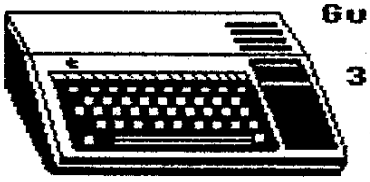
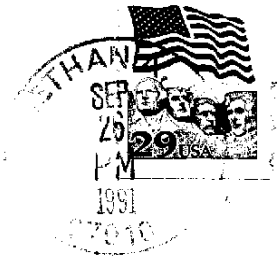
GUILFORD 99'ERS NEWSLETTER

SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER



GUILFORD 99'.ERS UG
3202 CANTERBURY DR
GREENSBORD NC
27408



TO:

George von Seth, Pres. (292-2035) Bob Carmany, Newsletter Ed (855-1538)
Tony Kleen, Sec/Treas (924-6344) Bill Woodruff,Pgm/Library (228-1892)
BBS: (919)621-2623 --ROS

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to:Tony Kleen c/o 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (Bob Carmany Ed)

OUR NEXT MEETING

DATE: October 1, 1991 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Program for this meeting will be a demonstration of the GRAMKRACKER by Roy West. In addition, there will be a demo of elementary Assembly Language programming. Stop by for a look at both--you might learn something!

MINUTES

The September meeting of the Guilford 99'er Users' Group was held on Tuesday the third at the Glenwood Recreation Center. There were five people present.

The meeting was two minutes late due to the late arrival of the Secretary/Treasurer (me). My normal 40 minute drive took an hour and a half! The Secretary/Treasurer reports were read and approved. as of this writing, the club has $168.29 in the treasury.

There was no old business.

New business: Bob mentioned that he now has the Epson MX80 GrafTrax chips on disk. This is the enhanced graphics modification for that series of printers. I mentioned that I was looking for RCA cables for my system. Radio Shack was suggested. Mack stated that he has a catalog at home. Later in the week, I called TI's 800 number for assistance in locating cables. Two phone references later, I talked to a company specializing in the TI99/4A. I ordered cables from them. If I receive their catalog prior to our next meeting, I'll bring it to the meeting.

Bob Carmany's topic for the evening was on the latest (and last?!?) version of Funnelweb (version 4.40). Bob paced us through the CONFIGURE software and gave an extremely good overview of the entire Funnelweb utility package. Thanks, Bob!

Respectfully submitted,

Tony Kleen

A/L MUSINGS

By Bob Carmany

Like a lot of people, I dreaded the thought of starting to learn Assembly Language. It seemed to be akin to trying to decipher an ancient Greek manuscript without a key. Unfortunately, some of it still is! Most of Assembly Language programming (hereafter A/L) has a more or less direct equivalent in XB or BASIC. That sure makes thing a lot easier, believe me!

The major advantage of using A/L is that you can do things with it that are beyond the capabilities of BASIC or XB programs. You can "talk" to the TI on more or less it's own level. This provides a tremendous advatage in speed! Nowhere is this clearer than in some of the sort programs that are around. By accessing the microprocessor directly, you can create the

most amazing graphics and sound. In fact, you can access the sound chip directly.

The disadvantage is that it takes lines of code to perform a relatively simple function. Unlike a high level language that might combine a couple of arithmetic functions into a single line of code, A/L might require 3 or more lines for the same thing. It isn't as bad as it seems. The lines of A/L code are short and compact and when the source code is assembled, it generally takes up less space than an equivalent XB program.

You will see what it is all about in the meeting. Basically, we will cover putting text on the screen and use some of the built-in routines provided in the E/A cartridge. As the evening progresses, I think that you will see that A/L isn't the monster that it appears to be. After all, the best way to learn something is to give it a try and learn from your mistakes.

On another related note, I had planned to re-write the disk initialization routines in PR-BASE so that they will work with all TI controllers in DS mode. I'm currently engaged in a three-way correspondence with Mike Dodd and William Warren to get my hands on the source code and see if the alterations can be made. It will probably show up here as the basis of an A/L tutorial if everything works out.

Just "hot off the press"!!! Those of you who have SPELLIT from Asgard Software might have noticed the monumental bug in the program that prohibitis the access of any RAMdisk above the address >1100. That means that if you have more than one RAMdisk (like I do), the second or subsequent ones couldn't be used by the program. Anyway, Ron Kleinschafer managed to rewrite the code to allow for the access of ANY disk drive --even RAMdisks above >1100. It works just fine. In fact, I even like the program better than the old Dragonslayer SPELLCHECK program!

GRAMKRACKER NOTES

GRAM KRACKER is a small box that just reeks of quality when you pick it up. The unit is quite heavy for such a small box. On the front are five switches which I will get to later. Between the box and where it plugs into the module port of the console is a module connector. The first switch is for turning GRAM KRACKER off (necessary for Atari cartridges and others without a reset line), and for resetting the computer back to the title screen. The second switch lets you use the operating system in the computer or using your own which is stored in an optional chip called GRAM 0. The third switch toggles between TI Basic and optional GRAMS 1 & 2. The fourth switch controls the banks that the modules are loaded into. The middle position of the switch is for write protection so you don't screw up what may already be stored. The final switch turns the internal software of GRAM KRACKER off, and this is necessary when using carts like PRK and STATISTICS.

When you turn on the computer, you will see GRAM KRACKER listed on your menu screen. If you select GRAM KRACKER, you are then presented with the following menu:

1 Load Module
2 Save Module
3 Init Module Space
4 Load/Save Console
5 Memory Editor

Selecting #1 will let you load a saved module. The screen will prompt you for enabling Banks 1 & 2 and when to turn on the write protection.

Option #2 lets you save a module that is either plugged into GRAM KRACKER or saved in GK's memory (you may have altered some of the data and want to re-save it).

The third option is for clearing out the space that the modules load into. It is possible to get garbage on the main menu screen if you do not do this. Of course, if you are loading multiple modules, you don't want to clear out the memory space.

Option #4 will take you to another menu which will let you Save or Load the console GROMS 0, 1, and 2. This allows you to alter the operating system and change TI Basic.

The final option brings up a screen similar to EXPLORER's memory window. You can use it to change information that is in memory, sort of like a super debugger.

GRAM KRACKER saves a module in 8K chunks, so with a cartridge like TI-Writer you will only have one disk file. On others like MULTIPLAN, you may have five. No matter which cartridge you save, each file will be 34 sectors in length. Once the

module is saved, you can then load it into GRAM KRACKER. The screen will prompt you to enable either Bank 1 and/or Bank 2 and when to turn on the write protection. You then return to the title screen and the module you just loaded will be listed. GRAM KRACKER will save/load with cassette, floppies, RAMdisk, or even a hard disk. The GRAM KRACKER's Save Module routine allows for chaining the loading of assembly language UTIL1 and Program Image type files along with the loading of the module space.

Included on the utility disk that comes with the unit is a program that lets you have the editor and the assembler load at the same time as your E/A cartridge. So, when you want to use the Editor, it's there instantaneously. You have to see it to believe it!!!!!

The section for loading/saving the console GROMS has eluded a successful trial for me, so I will have to jump over that. Documentation is listed as being temporary and this part of the GRAM KRACKER's operation is not explained real well. The three chips that are necessary for using this option can be purchased separately and installed by the user, or Craig will ship you a unit with them installed for a few dollars extra. I would suggest that you spend the extra $13 or so because you have to open up the box and separate two circuit boards, and this can be tricky for someone who is not experienced with electronic dodads.

The memory editor is useful for changing the contents of a saved cartridge. In the documentation, Craig tells how to convert TE2 to list 1200 and 300 baud on the menu instead of 110 and 300. Also he tells how to change the printer default in the TAX module to address PIO instead of RS232 and TP. The memory editor is a complete stand alone program and does not use any of the console GROM or ROM routines. This allows you to FILL, MOVE, or modify anything in any GRAM space, cartridge RAM space, or High Memory Expansion (>A000->FFFF).

As I said, the documentation is temporary, but Craig promises to have the final docs out in a few weeks. The docs also tell how to auto execute a module on power up (won't work with a CorComp controller). Also on the utility disk is a program which will change your copy of EXPLORER to be able to examine true GROM or pseudo GROM (the EXPLORER will not examine GRAM KRACKER as presently configured). There are also two files for letting you save a Basic program in GRAM KRACKER and having it appear on the title screen menu. This only works for Basic programs however.

There is a small lithium battery inside GRAM KRACKER which will retain the data. The battery appears to be the same one used by Mini Memory, so there should be no problem replacing it. The documentation also includes a complete memory map and some information for the "techies". Dealers are being limited in the quantities they can have on hand and Craig is having them built on a limited basis. (I understand that the first run was sold out a week after they started shipping) These things are so popular that Compuserve has added a special GRAM KRACKER section to the TI SIG.

GRAM KRACKER IDEAS

The documentation and utility disk included with GRAM KRACKER is only temporary and Craig promises to have the final stuff out soon. Among the items to be included are a catalog program which will reside in the GRAM KRACKER that will allow you to do a CALL CAT(DSK#.) from command mode of Extended Basic and not lose what you have in memory. Also included will be a special utility for TI-Writer which will let you chain the Editor and Formatter when you load it so they will be right there when you select from the TI-Writer menu. Some of the smart guys have already begun writing software for the GK and I think you'll see a lot of unique uses for it as time goes on. I predict that the GRAM KRACKER become standard equipment just like a loaded P-Box.

The docs and the utility disk that are included with GRAM KRACKER are temporary at this point. Craig tells me that final docs and disk should be out be around the middle of January.

FUN WITH THE GRAM KRACKER

After having received the GRAM KRACKER from Millers Graphics, the first order of business was to save all those valuable cartridges to disk and patch Extended Basic with the new "Call" subroutines provided by Millers Graphics. But there is more than that which the GK will do with its powerful memory editor and the ability to save the Operating System to disk and reload it as an editable GRAM.

One of the things that has irritated me for a long time is the character set provided with the 99/4A, particularly the lower case letters, which really are not lower case at all but rather a smaller version of the upper case set. For use with TI Writer and FastTerm I had patched in a disk file with a much better looking character set which was developed by the Titan

Users Group of Raleigh, NC. The objective now was to load this character set in GRAM so that it would be available permanently, also for Basic, Extended Basic and other applications. To locate the lower case character set table in GRAM proceed as follows:

From the GK main menu select 5) Edit Memory. Press Function 1 to select Grom/Gram, then Function= to toggle HEX display. Press Function 5, "Search" and for Start and Finish key in 0000 and 1000 respectively. Next press Function 9 and after disabling Write Protect key in the string 00003848484834. This is the Hex representation of the character "a" or ASCII 97. Move the cursor to the last "4" of the search string and press enter.

The Memory Editor should now show the start address of the LC character table. For my console this address was g0858. This was for a version 2.2 console. This address more likely than not will vary slightly from console to console.

Once having located the starting address, press Function 9 to place the cursor in the memory window at g0858 (or whatever your starting address may be) and carefully type in the following table of HEX values (without the addresses in HEX which are for reference only).

```
>0858 00003848484834 >085F 60203824242478 >0866 00003840404038 >086D
0C08384848483C >0874 000038447C403C >087B 18242070202020 >0882
00003C443C0438 >0889 60202834242424 >0890 1000701010107C >0897
08001808084830 >089E 20202428302824 >08A5 3010101010107C >08AC
00006854545454 >08B3 00005824242424 >08BA 00003844444438 >08C1
00007824382020 >08C8 00003048380800 >08CF 00005824202020 >08D6
00003C40380478 >08DD 20207820202418 >08E4 00004848484834 >08EB
00004444282810 >08F2 0000D454545428 >08F9 00004428102844 >0900
000044443C0478 >0907 00007C0810207C
```

A second change involves replacing the zero with a slashed zero (for this tip I am indebted to Millers Graphics). Following the search procedure described above, look for the string 0038444444444438 which should be located at g0720. Once located, replace it with 003844C54644438. Next look for 7C444444444447C which is the horrible looking squarish capital "O". Replace it with 0038444444444438 and now you have the original zero as a replacement for the square "O".

As a last step in customizing search for the letters TEX. (Should be at address g046E or thereabouts). Once located type in your first and last name. (If they are longer or shorter than Texas Instruments, abbreviate or key in blanks to make it fit). After having made all these changes, be sure to restore the GK write protect and return to the GK Main Menu and then to the title screen. If everything went fine, you should now have your personalized color bar title screen.

Select Extended Basic and key in: 100 FOR I=97 TO 122::PRINT CHR(I);::NEXT I

This is test to see that your lower case character set is ok. As a last step use the GK so save your customized operating system to disk. Have fun!!

[The previous article is a reprint of a 1986 column. The GRAMKRACKER is no longer avaialable --Ed.]

GRAM DEVICES

The GRAMKRACKER may not be available commercially but you might be able to find one as a 'previously owned' item. Ther are still some GRAM devices around --the most notable of which is the P-GRAM card produced by Bud Mills. The basic unit (72K) is $150 in kit form and $180 built. The 'souped up' version (192K) is $200 in kit form and $230 built and you can add a clok to either for an additional $20. It might be worth a look if the idea of a customized title screen and cartridges on disk appeals to you!

DISK CAT

By Herman Geschwind

For many of us with a sizeable program library the perennial problem is how to manage this library most efficiently: To

find a program quickly or to find a diskette with sufficient free space.

There are a number of disk cataloging programs on the market to assist in this chore which can be classified by (1) the program language used, Basic/Extended Basic or Assembler, and (2) the method that is being used, Snap-Shot or Perpetual Inventory. Disk cataloging programs based on Basic or Extended Basic commonly have two defects: (1) the number of disks that can be cataloged is restricted by memory limitations to fifty or sixty and the number of files to 500 or so. Unless the program does a good job of error trapping the danger always exists that the last file that was read in was one file too many and the whole system crashes. (2) Disk catalogers written in Basic tend to be slow and as memory fills up, these programs really slow to a crawl. A sort of file names which should be part of such a program can take as long as an hour for 500 files.

To manage a disk library of more than a few disks really calls for Assembler. If a program is well written up to 1000 file listings can be accommodated without straining the capacity of memory or disk. Likewise, internal processing is fast and sorts are a matter of minutes rather than an hour.

In terms of organization, a "Snap-shot" program means that all disks have to be read in in order to obtain a sorted listing of files and disks. A "Perpetual" program means that the entire library will be recorded in a disk file and from then on it is only a matter of deleting and reading back in those disks where changes have taken place.

The disadvantage of the "Snap-shot" method is that for one it is rather tedious to read in disk after disk and then it puts quite a strain on the disk drive mechanism to have disks inserted and removed in rapid succession . For ease of use the "Perpetual" method certainly is preferable .

Until now, even though there are any number of disk cataloging utilities, either commercial programs or "freeware" around, they were either of the "Basic" variety with their language constraints, or of the "Snap-shot" variety with its drawbacks.

A "Freeware" program "CATALOGING LIBRARY" by Martin Kroll, Jr. admirably succeeds in overcoming all these limitations: It is an Assembler program of the "Perpetual" type.

The opening menu of CATALOGING LIBRARY gives an indication of the many features that Martin managed to pack into this program:

A Add Disk to Catalog
B Delete Disk from Catalog
C Delete all "T"emporary Disks
D List Disk Summary
E List all Programs (Files)
F Search for and List a Disk
G Search for and List a Program
H Print Disk Summary
I Print all Programs
J Print all Program - No Page Division
K Search for and Print a Disk
L Change Printing Options
M Sort and Save Data
N Terminate Program

As can be seen by this menu, there is hardly an option not covered by this program: Any information desired is available either as printer hardcopy or as screen output. Printer setup is very flexible and not restricted to either PIO or RS232. Since all the information is memory resident, searches for either file or disk information are completed almost instantly.

If Martin can be faulted at all it is that there is no documentation for this program.

Thus there is no telling what the capacity limitations of this program are. I have tested it with 90 disks and 830 files without problems.

Another limitation of this program is that it requires the "Load and Run" option of either Editor Assembler or Mini

Memory. We can only hope that Martin will re-compile this program in Program image format so that it will be usable with an XB loader for a wider TI community.

A "cute" feature of this program is that whenever it branches to a sort, the message "By The Way..Have you sent $10 to.." appears on the screen rather than the more traditional "Now Sorting...". Certainly a nice way to remind the "free-loaders" among us that all programming efforts deserve their financial rewards.

All in all, CATALOGING LIBRARY is the best catalog program that I have seen yet, an I have tried many only to be frustrated either by lack of capacity, lack of speed or complex handling. Martin is to be commended for this excellent product for the modest price of only $10.

For further information or a copy, write to:

Martin Kroll, Jr., 210 Kaplan Ave., Pittsburgh, PA 15227

```
200 REM SPEECH SPLICER
210 REM
220 CALL CLEAR
230 PRINT "   +++WORD SPLICE
R+++": : : : :
240 PRINT "ENTER NUMBER OF Y
OUR CHOICE": :
250 PRINT :"  1 - JOIN TWO W
ORDS": :
260 PRINT "  2 - PRINT SPEEC
H DATA": :
270 PRINT "  3 - STORE NEW W
ORD ON DISK": :
280 PRINT "  4 - EXIT": : :
290 INPUT CHOICE
300 IF (CHOICE<1)+(CHOICE>4)
=-1 THEN 220
310 ON CHOICE GOTO 550,870,1
100,330
320 GOTO 220
330 CALL CLEAR
340 END

350 REM +++TUNCATE FIRSTWORD
+++
360 CALL CLEAR
370 INPUT "TRUNCATE HOW MANY
 BYTES?":BYTES
380 MAXBYTES=LEN(B$)-3
390 IF BYTES<MAXBYTES THEN 4
20
400 PRINT "TOO MANY BYTES. .
."
410 GOTO 370
420 IF BYTES>-1 THEN 450
430 PRINT "NO NEGATIVE NUMBE
RS"
440 GOD 370
450 L=MAXBYTES-BYTES
460 C$=SEG$(B$,1,2)&CHR$(L)&
SEG$(B$,4,L)
470 RETURN

480 REM +++SPEAK NEW WORD+++
490 CALL CLEAR
500 CALL SAY("",NEWDATA$)
510 INPUT "SAY AGAIN? (Y OR
N)":CHOICE$
520 IF CHOICE$="Y" THEN 500
530 RETURN

540 REM +++JOIN TWO WORDS SU
BROUTINE+++
550 CALL CLEAR
560 PRINT "ENTER FIRST WORD
TO JOIN"
570 INPUT FIRSTWORD$
580 IF FIRSTWORD$=LASTMADE$
THEN 610
590 CALL SPGET(FIRSTWORD$,B$
)
600 GOTO 620
610 B$=LASTDATA$
620 CALL CLEAR
630 PRINT "ENTER SECOND WORD
 TO JOIN"
640 INPUT SECONDWORD$
650 IF SECONDWORD$=LASTMADE$
 THEN 680
660 CALL SPGET(SECONDWORD$,D
$)
670 GOTO 690
680 D$=LASTDATA$
690 CALL CLEAR
700 PRINT "ENTER SPELLING OF
 THE NEW WORD"
710 INPUT NEWWORD$

720 REM
730 GOSUB 350
740 NEWDATA$=C$&D$
750 GOSUB 480
760 PRINT "   1 - CHANGE SOM
E MORE";"   2 - BACK TO MAIN
 MENU"

770 INPUT CHOICE
780 IF (CHOICE<1)+(CHOICE>2)
=-1 THEN 760
790 IF CHOICE=1 THEN 730
800 LASTMADE$=NEWWORD$
810 LASTDATA$=NEWDATA$
820 GOTO 240

830 REM  +++PRINT SPEECH DAT
A SUBROUTINE+++
840 REM
850 REM PRINTER STATEMENT
860 REM
870 OPEN #1:"PIO"

880 REM
890 CALL CLEAR
900 PRINT "ENTER THE WORD WH
OSE DATA YOU WANT TO PRINT--
   ": : :
910 GOSUB 1230
920 IF L=0 THEN 1070
930 VALUES$=""
940 PRINT #1: : :"THE WORD I
S **";WORD$;"**":
950 PRINT #1:"LENGTH=";L;"BY
TES": :
960 FOR I=1 TO L
970 VALUES$=VALUES$&STR$(ASC
(SEG$(F$,I,1)))
980 IF I/10<>INT(I/10)THEN 1
020
990 PRINT #1:"DATA ";VALUES$
1000 VALUES$=""
1010 GOTO 1030
1020 VALUES$=VALUES$&","
1030 NEXT I
1040 IF VALUES$="" THEN 1070
1050 VALUES$=SEG$(VALUES$,1,
LEN(VALUES$)-1)
1060 PRINT #1:"DATA ";VALUES
$

1070 INPUT F$
1080 GOTO 240

1090 REM  +++ADD NEW WORD TO
 VOCABULARY FILE+++
1100 CALL CLEAR
1110 PRINT "PUT THE DISK WIT
H ""WORDS""  FILE IN DRIVE
ONE."
1120 INPUT "PRESS ENTER WHEN
 READY ";X$
1130 PRINT : : :"ENTER THE W
ORD YOU WANT TO SAVE--  ": :
 :
1140 GOSUB 1230
1150 IF L=0 THEN 1200
1160 OPEN #1:"DSK1.WORDS",IN
TERNAL,APPEND,VARIABLE 254
1170 PRINT #1:WORD$
1180 PRINT #1:F$
1190 CLOSE #1
1200 RETURN

1210 REM
1220 REM  +++FIND WORD SUBRO
UTINE+++
1230 INPUT WORD$
1240 F$=""
1250 IF WORD$="" THEN 1300
1260 IF WORD$=LASTMADE$ THEN
 1290
1270 CALL SPGET(WORD$,F$)
1280 GOTO 1300
1290 F$=LASTDATA$
1300 L=LEN(F$)
1310 RETURN
```