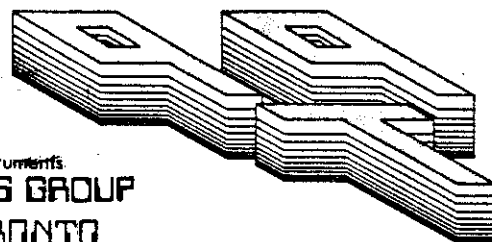


Newsletter Nine-T-Nine



MAY 1992 ISSUE

Texas Instruments
USERS GROUP
TORONTO

FOR THE TI-99/4A COMPUTER



65 Canada



FROM:
9T9 USERS GROUP
15 KERSDALE AVE.
TORONTO, ONT., M6M-1C9
CANADA

9T9 USERS GROUP

9T9 USERS GROUP EXECUTIVE COMMITTEE

PRESIDENT Steve Mickelson (657-1494)
VICE-PRESIDENT Neil Allen (236-0842)
SECRETARY/MEMBERSHIPS Randy Rossetto (469-3468)
TREASURER/OFFICER AT LARGE Cecil Chin (671-2052)

LIBRARY DIRECTORS

Gary Bowser (960-0925)
Aidy Parkinson (275-427)
Steve Findlay (416) 727-6807

NEWSLETTER EDITOR

Steve Mickelson (657-1494)

MEMBERSHIP FEES

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

All memberships are household memberships. A newsletter subscription is only for those who do not wish to attend meeting, 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 either our president, in writing, at the club address on the front cover or by phone.

The meetings are usually held on the last Wednesday of each month (exceptions are December's meeting date, usually mid-month and the months of July and August, when there are no meetings). Consult this issue of Newsletter 9T9 for the date and time of the next meeting. Meetings are usually held at Neil Allen's place, 52 Graystone Gardens, south of Bloor St., just west of Islington Ave., at 7:30 P.M. from 7:30 - 10:30 P.M.

BBS

The 9T9 Users Group supports the Toronto BBS. The TI Tower BBS # (416) 921-2731, 300/1200/2400 BPS, 24 hrs. Sysop. Gary Bowser.

MAILING ADDRESS:

9T9 Users Group, 15 Kersdale Ave., Toronto, Ontario, M6M 1C9, Canada

COMMERCIAL ADVERTISING

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

FULL PAGE (7" x 10") \$30.00
HALF PAGE (7" x 5") \$15.00

QUARTER PAGE (7" x 2 1/2") \$ 7.50
Please have your ads camera ready and paid for in advance. For more information contact the editor. Don't forget, that any member wishing to place ads, may do so free of charge as long as they are not involved in a commercial enterprise.

NEWSLETTER ARTICLES

Members are encouraged to contribute to the newsletter in the form of articles, mini programs, helpful tips, hardware modifications, jokes, cartoons and questions. Any article may be submitted in any form by mail or modem. We welcome the reprinting of any article appearing in this newsletter providing credit is given to the author and 9T9. If more information is required, call the editor. The names, 9T9, Nine-T-Nine Newsletter, 9T9 Users Group, and Nine-T-Nine Users Group are Copyright, (c) 1982-1992, by the 9T9 Users Group of Toronto, Canada. All rights reserved.

DISCLAIMER

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

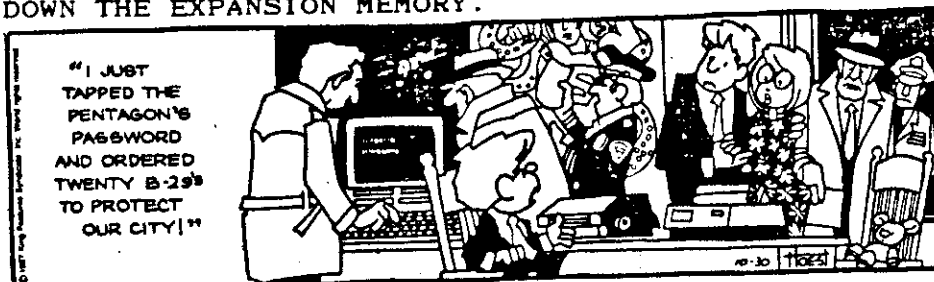
TRIUMPH OVER FUNCTION QUIT FROM TINS, ORIGINAL FROM 9900 U.G.

REMEMBER THE TURNING OFF OF FCTN QUIT? THE CALL LOAD WAS (-31806,16). HOWEVER, WHAT IF YOU DIDN'T DO THAT OR YOUR COMPUTER LOCKS UP AND YO ARE FORCED TO GO TO THE MAIN TITLE SCREEN. THERE IS HOPE. YOU REQUIRE 32K, AND E/A, MINI-MEM, OR XBASIC. YOU CAN ALSO DO THIS WITH THE CORCOMP DISK CONTROLLER CARD FROM BASIC.

THERE ARE 2 LOCATIONS IN THE PAD THAT ARE POINTERS TO THE FIRST AND LAST LINES OF A PROGRAM. THESE VALUES ARE THE ENDS OF THE LINE NUMBER TABLE. EACH ENTRY IN THE LINE NUMBER TABLE IS 4 BYTES. 2 FOR THE LINE NUMBER, AND 2 FOR THE LOCATION WHERE THAN LINE NUMBER IS IN MEMORY. THE LINE NUMBER AND THE ASSOCIATED STATEMENTS ARE IN DIFFERENT LOCATIONS IN YOUR COMPUTER.

WHEN YOU FCTN QUIT TO THE MAIN MENU ONLY THE POINTERS AND OTHER VALUES IN PAD (CPU RAM IN YOUR CONSOLE) ARE CLEARED. EVERYTHING ELSE, LINE NUMBER TABLE AND STATEMENTS ARE STILL UNTOUCHED IN TE 32K RAM. THIS IS TRUE AS LONG AS YOU DON'T POWER DOWN THE EXPANSION MEMORY.

What A Guy





TIDBITS

#59

**-By Steve Mickelson, President 9T9 Users Group
CompuServe 76545,1255; Delphi SMICKELSON; GENIE S.MICKELSON**

M.U.G. Musings:

The 1992 Edition of the TI 99/4A computer Multi-User Group Conference seemed to be a success. Attendance was fairly steady, though one vendor said sales of hardware and software were down, from the 1991 meeting. There were the regular series of speakers, from 8:00 AM to 6:00 PM.

The conference began earlier than last year, at 7:30 AM, and continued throughout the day. One vendor, remarked to me, that many people who had sold off their TI systems to buy another, returned to buy back a TI system, finding that "grass isn't always greener"!

On behalf of the Canadian contingent, of which there were many on Victoria weekend, our thanks for the hospitality and obvious hours of preparation from Charlie, Dave and the rest of the Lima group. Although I was at the conference several hours, before I discovered Crystal Software and another TI vendor tucked-away in a small alcove. However, I did not see representation by Rave, at the conference, to answer some questions regarding my PE/2 card kit. Rave did manage to make the Chicago International Fair and Lima, last year. MICROpendium, too, was conspicuous by their absence.

Myarc and company:

Lee Bendick, had on display both a TI-99/2, (using 9995 CPU and 9928 monochrome video display chip), as well as a TI-99/8, (with 9995 CPU, built-in P-Code firmware). The 99/2, was the first that I had seen, was light weight, using the hexbus peripheral system interface. The 99/8, had one of four Winchester hard-drive controllers owned by Bendick. The fourth, apparently was given to Lou Phillips, by TI, when Myarc attempted to adapt the Myarc-Winchester controller and drive for pre-production 99/8 computers. This was the second 99/8 I had seen, the other being at the (Washington) D.C. Fest, a few years back. A side-bar to the 99/8 was that the fact that Myarc had been given a pre-production 99/8, to develop hard drive and interface, and were allowed to keep the 99/8, after TI pulled out of the home computer market. Apparently, Lou Phillips, reversed-engineered the 99/8, and then approached TI, regarding getting a license for the 99/4A's operating system, for the Myarc computer, later to be called the Geneve or model 9640 computer on a card. Some of the early demonstration software used by Myarc had the 99/8 name on some of the display screens. Perhaps the 9640 gate array chip, made by Yamaha overseas, would reveal a great similarity to the 99/8 operating system.

The fact that Beery Miller has undertaken a drive, to buy-out Paul Charlton and Lou Phillips, to obtain all source codes and rights to MDOS and related software, may well indicate where Myarc stands in the TI community. The only fear I have, is like the various patches we have the MDOS object code, every writer or hacker may start producing different versions of MDOS, without proper beta testing, for compatibility, with all software and hardware configurations, once the source code becomes available. However, the relative scarcity of Geneve Assembly writers, may make my concerns unfounded.

By the way, Lee Bendick told me, that both he and Barry Boone are working on a new product for the TI-99/4A community, that has yet to become available. This "secret" product may be ready for debut in Chicago, this fall.

Last meeting:

The 9T9 Users Group had a hardware/software auction, last meeting, to raise funds for the club. Any donations of unused items to the club will be welcome. Proceeds go to 9T9 Recall, the TI reunion, tentatively scheduled, for Saturday, October 24. More information to follow, stay tuned!

That's all for this month!

SOME THINGS THAT COULD MAKE YOUR CORCOMP RS232/PIO CARD PERFORM MORE RELIABLY..... by John F. Willfont

A couple of days ago, Frank Foster came to me with an interesting problem. He had only a day earlier, purchased a FASTEXT 80 printer, a special strobe inverting cable, and on putting these together with his CORCOMP RS232/PIO card, he sat in disbelief when nothing happened. He tried a regular cable that had been used on an EPSON printer, and still no printing occurred. The printer would print selftest, however.

Since I use a CORCOMP card, I asked Frank to bring the printer, card, and cables over so we could have a look/see.

Frank was right his card/cable(s)/printer combination did nothing. Since I had his PIO controller already in the PEB, I hooked up my printer (an EPSON), to his card. It worked. It seemed that with the EPSON, that all the cables worked, and both his and my CORCOMP cards drove the printers. THIS WAS STRANGE.

I had heard that several years ago that the CORCOMP RS232 cards had undergone some improvements, and I couldn't find the information to verify what they were and what they were for. Well Frank left and I said I would call CORCOMP and investigate the problem. Problem was I got busy and about two weeks later Frank called, and alas there I was no further along than when he had dropped it off.

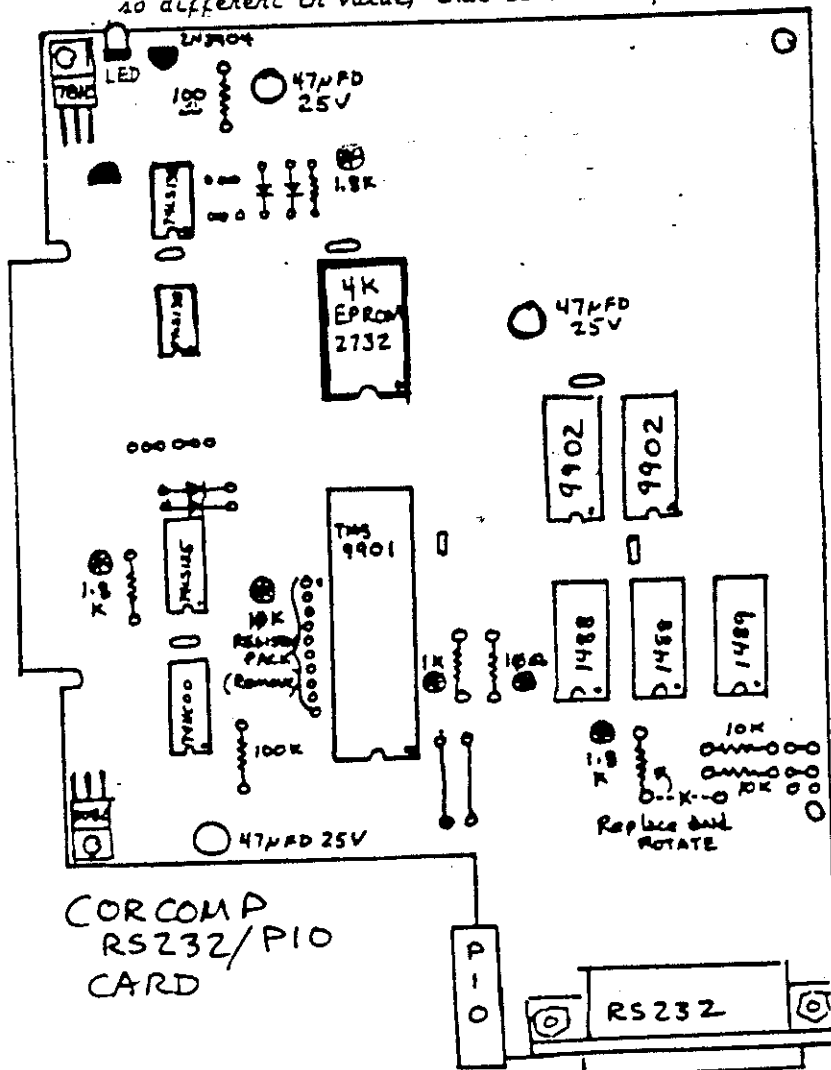
Now for the desperate part. I removed both cards from their shells, and noted that the artwork (physical board layout) was the same. His did however have physical jumpers for multiple board strapping, which mine did not. My board was the newer, and therefore had reaped the benefits of corporate cost cutting that is the plague of all good things. So now looking further, and having already verified that the ROM information was the same in his card as in mine, I started checking chip types and vendor types. There were some differences, and if nothing further would show up I would resort to changing the few of them that were different. I had heard of a problem in this area.

Well on closer investigation, I found a very astounding difference. The RESISTORS were so different in value, that it is hard for me to believe that the card could have worked.

I drew a sketch of the card and on it placed the main components for reference. Altogether five resistors & a resistor pack may be removed, and no apparent problems found, if they are replaced (in the case of the individual resistors) with the values indicated on the drawing. You will note that the resistor just above the RS232 connector is rotated to solder into a hole that exists in a trace already. The resistor pack is a 10K pack, as are the five discrete resistors that I removed. I placed a (⊗) symbol at the location where you may have to remove/replace a component.

This may fix problems that you have had with this controller. I've been using mine since early 1984, and have never had trouble with it on any type printer or cable. This is not to say that there have not been some changes to the card. You should always check with CORCOMP. They will be happy to help you. I hope that this will help someone who has had strange and hard to define problems with this card.

This is my contribution for this month, You may need (5) 1.8K 1/4 W. resistors, (1) 1K 1/4 W. resistor, and (1) 10 ohm 1/4 W. resistor.



GEN-TRI Upgrade

From : JERRY COFFEY
 To : ALL
 Subject: GEN-TRI v1.02
 Folder : B, "9640 Message Folder"

GEN-TRI VERSION 1.02 (FOR THE GENEVE) IS SHIPPING SINCE OCTOBER 3, 1991. Version 1.02 of GEN-TRI includes the spellchecker that was an integral part of the original design and has several changes in response to bug reports and user suggestions. Minor changes include: --- correcting the MACRO function of the word processor to perform as designed; --- correcting the handling of blank lines by the reformat command in WP; --- adding the ESC character to those that can be passed to the remote host in terminal mode (useful for some PC BBSs); --- adding some delay loops to the ymodem routines to offset the slow performance of some clones (direct transfers now work up to 19,200); --- temporary fix for a directory bug on very large program files; and --- improved Find and Replace functions.

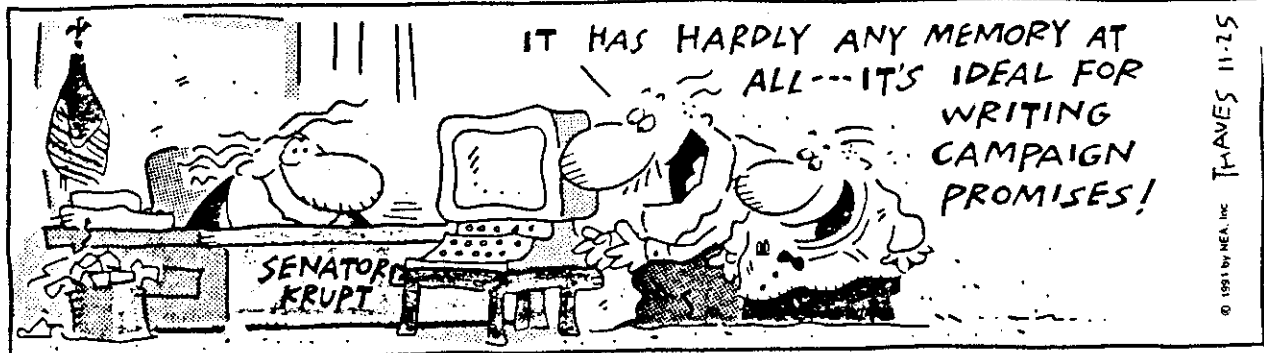
The spellchecker is a completely original approach with extremely quick response. The standard word list (or "dictionary") is about 30,000 common words written in compact form in a 718 sector file. This file can be expanded on a hard drive or 720k disk to three times this size and filled with additional words. A utility program is provided to add new words in the efficient coded form developed by Wayne Stith. The utility accepts words from the keyboard or any DV80 file, scans the dictionary for them and adds them to the dictionary if not found. If your file and dictionary are on ramdisk the dictionary can be searched in one fifth of a second (300 searches per minute). This also means that it takes only a fraction of a second to check a single word in a document you are writing or editing. If you are checking an entire document, words not found will be highlighted and then you have the option to "Ignore" (if you aren't sure of the spelling) or "Add" them to the dictionary. The basic dictionary takes up all of a DSSD disk, but Wayne will provide an abbreviated 358 sector dictionary on request.

To order send \$49.95 to Jerry Coffey 9119 Tetterton Ave. Vienna VA 22182. To upgrade your v1.0 program, send your original disk and \$1 for postage to the address above.

In either case please indicate if your system can handle 1440 sector disks (DSDD), otherwise it will be shipped on single density 720 sector disks (DSSD). SSSD disks require a special order because of the size of the dictionary.

(THERE WAS TALK OF THIS AT THE LAST WEST PENN MEETING)...

9640 BBS PHONE 901-368-0112 (RUNNING AT 300/1200/2400 BAUD 8N1)
 UGOG99BBS PHONE 714-751-4332 (RUNNING AT 300/1200 2400 BAUD 8N1)



FRANK & ERNEST

WORD PROCESSING WITH MULTIPLAN

Word processing with Multiplan? Why not? Multiplan has many advantages over TI-Writer and the Editor/Assembler Editor. For instance, Multiplan will allow you to format your document in a columnar layout and print it in condensed text, providing for a larger amount of text on a given page. In addition, Multiplan will center your text where desired, and allow for the movement of blocks of text in a much more flexible format.

Using Multiplan as a word processor does have it's drawbacks. Among these are the lack of a global editor, editing of text is a bit more difficult (you can't simply type over your text), and fast typists will have to learn to slow down a little due to the programs relatively slow processing speed.

Despite these drawbacks, however, for many applications Multiplan may be the easiest way to solve the problem at hand.

I don't propose to go into a full tutorial on the use of Multiplan, for that I would refer you to the Multiplan Manual. I realize that many people find this a formidable document, but for use as a text processor, only a general knowledge of the use of Multiplan is necessary. Therefore, in this discussion, I will merely cover what I have found to be the easiest steps to follow in setting up and using the worksheet.

Starting with an empty worksheet, your first step should be to select the OPT or OPTIONS command and turn off the recalc option. Since you will be doing no mathematical

calculations, this will eliminate the considerable delay incurred as the program searches for mathematical cells.

Next, select the FORMAT option, then DEFAULT on the sub-menu, and finally WIDTH on the next menu, and set the default column width at 30 columns. I realize that it is possible to set the width up to 32 columns, but by setting it at 30 we will later be able to widen it to 32 to allow for a buffer between columns of text.

The next setup step that is advisable is to again select the FORMAT, DEFAULT option, but this time select the CELLS option on the third menu. In the alignment column select L for Left. Remember, when Multiplan is displaying the ALPHA/VALUE prompt, hitting a number as the first character in a line will select the VALUE option rather than ALPHA. Therefore, if the first character in a line is a numeric one, you must first hit enter twice to specifically select the ALPHA command. In case you forget, however, and the only characters entered on that line are numeric ones, this will prevent them from being right justified or otherwise skewed.

The final setup step I use is to select the WINDOW option and place a border around the one open window. You may then use this border as a line length guide while typing. You may type up to but not including the column containing the right border without having the end of your text cut off.

You are now ready to begin entering your text. Start at row one, column one, and enter

one line after the other in column one. I prefer to enter all of my text in column one and format it later, since this makes it somewhat easier to move data about. Another advantage is that you don't have to worry about keeping track of where you are located on the page.

Once you have finished entering your text, you are ready to format the data into columns. Since the maximum column width on the TI printer is 132, we will divide the text into 4 equal columns of 32 characters each and have a 2 column border on the left and right margins.

Assuming we're working with one page as an example, there are two ways you can format the text. One would be to simply divide it into 54 rows per column (assuming your page length is 66), and leave whatever may be left over in the fourth column. You may also decide that you would like the columns to be of even length, in which case you would simply divide the total number of rows by four, and make each column that length.

For example, let's assume the total number of rows, when the document is formatted in one column is 200. 200 divided by 4 equals 50. We would therefore make each column 50 lines long.

To do this, we would copy from row 51 to 100, and place the copy in row 1, column 2. Next we would copy from row 101 to 150, and place the copy in row 1, column 3, and finally, we copy from row 151 to 200 and place the copy in row 1, column 4.

You now have the entire document in rows 1 through 50 and columns 1 through 4, but

you still have copies of columns 2 through 4 below row 50 in column 1. To get rid of these use the delete command. Now change the default width to 32 to provide spaces between columns.

You are now ready to print the file. To do this, first, save the file to disk. Next, exit Multiplan and select TI BASIC, then enter the following commands:

```
OPEN #1:'P10.CR' (Use double quotes)
```

```
PRINT #1:CHR$(13)
```

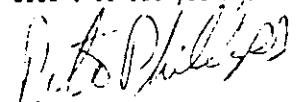
```
BYE
```

If you're printer is not connected to the parallel I/O interface, you will have to supply the proper file-name. This procedure sets up the TI printer to print in condensed text.

Next, re-enter Multiplan and select PRINT, OPTIONS. Enter your printer name in the setup field and return to the PRINT menu. Now, select MARGINS and set the left margin to 2 and change the print width to 132.

All that need be done now is to select the PRINTER command and your document should come out in 4 even columns.

I'll admit that this procedure is a bit tedious, but it is the most flexible means I know of to format text into columnar form. I have made several attempts to devise a program to translate a TI-Writer file into a Multiplan file using the symbolic link file format, but so far all of my attempts have proved to be fruitless. I'm still working on it, so if I have any success I'll let you know.



SAN ANTONIO AREA 99ERS

 * WORD PROCESSING USING *
 * EDITOR / ASSEMBLER *
 * BY PHIL BENNIS *
 * NEW HORIZONS COMPUTER CLUB *

BY NOW MOST OF YOU HAVE ALREADY DISCOVERED THAT THE E/A MODULE CAN BE USED TO PLAY SOME GREAT ARCADE STYLE GAMES, BUT DID YOU ALSO KNOW THAT THE VERY SAME E/A MODULE CAN ALSO BE USED AS A VERY CAPABLE WORD PROCESSING PROGRAM?

THIS AUTHOR JUST RECENTLY FOUND OUT ABOUT THIS CAPABILITY THANKS TO THE HELP OF JOHN CLULOW. USING HIS INSTRUCTIONS, I WILL TRY TO HELP YOU TO ALSO BE ABLE TO ENJOY THIS SUPER APPLICATION.

- 1) INSERT THE E/A MODULE AND THE E/A MASTER DISK #A. THIS DISK HAS TO BE IN DRIVE #1 TO WORK.
- 2) SELECT OPTION 2 (E/A).
- 3) AT E/A MASTER SCREEN SELECT #1 (EDIT). EDIT PROGRAM WILL LOAD.
- 4) ON NEXT SCREEN SELECT #2 (EDIT). YOUR NEXT SCREEN WILL SHOW "#E OF (VERSION 1.2)".
- 5) PRESS FUNCTION B (INSERT LINE) TO MOVE THIS HEADER DOWN 1 LINE.
- 6) PRESS FUNCTION S (LEFT ARROW) ONCE TO DISPLAY LINE NUMBERS AT LEFT OF YOUR SCREEN. THESE WILL BE NEEDED LATER IF YOU WANT TO MOVE A LINE OR ENTIRE PARAGRAPH.
- 7) IF YOU PLAN TO SUBMIT YOUR ARTICLE TO THIS NEWSLETTER, TRY TO KEEP YOUR LINE LENGTH TO ONLY 33 CHARACTERS. YOU CAN GO ALL THE WAY OVER TO 74 CHARACTERS FOR USE IN OTHER APPLICATIONS.
- 8) TYPE IN YOUR LETTER USING THE ABOVE INSTRUCTIONS. YOU MAY USE THE FUNCTION ARROW KEYS ANYTIME YOU NEED TO MAKE CORRECTIONS OR



TO MOVE THE CURSOR ANYWHERE YOU WANT.

9) TO SAVE YOUR LETTER, PRESS FUNCTION 9 (BACK) TWICE TO RETURN TO THE EDITOR SCREEN.

10) PRESS OPTION 3 (SAVE) AND YOU WILL BE ASKED IF YOU WANT TO SAVE YOUR PROGRAM IN THE VAR 80 FORMAT (Y/N). TYPE Y (YES).

11) NEXT YOU WILL BE ASKED FOR A "FILE NAME". TYPE DSK_

AND YOUR PROGRAM WILL BE SAVED TO DISK. IT IS A GOOD IDEA TO SAVE YOUR PROGRAM FROM TIME TO TIME AS YOU WRITE IT TO KEEP FROM LOSING IT ACCIDENTLY.

12) TO RETURN TO YOUR PROGRAM IF YOU WANT TO CONTINUE ENTERING YOUR PROGRAM, JUST PRESS OPTION 2 (EDIT) AND YOU ARE BACK AT THE BEGINNING OF THE PROGRAM.

13) MOVE THE CURSOR TO WHERE YOU LEFT OFF BY USING FUNCTION 4 (ROLL UP) OR FUNCTION X (DOWN ARROW).

THIS SHOULD BE ENOUGH TO GET YOU STARTED. HAVE FUN AND ENJOY THE MYSTERIES OF THIS MARVELOUS MACHINE OF OURS.

HAPPY COMPUTING
 PHIL BENNIS

**MYSTERY PROGRAM
 EXTENDED BASIC**

```

90 CALL INIT
100 FOR A=1 TO 255::CALL LOAD(-31744,-A)::NEXT A
110 FOR A=1 TO 255::CALL LOAD(-31744,A)::NEXT A
120 FOR A=1 TO 255::CALL LOAD(-31744,A)::NEXT A
130 FOR A=1 TO 255::CALL LOAD(-31744,-A)::NEXT A
140 FOR A=1 TO 255 STEP 4::CALL LOAD(-31744,-A)::NEXT A
150 CALL SOUND(100,110,0)
160 FOR B=1 TO 2
170 FOR A=1 TO 975 STEP 4::CALL LOAD(-31744,A)::NEXT A
180 FOR A=1 TO 255 STEP -5::CALL LOAD(-31744,A)::NEXT A
190 CALL SOUND(1,110,0)
200 FOR A=1 TO 255::CALL LOAD(-31744,-A)::NEXT A
210 NEXT B
220 CALL SOUND(1,110,0)
230 END
  
```

This program seems to have originated in the A9CUG (Atlanta) and comes to us from the Southwest Ninety Niners Newsletter.

TWO PROGRAMS IN ONE
BY GRAHAM HILTON ENGLAND

HAVING TWO PROGRAMS IN MEMORY AT ONCE. YOU WILL NEED THE 32K RAM AND EXTENDED BASIC.

1. SELECT EXTENDED BASIC
2. OLD THE FIRST PROGRAM. IT WILL GO INTO 32K
3. CALL PEEK(-31952,A,B,C,D).
4. PRINT A;B;C;D. NOTE DOWN THESE LINE NUMBER TABLE POINTERS.
5. MAKE 32K INVINCIBLE WITH CALL LOAD(-31868,0,0) WHICH DISABLES THE 32K MEMORY.
6. OLD THE SECOND PROGRAM. IT WILL GO INTO VDP RAM.
7. WHEN FINISHED RUNNING, NEW, AND RESTORE THE 32K RAM USING CALL LOAD(-31868,255,231).
8. RELOAD LINE NUMBER TABLE POINTERS CALL LOAD(-31952,A,B,C,D).
9. NOW RUN THE PROGRAM IN 32K.
10. YOU CAN EVEN QUIT OR BYE OR EVEN SWITCH OFF THE CONSOLE WITHOUT LOSING THE PROGRAM IN THE 32K.

NEW EDITING FOR BASIC PROGRAMS

The following article appeared in the Lehigh 99'er Computer Group Newsletter Vol. II, No. 8.

BASIC: Full screen editing is finally possible!

John Hamilton, writing a column called 99 Tips in the Central Iowa 99/4A UG "The 4A Forum", (c/o Robert Utter, 3013 E. 32nd St., Des Moines, IA 50317), is up to tip 81. We don't know about the first 80 but the last is a real humdinger.

John's crucial insight into TI's MERGE (XBASIC) command makes this program tick. What he noticed, (and TI didn't document, naturally), was that the MERGE command doesn't check for syntax on the way back in (from the disk-Ed).

In other words, IF YOUR DISK FILE IS:

1. DISPLAY, VARIABLE 163
2. EACH RECORD STARTS WITH A LINE NUMBER (followed by a space-Ed)
3. THE LAST RECORD IS HEX FFFF (CHR\$(255) twice)

. . . YOU HAVE A MERGE FORMAT FILE.

Notice that the file DOES NOT require any BASIC syntax! You could create a DISPLAY 163 file that consists of a grocery list and it will MERGE!

Why is the ability to create a MERGE file so earthshaking? Because now you can use TI WRITER or the EDITOR/ASSEMBLER to write your BASIC programs. You can have all of the features of their editors like Find String, Move, Copy, Include Files, and so on, and still be able to run the programs. You can also run a LISTED program (obviously it has to be on a storage device, not a printer). Other uses include being able to run screen dumps from the Terminal Emulator II environment.

TRANSL

VAR 80 to 163 conversion program

```
1 CALL CLEAR :: OPEN #1:"DSK1.TESTR" :: OPEN #2:"DSK1.OUTR",  
  VARIABLE 163  
2 LINPUT #1:LS :: S=POS(LS," ",1) :: ON ERROR 6 ::  
  N=VAL(SEG$(LS,1,S))  
3 ON ERROR 5 :: A=INT(N/256) :: A$=CHR$(N/A6) :: PRINT LS  
4 PRINT #2:CHR$(A);A$;CHR$(131);SEG$(LS,S+1,80);CHR$(0) ::  
  GOTO 2  
5 PRINT #2:CHR$(255);CHR$(255) :: CLOSE #2 :: END  
6 ON ERROR 5 :: RETURN 2
```



How to use TRANSL: The program expects a standard DISPLAY, VARIABLE 80 file. Each line in the program text must begin with a line number. Error trapping on lines 2 and 6 will discard any line that has no line number. The program text line numbers DO NOT have to be in order; the MERGE command will put them where they ought to go.

If you use TI WRITER, be sure to use the fixed cursor mode. Otherwise each line will have a carriage return affixed to it and you'll get a syntax error. TI WRITER will also save a TAB line if you use word-wrap mode. Keep lines shorter than 80 characters because TRANSL will clip off the remainder (it is possible to get around that, so go ahead and change it).

You can modify line 1 to open different files. As the program runs, each line is displayed to the screen. Watch for truncation on a LISTed or down-loaded file. When XBASIC returns READY, type the following:

```
KEY  
MERGE "DSK1.OUTR" (or whatever)
```

If you run the program now, nothing will happen. Each line is a tail REMark. Edit the program from the top down using FCTN X and FCTN 1. As you delete each "!" and cursor down, the XBASIC system retokenizes each line into a RUNable statement. If you get a syntax error, look first for a truncated line. Second, insure that you didn't use word-wrap mode in TI WRITER. A quick check from the XBASIC environment:

```
CALL SCREEN(14) :: FOR A=0 TO 12 :: CALL COLOR(A,16,15) ::  
NEXT A :: ACCEPT AT (4,4):A$
```

Don't answer ACCEPT, but rather Clear it with FCTN 4. Now list the program. Anything that looks like a red box is probably bad.

If you've a routine in one program, just LIST "DSK1.TESTR": (line numbers). and RUN TRANSL against it. Voila! Here's your routine in a MERGE format.

>Frederick Hawkins

[Editor's Note - I've copied line 3 of TRANSL exactly as it occurred in the SOUTHWEST NINETY-NINERS NEWSLETTER even though there is no previously declared A6 variable for the statement A\$=CHR\$(N/A6). The next newsletter said it should be N-A6. It still uses the A6 variable. You will have to figure out what it should be.]



"All he's learned so far is
— it doesn't help to hit or
kick it."

DISK DRIVES
By Jim Ness, LAUG

It's funny (at least to me), but there are lots of people who seem to know lots of stuff about their computers, and all those tiny chips, and how the bits and bytes are handled. And there seems to be next to nobody that knows anything about disk drives, and how they work. Sensing this huge gap in man's knowledge, I decided to figure out what makes them tick.

The great thing about disk drives is that they can find files buried randomly within a huge field of data, and they do it pretty fast. Actually, they can do it so fast because it's not at all random.

The mechanical concept is not all that complicated. A small motor spins at 300 rpm (at least in this country with its 60 hz power supply), and there is a tiny stepping motor attached to a read/write head. A stepping motor is a common item in indexing applications, where you want a motor to move a precise distance and stop on a dime. The read/write head is just a smaller version of what you have on a cassette recorder.

The stepping motor "steps" the head from track to track on a diskette. The tracks are concentric circles, not a long spiral as you would have on an album.

All of this is ultimately controlled by the disk software provided with your computer. Usually this is located in ROM within the machine. In most machines, the ROM is only sophisticated enough to load in the official Disk Operating System (DOS) which is located on the disk in the drive when the machine is turned on. The DOS contains all the file handling software, copying software, etc. and because it is on disk, it can be easily modified and/or updated as time goes by.

Our friends at TI decided to put the whole thing in ROM, which has a few bad side-effects. First, it makes it hard to update and improve the software, which is located in the Disk Controller Card. Second, although the machine is a 64K machine, just like all the others, TI has set aside so much memory for special purposes, that there is only 32K left to play with. They set aside 8K for cartridges, 8K for disk drives, 8k for RS232/PIO, 8K for the Operating System (can't complain about that one), and 8K for various interfaces (speech, sound, VDP). OK, those are all good applications to have, but if you don't use them, you still can't use that memory for other things.

Anyway, all of the controlling software for the TI is located in the ROM card, as I said. This software tells the step motor when to step to the next track, when to return to the beginning, etc.

There is no STANDARD for how a computer keeps track of data. In the case of the TI/4A there is a directory of existing files, and a map of where they are located, at the beginning of each disk. These files are not necessarily all in complete groups. If you delete a 12 sector file from a disk, there is a 12 sector gap recorded in the map. Then if you add a 20 sector file, the software will put the first 12 sectors in the gap, and put the rest in the first available spot. When you ask for a file that is broken up this way, you can hear the disk head scooting along to read each individual segment.

Because the disk drives themselves are pretty STANDARD, there are a few things that don't change. For instance, there are 48 tracks per inch in most 5-1/4" systems. (There is a new 96 TPI system around, not TI compatible). Most systems use only 35 or 40 of the available 48 tracks. There are either 9 or 18 sectors per track (single or double density). Each

sector holds 256 bytes of data. The standard design allows 250,000 bits per second to be written.

The following is a complete, and to the best of my knowledge, accurate description of the Disk Directory format and file storage allocation used by the TI 99/4A computer.

SECTOR 0 - Volume Information Block
Address Contents

0000-0009 Disk name, up to 10 char

000A-000B Number/sectors on disk
 >0188=360, >02D0=720,
 >05A0=1440

000C Sectors/track 09sd,12dd

0000-000F 'DSK' 44534B

0010 >50 Protected, >20 Not P

0011 Tracks/side 28=40, 23=35

0012-0013 Sides/Density 0101
 SS/SD, 0201 DS/SD
 0102=SS/DD,0202= DS/DD

0038-end Sector allocation bit
 map

This is a sector-by-sector bit map of sector use; 1= sector used, 0= sector available. The first byte is for sectors 0 through 7, the second for sectors 8 through F, and so on.

Within each byte, the bits correspond to the sectors from RIGHT to LEFT. For example, if byte >0038 contained >CF00 then the byte equals 1100 1111. This means that sectors 0 through 3 are used, sectors 4 and 5 unused and sectors 6 and 7 used.

Information for sector 168 starts at >0065. Therefore, if your disk is SS/SD, all addresses from >0065 to end should be FFFF if it was formatted by DISK MANAGER and has not been tampered with.

SECTOR 1 - Directory Link

Each 16-bit word lists the sector number of the File Descriptor Record (FDR) for an allocated file, in alphabetical order of the file names. The list is terminated by a word containing >0000; therefore, the maximum number of files per disk is 127 [(256/2)-1]. Any addresses past >0000 will not catalog, but will still be accessible. If the first address is >0000, move all addresses four digits to the left, (eliminating this false address), then the disk will catalog. If the alphabetical order is corrupted (by a system crash during name change, for instance), the binary search method used to locate files will be affected and files may become unavailable,

SECTOR >2 to >21 - FDRRecords

0000-0009 File name, up to 10char

000C Filetype: >01=Program
 (memory image)
 <00=Dis/Fix
 >02=Int/Fix
 >80=Dis/Var
 >82=Int/Var
 File | deletion
 protection invoked by
 DM2 will be shown by
 >08 added to the
 above.

000D (MAXRECSIZE)
 Records/sector

000E-000F #/sectors to file
 DM2 will list one more

0010 For memory-image
 program files and
 variable-length data
 files; this contains
 the number of bytes
 used in the last disk
 sector. This is used
 to determine
 end-of-file.

0011 MAXRECSIZE of data file
 >50=80, >FE=254, etc

0012-0013 File record count, but

with the second byte being the high order of the value.

001C Block Link (see note)

NOTE on file storage:
Files are placed on the disk in first-come / first-served manner. The first file written will start at sector >0022, and each subsequent file will be placed after it. If the first file is deleted, a newer file will be written in the space it occupied. If this space isn't big enough, the file will be 'fractured', and the remainder will be placed in the next available block of sectors. The block link map keeps track of this fracturing. Each block link is 3 bytes long. The value of the 2nd digit of the second byte, followed by the 2 digits of the first byte is the address of the first sector of this extent. The value of the 3rd byte followed by the 1st digit of the 2nd byte, is the number of additional sectors within this extent.

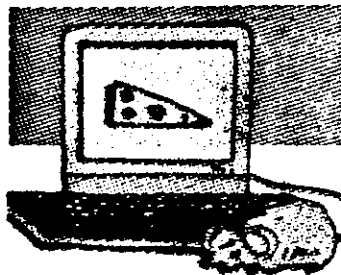
Sectors >2 through >21 are reserved for File Descriptor Records and are allocated for file data ONLY IF no other available sectors exist. If more than 32 files are stored on a disk, additional FDR's will be allocated as needed, one sector at a time, from the general available sector pool.

QUIET DISK DRIVES AND FANS FOR P-BOX

DSDD FH DISK DRIVES FOR TI
AND CORCOMP CTRLS. ONLY
QUIET BOX FANS FOR P-BOX
BOTH WILL REPLACE ORIGINAL
30 DAY WARRANTY

TESTED
DISK DRIVES \$20.00 + \$3.00 S&H
FANS (NEW) \$9.95 + \$3. S&H
SEND CHECK OR MONEY ORDER TO

DELBERT WRIGHT
185 N. POST RD.
INDPLS. IND. 46219
317-895-1765



Something's cheesy around here

from - HUG

```
! REM---saved as SPEAK2ME---
10 REM *****
20 REM *   SPEAK TO ME   *
30 REM * by Chick De Marti *
40 REM * from an idea by *
50 REM * Sue Harper of the *
60 REM * Pittsburgh U.G. *
70 REM * Terminal Emulator *
80 REM *   is required   *
90 REM *****
```

```
100 CALL CLEAR
110 OPEN #2:"SPEECH",OUTPUT
120 GOSUB 330
130 PRINT "Enter Q and a com
ma to Quit."
140 PRINT "Enter J or M, a c
omma, and":"a message."
150 INPUT " ":N$,MSG$
160 IF N$="Q" THEN 240
170 IF N$="J" THEN 210
180 PRINT #2:"//25 80"
190 PRINT #2:MSG$
200 GOTO 140
210 PRINT #2:"//45 144"
220 PRINT #2:MSG$
230 GOTO 140
240 CALL CLEAR
250 PRINT TAB(12);"THE":TAB(
12);"END"
260 PRINT : : : : : : : : : :
: :
270 PRINT #2:"//45 144"
280 PRINT #2:"SO ^LONG"
290 PRINT #2:"//25 80"
300 PRINT #2:"BY 4 <NOW!"
310 CLOSE #2
320 END
330 PRINT TAB(5);"TURN CAP=L
OCK ON!!"
340 FOR DELAY=1 TO 600
350 NEXT DELAY
360 CALL CLEAR
370 RETURN
```

BBS

Hoosier Users Group
Baud rate: 300/1200/2400
On Line 24 Hours Daily

317-782-994A

Now with a Hard Drive
40 MEG ON LINE

SOME RECENT PRODUCTS REVIEWED

Compiled from here and there

DIGI-PORT-digital sound adaptor and player; a cable that plugs into your 4A or Geneve; lets you play 8 bit digitized sound from the IBM, MAC, AMIGA, ATARI ST, etc. Software is supplied with the PIO adaptor cable and 10 disks configured for your particular drive. \$39.95 US dollars

OPA 432 Jarvis St, Suite 501-502
Toronto, Ontario, Canada M4Y 2H3

TIM (TI Image Maker)-80 column device, produces printout of GIF pictures \$179

OPA--see above and
Bud Mills Services 166 Dartmouth Drive
Toledo OH 43614-2911 \$39.95 US dollars

LinEditor-a text editing program for 4A and Geneve; allows loading and editing of a text file larger than can be fitted into the computer's memory. \$14.95 + \$3 s&h

Asgard Software PO Box 10306 Rockville MD 20849

Asgard Mouse Developers Package-extensively documented routines with source code for Assembly, c99, Fortran, XB programmers. Requires an Asgard Mouse. \$14.95 + \$3 s&h

Asgard-see above.

Thumbnails-organize, catalog and convert MacPaint pictures

Starbase Raiders-arcade style game

Gofer-a utility for use with Page Pro 99

Each is \$12.95 & \$3 s&h

Each is from Asgard-see above

Bride of Disk of Dinosaurs-TI Artist format \$12 & \$1 s&h

Fonts and Borders

Disk of Horrors

Each is by Ken Gilliland

From Notung Software 7647 McGroarty St.
Tujunga CA 91042

Scud Buster-uses joystick

Code Breaker

Harrison Software 5705 40th Place
Hyattsville MD 20781

Smart Connect-transfers files between TI and PC computers; automatically splits large PC files into increments small enough to be loaded into TI Writer. \$10 includes s&h

Harrison Software--see above

Sound F/X-by Barry Boone-plays true digital sound through the 4A or Geneve without additional equipment; disk comes with several sound files \$14.95

Texaments 53 Center St. Patchogue NY 11772

Sound Bytes-disks with F/X Sound files; 2 SSSD each(sounds of President Bush, cartoon characters, etc. 1 pkg \$2.95
Texaments--see above

The Organizer for TI Base by Bill Gaskill-an informative management system, completely menu driven; is designed to work as part of TI Base, but is also a good way to learn TIBase itself. \$14.95
Texaments--see above

DM1000 version 5.0-extensively revised by Jack Mathis of Southwest 99ers; has 10 major changes. \$2.00
Southwest Ninety Niners PO Box 17831
Tucson AZ 85730
(Donations should be sent to Ottawa Users Group)

Fairware Author-Ray Kazmer (famous for his Woodstock's Christmas)
Full Animation Disk #1 and #2
Valentine-a game
The 12 Dungeons of Remzak-Texas Rangers, Cannonball Chess,
Charpat, etc.
XB to Artist Converter
Contribution asked of \$5 and \$10
Infocom Rapid Loader-loads Infocom games in seconds. Asking \$7
All above-Ray Kazmer 8614 Foothill Blvd Apt.#221
Sunland CA 91040

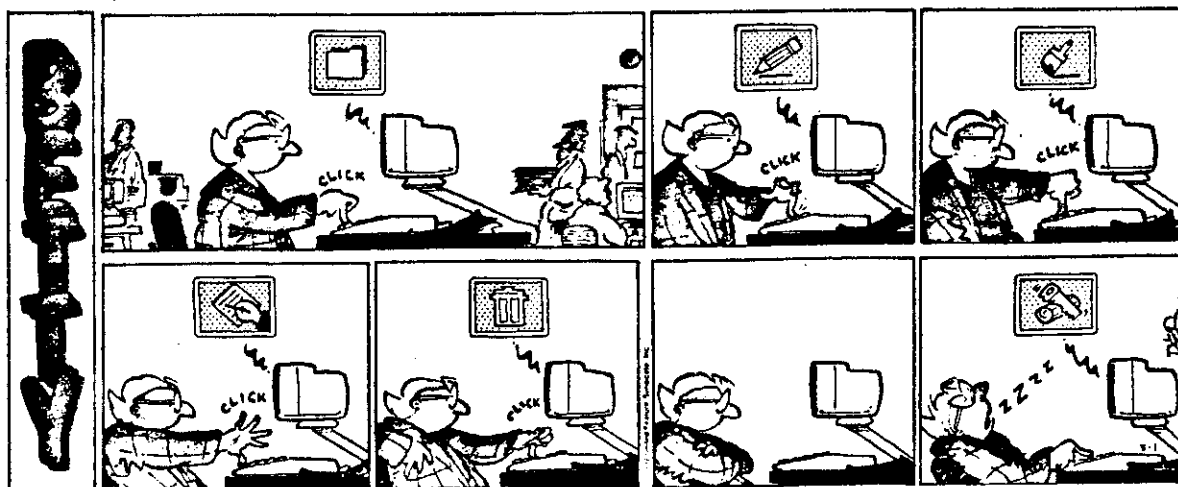
Foreign language drill programs-some 30 languages using TI Basic; type in a new vocabulary; save lists; load lists; edit, delete lists. Also provides graphic and musical rewards. Each disk has at least 1 starter vocabulary that can be added to. Information from programmer: Mr. Don Shorock P.O. Box 501
Great Bend KS 67530

Triad-terminal emulator, disk manager and 40 column text editor by Wayne Stith. For 4A and Geneve. \$20
Jerry Coffey 9119 Tetterton Ave.
Vienna VA 22182

Chainlink Solitaire-by Wayne Stith and Walt Howe \$12
Jerry Coffey--see above

Note: Jerry Coffey is now the distributor for J P Software

Not responsible for prices as shown here. Check with vendor.



**A Real Time Clock
for the
CorComp 9900 MES**
by John Clulow and Ron Gries, New Horizons

Because one of the primary uses of our CorComp 9900 Micro-Expansion System will be in our local TI COMM Bulletin Board, we wanted to build a real-time card that could be used with the MES. The initial version of our BBS used an interrupt driven "real" time clock, but the intermittent disabling of interrupts in the course of program execution greatly affected the clock's accuracy. The next step was to build a real-time clock card for the PEB. We based our card upon a design for a clock and A/D converter by Gary Emmich and Tony Albanese of the No. NJ 99er User's Group with modifications in the power supply circuit and address decoding (required for the CorComp DSDD controller).

This design subsequently evolved into the one presented here. The device plugs into the 44 pin bus and may be used with the console and Mini Memory or Editor Assembler although it is designed specifically for use with the 9900 Micro-Expansion System. Used with the MES, the card may be accessed through TI BASIC or Extended BASIC. CALL LOAD statements may be used to set the clock registers and CALL PEEK to read the time and date. In this case conversion from BCD to decimal is required within the basic program. Alternatively, an assembly language program (below) may be used to access the clock with simple CALL LINK("SET",...) and CALL LINK("TIME",...) statements. The particulars are explained in the A/L source code.

To access the clock through BASIC, use the decimal versions of the addresses shown in the program (e.g., Seconds = -31228). The byte you write or read at each address must be in BCD format. That means that each digit of the number is expressed separately in hexadecimal. For example, the number 31 would be equivalent to 3*16 + 1 or 49. So if you wanted to write a 31 to the Day of Month register, you would write 49 instead of 31. Similarly, the month 12 would be represented as 18 (1*16 + 2). To convert a number back from BCD to decimal, divide 16 into it. The integer result is the first digit and the remainder the second. For instance if you obtained an 88 from the seconds register, INT(88/16) is 5 with a remainder of 8 so the decimal number is 58 seconds.

If you have had some previous experience building circuits such as this, you will not find the board particularly difficult to construct. However, we do not recommend that this project be undertaken if you don't feel sure of yourself and in any case we assume no responsibility for any damage to your equipment or consequential damage arising from use of the clock card. If you want to construct a card and feel uncertain about some aspect of its construction, you may be able to get help from someone in your users group who is more familiar with electronics.

The parts list is shown below. The 22/44 pin edgcard connector must be modified by cutting off the tabs on the ends (so it will fit into the 44 pin bus on the right side of the MES). The solder tails should be spread apart and the ribs in the slot on the back of the connector removed so that the grid board can be epoxied to the connector. It will be necessary to cut a 1/4 inch notch in each side of the grid board so it will fit. Once the board is constructed, it will not be hard to put the card together. We recommend the use of wire-wrap wire even if you plan to solder because it allows for more dense circuit configuration. Note that three "AAA" Nickel-Cadmium cells are used in series as battery backup for the clock.

DEF SET,TIME

* This program allows access to the clock. To set
* it use: CALL LINK("SET",H,M,S,DW,M,DM) where the
* parameter list is Hours, Minutes, Seconds, Day of
* the Week, Month, and Day of the Month. To read
* the time use: CALL LINK("TIME",H,M,S,DW,M,DM,D\$,M\$)
* where D\$ and M\$ are return variables for the day
* of week and month names (e.g., "Saturday" & "June").

* Equates used are for Ext. BASIC
* Change as required

	XBASIC	MINIMEM	ED/ASSM w/BSCSUP
NUMREF EQU	>200C	>6044	REF NUMREF
STRREF EQU	>2014	>604C	REF STRREF
NUMASG EQU	>2008	>6040	REF NUMASG
STRASG EQU	>2010	>6048	REF STRASG
XMLLNK EQU	>2018	>601C	REF XMLLNK
CFI EQU	>12B8	>1200	>1200

* Equates	hex	decimal
THOUS EQU	>8600	-31232
HUNDR EQU	>8602	-31230
SECONDR EQU	>8604	-31228
MINUTE EQU	>8606	-31226
HOUR EQU	>8608	-31224
DAYW EQU	>860A	-31222
DAYM EQU	>860C	-31220
MONTH EQU	>860E	-31218
STATUS EQU	>8688	-31096
RESET EQU	>8684	-31100
FAC EQU	>834A	-31926

* Buffers and data

WS	BSS	32
HR	BYTE	0
MN	BYTE	0
SC	BYTE	0
DW	BYTE	0
MO	BYTE	0
DM	BYTE	0
SAVRTN	DATA	0
D10	DATA	>000A
D6	DATA	>0006
FAC1	DATA	>4000
BUFFER	BSS	10
SUND	TEXT	'Sunday'
MOND	TEXT	'Monday'
TUES	TEXT	'Tuesday'
WEDN	TEXT	'Wednesday'
THUR	TEXT	'Thursday'
FRID	TEXT	'Friday'
SATU	TEXT	'Saturday'
JAN	TEXT	'January'
FEB	TEXT	'February'
MAR	TEXT	'March'
APR	TEXT	'April'
MAY	TEXT	'May'
JUN	TEXT	'June'
JUL	TEXT	'July'
AUG	TEXT	'August'
SEP	TEXT	'September'
OCT	TEXT	'October'
NOV	TEXT	'November'
DEC	TEXT	'December'
WEEK	DATA	SUND,MOND,TUES,WEDN
	DATA	THUR,FRID,SATU
MNTH	DATA	JAN,FEB,MAR,APR,MAY
	DATA	JUN,JUL,AUG,SEP,OCT
	DATA	NOV,DEC,WEEK

* Read data from XBASIC: Set Clock

```
SET      MOV  R11,@SAVRTN
          LWPI WS
```

* LINK Parameter List Format:
* Hr,Min,Sec,Day/W,Mo,Day/M

* 1) Get XBASIC LINK parameters
* and convert to BCD

```
GP1      CLR  R0
          CLR  R1
          LI   R5,6
          LI   R4,HR
          INC  R1
          BLWP @NUMREF
          BLWP @XMLLNK
          DATA CFI
          MOV  @FAC,R3
          CLR  R2
          DIV  @D10,R2
          MPY  @D6,R2
          A    @FAC,R3
          SWPB R3
          MOVB R3,#R4+
          DEC  R5
          JNE  GP1
```


* Return Day and Month strings

```

*
INC R1
LI R2,BUFFER
CLR R3
MOV @DW,R3
SWPB R3
DEC R3
SLA R3,1
MOV @WEEK(R3),R4
MOV @WEEK+2(R3),R5
BL @SUBR
INC R1
CLR R3
MOV @MO,R3
SWPB R3
DEC R3
SLA R3,1
MOV @MONTH(R3),R4
MOV @MONTH+2(R3),R5
BL @SUBR

```

* Return to XBASIC

```

*
RETURN LWPI >B3E0
MOV @SAVRTN,R11
BL *R11

```

* SUBROUTINE

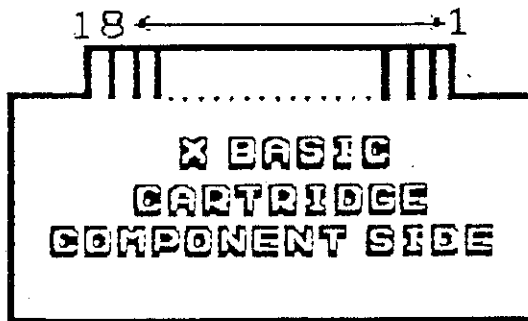
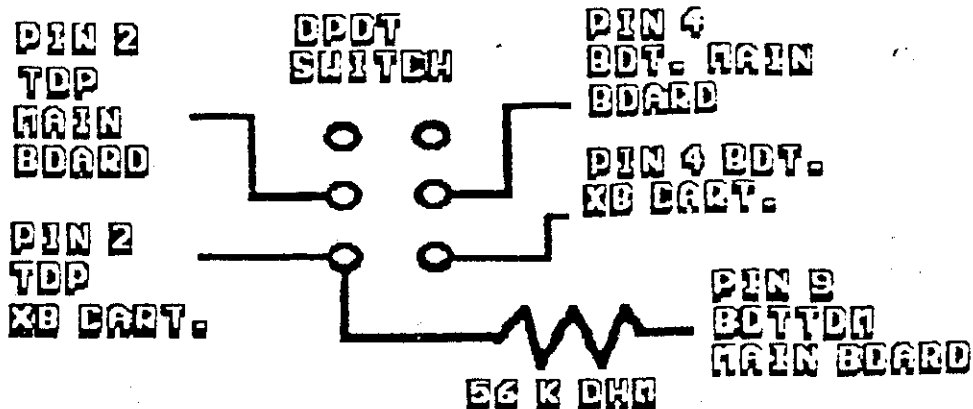
```

*
SUBR S R4,R5
SWPB R5
MOV @BUFFER,R5
LI R6,1
SWPB R5
DLP MOV @R4+,@BUFFER(R6)
INC R6
DEC R5
JNE DLP
BLWP @STRASG
B *R11
END

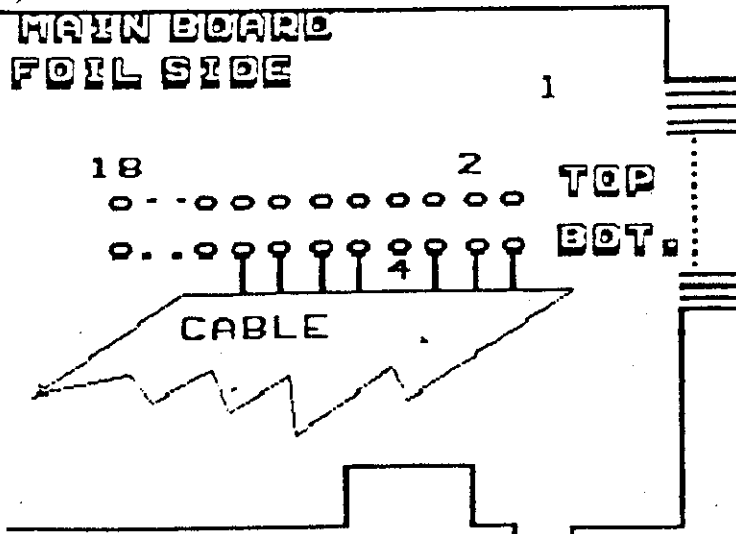
```

CONSOLE XBASIC

BY CHUCK REINHART



TOP= COMPONENT SIDE
BOT= FOIL SIDE



CONSOLE XBASIC by CHUCK REINHART-LTI USERS GROUP

XBASIC IN THE CONSOLE PROJECT INTRODUCTION:

Since more and more programs are loading from XBASIC (example: TI-WRITER, DM 1000, MENU), it would make good sense to put the XBASIC cartridge in the console. This would

also reduce lockups that are due to a dirty cartridge port. (The XBASIC cartridge causes most of the lockups)

The following project will mount the XBASIC CARTRIDGE in the console. In the project 2 ribbon cables are soldered to the cartridge port pins on the main board. The XBASIC cartridge circuit board is removed from it's case and soldered to the other end of the ribbon cables. The cables are then routed around the back of the main board and the XBASIC cartridge is mounted on the top of the metal shield. There is plenty of clearance to the left of the cartridge port. A switch is also installed to allow operation with XBASIC or an intalled cartridge. XBASIC will not function if a cartridge is installed in the port. The switch can be installed in the back center or top of the console cover.

PARTS REQUIRED:

- 1 56K RESISTOR 1/4 WATT
- 1 DPDT MINI SWITCH (Radio Shack 275-626)
- 2 8 in PIECES OF RIBBON CABLE WITH 17 CONDUCTORS IN EACH (Radio Shack 278-772)
- 5 8 in PIECES OF WIRE

NOTES:

1. I do not accept responsibility for problems resulting from this project. The risk is yours.
2. *** This is not a simple project *** Do not attempt this project unless you are familiar with electronics and are experienced in soldering.
3. Use solder sparingly. There are land patterns that run between the pins. Also use a small soldering iron.
4. After this modification you will not be able to have a cartridge installed while you are running XBASIC.
5. Read the instructions fully before starting the modification.

INSTRUCTIONS:

-- Remove the main board from the console. -- Remove the cartridge port and metal shield from the main board. -- Remove the XBASIC circuit board from it's case.

-- Take the two pieces of ribbon cable and separate the wires in the four ends back 1 inch. Then strip all of the wires 1/8 inch and tin the bare ends. Mark one cable TOP and the other BOTTOM.

-- Place the main board component side down with the side port connector on the right. Locate the two rows of pins that go to the cartridge port (see main board diagram).

-- Take the cable marked BOTTOM and mark a 1 on the edge at both ends. Then solder the wires from one end of the cable to the bottom row of pins skipping pin #4. Keep the wires in order with pin #1 on the right (see main board diagram). Solder one of the 8 in wires to pin #4 and one 8 in wire to pin #9 (pin #9 will have 2 wires). Place a piece of black electrical tape on the circuit board, under the cable to prevent shorts.

-- Take the cable marked TOP and mark a 1 on the edge at both ends. Then solder the wires from one end of the cable to the top row of pins skipping pin #2. Keep the wires in order with pin #1 on the right (see main board diagram). Solder one of the 8 in wires to pin #2.

-- Place the XBASIC circuit board with component side up and connector facing the cable marked TOP. Solder the wires to the connector skipping contact #2 (see circuit board diagram). Solder an 8 in wire to contact #2.

-- Turn the circuit board over and solder the wires from the cable marked BOTTOM skipping contact #4 to the contacts on the foil side of the XBASIC cartridge (pin #1 TOP should line up with pin #1 BOTTOM). Solder an 8 in wire to contact #4.

-- Solder the 5 wires and 56k resistor to the switch (see switch diagram).

-- Bend the edge of the metal shield to allow room for the cable to pass. Mount the XBASIC cartridge on a piece of cardboard and tape it to the top of the metal shield to the left of the cartridge port. -- Mount the switch in the back of the console cover near the center.

-- Check the wiring with an OHM METER from the cartridge port to the XBASIC circuit board connector. -- Clean the side port and cartridge port. -- Reassemble the console and test the switch in both positions.

..... THE END ** GOOD LUCK

PROGRAMMER'S DILEMMA

By Don Lester, Vancouver, BC
From ROM User's Group
Huntington Beach, CA
Copied from LA 99ers, Feb 92

I sit before my 4a
The screen is cold and
black
I push the keys I think will
work
But nothing's coming
back.

I know it's not the RAM or
ROM
Since they were both just
tested.
Maybe it's hung up
In some deep loop I'd
nested?

The floppy drives sit silently
Their little lights are
out.
I search the screen for any
clue
To what it's all
about.

Could it be a vicious virus
Deep down in the
root?
All else fails, I push the button
To go for a
reboot.

But nothing works!! Is there no
cure?
I must seek out this
bug.
That's when I look down and see
That someone's pulled the
plug!
