SOONER 99ERS

This newsletter is the official publication of the SOONER 99ers
POB 61061
Oklahoma City, OK 73146

Sooner 99ers BBS#:
(405) 672-8270 300/1200 8N1

MARCH 1989 NEWSLETTER

Hello, friends, and welcome to the next edition of our newsletter. I have tried to be responsive and find articles that you might be interested. Spring is the traditional time to start in on things you have intended to do, so it is with our club! I have seen many people lately that were missed for quite some time. We must be doing something right to continue so long; that something is to support the members. If you have a question or a problem, bring it to the next meeting. It always amazes me that so many people know so much but write so little! Newsletter articles are always appreciated, don't think you have nothing to offer; I learn some new trick, fact or trivial detail at each meeting. How about those who don't make the meetings? How will they pick up these little gems that might save them hours of work. We always talk about 're-inventing the wheel' as something that should not be done. How is anybody to know about the wheel you have created unless you WRITE about it?

This month; read about MacFlix in an article by Deanna Sheridan of the Northcoast 99ers of Cleveland, Ohio. MacFlix gives us more ability to use files which were created on another type of computer, more compatibility! If you are "into" graphics, there are about a zillion programs or methods you can use to get exactly what you want. Where there's a will, there's a way...Especially with a computer!

Also, there are many hardware modifications being made to our trusty machine; I was asked about changes to the TI disk controller and the RS-232 card. Read on! Also, I have found info about another board you can add for just a few dollars and expand your computing horizons.

Garth has written an article about the one thing few of us admit to doing; games on the computer. He has conducted extensive research on the subject and interviewed some of the most competent authorities on the subject and reveals his findings to us in the pages of this newsletter.

I have decided to write an article about disk information. I don't know how much space I will have, but don't want to drag it out or spit it up into another issue.

One last topic; is it me or has the supply of freeware dried up? It seems like we used to see new software constantly; games, utilities, some great, some not so great. Has the well gone dry? Have the "brains" of the TI community given up? Do we have all the software we can ever need? Stay tuned next month for the answers to these questions.............. BP

A KIDS' EYE VIEW OF GAMES ON THE TI

Perhaps if we had played computer games as children those Johnny-Come-Latelys' of us would not have had any "computerphobia" to overcome in our adult years. Such is not the case in the Potts household. Three youngsters (with a fourth eying this "funbox" rabidly) enjoy the many offerings available on the TI.

With their help, mini-reviews follow of their favorites. I note that all of these are available in our club library.

The first and perhaps, most favorite, is POLE POSITION. Put out by Atarisoft, this nifty piece of software puts you behind the seat of a beautifully drawn race road up a mountain. Roaring along at speeds up to 195 mph, avoiding slower moving competitors and roadside obstacles, you compile points for speed, and safety. This game fulfills the 11 yr. old's fantasy of "driving a real car." (Probably more safely, too). Great sound effects!

Also on the same Atari disk is MOON PATROL. Your lunar vehicle bumps and bounces along the rugged moonscape navigating a careful course away from space ship and spacestation bombings, landmines, and huge meteor pocks. You can fire back at your enemies, but you've got to watch your fuel guage or a long and cold vigil awaits you on the moon watching endless earthrises.

JUNGLE HUNT is another Atari offering for the TI. The object of this game is to save a young woman suspended over a boiling stewpot surrounded by carniverous cannibals. To get to the rescue scene you must swing across a wide forest crevass by rope, swim through a wide jungle river infested with alligators and other predators, endure a rock pelting en route to mano a mano combat with the nasty cannibals. All of this is set to the background sounds of jungle drums and appropriate music. Pretty heady stuff!

TI offers a classic PINBALL game as one of its options on its Videogames module. Complete with appropriate sounds just like an arcade, the lifelike movement of the ball simulates the real thing very effectively.

VIDEO BOWLING, done for TI by the Software Exchange, is a fine game for up to 4 players. You can throw your ball straight as a string, a hook or a Brooklyn at the little pins. The gameboard keeps score for all players. My kids' only complaints are that the program doesn't include joystick capability.

The popular FROGGER for TI includes many of the features you come to expect. In order for you to get your little green slimy amphibians safely to rest on the other side of the river, the critter must dodge assaults from cars, trucks, and tractors on land and logs, turtles, alligators and snakes on water. The graphics are super.

Finally, in keeping with the popular movie series, a complex dungeon escape game is available for aspiring NINJAs. You have to find your way through a maze of the castle fighting off the loathful guards, amassing experience points and avoiding damage points. It helps to collect "Gold" along the way, although my kids can't tell me why. Maybe it's to bribe your way out of this mess.

There are literally hundreds of games available for our computer ranging from these rather elementary ones with the emphasis on graphics and action to those of the "Adventure" (word game) variety to puzzles, etc. The built-in graphics capability makes the TI easy for kids to start their TI careers. Of course, they will quickly want to abandon these "frivolous pursuits" for more esoteric functions like wordprocessing, etc. Like us, right!.... and eventually "graduate" to the more sophisticated functions like we adults....sure. GARTH POTTS

GETTING A LINE ON YOUR PROGRAM

Have you ever wondered how your TI console stores and reads the BASIC program you type in? Well, I have! It all started when I began to have problems with one of my consoles. Every time I EDITed an existing line, other lines got "messed up" or even moved to some other part of he program! Eventually, I had to have the console replaced, but it didn't keep me from becoming curious. I was determined to figure out how this could happen. When I finally found the answer, I learned a great deal about how the TI keeps track of the program we type into it. Below is some info that you can use to get "computer's-eye-view of the program you have typed in.

Any BASIC operating system has to keep track of every line of code you type in. It not only has to keep track of the program data, but it also has to keep track of the line numbers, too! TI BASIC uses two different areas for this; 1) the Line Number Table and 2) the Program Area.

All this info is stored in the highest memory available to the console. For TI BASIC that will usually be somewhere around address 14228 and for X-BASIC with memory expansion, somewhere around address -28. (This all depends on the number of files open, whether the disk drives are attached, etc.)

The actual lines of program code are stored first. Each line of code is stored as a series of bytes. Most all reserved words like PRINT, INPUT, REM, etc. are stored as a single byte value called a TOKEN. For instance, the TOKEN values for the the above three words are 156(>9C), 146(>92) and 154(>9A). TI BASIC also keeps track of variables, constants and jump references in a similar way. Each line is preceded by a length byte and, at the end of each line, the value 0(zero) is placed to indicate the end of the line.

As each line is typed in, the BASIC operating system translates it into these byte values and stores it at the top end of the available memory. When you EDIT a line or INSERT a new line between existing ones, that new line is stored at the "end" of the Program Area, not inserted into the middle. This is important to remember when trying to locate a line of code! Note also that the line numbers (i.e. 100,110,120,etc.) are NOT stored with the line of code. This allows the TI system to easily RESequence your program, and allows the system to place lines at any available memory location instead of in execution order only. So where are the line numbers? In the Line Number Table, of course!

After all the lines of code are stored, the Line Number Table begins. Each line of code has 4(four) bytes of information in the Line Number Table; two bytes for the line number (>0064 for line 100), and two bytes for the memory address where the line of code is stored (>FFE7=-28). When the BASIC interpreter runs the program it looks at the Line Number Table, gets the line number and the line address, branches to the address, gets the actual program line and THEN performs the line of code! Not so simple, eh?.

Since the Line Number Table is stored at the end of all the program lines, each time you add a new program line, the entire table has to be "shifted" downward into lower memory. This is why, especially in a long TI BASIC program, it takes quite a while for the cursor to "come back" after you have EDITed a line. The operating system must code the line, shift all the line numbers and addresses downward and then insert the new line number and address into the table. That would take anybody a little time!

Let's look at the operating system as it does all this stuff! Here's what we'll do:

1) Enter X-BASIC (or TI BASIC with ED/A or MINIMEM module) and type:

```
100 PRINT "THIS IS A TEST"
```

2) Find Line Number Table: To find the Line Number Table, PEEK into the CPU RAM PAD at address -31950; get two consecutve bytes; convert these bytes to decimal; subtract 3 bytes for overhead and, if you are using X-BASIC with memory expansion, subtract 65536; you now know the address where the line table starts!

```
CALL PEEK(-31950,A1,A2)
LT=A1*256+A2-3-65536
```

In X-BASIC, the value should be -47 and in TI BASIC it should be 14273. As each line of code is typed in, the table will move farther down into memory.

3) Get the line and address: Using the start of the Line Number Table, get the first line of code and the address where it is stored. In X-BASIC this is stored in memory expansion so you use the following:

```
CALL PEEK(LT,L1,L2,A1,A2)
LN=L1*256+12
LA=A1*256+A2-65536
```

LT=Line Table Address (-47)
LN=Line Number (100)
LA=Line Address (-42)

In TI BASIC, the program is stored in the console itself, therefore you must look into the VDP area with the command PEEKV:

```
CALL PEEKV(LT,L1,L2,A1,A2)
LN=L1*256+L2
LA=A1*256+A2
```

LT=Line Table Address (14273)
LN=Line Number (100)
LA=Line Address (14278)

4) Get the line length: Each line of program code is preceded by a length byte. This tells the system how many bytes to interpret. To find the line length, read the byte PRECEDING the Line Address:

```
X-BASIC: CALL PEEK(LA-1,LL)
TI-BASIC: CALL PEEKV(LA-1,LL)
```
In both cases, LL=18

5) Read the Program Line: Now all we have to do is read in 18 bytes of code starting at the Line Address (LA):

X-BASIC:
```
CALL PEEK(LA,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R)
PRINT A;B;C;D;E;F;G;H;I;J;K;L;M;N;O;P;Q;R
```

TI-BASIC:
```
CALL PEEKV(LA,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R)
PRINT A;B;C;D;E;F;G;H;I;J;K;L;M;N;O;P;Q;R
```

In both cases, the values A-R will be:
156 199 14 84 72 73 83 32 73 83 32 65 32 84 69 83 84 0

156 - Token value for PRINT
199 - indicates a string value
14 - length of the string

The rest of the bytes are ASCII values for the string (84=T, 72=H, 73=I, 83=S, etc.).

Now let's see if we can't reverse the process. This time we'll do a few POKES into memory that will make the computer think you typed in a line of code! Here's what we'll do:

1) Clear memory and type CALL INIT.

2) Set up Line Table: The system needs to know where the line table starts, so:

```
          CALL LOAD(-31950,A1,A2)
X-BASIC:                     A1=255, A2=220
TI-BASIC                     A1= 55, A2=204
```

3) Insert line # and address: Next we need to fill up the line table with

line and address data:

X-BASIC: CALL LOAD(-39,0,100,255,222)

TI BASIC:
CALL POKEV(14281,0,100,55,206)

4) Insert line length and line code. Now all we need to do is insert the line data:

X-BASIC:
```
CALL LOAD(-35,10,156,199,6,72,69,76,76
,79,33,0)
```

TI BASIC:
```
CALL POKEV(14285,10,156,199,6,72,69,76
,76,79,33,0)
```

5) LIST YOUR PROGRAM! Because the system has not been properly informed as to where the program ENDS, this is not "RUNable" code. But at least you can see your work!

SAMPLE PROGRAM

Below is a program that will "read itself." Using the tools we learned above, this routine will read the Line Number Table, and find the Line Address and Line length. After each line is found, the program waits for you to hit a key before it reads the next line.

NOTE: This program is written for TI BASIC with ED/A or MINIMEM modules. For use with X-BASIC/memory expansion, change the following lines:

220 add -65536 to the end of the line.
260 change PEEKV to PEEK.
290 add -65536 to the end of the line.
330 change PEEKV to PEEK.

Armed with this info, you can begin to learn more about how the TI system works and you can use these tools to help read AND write your own programs. A number of programmers place routines like this into their programs as a way to protect valuable code sequences or alogorithms from prying eyes. It's usually called "imbedded code." You can also use routines like this to translate a text line into program format once you learn to decode all the token values!

```
100 REM  *****************
110 REM  *               *
120 REM  * PROGRAM PEEKER *
130 REM  *               *
140 REM  *****************
150 REM
160 REM
170 REM
180 REM
190 CALL INIT
200 CALL CLEAR
210 REM
220 REM  *FIND LINE TABLE*
230 REM
240 CALL PEEK(-31950,A1,A2)
250 LT=A16+A2-3
260 REM
270 REM  *GET LINE  ADR*
280 REM
290 CALL PEEKV(LT,L1,L2,A1,A2)
300 LN=L16+L2
310 IF LN=0 THEN 570
320 LA=A16+A2
330 REM
340 REM  *GET LINE LENGTH*
350 REM
360 CALL PEEKV(LA-1,LL)
370 REM
380 REM  ***************
390 REM  *PRINT THE DATA*
400 REM  ***************
410 REM
420 PRINT "LINE";LN
430 PRINT "ADDR";LA
440 PRINT "LEN ";LL
450 PRINT
460 CALL SOUND(150,1400,0)
470 CALL KEY(0,K,S)
480 IF S=0 THEN 470
490 REM
500 REM  *UPDATE LT*
510 REM
520 LT=LT-4
530 GOTO 290
540 REM
550 REM  *END*
560 REM
570 PRINT "END OF PROGRAM"
580 PRINT
590 END
```

DISK TIPS
by
Barry Peterson

This will be nothing new, but since many of our newer club members seem to need help with the layout of data and files on the disk, I decided to "get it in writing". Old-timers might trip me up now and then; I don't claim to be an expert, but this article will be a summary of what I have learned and guessed through the years.

There are three controllers that are widely available for the TI-99 system. They were/are produced by TI, CorComp, and Myarc. I own one of each, but this will be more of a Joe Friday approach. (Just the facts, ma'am)

The TI controller was the first, and for a while, the only game in town. TI played it safe; although other disk formats were possible; used by Apple, Commodore, Atari, etc., they also required special drives. All TI controllers can use almost any drive: IBM-compatible means TI-compatible.

A SSSD disk is capable of holding 125K bytes of data, although we do not have access to all 125K. Some disk space is used in the initialization process as a "road map" specifying the organization of the disk. This results in the usable disk space of 90K bytes. Using double sided drives, of course, means we have 180K of disk space. A double-density controller, means 180K on a single-sided drive and 360K on a double-sided drive. (Just like IBM, almost) Myarc came up with a double-density controller with 160K and 320K capacities. I'll go into the differences later.

TI drives are '40-track', a track being an invisible magnetic circle where data can be stored. This is similar to a phonograph record. (but it's really one spiral on a record) TI chose to subdivide each track into 9 sectors of 256 bytes each. This conflicts with the IBM standard of 512-byte sectors, but that's another story.

When you buy a disk, it is blank, uninitialized, unformatted or empty; sort of like if you bought a blank phonograph record with no grooves. You could use the same diskette on almost any computer; but before it can be used, it must be initialized or formatted for that computer. An uninitialized diskette is useless. A disk which has been formatted on another type of computer is generally useless to the TI-99.

The standard SSSD disk consists of 360 sectors..... BUT you can only use 358 of them! Have you ever seen a catalog of an empty disk? Did it say there were 2 sectors used, and 358 available but the disk is empty? I will now attempt to explain the trick that seems to rob you of 2 sectors from every disk: sector zero is used for disk information and sector 1 is used for file directory information. Those first two sectors can never be used for another purpose. (I know, never say never!) And most catalog programs subtract those sectors from the total; they can not be used for file storage, only for file location information.

Much information is stored in the first sector; the first ten bytes contain the disk name. The name is generally required when the disk is initialized, but can be changed later with a sector editor. Spaces, periods or whatever you want can be in the disk name, although they cannot be useful. Also in the first sector is, in hexadecimal, the number of sectors available on the disk; 168 for SSSD, 2D0 for DSSD/SSDD or 5A0 for DSDD. (Myarc's DD capacity varies) In the first sector are the 3 letters DSK. Many programs look for this code, and if it's not there, the disk cannot be used. (Some companies have changed this to prevent their disks from pirates.) Also in the first sector is a byte which indicates how many sides are used; another indicates density (single, double, quad); another byte contains a "P" if the disk has been protected, a space otherwise. This disk protection is ignored by many programs but is another method of foiling pirates. The final piece of information in sector zero is a map of the disk: one bit for each sector- a zero if that sector has been used, a one if it is available. The layout for this disk map is used when data is written to the disk in order to ensure that previous data is not destroyed, also it is updated when a file is deleted. (Files are not really erased, and often can be unerased.)

Sector 1 contains the file directory information. On a formatted disk, (no files) sector 1 contains all zeros. Files consist of two parts; a single directory sector and one (or more) file sectors. If you find a file with size=1, you have no file! All you have is a directory sector with the name and file type. When a disk file is opened, the file information is written on the first available directory sector. Directory sectors normally begin at sector 2 and are mostly unused space. When data is written to the file, file sectors normally begin at sector 34, the directory sector is updated with information telling where the file begins. Normally file sectors are contiguous, but not necessarily so; file data is written to the lowest numbered available data sector. If all sectors from 34 to the end of the disk are used, directory sectors are used. If all directory sectors are used and you still have more data to be written, an error message results; the disk is full.

In the process of writing a file to disk, the controller writes the file information to sector 2 (name, file type), changes the sector zero bit map to show that sector 2 is now used, and writes a 2 in sector 1 to indicate that sector 2 is a directory sector. When data is written to the file, the controller begins writing data in sectors 34, 35, 36, etc. using as many sectors as necessary. When the file is closed, information is written to the directory sector showing where all file sectors are and the sector zero bit map is updated to show the sectors which have been used. If something happens to interrupt the process before the file is closed, the data is on the disk BUT you can't get to it! (Never say never)

When you catalog a TI disk, the list is in alphabetical order; not the files, just the list. The files are on the disk in the order they were created; first come, first served. Sector one contains numbers; the sector numbers where file directory sectors can be found, maximum 127. (There are 256 bytes or 128 16-bit words, but 0000 MUST be at the end of the list)

When a new file is opened, the filename is written in the next available directory sector. Then, each word in sector one is examined; if non-zero, that word tells where a directory sector can be found. The directory sector is read to obtain the filename and compared against the new filename. If the new filename is alphabetically before the previous filename, sector one is rewritten with the new information.

When a file is deleted, only sectors zero and one are changed! Sector zero to make the files sectors available, and sector one to remove the file from the directory. (Directory sector numbers are shifted down to fill the open slot) To restore a file, all that is needed is to find the directory sector, restore sectors zero and one. This can only be done if no more files are written to the disk.

When a file is renamed, only two sectors are changed! The file directory sector, of course, because that's where the new filename is stored. Sector one will be revised to position the new file (alphabetically) in the disk catalog. That is another protection trick; rename the file, then when the program runs, check for the directory sectors to be in the proper places. A file-by-file copy might have rearranged the sequence of the files. Try this: copy a disk that has a number of files on it; then check sector one with a sector editor, sector one will look something like this; 000200030004000500060007...etc.

Some disks have had their bit-map altered. Barry Traver, in the first issue of his disk magazine Genial Traveler, put a message on one of the sectors because he had 357 sectors of files. He then changed sector zero to show that sector was used. Nobody noticed that the files added to 357 but the catalog showed 358 sectors used. Plato disks are similarly modified; look at a catalog and you wil see one small file, although all sectors are used.

TI Forth disk contains three files: FORTH, FORTHSAVE, and SYSSCREENS. These files occupy the entire disk, SYSSCREENS overlaps beyond sector 357 (the normal end of disk space) and fills sectors 5 to 33. Notice that although these are usually directory sectors, they can also contain data.

Those sectors beyond 34, normally data sectors, might also be directory sectors. Check a disk with more than 32 files on it, looking at sector one, you will see that other sectors can be directory sectors. The only rule is regarding the first two sectors; they must be used for the disk directory. BUT..... PRBASE and TI-FORTH users know that there are always exceptions. As I previously mentioned, many programs look at sector zero for the bytes that should contain DSK. If the formatting process is interrupted before DSK is written, a disk manager program will say it's not formatted. Try this; format a disk but open the door on the drive before it finishes. Then catalog the disk, paying close attention to the 'not initialized' message. It will come up quickly because the manager recognizes that it is formatted but bogus.

One last topic before your assignment; fractured files. No, not broken files, blown files, but fractured files. These are fairly normal, relatively harmless, and impossible to avoid. A file is fractured because of how data is written to disk; the next available data sector is used. In other words, the lowest-numbered data sector that is available according to the bit map will be used for new data. A file can be fractured if there is a 'gap' in the bit map that is smaller that the size of the data to be written. A file can become fractured if more data is added to it and there is another file immediately after it on the disk. A file can also be fractured if it runs past sector 357 and directory sectors will be used. Fractured means only that the file is not written to consecutive sectors. It is a slight headache since the file directory sector must hold information about where all of the file is. The easy cure for fractured files is to do a file copy of the disk. This also has the benefit of re-ordering the directory.

Here is a project for you to try if you dare! USE ONLY COPIES OF DISKS OR BE PREPARED TO LOSE THEM. Us a sector editor (DISCO, DISKMASTER, DISK+AID or DISK UTILITIES come to mind) and check your disks. You might also put write protect tabs on disks you can't afford to lose. Look around, see what you can find; see where the files are and compare a disk with its' copy. When you get brave; ON A COPY, try writing different info to sectors zero, one or find the file directory sectors and 'fiddle' with them. You can't hurt a thing as long as you are working on a copy. Try to restore a file-the hard way- try to delete a file. See what happens. The worst you can do is wipe out the data on a disk and that won't hurt a thing. GOOD LUCK.....BP

KEYBOARD STRIPS

G.Crawford, N.O.V.A.

Do you have a program with special functions and you don't have the keyboard strip for it?

If you have acquired programs by way of club auctions, garage sales or other persons selling their entire systems, the way I have, the completeness of what you buy is dependent upon the sellers personal organization. Of the two things that are purchased with missing paraphernalia, the keyboard strips are almost a sure bet to be gone.

Several of our members have asked our librarian for these strips and although our library does have a program which allows you to create your own keyboard strips, it doesn't tell you what belongs on the strip for each specific program.

This page is for you. It has several strips than can either be cut out or duplicated for your personal use. The black dot is the line that uses the **CTRL** key and the other one uses the **FCTN** key.

OOPS!	REFORMAT	SCREEN COLOR	NEXT PARAGRAPH	DUPE LINE	LAST PARAGRAPH	WORD TAB	NEW PARAGRAPH	NEW PAGE	WORD WRAP		● TI-
DEL CHAR	INS CHAR	DEL LINE	ROLL DOWN•	NEXT WINDOW	ROLL UP•	TAB	INS LINE	COMMAND/ ESCAPE	LINE #	QUIT	WRITER

A PRINTER COM	B BLOCK	C CENTER	H TAB	J RED SPACE	L NEWLINE	N NEW PAGE	M MID LINE	P PARAGRAPH	·MOVE· COPY·DELETE	UNDERLINE	● COMP-
DELETE & CANCEL PRT	INSERT	ERASE & HIGHLIGHT	↓ V	BEGIN	^ ↑	MENU	END	EDITOR COMMANDS	V COLOR	QUIT	ION

HOME	TAB	NXT UNL CELL	FORWARD CHAR	FORWARD WORD	CHANGE WINDOW	REL ABS REF				CANCEL	● MULTI
LOWER RIGHT			BACK CHAR	BACK WORD		HELP	RECALC	BACK SPACE	DELETE FORWARD		PLAN

DELETE CHAR	INSERT CHAR	DELETE LINE	ROLL UP	NEXT SCREEN	ROLL DOWN	TAB	INSERT LINE	ESCAPE		QUIT	● ES/
DELETE	INSERT	ERASE	CLEAR	BEGIN	PROCEED	AID	REDO	BACK		QUIT	AS

					I		INSERT BLANK LINE				● FORTH
DELETE	INSERT	ERASE	NEXT SCREEN	NEXT WINDOW	LAST SCREEN	ERASE TO END OF LINE	COPY LINE	EXIT EDITOR			

			G	R	A	P	H	X			●
SLOWER	FASTER	DRAW	ERASE	NO HELP	ZOOM	COLORS	LINES	CIRCLES	COPY	MENU	

A°CLEAR I	B°CLEAR C	C°INPUT D	D°DRAW	E°ALPHA N	F°FILL	H°H OR V	I°INVERT	K°K-LINE	L°LINE	M°MIRROR	● TI
N°SWAP	O°CIRCLE	P°POINT	Q°DISC	R°RAYS	S°STORE	V°FRAME	X°BOX	Z°ZOOM	FCTN °PEN	FCTN.°SPEED	ARTIST

MODEM BAUD RATE	SPOOLER ON OFF	MODEM PARITY	MODEM PORT	PRINTER PARITY	PRINTER PORT	PRINTER BAUD RATE			TOGGLE FREEZE ON OFF		● FAST
NEXT SEND	# DUMP TO DISK	V CLEAR LOG	CANCEL	WINDOW	X TIMER	FORE GROUND	BACK-GROUND	LOG ON OFF		QUIT	TERM

SPEAK	OUTPUT	CANCEL	TRANS	WRAP	CASE	PAGE	.		EXIT		● TE II
										QUIT	

THE "ZENOBOARD"............... (A new piece of hardware for the TI-994A)
by Eric Zeno (West Penn 99'ers) December, 1988
(written by John F. Willforth)

Have you had your console lock-up after you had just about finished keying in a long XBasic program, or have you had a game running under XBasic just stop when you were about to get your all time best score, or has a utility stopped as you were just about done entering the last of the names and addresses? The "ZENOBOARD" (as I call it) some reference to "ZUCKERBOARD" will accept the chips from your XBasic cartridge, as well as a 32K Byte Static Ram chip, and a battery backed clock circuit and the chips from your speech synthesizer. It will also have GROM chip locations, so you can install your most used GROM based cartridges right in your console. The Extend Basic is a very common cause of lockups, and can now be installed inside the console, almost eliminating lockups. Included , you'll get the installation instructions to aid in the installation of this board inside the console and the above mentioned items on this board.

Eric plans to offer this board for less than $15. The intent here is to find out if there is enough genuine interest or need for the board for him to continue. If you would like to see one of these, and would support it write or call Eric at the address below.

Specifications:

* Fits inside console above CPU board and solders directly to back of GROM conn., with just a few wires to the CPU board.
* Requires no additional power.
* Includes RESET circuit
* Can be expanded or configured as the user requires.
* Supports 32K STATIC RAM
* Supports Battery-backed CLOCK
* Supports SPEECH SYNTHESIZER
* Supports EXTENDED BASIC
* Supports additional switch selectable GROM
* Do-It-Yourself low cost

>>> SOME TECHNICAL ASSEMBLY REQUIRED<<<

DO not order at this time, because the idea is quite attainable, but there may not be enough demand to complete the project. Write/call:

ERIC ZENO (412) 371-4779
414 HIGHLAND RD.
PITTSBURGH, PA 15235 (SASE Please!)

NOTE: I didn't have a more recent drawing of the board at this printing, but I didn't want to delay passing this new hardware effort to you until January. Eric needs to know soon so that he can take advantage of the longe winter nights to finish the board and get it out to you.
JFW

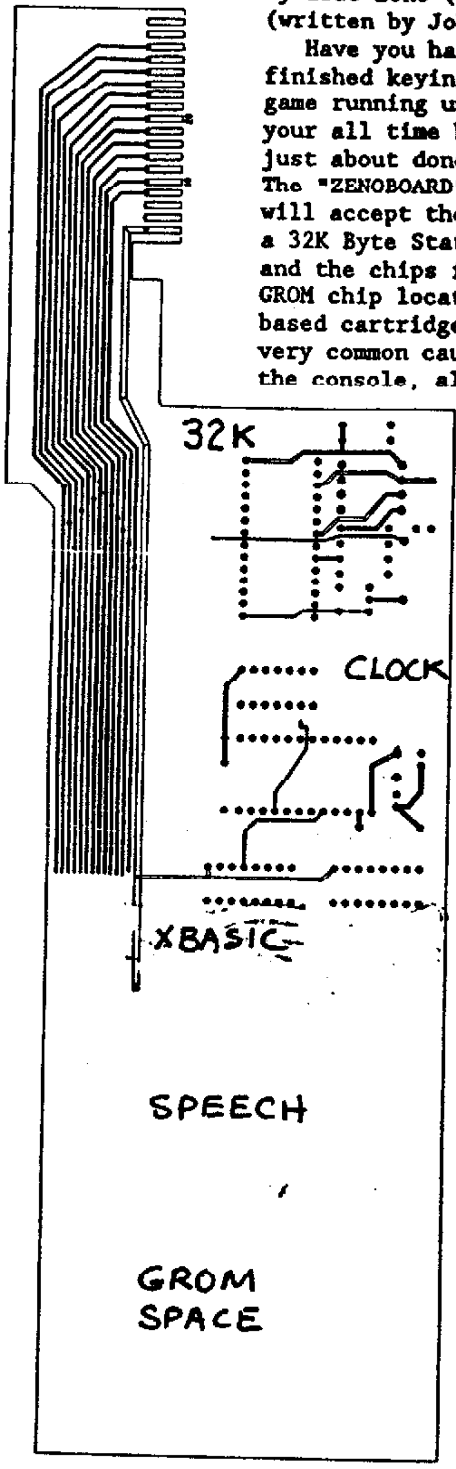

FIG. 1 THE ZENOBOARD

From NUTMEG TI-99ERS NEWSLETTER

DO YOU HAVE A TI DISK CONTROLLER CARD?

Upgrade your system with a

DISK CONTROLLER UPGRADE KIT

Users of the TI disk controller card can now enjoy some of the features of the newer disk controllers at a fraction of the cost! By installing a few components, the TI card can be upgraded without the worry of software compatibility or the purchase of a new disk controller.

The TI disk controller upgrade kit adds the ability to connect a fourth single or double sided floppy disk drive to the system, just like the newer cards. The ability to use lower case drive names like "dsk1" is not only more convenient, but it allows the user to access a floppy disk drive if a RAM-Disk has been set up to use the same drive number. As an option, the head step time of the drives may also be decreased from 20 milliseconds to 12 milliseconds to reduce the time required for some disk operations. This upgrade does not provide double density.

This upgrade is for programmers, data base users, bulletin board operators, or anyone who needs access to more online disk space and extra features but does not want the extra cost of a non-TI card. With the low cost of disk drives (especially drives removed from a system upgraded to half-height units), upgrading the TI disk controller can be the most cost effective way to enhance your system. The cost of the upgrade kit is substantially less than the $150-$175 you can pay for a new disk controller (not to mention the cost of your TI card which then gets to sit on the shelf).

Installation of the upgrade kit is accomplished by replacing the two ROM chips on the TI disk controller and stacking a few additional chips. The ability to solder and desolder components is required. No modifications to the disk drives are needed. An illustrated installation manual is included with the kit.

If the faster head stepping is desired, the drives used with the system MUST be capable of handling the faster speed. Faster head step times should not be requested unless the user knows that the head step time can be accommodated by ALL the drives in the system. There is no extra charge for faster head stepping.

To order, send $19.95 (cash or check, please) to:

John Guion
11923 Quincy Lane
Dallas, TX 75230

Please state whether normal or fast head stepping is desired.

DO YOU USE A TI RS232 CARD?

Upgrade your system with an

RS232 UPGRADE KIT

With an RS232 upgrade kit, users can expand the software compatibility of their systems as well as add convenient features. Once installed the upgrade kit provides two new devices:

"TP" - In addition to the PIO and RS232 devices, TP is added to provide full emulation of TI's Solid State (thermal) Printer by an Epson compatible printer. This allows the user to run software which is set up to access only the Solid State Printer. Module software that was previously unable to print with a parallel or serial printer will use these devices as though the Solid State Printer was attached. Additionally, program listings can be made in 60, 32, or 28 columns so the printout is just as it appears on the screen. Any option available to the original Solid State Printer may be used with the TP feature. The TP option even allows the user to write one-line screen dumps for Extended BASIC programs, complete with graphics!

"SIO" - Either serial port may be set up to respond as SIO. The port, baud rate, number of data bits, and parity may be specified when the upgrade kit is ordered. Instead of having to enter a long device name such as "RS232/2.BA=4800.DA=8.PA=O" each time a port is accessed, "SIO" may be used. Not only is this more convenient than typing a long device name each time the device is accessed, it allows the user to easily modify programs that previously used only PIO for printer output. All the user needs to do is change occurrences of PIO in the program to SIO. Software switches such as ".LF" and ".CR" may be added to SIO if required by a particular program.

The TP, SIO, and PIO device names may also be entered in lower case so that errors caused by having the Alpha Lock in the wrong position are eliminated. None of these features interfere with the normal operation of the RS232 card.

The parameters for both "TP" and "SIO" ports must be specified when ordering to correspond with the configuration of the user's system. Installation of the upgrade kit requires the ability to desolder and replace one ROM chip on the TI RS232 card.

To order, send $14.95 (cash or check, please) to:

John Guion
11923 Quincy Lane
Dallas, TX 75230

Please use the enclosed form to specify your system configuration.

MACFLIX

Deanna Sheridan - Northcoast 99ers

We thought we had "arrived" when Travis Watford developed MAXRLE and we could view and download the many digitized RLE pictures available on CompuServe, GENie, etc. MACFLIX, written by J. Peter Hoddie and distributed by Genial Computerware gives us another powerful graphic viewing program for MacPaint pictures.

These are usually full-page pictures and even on my MSDOS machine with 80 columns, I am unable to see the entire picture at one time. I have downloaded lots of these pictures, but was unable to find any real use for them, I have been unable to find at least on public bulletin boards, any utilities to transfer them into other programs where they could be used for clip art.

MACFLIX for the TI lets you view them, print them and clip them. Just as on my Leading Edge, it is impossible to see the entire screen at one time, and you must scroll across and up and down. The best way to get an idea of what it looks like is to make a printout. The program supports Epson compatible and Prowriter printers. You can print in your choice of three densities, but are warned that option 1 will cut off part of the picture, and option 3 will make it look elongated. So, with option 2, print out and see what you have.

You can save the portion of screen in view to disk in TI-Artist format. I found a disk of Christmas characters which I did just that and retrieved 8 clips for a Christmas disk. There seems to be a wide variety of pictures available and we will no doubt soon have a special section in our library just for Mac-Paint pictures (who would have ever thought?). Most of the ones I have are drawings rather than digitized pictures like the RLE's. Thus, those which are "clippable" are much clearer and of more use than RLE's.

The docs state that if you have a CorComp or Myarc disk controller and PC Transfer, you can take IBM disks with MacPaint pictures and transfer them for the TI. I don't have the right disk controller or PC Transfer, but I do have a cable between my TI and my Leading Edge. I fired up both machines with a terminal emulator program in each. I sent some MacPaint pictures over via Xmodem, which results in a Dis/Fix 128 format. I held my breath, fired up MACFLIX and tried loading one of the files. There it was, just the same as on my other machine. Suddenly I found myself with 3 disks of MAC pictures for my lowly TI. I will download some more from the local bulletin board to which I subscribe and we should soon have a good MAC library for the club.

This program is written in assembly and only $10.00 plus $1.00 SH from Genial Computerware, P.O. Box 183, Grafton, MA 01519. Note: I sent a personal check because I was in no hurry to get the program and it took six weeks. If you want faster delivery, I would suggest a bank check or a money order.

When I first wrote the above, I had not explored all the possibilities this program offers for us TIers. Did you ever think that there would be a day that you could utilize the various graphics for PrintMaster, PrintShop, Newsroom, etc. on your TI? I have even discovered that I can reverse the procedure and use my TI graphics on those MSDOS programs.

I found an MSDOS program called "ICONVERT". This program converts PrintMaster, PrintShop, Newsroom, MacPaint, RLE's and many more from any of the above to any of the above. I have several libraries of PrintMaster graphics and decided to give it a whirl. ICONVERT will take a set of PrintMaster graphics which usually are 120 individual graphics and automatically convert the first 50 of them to MacPaint format. I can convert the remaining by choosing the graphics individually. Thus it takes 3 files of MacPaint to use up one set of PrintMaster graphics. You can send them over just as described above. They are saved on a sheet which can be "clipped" out to TI-Artist. I have 26 of these files already and am just getting started. By next meeting, I may have 20 disks of new clipart.

Since this worked so well, I wondered if I could send some of my TI graphics over to the Leading Edge for use with PrintMaster (the only program I have). I took some TI-Artist files in program format. Loaded them into MAXRLE and resaved them in DF/128. I used the same method as above to send them over to the LE. I was able to view them with one of the RLE viewers I have for that machine. With ICONVERT I can put them in PrintMaster format and use my TI graphics over there.

People who got rid of their TI's when they got MSDOS machines are going to be sorrrry.

THE UGLY DUCKLING

Monitoring the Geneve

Word processers and spreadsheets are useful. Telecommunications programs are fun. Plato is educational. And graphics programs run on the Geneve are downright spectacular.

Unfortunately, the graphics capability of our machine is wasted without the proper monitor. The search for the correct monitor can be a bit trying but the results are certainly worth the effort.

Myarc designed the Geneve to support the standard TI video display cable and modulator and provided for an RGB monitor upgrade. If a black and white or color TV is used for a video display with the Geneve, the results are marginal at best. As a fairly serious TI'er I'd been using a composite color monitor with my system but I found that the composite monitor presented some serious problems when I used it with the Geneve. The graphics displays were good but the composite monitor could not resolve the 80 column text available with the Geneve. Since the 80 column display was one of the major selling points of the 9640, it was time to resort to plan B.

I found a cheap Samsung monochrome monitor at the Dallas flea market and it worked well with the 80 column display. However the Samsung monitor lacked a speaker and when I wanted to fire up Myart, an Extended Basic game, or some other program that relied on color graphics I had to reach into my rat's nest of cables and switch the Geneve's output back to the color monitor.

This year I decided to upgrade my system by adding an RGB monitor. I learned that the RGB output from the Geneve is an ANALOG signal which serves to complicate matters. Since the rest of the world is IBM compatible the major task seemed to be to find out what "analog RGB" meant to the IBM community. The IBM users I work with talked a great deal about a Greek strong man but I was eventually able to determine that analog RGB meant VGA or Video Graphics Adapter in IBMese. I also discovered that VGA stood for Very expensive Graphics Adapter and most VGA monitors cost 400 to 450 dollars. Maybe the amber Samsung monitor wasn't so bad after all.

Andy Lee had good success with a Radio Shack CM-8 monitor with his Geneve so I took a ride over to Sooner Mall. The CM-8 is an analog RGB monitor designed to be used with the CoCo (of all things). It has an internal speaker and is reasonably priced at about 300 dollars.

The RS monitor requires a split horizontal and vertical synch signal while the Geneve provides a composite synch signal for both horizontal and vertical. Andy designed and built a simple circuit to split the Geneve synch signal into its horizontal and vertical components. I'm not much of an electronics technician - I think that a transistor is a can with smoke inside it - but after two or three tries I managed to get Andy's circuit to work. The directions to build the synch circuit can be found on our bulletin board and I'd like to praise Andy for the excellent job he did on them. The problems I had with the set-up were due to my incompetence and not Andy's docs.

The CM-8 monitor has "pretty good" resolution. Its not as good as the very high dollar monitors I've seen on the Computer Aided Drafting systems at work, but for the price it's a reasonable alternative to looking at graphics in amber or trying to make out 80 column text on the composite color monitor.

Dave Lewis SOONER 99ers

"OOPS - HERE'S THE PROBLEM. SOMETHING'S CAUSING SHORTS IN THE MAINFRAME."

FLONG, I'VE JUST BEEN LOOKING OVER YOUR VIDEO GAME SCORES. PLEASE, HAVE A SEAT.

Carl learns the risks of outscoring his supervisor on computer software games

Sooner 99ers
PO Box 61061
Oklahoma City, OK 73146

TI-99/4A
HOME COMPUTER
TAKE THIS "ORPHAN" OUT OF THE CLOSET AND JOIN THE SOONER 99ERS USERS GROUP!
UNLEASH THE TREMENDOUS POTENTIAL OF THIS ONCE UGLY DUCKLING WHICH HAS BECOME A SWAN
WITH THE TI-99/4A YOU CAN DO ~
*WORD PROCESSING *GRAPHICS *GAMES
*SPREADSHEET *MUSIC & SPEECH
*PROGRAMMING IN 6 LANGUAGES
*CLUB MEETINGS 2X/MONTH *$12/ANNUAL FEE
*SOFTWARE LIBRARY *WORKSHOPS & TUTORIALS
*BULLETIN BOARD
CONTACT
DR GARTH POTTS (CLUB PRES.)