEN checks if a specified value falls within a bounded range. See ASCII for an example of its usage)

```
: BETWEEN ( n\low\high - n )
  >R OVER > SWAP R> > OR NOT ;
```

(ASCII and CONTROL are two little words used to save the poor programmer from checking ASCII tables in the back of some book).

```
: ASCII ( - n )
  BL WORD HERE 1+ C@
  STATE @ IF [COMPILE] LITERAL ENDIF ;
  IMMEDIATE

: CONTROL ( - n )
  BL WORD HERE 1+ C@  ASCII @ -
  STATE @ IF [COMPILE] LITERAL ENDIF ;
  IMMEDIATE
```

(UPC converts a potentially nasty lowercase character into its uppercase equivalent. This comes in handy when accepting user specified input in a case insensitive way.)

```
: UPC ( n - n : convert to uppercase )
  DUP ASCII a  ASCII z BETWEEN
  IF BL - ENDIF ;
```

(Even BASIC has a beep)

```
: BEEP ( - )
  CONTROL G EMIT ;
```

(CURRENT_MENU is used to maintain which menu we are currently referencing since things can get confusing when dealing with nested menus. Yes nested menus are supported !!)

```
  0 VARIABLE CURRENT_MENU
```

(MENU_ITEMS contains the number of possible selections for the current menu)

```
: MENU_ITEMS ( - # menu items )
  CURRENT_MENU @ ;
```

(MENU_START simply returns the address of [i.e. points to] the start of the routine addresses which represent our possible menu choices. We add two to skip over our MENU_ITEMS count)

```
: MENU_START ( - ptr to 1st menu item
)
  CURRENT_MENU @ 2 + ;
```

(GOTO_LINE is very WYCOVE specific. Its sole purpose in life is to position the cursor at the begining of the specified line where the top line is line 0. This one is nice in that it works in both TEXT and GRAPHICS mode)

```
: GOTO_LINE ( n - )
  SCREEN-WIDTH @ * COLUMN-SKIP @ +
  CURSOR-POS ! ;
```

(MENU_LINE is just like GOTO_LINE except that it assumes the first 3 lines have been reserved for a header i.e. the *Select on...* and two blank lines)

```
: MENU_LINE ( n - )
  3 ( header lines ) + GOTO_LINE ;
```

(MENU_LIST is the guy who converts those routine addresses I alluded to earlier and converts them into their associated name. I tried just displaying the hex addresses but people complained. Picky, Picky, Picky!!)

```
: MENU_LIST ( - )
  CLS ." Select one of the following:
"
  CR CR  MENU_START
  MENU_ITEMS @  0
  DO
     DUP @ 2+ NFA ." * " ID. CR  2+
  LOOP DROP ;
```

(The following is just a fabricated example to use the fancy ASCII word we defined. Remember... Want not...waste not)

```
  ASCII E CONSTANT UP
  ASCII X CONSTANT DOWN
```

(MENU_SELECT is used to track the goings on of the user typing at the keyboard. Only three keys are enabled: the E-up, X-down and the SPACE-select. All other keys cause a charming BEEP. Its worth noting the nested structures: a CASE within a BEGIN WHILE REPEAT. Lets see you try that in BASIC)

```
: MENU_SELECT
  0  ( start at first entry )
  BEGIN
  0  MENU_ITEMS @ 1-  BOUND
  DUP MENU_LINE
  KEY UPC DUP BL -
  WHILE ( selection not made )
     CASE
        UP   OF 1- ENDOF
        DOWN OF 1+ ENDOF
       ( else ) BEEP
     ENDCASE
  REPEAT
  DROP ( blank key )
  CURRENT_MENU @ >R
  2* MENU_START + @ EXECUTE
  R> CURRENT_MENU ! ;
```

(MENU_EXIT is a flag variable used to communicate when we are finished with the current menu level)

```
  FALSE VARIABLE MENU_EXIT
```

(This is one of those neat BUILDS DOES words you've read so much about. What it actually does is generate a table of routine addresses prefixed with the number of such addresses. When a word created by this *definer* is executed, it displays the *word name* associated with each such address and then allows the user to select one by using the up/down arrow keys to position the cursor and the SPACE bar to select the entry. What could be simpler?)

```
: BEGIN_MENU
```

```
<BUILDS
   HERE CURRENT_MENU !
   0 , ( max entries )
   ] SMUDGE
DOES>
   CURRENT_MENU !
   MENU_EXIT OFF
   BEGIN
      MENU_LIST
      MENU_SELECT
      MENU_EXIT @
   UNTIL
   CLS ;
```

(Previous_Menu just sets a flag to let the LOOP in BEGIN_MENU that is time to exit the current level)

```
: Previous_Menu
  MENU_EXIT ON ;
```

(END_MENU is responsible for adding the last *Previous_Menu* address and calculating the number of menu entries... very useful if you are like me and can't count past six)

```
: END_MENU
  COMPILE Previous_Menu
  HERE CURRENT_MENU @  - 2 / 1 -
  MENU_ITEMS !
  [COMPILE] ; ; IMMEDIATE
```

(The following definitions are nothing more than a simple test routine to show that the above definitions do anything at all... a very poor reason for existence if I do say so)

(PAUSE flashes a polite message and waits for the user to hit a key)

```
: PAUSE
  23 GOTO_LINE
  ." Hit any key to continue..."
  KEY DROP ;
```

(The following words simply redefine the standard FORTH EDIT and LOAD screen commands. I used lowercase names since I find them more attractive in menus)

```
: Edit ( current screen )
  SCR @ EDIT ;

: Load ( current screen )
  SCR @ LOAD
```

(RUN is just a tricky way to execute the last word defined... see your local FORTH guru if you want to know how it works)

```
: RUN
  CLS
  CONTEXT @ @ PFA CFA EXECUTE PAUSE ;
```

(And now for the great finale... dah...dah!!!)

```
BEGIN_MENU DEMO
   Edit
   Load
   Run
END_MENU

DEMO ( auto start )
```

Well, that wasn't so bad was it? The next time perhaps we will check out some shorter FORTH examples in order that we might encourage a few more of you BASIC programmers out there to try your hand at the wonderful world of FORTH.

Until then if you have any questions, please feel free to drop me a line in care of the TINS newsletter, or better still why not show up at the local FORTH users group meeting on the fourth day (*Wednesday*) of the fourth week of each month at DYMAXION RESEARCH at 7:30. There is no fee and new members are welcome.

Recovering a Damaged Disk Directory
Frank Geitzler

Several nights ago I finished editing an article for TINS, and hastily shut down my system and went to bed. The next day, I loaded in a quick catalog program to find out what file name I had used for the article. The catalog program promptly said:

```
I/O error 26 in 190
```

After several attempts to run the basic catalog program, which was failing to read sector 0 of the diskette, I gave up and decided to try the 'SD' function of FUNLWRITER, since I wanted to use that package anyway. FUNLWRITER responded with:

```
DISK ERROR has occurred -
Press ENTER for Editor
```

I began to get desperate. I loaded up Disk Manager 1000, and requested a catalog. It answered with:

```
DISK ERROR
Device Error
```

As a last resort, I put in Millers Graphics ADVANCED DIAGNOSTICS, and told it to provide a Disk Directory. It did so, showing that I had eight files on the disk. I recognized that five of those files were TI-Writer files, and three were program files. Then I selected the Check Disk function from ADVANCED DIAGNOSTICS and was rewarded with:

```
Crc error on a Read
```

Edit Sector 0 produced the same result, but I was able to edit the remaining sectors of the directory (sectors 1 through 9). Now I was getting somewhere. I might not have to type in that whole article again, nor that program with 10 lines of sprite definitions.

Here is how I managed to recover every file on the diskette, using two simple tools: ADVANCED DIAGNOSTICS, and DISK MANAGER 1000. There are, of course, other ways to do so, but the following method did not require that I write anything on the damaged disk, avoiding possible further damage. In fact, I put a write protect tab on the damaged disk before I began.

1. I formatted and verified two diskettes, to make sure that they had no bad sectors, and that they had empty directories.

2. I copied sectors 1 through 9 from the damaged diskette to one of the newly formatted diskettes. This was made easier with two drives, but could have been caried out with one, by swapping diskettes in place of the 'SD n' (Set Drive to disk n) command. The actual sequence of commands, with the damaged diskette in drive 2 and the new diskette in drive 1, was:
SD 2 ES 1 (Set Drive 2 Edit Sector 1) and press Function 9 to return to command mode, then
SD 1 WS 1 (Set Drive 1 Write Sector 1). This sequence was repeated 9 times, editing (and writing) sectors 1 through 9. A similar process could have been followed if the bad sector had been in sectors 2 through 9, but it would have been necessary to avoid editing the bad sector, and to write a phony directory entry in its' place. For example, if sector 5 were bad, sectors 1 through 4 would be

copied as above, then sector 4 edited again, a change made to the entry so that it would be unique (change the filename to 'BADSECTOR', for example), and sector 4 would be written out as sector 5 (WS 5). I now had a disk which had a good directory, and which matched as nearly as possible the damaged directory.

3. I put a write protect tab on my new directory disk, and put it in drive 2. I loaded up Disk Manager 1000, and selected File Copy, from drive 2. It happily read in my newly created directory, and said that only 2 sectors of the 360 sectors on the disk were used. No problem -that information was on sector 0, which represented an empty disk.

4. I removed the directory diskette from drive 2, flipped it to the unused side, and put it in drive 1, to be used as the target for the copy. I put the damaged diskette in drive 2, and selected every file for copying. Disk Manager 1000 performed as intended, and I had a new diskette, with all my files intact. I quickly put a write protect tab on that diskette, loaded each program file and ran it, loaded FUNLWRITER and checked each of the five text files, and breathed a huge sigh of relief.

5. Before I forgot, I reformatted my directory diskette again, to get rid of the phony directory entries, and discarded the disk which was the cause of the problem.

How did it happen? I have two possible causes. First, the diskette which went bad had already showed up with several bad sectors, but I used it as a work diskette anyway. That was probably a mistake. Second, I was in a hurry when I finished writing my article the other night, and failed to remove the diskette from drive 2 before I shut down the system. In fact, I have a feeling that I turned off the power bar first, and then the console, whereas I usually shut off the console first. Anyway, I think I learned several lessons:

a) Don't trust good data to bad diskettes.
b) Don't shut down in a big hurry.
c) Just to be safe, remove the diskette before shutting down.

T.I. Artist II
Ron Weagle

It is my understancing that some members have been having some trouble with the use of slides in the TI Artist II package. This is a short article to explain just how simple they are to use.

For the benifit of readers who do not have this package I will explain that slides come in the enhancement package of TI Artist II. They consist of a series of small designs (electrical symbles encluded with the package) which apppear around the primater of the enhancement screen. By pressing 'S' on the keyboard or the firebutton with the curcer on the block marked slides once into the enhanced portion of the package you are supplied with a menu with which to:

1-Define a slide.
2-Erase slides.
3-Rotate slides.
4-Load Slides file.
5-Save slides file.
6-Load an instance.
7-Save an instance.

I will not get into Instances in this article.

Lets get started, 1st of all to use a slide simple place the cursor on the desired symble or into the block it is contained in and press the fire button. The cursor will appear on the drawing surface. Press the fire button and where ever the curser is it will draw an immage of the symble. You may deposite as many as you like until the space bar is pressed returning you to the main menu.

The slides menu:

Define a slide-simply press #1 while your cursor is over the box you wish to define. Then a box appears on the screen you move it over the figure you previously defined and want to form into the slide. Press the fire button and the slide menu appears with the figure defined in the choosen box.

Erase a slide, simply press 2 place the cursor over the box containing the slide you wish to erase and press the fire button. The slide will be gone. The box may then be filled with a new slide.

Rotate a Slide, press 3 from the menu and place the cursor on the slide you wish to rotate. Every time you press the fire button the slide will rotate 90 degrees.

Load Slide Files, Press 4 from the menu and the screen will ask you to name a file previously created as a slide file. Once it is loaded the symbles it contained will replace the old slides for that session.

Save Slide Files, Press 5 on the menu and a file name will be requested. type in a file name and it defaults to drive 2 or redefine as required and any new slides files you created will be saved by this name for later reuse.

I hope this information will be of some use to any one having dificulty with this feature of what appears to be a very worth while package . I would recommend buying it.

9938 Video Chip
[Courtesy of Ryte Data]

One of the most exciting things about the Myarc machine(and the long awaited 80 column display unit) is the Yamaha V9938 Video Display Procesor. This LSI (large scale integration) chip (a 64 pin dual in-line package) is fully software compatible with the old 9918A contained in the 99/4A.

We [Ryte Data] have chip samples, spec books and programming guides here for the 9938. This is one hot chip.

Seven, yes seven, different modes; mouse and light pen inputs, up to 192k of Video RAM, 512 colours, superior bit map graphics, 512x400 pixel resolution and other interesting features: logical operations, up to 16 sprites per line, full buss speed operation, etc.

Okay, what does all this mean? How does advanced HIGH resolution bit-mapped graphics sound? You can programme each and every pixel (512x400) in a different colour. With the supported 192k (128k main and 64k expansion) of video RAM, you can do page switching where a full page of display is put on the screen in less than 1/30th of a second. When the display is shut off or idle, logical operations can be programmed using the on-chip logic. With fairly standard programming the chip can perform operations such as high speed move CPU to VRAM, VRAM to VRAM, logical move CPU to VRAM, logical VRAM to CPU etc. Other commands will search for border colours, draw lines and points in video or expansion RAM.

Interfacing the 9938 chip to the TI 99/4A has proven to be a non-trivial task. In their design, TI laid out the video interface to acommodate users with TV sets or composite monitors, rather than allowing different video displays to be implemented.

Apparently a few other individuals have attempted to interface the 9938, usually on a PE Box card, to the 99/4A without success. The approach which has worked has been a combination of using the I/O port connectors and a connector to the 9918A video chip on the 4A motherboard. This has the added advantage of being useful for all owners from console and cassette to full scale expansion systems.

The approach which we are licensing [Ryte Data] is designed to fit inside the console without a stand-alone case on the side. This will prove to be easier and less expensive. The modifications required will be simple to accomplish for average owners. Those who do not wish to perform such modifications will be offered the upgrade service.

We would like to see letters from all of you to Ryte Data [address below] discussing this, stating your preferences and demonstrating the viability of this idea.

Ryte Data
Box 210 Mountain St.,
Haliburton, Ont, KOM 1S0
Canada.

[mention you saw it here]

```
.CO EXPERIMENT WITH VARIABLE INPUT

.DF 1:TEST SYMBOL
.DF 2:TEST ASCII VALUE
.DF 3:REPLACEMENT SYMBOL
.DF 4:REPLACEMENT ASCII VALUE

THIS IS A TEST WITH THE '*1*-SIGN'.
THIS IS A SECOND TEST WITH THE '*1**1*-SIGN'

.CO REPLACE ONE '&' SIGN WITH TWO '&' SIGNS
.TL *2*:*2*,*2*

THIS IS A THIRD TEST WITH THE '*1*-SIGN'.
THIS IS A FOURTH TEST WITH THE '*1**1*-SIGN'

.CO REPLACE ONE BACKSLASH ('\')-SIGN WITH ONE '&' SIGN
.TL *4*:*2*

THIS IS A FIFTH TEST WITH THE '*3*-SIGN'.
THIS IS A SIXTH TEST WITH THE '*3**3*-SIGN'

.CO REPLACE ONE '\'-SIGN WITH TWO '&'-SIGNS
.TL *4*:*2*,*2*

THIS IS A SEVENTH TEST WITH THE '*3*-SIGN'.
THIS IS A EIGHTH TEST WITH THE '*3**3*-SIGN'

.CO REPLACE ONE '&'-SIGN WITH ONE '\'-SIGN
.TL *2*:*4*

THIS IS A NINETH TEST WITH THE '*1*-SIGN'.
THIS IS A TENTH TEST WITH THE '*1**1*-SIGN'
```

Connecting to Atari Joysticks

by Brian Bourbonniere

The joystick port on the side of the TI is similar to that of an Atari type joystick. However, an adapter is required to use Atari joysticks with the TI as the pins in the plug are wired differently. There are at least two commercial adapters available, but if you have the required parts on hand it may be cheaper to construct your own.

Parts list:

2 strands of 6 conductor wire
2 9 postion Sub-D male plugs
1 9 position Sub-D female plug
3 hoods for the above connectors(if not included

(If you can get a set of TI joysticks that have the plug and cable in good condition, but have a handle broken off, you've got most of the parts and only have to wire the two strands.)

The plugs are numbered:

```
 -------------
\ 1 2 3 4 5 /
 \ 6 7 8 9 /
  ---------
```

TI joysticks		Atari-compatibles	
pin #	use	pin #	use
1	n/a	1	n/a
2	joystick #	2	right
3	up	3	left
4	button	4	down
5	left	5	up
6	n/a	6	n/a
7	joystick #	7	common
8	down	8	n/a
9	right	9	button

wire the pins:

TI console	Atari #1	Atari #2
2	n/a	7
3	5	5
4	9	9
5	3	3
7	7	n/a
8	4	4
9	2	2

(Be careful not to form bridges when you solder as these could cause strange results and odd variables to be returned to the console port.)

This arrangement should work for all Atari/Commodore compatible joysticks.

Users are cautioned against any "home brew" units they connect to thier console. Neither the author nor TINS can be held resposible for any damages caused by the information presented in this article.

TI 99/4A ERROR CODE REFERENCE CHART

The following error code listings were printed in the February, 1986 issue of HOCUS, Newsletter of the Milwaukee Area 99/4 User Group of Wauwatosa, Wisconsin.

EXTENDED BASIC

10	Numeric Overflow
14	Syntax Error
16	Illegal after Sbrtn
19	Name too long
20	Unrecognized Char
24	$/# Mismatch
28	Improperly used name
36	Image error
39	Memory Full
40	Stack Overflow
43	NEXT without FOR
44	FOR-NEXT nesting
47	Must be in Sbrtn
48	Recursive Sbrtn CALL
49	Missing SUBEND
51	RETURN without GOSUB
54	String Truncated
56	Speech $ too long
57	Bad Subscript
60	Line not found
61	Bad Line #
62	Line too long
67	Can't CONtinue
69	Command illegal in prgrm
70	Only legal in prgrm
74	Bad Argument
78	No program present
79	Bad value
80	Nil
81	Incorrect argument list
82	Nil
83	Input Error
84	Data Error
97	Protection Violation
109	File Error
130	I/O Error
135	Sbrtn not found

DISK MANAGER ERROR CODES

#:	First #	Second #
1:	OTHER	Rec not found
2:	SEEK/STEP	Cyclic Redundancy
3:	INPUT	Lost Data
4:	PRINT	Write protect
5:	NIL	Write fault
6:	NIL	No Disk Drive
7:	NIL	Invalid input
8:	NIL	
9:	Special Error Code for Comprehensive Test	

I/O ERRORS

#:	FIRST #	SECOND #
0:	OPEN	Device not found
1:	CLOSE	Write Protected
2:	INPUT	Bad Open Attribute
3:	PRINT	Invalid I/O Command
4:	RESTORE	Out of Space
5:	OLD	EOF
6:	SAVE	Device Error
7:	DELETE	File/Data Mismatch

TI BASIC ERROR CODES PERTAINING TO DISK SYSTEM

#:	FIRST #	SECOND #
0:	OPEN	Can't find specified Disk Drive
1:	CLOSE	Disk or program is Write Protected
2:	INPUT	Bad Open Attribute
3:	PRINT	Illegal Operation
4:	RESTORE	Disk full or too many files opened
5:	OLD	Attempt to read past EOF
6:	SAVE	Device Error
7:	DELETE	File Error
9:	[illegible]	

EDITOR/ASSEMBLER ERROR CODES

X.B. ERROR EQUATES

ERRNO	)0200	2	Numeric Overflow
ERRSYN	)0300	3	Syntax Error
ERRIBS	)0400	4	Ill. after Sbprgm
ERRNQS	)0500	5	Unmatched Quotes
ERRNTL	)0600	6	Name too long
ERRSNM	)0700	7	$/# Mismatch
ERROBE	)0800	8	Option Base Error
ERRMUV	)0900	9	Improperly used name
ERRIM	)0A00	10	Image Error
ERRMEM	)0B00	11	Memory Full
ERRSO	)0C00	12	Stack Overflow
ERRNWF	)0D00	13	NEXT without FOR
ERRFNN	)0E00	14	FOR-NEXT nesting
ERRSNS	)0F00	15	Must be in Sbprgm
ERRRSC	)1000	16	Recursive Sbprgm
ERRMS	)1100	17	Missing SUBEND
ERRRWG	)1200	18	RETURN without GOSUB
ERRST	)1300	19	String truncated
ERRRBS	)1400	20	Bad subscript
ERRSEL	)1500	21	Speech $ too long
ERRLNF	)1600	22	Line not found
ERRBLN	)1700	23	Bad line number
ERRLTL	)1800	24	Line too long
ERRCC	)1900	25	Can't Continue
ERRCIP	)1A00	26	Illegal in program
ERRGLP	)1B00	27	Only legal in program
ERRBA	)1C00	28	Bad argument
ERRNPP	)1D00	29	No program present
ERRBV	)1E00	30	Bad value
ERRIAL	)1F00	31	Incorrect argument list
ERRINP	)2000	32	Input error
ERRDAT	)2100	33	Data error
ERRFE	)2200	34	File error
ERRIO	)2400	36	I/O error
ERRSNF	)2500	37	Subprogram not found
ERRPV	)2700	39	Protection violation
ERRINU	)2800	40	Unrecognized character
WRNNO	)2900	41	Numeric overflow
WRNST	)2A00	42	String truncated
WRNNPP	)2B00	43	No program present
WRNINP	)2C00	44	Input error
WRNIO	)2D00	45	I/O error

LOADER ERROR CODES

0-7	Standard I/O
8	Memory overflow
9	Not used
10	Illegal tag
11	Checksum error
12	Unresolved ref.

TI WRITER ERROR CODES

0	- Indicates Disk Controller not on; OR: Diskette not Initialized
6	- No Disk in Drive; OR: Is upside down; OR: Drive is not turned on
7	- No Disk in Drive
00	- Illegal use of LoadF, PrintF; OR: Error in using those commands
02	- No file in Diskette with Filename used
04	- Disk is full
06	- PrintF Command in progress was interrupted; OR: Disk Door was opened while Red Light was on.
07	- Invalid Filename (I.E. Name too long or using invalid characters)
15	- Invalid Disk Drive Number, or Device

EXECUTION ERRORS

0-7	Standard I/O
08	Memory Full
09	Incorrect Statement
0A	Illegal Tag
0B	Checksum Error
0C	Dup. Definition
0D	Unresolved Ref.
0E	Incorrect Statement
0F	Program not found
10	Incorrect Statement
11	Bad Name
12	Can't CONtinue
13	Bad Value
14	Number too big
15	String-Number
16	Bad Argument
17	Bad Subscript
18	Name Conflict
19	Can't do that
1A	Bad Line Number
1B	FOR NEXT Error
1C	I/O Error
1D	File Error
1E	Input Error
1F	Data Error
20	Line too long
21	Memory Full
22-	Unknown Error Code

RS232c ERRORS

OPEN:	00	Device cannot be opened
	02	Software Switch Error
	06	Hardware Error
INPUT:	24	Internal Data too large for buffer
	26	'CLEAR' pressed or Hardware Error
PRINT:	36	'CLEAR' pressed or Hardware Error
OLD:	50	Can't load from specified device
	52	Can't use software switch with 'OLD'
	54	Program too large to load
	56	'CLEAR' pressed or Hardware Error
SAVE:	60	Can't save to specified device
	62	See 02. Can't use with SAVE
	66	'CLEAR' pressed or Hardware Error
MISC:	43,73,83,93,	Executing Illegal Command