692 9003

Vol.8 No.03 March 1990

NEW HORIZONS

NEWS LETTER

NORTHWEST OHIO COMPUTER CLUB FOR THE TEXAS INSTRUMENTS 99/4A

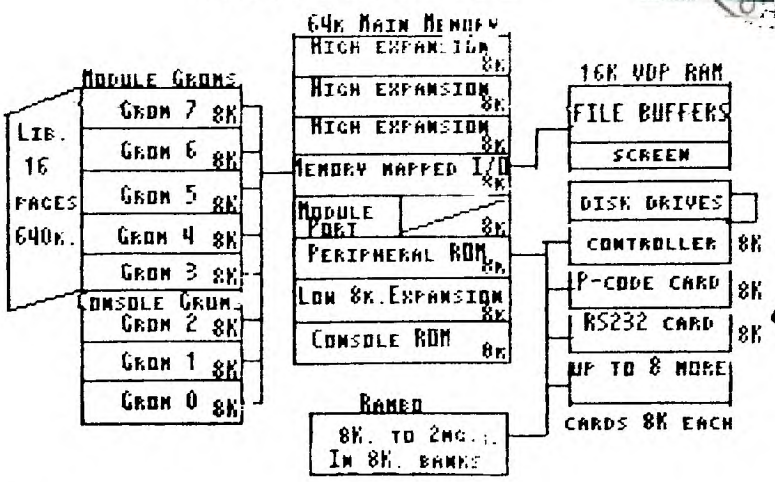
AND THE MYARC GENEVE 9640 PERSONAL AND HOME COMPUTER

This newsletter is published by New Horizons TI-99/4A Home Computer User's Group. Material may be reproduced without permission provided that the Author and the source are Acknowledged. For more information consult one of the following officers: Yearly Dues $15.00 per Family or Individual.....

THIS MONTHS MEETING MAR.10,1990 SATURDAY AT UNITY CHURCH 12:30 PM.
Behind Wendy's off Secor Road on Executive Dr.

TI-COM BBS.....	1-419-385-7484
PRESIDENT....BILL TIEF ADDRESS 5926 RAMBO LN TOLEDO. OHIO 43623	1-419-475-1775
VICE-PRES...CHARLES STROBELL	1-419-474-4128
SECRETARY..MARILYN SCHAFSTALL	1-419-882-6870
TREASURER..EARL W.HOFFSIS	1-419-475-0461
SOFTWARE LIB.JOHN&CHRIS DEWEY	1-419-475-3871
NEWSLETTER LIB. BURK MALLORY	1-419-882-6769
EDITOR....ROGER FEIMAUER	1-517-263-6
COMPILER..JUDY FEIMAUER	" .... "

MEMORY MAP FOR THE 99/4A



AIR MAIL

AIR MAIL

N.W. OHIO 99'ERS USER GROUP
%FIRST CHURCH UNITY
3535 EXECUTIVE PARKWAY
TOLEDO OHIO 43606
ATT EARL W. HOFFSIS

Edmonton 99'ers Computer
PO Box 11983
Edmonton, Alberta

PRESIDENT'S CORNER

Bill Tiep

Well, it's lent and I will give up my PRESIDENTIAL appointment. HUMOR. HUMOR. Please have a safe and sober St. Paddys day. This months CLUB disk is games. Some are new, some are old. Most should be enjoyable by someone. A. Andrews has suggested a questionnaire be sent to PREVIOUS members. This might help us to get a handle on our problems. If any one has software problems, bring it to club and we will try to get an answer for you. Someone may just have the correct answer. I had a message left on the board from GARY BOWSER of TORONTO, ONT. He said that the project to use COLECOVISION and SEGA cartriges was put on a back burner. These cartriges both use the same vidio ans sound chips as the T.I.. The hardware has turned out to be as big a project as a GENEVE and his resources are limited. He did develop a software version for the GENEVE except it only ran at 60% of normal speed. REMEMBER that the meeting of NEW HORIZENS will begin at 12:30 on MARCH 10, 1990 at UNITY CHURCH. HAPPY COMPUTING.....

```
100 CALL SCREEN(2):: FOR P=3
 TO 30 :: CALL CLEAR :: CALL
 COLOR(2,P/2+1,1):: FOR A=0
TO 5*PI STEP PI/36 :: R=11*S
IN(P*A/2):: CALL HCHAR(12-R*
SIN(A),17+R*COS(A),42):: NEX
T A :: NEXT P
```

Don Turner

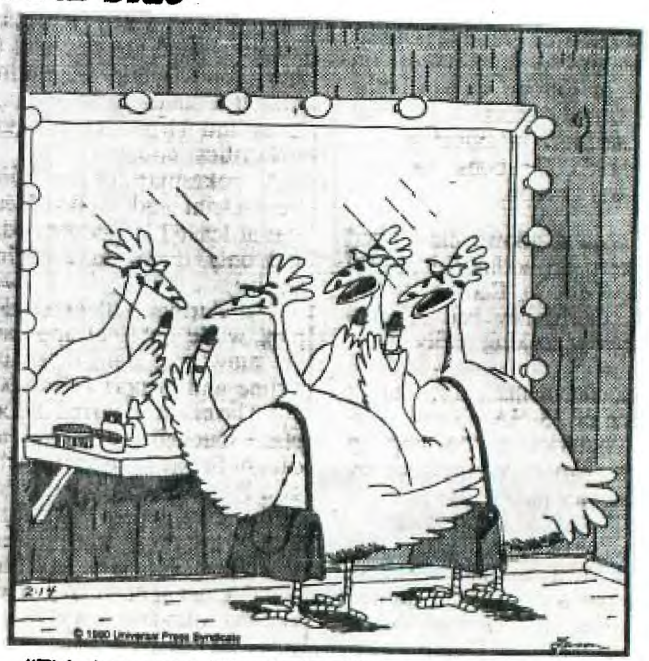
Modeming

Several days ago I needed to get a few files from Roger Feinauer. I could have drove to his house to get them but I really don't care for driving much. I needed them quite quickly soooo I called Roger up and we transfered them.

Seems like a simple operation but sometimes it gets rather tricky. At first We could not get the modems to synch. We fudged around trying all the modem commands to get them (the modems) to realize we wanted to let our computers do the talking. We both have Packard Bell modems. One would think that would make things easy. This was not true.

We used a little cleverness and set one modem to auto answer. Walla! We were in business. A two hour drive turned into a 15 minute call. We both used our favorite terminal programs and transfered the files.

The Far Side



"This is no use, Wanda. It's like they say — we just don't have lips."

Well another month has passed, and spring is just around the corner. Which brings us to this months article. There was a lot of discussion of what we could do in the meetings to make the meetings more interesting. With not very much resolved. I, Guess the bottom line is what you as members find interesting? If it isn't brought to the attention to the Officers of the Club how can they possibly know with out the members input. There was a suggestion that we go back to the basics. A lot people in the club wanted us to go back and review some of the old club disk, but which ones we at the club as a unit must must make that decision. Well, with this I leave this to the membership bring your ideas,wants,needs,and of course help to the March meeting.

A lot of persons brought up that they have a lot of trouble working with disk files. There are a lot of different formatts and how can you tell what software will run from what? Example how do you know what will run what program? All the following show up as program on a catalog;an XB program,EAopt#5,TI-BASIC program,the Adventer programs,also some files for other programs can show up as program on a catalog. So whats a person to do? Don Turner said last month that if the people where interested he would give a demo on the TI-DISK-OPERATING-SYSTEM. So this alone showed make you interested in coming to the March 10 meeting.

Last month I said I had a nunber of differnt programs running from Gram space. Well untill the last few years this area of our computers wasn't accessible. But, with advent of devices such as Gram Kracker,P-Gram,and the Gramulator. we now have access to a very large and powerfull area of memory, that only TI could access. My filling on this was when TI made the computer they new things would need changing and upgrading from time to time so they added a place in memory you could add to the system without having to rewrite the whole system over each time it was up graded. So this is why they put the little talked about Module Library in the system . So to upgrade the computer you just add to the library, without rewritting the whole system. Then by adding the proper Calls to the new routines you have expanded the computer.

To give you a little insight here is a sample of the memory map of grom/gram space used by all TI's;

CONSOLE GROM/GRAM

GRAM 0 >g0000 >g1FFF
TI OPEATING SYSTEM

GRAM 1 >g2000 >g3FFF
TI BASIC

GRAM 2 >g4000 >g5FFF
REST OF TI-BASIC

CARTRIDGE GROM/GRAM

GROM 3 >g6000 >g7FFF

GROM 4 >g8000 >g9FFF

GROM 5 >gA000 >gBFFF

GROM 6 >gC000 >gDFFF

GROM 7 >gE000 >gFFFF

BANK SWITCH ROM/RAM
USED BY XB, 4A/DOS,
AND THE SUPERCART ECT.

BANK 1 >c6000 c7FFF
can be used as rom/ram

BANK 2 >c6000 c7FFF
can be used as rom/ram

---

note bank 1 can also
be call GROM 9,
AND bank 2 can be
call GROM A

Well I said last month I had the EA and Editor and assembler files along with 4A/Dos , Boot all in a space that PG-Ram card could use. Well here is how I did it. To start with load E/A cart which will load at GRAM 3 g>6000 thru g>7FFF, and load E/A utility files Editor and Assembler at Grams 4 and 5 address g>8000 thru g>BFFF. To do this last part you need a program call EARAMDISK put out by M.G. for the Gram Kracker. or know some one who has a G.K., by running this file it will prompt you to put in the Editor and Assembler files in drive #1, and load them to Grams 4 and 5. After this call up the EA option #3 loader and load the 4A/DOS loader files, next load Boot with EA opt #5 loader. After it is loaded press FCTN 5, then FCTN 9, then FCTN 9 one more time. The software will ask what Gram ? Answer #6 this will put Boot at a Starting address of g>C000 . One last item 4A/DOS use CPU Ram c>6000 the bank switch ram so if you look at the memory map you see how everything ties in.

One last item I have been able to download files from TI-COMM , but still have been unable to upload at this time. If you have a short article I can still get it if you leave it to me in the E-Mail section of the board. But remember this area, you must type it in while you are on line. If you are calling longdistance could be expensive.

Remember all newsletter article must be in my hands no latter the 14 days before the next meeting. send them to Roger Feinauer, 166 S. Mckenzie ST. Adrian, MI. 49221.

```
1 CALL CLEAR :: CALL QUITOFF
 :: ON BREAK NEXT :: ON WARN
ING NEXT :: PRINT "     FOUR
 IN A ROW,OR MOST COMMONLY K
NOWN AS'CONNECT FOUR'" :: PR
INT
2 PRINT "     THE GAME CONSIT
S OF STACKING X'S AND 0'S (T
HE COMPUTER IS '0') UNTIL ON
E OF THE PLAYER GETS FOUR IN
 A ROW V"
3 PRINT "ERTICALLY,HORIZONTA
LLY,OR DIAGONALLY" :: PRINT
4 DIM B$(8,8),V(16):: FOR Z1
=1 TO 16 :: READ V(Z1):: NEX
T Z1 :: X$="X" :: O$="0"
5 DATA 1,100,500,1E20,1,800,
4000,1E20,1,75,900,1E18,1,45
0,3000,1E18
6 FOR I=1 TO 8 :: FOR J=1 TO
 8 :: B$(I,J)="-" :: NEXT J
:: L(I)=0 :: NEXT I :: IF H=
0 THEN PRINT "DO YOU WISH TO
 GO FIRST(Y/N)" :: CALL KEYS
("YyNn",C):: CALL CLEAR :: G
OSUB 7 :: ON C/2 GOTO 9,12 E
LSE GOSUB 7 :: ON H GOTO 9,1
2
7 FOR I=8 TO 1 STEP -1 :: FO
R J=1 TO 8 :: CALL HCHAR(9-I
,J+2,ASC(B$(I,J))):: NEXT J
:: NEXT I :: DISPLAY AT(9,1)
:"12345678" :: DISPLAY AT(10
,1):RPT$(" ",128):: RETURN
8 DISPLAY AT(10,1):"ILLEGAL
MOVE,TRY AGAIN" :: FOR T=1 T
O 500 :: NEXT T
9 DISPLAY AT(10,1):"A NUMBER
 BETWEEN 1 AND 8:" :: ACCEPT
 AT(10,26)VALIDATE(DIGIT)SIZ
E(1):M :: IF (M<1)+(M>8)THEN
 8 ELSE H=L(M):: IF H>7 THEN
 8 ELSE L(M)=H+1 :: H=H+1 ::
 B$(H,M)=X$ :: GOSUB 7 :: P$
=X$ :: GOSUB 25
10 FOR Z=1 TO 4 :: IF S(Z)<4
 THEN 11 ELSE DISPLAY AT(12,
1):"YOU WON" :: H=1 :: GOTO
33
11 NEXT Z
12 M9=0 :: V1=0 :: N1=1 :: F
OR M4=1 TO 8 :: H=L(M4)+1 ::
 IF H>8 THEN 21 ELSE A=1 ::
P$=O$ :: W=0 :: M=M4
13 GOSUB 25 :: FOR Z1=1 TO 4
 :: N(Z1)=0 :: NEXT Z1 :: FO
R Z=1 TO 4 :: U=S(Z):: IF U-
W>3 THEN 23 ELSE T=U+F(Z)::
IF T<4 THEN 14 ELSE A=A+4 ::
 N(U)=N(U)+1
14 NEXT Z :: FOR I=1 TO 4 ::
 O=N(I)-1 :: IF O=-1 THEN 15
 ELSE I1=8*W+4*SGN(O)+I :: A
=A+V(I1)+O*V(8*W+I)
15 NEXT I :: IF W=1 THEN 16
ELSE W=1 :: P$=X$ :: GOTO 13
16 H=H+1 :: IF H>8 THEN 19 E
LSE GOSUB 25
17 FOR Z=1 TO 4 :: IF S(Z)>3
 THEN A=2
18 NEXT Z
19 IF A<V1 THEN 21 ELSE IF A
>V1 THEN N1=1 :: GOTO 20 ELS
E N1=N1+1 :: IF (RND*1)>1/N1
 THEN 21
20 V1=A :: M9=M4
21 NEXT M4 :: IF M9<>0 THEN
22 ELSE DISPLAY AT(11,1):"TI
E GAME" :: GOTO 33
22 M=M9
23 DISPLAY AT(10,1):"I PICK
COLUMN";M :: H=L(M)+1 :: L(M
)=L(M)+1 :: B$(H,M)=O$ :: P$
=O$ :: GOSUB 7 :: GOSUB 25 :
: FOR Z=1 TO 4 :: IF S(Z)<4
THEN 24 ELSE DISPLAY AT(11,1
):"I WON" :: H=2 :: GOTO 33
24 NEXT Z :: GOTO 9
25 Q$=X$ :: IF P$=X$ THEN Q$
=O$
26 D2=1 :: D1=0 :: Z=0 :: GO
SUB 27 :: D1=1 :: D2=1 :: GO
SUB 27 :: D2=0 :: D1=1 :: GO
SUB 27 :: D2=-1 :: D1=1 :: G
OSUB 27 :: RETURN
27 D=1 :: U=1 :: T=0 :: Z=Z+
1
28 C=0 :: FOR K=1 TO 3 :: M5
=M+K*D1 :: L1=H+K*D2 :: IF (
M5<1)+(M5>8)+(L1<1)+(L1>8)TH
EN 31 ELSE D$=B$(L1,M5):: IF
 C=0 THEN 30
29 IF D$=Q$ THEN K=3 :: GOTO
 31 ELSE T=T+1 :: GOTO 31
30 IF D$=P$ THEN U=U+1 :: GO
TO 31 ELSE C=1 :: GOTO 29
31 NEXT K :: IF D=0 THEN 32
ELSE D=0 :: D1=-D1 :: D2=-D2
 :: GOTO 28
32 S(Z)=U :: F(Z)=T :: RETUR
N
33 DISPLAY AT(14,1):"PLAY AG
AIN?(Y/N):" :: CALL KEYS("Yy
Nn",C):: ON C/2 GOTO 6,34
34 CALL CLEAR :: PRINT "THAN
KS" :: PRINT TAB(8);"FOR" ::
 PRINT TAB(12);"PLAYING" ::
PRINT TAB(20);"WITH" :: PRIN
T TAB(25);"ME" :: 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 mentibned, 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