-------WICHITA 99er's NEWSLETTER-------

----------WICHITA KANSAS----------

JCTOBER 1985 EDITOR GUY HULSEY

The next meeting of the WICHITA AREA 99ers USERS GROUP will be on FRIDAY the 18th. of OCTOBER 1985. The meeting will be held at the TRIANGLE BARBER SHOP at CENTRAL AND EDGEMORE. The meeting will start at 7:PM.

SUBJECTS TO BE COVERED THIS MUNTH

EXTENDED BASIC * PROGRAMING TIPS #1

FORTH * PART 2 BSAVE + LOADER PROGRAM

TI-WRITER * SPECIAL CHARACTORS

MULTI-PLAN * PRINTER CONTROLS + WORD PROCESSING

ASSEMBLY LANGUAGE * PART 2

P-CODE * SYSTEM REVIEW

PROJECT * BUILD LIGHT PEN

REVIEW * GRAPHX

NEWS * 9938 VDP

NEW PRODUCTS * GRAM KRACKER

FREEWARE PROGRAM LIST

There will be a software swap held at the home of Guy Hulsey on Saturday the 19th. of OCTOBER 1985. You are all invited to bring your programs and blank disk. The meeting will start at 9AM. and last till everyone goes home. I will have two systems running,one with two double sided disk drives the other with one double sided disk drive. I have aprox. 80 disk full of every kind of program you might want so come on down. To find my house come to Winfield KS. go South out of Winfield on Highway #77 cross the Walnut River go on south aprox. 1/4 mile, turn West cross rail road tracks continue west to the second road that goes south (Medowlark lane) turn south on this road, My house is the first house on the East side of the road. If you can't find it call me at 221-7148 and I'll come and lead you in.

Has anyone thought about Jerry's idea about the users group writing a program. It was suggested that each member would write a subprogram that could be merged into one large program. I think this sounds like alot of fun! If you think so to let Jerry know about it at the next meeting.

XTENDED BASIC TIPS, part 1

by Lee Wilkerson

If you write programs for the TI 99/4a, you are probably aware of how much n improvement TI Extended Basic is over console Basic. This extension of the esident interpreter permits the creation of faster, more efficient and more ersatile software. Here is a descussion of certain features of this language, ith some effective and efficient ways to use them, plus a few things that exas Instruments forgot to tell you. These tips, suggestions and examples xpand on the material in the Extended Basic manual. Certain topics covered lso apply to console Basic, as well as to Basic programming in general.

ENERAL GUIDELINES

Most techniques which improve a program's performance on other computers lso apply to the TI 99/4a. Logical structure,efficient algorithms, reduction f the number of program lines and variables, and concise coding are important o practice. There are, however, some differences that are helpfull to emember. Most importantly, Extended Basic hase many statements and subprograms hat have no equivalent in other common implementations of Basic. Some examples re ACCEPT, CHARSET, SUB, RPT$, PI, SPRITE AND COINC. These can sometimes eplace groups of statements or even entire subroutines. As a result you get aster operation as well as listings that are easier to understand. But emember, using these unique features may make it difficult to convert your rograms to the version of Basic used by other computers.

Another difference is the relationship between the location fo lines in a rogram ad the speed of their execution. Many popular computers exicute tatements faster when they are located mear the beginning of a program, but he 99/4a is fifferent. The fastest run times will be obtained if the routines hich are used most often are placed near the end.

ULTI LE STATEMENT LINES

One of the more obvious shortcomings of TI console Basic is that it allows nly one statement per program line. Extended Basic remedies this by allowing ultiple statements on any line, with the exception of IMAGE statements. The nly limitation is that the length of the condedsed program line does not xceed 163 bytes. The computer will give a warning if you try to exceed this imit, and the line will be rejected. But don't worry, it will not be lost if ou recall it with "REDO" (FCTN 8). Shorten the line and then re-enter it.

Multiple statements can be used as direct commands to give instructions to d extract information from the computer. In fact, a command can be the quivalet of a short program. For instance, the following command could be used o help debug a hypothetical program which uses a nemeric array named ARR(). irst, break into the running program using FCTN 4, then enter this line to rint out the contents of the array on a parallel interface printer. fterwards program execution can be continued:

```
PEN #5:"PIO" :: FOR ZZ=0 TO 50 :: PRINT #5:ZZ,ARR(ZZ) :: NEXT ZZ :: CLOSE #5
```

Similary, programming can be made much more efficient and visually logical ycombining related statements on a single line. The following wxample waits ntil the "N" or "Y" key is pressed, then branches to an appropriate ubroutine:

```
40 CALL KEY(3,K,S) :: IF K<>78 AND K<>89 THEN 440 ELSE ON SGN(K-78)+1 GOSUB
000,2000
```

The next example reads data and displays it sequentially on the screen. Some nes can become quite complex, so Extended Basic allows terminal remarks for ocumentation:

20 CALL CLEAR :: FOR I=1 TO 21 STEP 2 :: READ A$ :: IF A$="X" THEN 940 ELSE
ISPLAY AT(I,1):A$! DISPLAYS INSTRUCTIONS ON THE SCREEN.

Such programming techniques eliminate extra work when keying in programs,
nd result in software which executes faster and is more memory efficient.
emember, each program line takes up six bytes in the computer's RAM, in
ddition to the statements in the line itself. Multiple statement lines can
esult in significant savingas of memory space, ad when used properly they
roduce more easily deciphered and professional listings. But don't try to
tuff as many statements as possible into every line. You will end up with
istings that are crowed and confusing, as well as very fifficult to debug or
evise.

AKEN FROM THE CHARLOTTE TI 99 USER'S GROUP NEWSLETTER
==========

I FORTH PART 2 BY HANK ELLERMAN (TI FDL BBS)

Now that you have had some time to play with the system and maybe become
omfortable with FORTH, It's time to take the options you like to use and turn
hem into a fast loading BSAVE program, We will add a couple of words that I
se all the time, PAGE is nice to clear the screen and sets the curser at upper
eft. FREE is a good one, it will tell you how mush memory you have free. Maybe
y now you have tetined some of your own words.

Lets define the word PAGE:

: PAGE 0 0 GOTOXY CLS :

Just as you see it on the above line. Thats all there is to it. You have a
ew word.

Lets define the word FREE:

: FREE SP HERE -. ;

Now you can see the amount of free memory anytime.

Now we have to find a place on a scr for these new words. You can pick a
mpty acr or add them to scr#3 latter, because we have to update scr#3 any way.
he only option you need to put these words in the vocabulary is -SYNONYMS. So
oad -SYNONYMS and define these words and see how they work.

Lets do the job now. BSAVE. The first thing is to put the backup disk in
rive 1. Better yet make another copy of the backup disk, because we are going
o write on this disk and on some of the scrs that you may want to look at in
n future. Now we have a 3rd copy. The first is the original with the write
rotection tab on it. The second is the backup that has scr72 corrected "the
ug removed and set for pio". Now lets work on the third.

Put it in drive 1 and load FORTH. If you are already in FORTH, go to command
ode and type:

TEXT COLD

press enter

This will reboot the FORTH system to make sure we start from scratch.

Now we have the flashing curser under the word TI FORTH.

Lets load our options. The ones you like to use or these,type:

-GRAPH -DUMP -VDPMODES -COPY -PRINT -BSAVE -64SUPPORT

press enter

The -64SUPPORT maybe -EDITOR which one you like better. REMEMBER you can not
se both.

In my system I have loaded everything but -EDITOR -TRACE -ASSEMBLER -CRU
FLOAT,to save memory. They can be loaded after the BSAVE is in memory.

After the of is displayed and the curser is flashing again we are ready to
o our BSAVE type:

' TASK 51 BSAVE

press enter

(NOTE the first character in the line is the apostrophe)

: FORTH PART 2 CONTINUED

Now we have our fast loading program on disk. We just have to tell FORTH
ere it is at. type:
EMPTY-BUFFERS 3 EDIT
press enter

Now we have to change scr#3 to tell FORTH where the BSAVE is at and also put
a little more information. SCREEN#3 is the welcome scr. If you are using the
column editor you have line numbers, if -64SUPPORT you have no line numbers
d have to count them for yourself. This is what scr#3 should look like after
l changes.

```
0(WELCOME SCREEN) BASE->R HEX 10 SYSTEM (CLears Screen)
10 0 GOTOXY ."TI FORTH BOOTING" CR 10 83C2 C! ( QUIT off)
2DECIMAL 51 BLOAD 16 SYSTEM MENU
31 VDPMDE ! ( Tells FORTH you'r in TEXT mode)
40 DISK_LO ! (Allows EDIT/COPY on all SCREENS)
589 DISK_HI ! (Sets highest scr number)
689 DISK_SIZE ! (Set single sided disk)
7:FREE SP HERE-. : (Free memory)
8:PAGE 0 0 GOTOXY CLS : (Clear screen)
9 FREE
10
11
12
13
14
15R->BASE
```

Now don't get the line numbers mixed up with the text, and don't type in the
ne numbers, The line numbers are in column 1, they are just there for
ference.
Now hold down the function key and press 9. Back to the command mode. type:
FLUSH
press enter

Our new scr is written and we can try it out. type:
TEXT COLD
press enter

That was quick, we just loaded everything that you have been doing.
Try it again, this time turn the computer off and start from the master
reen. FAST ha?
Now all you need is the program "CLONE" and you can copy ANYTHING in ti
nguage. You can do it also with FORTH command "FORTH-COPY".

ST OF THIS INFORMATION I READ FROM MILLERS GRAPHICS, THE SMART
OGRAMER",1475 W. CYPRESS AVE. SAN DIEGO,CA. 92173 $12.00 A YEAR. IT IS A
NTHLY PUBLICATION, MAYBE.
==========

I-WRITER: Using the Special Character Mode by ABDALLAH CLARK
RIVERSIDE USERS GROUP)

The special character mode of the TI-WRITER software is a method of sending necessary commands to a printer in order to activate the many functions, such s different character fonts. Those who are familiar with using the Text ormater of TI-WRITER may already know of using the Tansliterate command to do ust this. With special character mode. the TL commands are not used. INstead, number of "Special" characters, other than the normal ASCII range of 32 space) to 127 (DEL),are generated and sent to the printer upon printout with ither the PF command or the Text Formatter.

We have all seen the symbol that represents a Carriage return installed at hend of a sentence or paragraph in our document, this is one of those SP-CH"'s. The ASCII value of a carriage return is 13, and to send a CR to a rinter you must send character number 13 (not the value 13).

To activate the special character mode, you hit a CTRL-U, and the cursor ymbol changes to an underline character. A second CRRL-U puts you back in ormal mode.

To install a SP-CH in your document, you must first know what function of ouprinter you wish to invoke. Most printer manuals have a chart that list the rious funtions and the codes needed to activate them, and it is very handy to ave a copy of this list nearby when formatting a document. For instance, the equence ESCcape-"E"(ASCII values 27 and 69) will invoke emphasized print on pson printers,and if we send an ASCI value of 27, then an ASCII value of 69, he printer switches to emphasezed print. We know that ASCII-69 is a capital E, ut ASCII-27 (ESCape) is not a "typeable" character. Now we go to the list of pecial Characters on page 146 of the TI-WRITER manual, and we see that SCII-27 can be generated by typing the FTCN-R while in the SP CH mode. The ymbols generated by each SP CH are also listed, and almost all are the exadecimal value of the ASCII code, compressed down to take up the space of nly one character, ASCII-27 is HEX-1B and you will see a little "1b", right fter the "ESC" character, you type the "E" for ASCII-69, In summary, the equence to send the control codes for Emphasized print would be:

TRL-U
FTCN-R)E
TRL-U

These special characters can be installed anywhere in the text, as they do ot print upon output, just as the "re-defined" characters used with the ransliterate command are "invisible."

There are pluses and miuinuses to using the special character mode against sng the transliterate command, The TL commands are more versatile, and can asily be made to send a complicated sequence of ASCII values, where using p-Ch mode would get quite tedious each time a lengthy code was sent, Also, a umber of TL commands can be stored in a separate file, ad linked to the ocument upon printing, thus saving having to re-write them each time. On the ther hand, when a relatively short code sequence is needed, SP-CH is much impler, and the biggest advantage is that you need not load and run the Text ormater, which can be a mojor obstacle to many.

As an example, when I want to just write a little note, and I want it in mphasized, I can simply start with a CTRL-U/FTCN-R/"E"/CTRL-U and when I use he PF command, I have a nice dark print. Another widely used area is when e-formatting paragraphs, such as when modifying margtgins, and you need to nstall a carriage-return symbol at the end. One way is to move the cursor to h point where you need the symbol, hit CTRL-8 (new paragraph) and then edit ut the extra line and spaces. a much simpler way is to just locate the cursor d hit the CTRL-U/"M"/CTRL-U, Which will generate a CR symbol.

MULTI-PLAN: SENDING PRINTER CONTROL

Have you ever wanted to gain control of your printer through Microsoft Multiplan? You know, send printer controls using spreadsheets created with Multiplan? Well, according to Curtis Ringold of the Mid-South Users Group it is possible, though it takes a little time. The following information is a condensed version by Walt Maes of Ringold's solution, which appeared in The Suncoast Beeper, the newsletter of the Suncoast 99'ers

Requirements are Multiplan and a program that allows single sector disk access. We'll do this in steps to minimize the confusion. It is recommended that the following be done using a newly initialized disk to make it easier.

STEP 1: GET INTO MULTI-PLAN

STEP 2: IN ROW 1,COLUMN 1,USING ALPHA, ENTER ABC

STEP 3: PRESS T FOR TRANSFER. PRESS O FOR OPTION, SELECT SYMBOLIC AND PRESS ENTER.

STEP 4: PRESS T THEN SAVE, ENTER A FILENAME AND PRESS ENTER. THE ABC SHEET HAS NOW BEEN SAVED.

STEP 5: LEAVE MULTIPLAN AND LOAD YOUR SINGLE-SECTOR DISK ACCESS PROGRAM. ASSUMING THAT YOU USED A NEWLY INITIALIZED DISK, USE THE PROGRAM'S EDITOR TO GO TO SECTOR 022. DEPENDING ON THE CAPABILITIES OF YOUR SECTOR ACCESS PROGRAM, YOU MAY SEE THE CHARACTERS ABC OR, IF IT DOESN'T DISPLAY ASCII,YOU WILL SEE HEX CODES FOR ABC, THE HEX NUMBERS REPRESENTING ABC SHOULD APPEAR AS 41,42 AND 43. REPLACE THESE NUMBERS WITH THE FOLLOWING: 0F,1B,47.

STEP 6: SAVE THESE CHANGES TO DISK.

STEP 7: REENTER MULTIPLAN. PRESS T, THEN SELECT SYMBOLIC AND ENTER. PRESS T, THEN LOAD THE FILE YOU SAVED. EXAMINE R1C1. INSTEADAD OF ABC YOU SHOULD SEE A CONTROL CHARACTER AND THE LETTER G.

STEP 8: PRESS T, THEN OPTION AND MOVE THE CURSOR TO NORMAL. PRESS ENTER.

STEP 9: RESAVE THIS FILE UNDER A NEW NAME. THIS CELL MAY NOW BE CALLED UP AND PLCED ANYPLACE IN A SPREADSHEET TO SEND A CONTROL TO THE PRINTER, IN THIS CASE A GEMINI. THE CONTROL HERE CREATED IS FOR CONDENSED, DOUBLE STRIKE.

STEP 10: TO INSERT THIS PRINTER CONTROL INTO A SPREADSHEET, TYPE X FOR EXTERNAL AND C FOR COPY. ENTER THE NAME OF THE SECOND FILE YOU CREATED CONTAINING THE CONTROL CHARACTER. PRESS CONTROL A. TYPE IN R1C1. PRESS CTRL A. TYPE IN THE CELL LOCATION YOU WANT THE CONTROL CHARACTER TO APPEAR IN THE SPREADSHEET. PRESS CTRLA. AT THE PROMPT FOR LINKED SELECT NO AND PRESS ENTER. THE COMMAND SHOULD NOW APPEAR IN THE WORKSHEET AT THE CELL LOCATION YOU ENTERED. IT IS NOW READY TO TAKE CONTROL WHEN THE SPREADSHEET IS SENT TO THE PRINTER.

THERE IS NO REASON WHY YOU SHOULD STOP WITH ONE PRINTER CONTROL. YOU CAN ENTER A SERIES OF PRINTER CONTROL COMMANDS AND INSERT THEM WHEREVER YOU LIKE IN YUR SPREADSHEET. YOU MAY DO THIS BY ENTERING A SERIES OF SINGLE-CELL REFERENCES INTO A SINGLE FILE THAT WOULD BE CALLED UP BASED ON CELL LOCATION OF THE COMMAND OR ; YOU MAY FILE THEM INDIVIDUALLY. REGARDLESS OF HOW YOU GO, YOU NOW HVE THE ABILITY TO TAKE FULL CONTROL OVER YOUR PRINTER WHILE USING MULTIPLAN, A VERY USEFUL ABILITY INDEED. TAKEN FROM MICROPENDIUM SEPT. 1985

ULTI-PLAN: WORD PROCESSING

Word processing with Multi-plan? why not? Multi-plan has many advantages over TI-WRITER and the Editor-Assembler Editor. For instance, Multiplan will allow you to format your document in a coulmnar 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 relativily slow processing speed.

Despite these drawbacks, however, for many applolications 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. Reamber, 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 intered on that line are numeric ones, this will prevent them form being right justified or otherwise screwed.

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 withort 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 nto columns, Since the maximum column width on the TI printer is 132, we will levide the text into 4 equal columns of 32 characters each ad 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 forth 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 legth.

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 lines long.

MULTI-PLAN: WORD PROCESSING CONTINUED

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:"PIO".CR"

```
PRINT #1:CHR$(15)
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 lynk 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.

TAKEN FROM THE RIVERSIDE USERS GROUP VOL.2 NUM.3

LEARNING TI ASSEMBLY PART 2

Well part 2 is finally here. I was busy doing update to the bbs than to write this article.

My articles are now being reprinted in the user group newsletter, any questions to me should be mailed to:

ARTICLES
C/O NICK IACOVELLI JR.
1411 N. 36th.
MELROSE PARK,IL. 60160

Or should be left on the TI-WEST BBS at 312-766-2797 for quicker response all questions will be printed and answered in the next article.

In the last article I told you about the utilities and how to put a character on the screen. This article will show you how to put more than 1 character on the screen move the character when the a key is hit and fill up the screen with that character (call hchar).

LAST ARTICLE SHOW THIS

```
    DEF START
    REF VSBW
START   LI R0,400
    LI R1,>4100
    BLWP @VSBW
```

this put a single character

LEARNING TI ASSEMBLY PART 2 CONTINUED

Now lits try filling up the screen with charcters,

```
     DEF START1
     REF VSBW
START1    LI R0,0
     LI R1,>4100
NEXT      BLWP @VSBW
     INC R0
     CI R0,768
     JNE NEXT
     END
```

The LI R0,0 set the screen location to 0.
The INC R0 adds 1 to the screen location so the next location I want to write to is 1.
The CI R0,768 check R0 for maximum screen location.(screen location is from 0 to 768)
The JNE NEXT will tell the machine to goto NEXT if R0 does not equal 768 and wil write the next character on the screen.
JNE means jump not equal.

WRITING MORE THAN 1 CHARACTER TO THE SCREEN

The new command is the TEXT command this will allow you to write a sentance and display it on the screen.

THE PROGRAM

```
   DEF START2
   REF VMBW
TI TEXT 'THIS WILL SHOW UP ON THE 'TEXT' SCREEN'
START2 LI R0,0      *SCREEN POSITION
   LI R1,TI  *LOCATION OF TEXT
   LI R2,34   *NUMBER OF CHARACTER
   BLWP @VMBW *DOES THE WRITE
   END
```

THE USE OF THE KEYBOARD

Just like the call key command in basic assembler works the same way with it BLWP @KSCAN command. In order to map the keyboard in basic a CALL KEY(#1,J,S) the #1 determines what keys are active(more information is in your TI Basic manual).In order to map the deyboard in assembler we must move a 1 to a location in memory. This location is >8374 the way to move to that location is:

```
LI R1,>0100
MOVB R1,@>8374
```

The LI put the number we want in the memory. The MOVB (which stand for MOVE BYTE) move the first numbers to the location in the second half or loc. 8374 in this case.

```
ETS MOVE ONE CHARACTER EVERY TIME WE HIT A KEY

  DEF START3
  REF VSBW,KSCAN
TART LI R0,0       *SCREEN POSITION
  LI R1,>4100   *CHAR 'A'
  BLWP @VSBW
ITKEY LI R1,>0400   *MAP THE KEYBOARD
  MOVB R1,@>8374   *IN BASIC MODE
  BLWP @KSCAN
  LI R1,>FF00
  CB @>8375,R1      *CHECK TO SEE IF KEY
  JEQ HITKEY        *WAS HIT NO GO BACK
  LI R1,>2000       *CHAR ' ' A SPACE
  BLWP @VSBW        *WRITE TO SCREEN
  INC R0            *ADD 1 TO SCREEN POS
  CI R0,768 *LAST POSITION JEQ START3 *YES START OVER
  LI R1,>4100       *NO WRITE TO NEW POS
  BLWP @VSBW
  JMP HITKEY        *GOTO HITKEY
  END
```

e 'C','CB','CI' are compare commands this command is use with the
IP,JNE,JEQ,JGT,JLT command, think of these commands as the IF THEN ELSE
atement in basic.

AMPLE
BASIC IT SHOWS LIKE THIS
REM
 IF A=1 THEN 5 ELSE 11

 ASSEMBLY
P
 C @A,@D1
 JEQ TOP
 A and D1 would be labels to a data statement in assembly put in the assembly
ke this.

)ATA >0001
 DATA >0001

 You could change A or D1 by adding to it,subrtacting from it or bye moving
 it. I will continue this in my next article and explain most of the
structions for moving data.

==========

A GUIDE TO THE p-[illegible]
ON THE TI 99/4a

[Note: this is the first draft of this cument...please make comments via MUSUS, the TI Forum or AIL (my # is 72406,2754), or US Mail, or via TI-Writer sks. My address is:

1540 Florida Ave. #205
Modesto, Ca 95350

I especially request help and comments at the places ere I have placed percent signs (ZZZ)]

Introduction

When TI discontinued the machine in October, 1893, few the over two million 99/4 owners were using the p-System; all 99/4 owners felt like orphans, we p-System users were ubly so. But however isolated we seem there are vantages: the p-System on the 99/4a is a full plementation of a "serious" operating system and as such are able to do much more with it than TI-BASIC. Also, we n get help from any p-System user, regardless of their rdware. Finally, when the day comes when our machines ther die or are outgrown, our programs will be portable to rtually any other machine on the market.

The purpose of this guide is to orient the TI 99/4a er to the p-System in general, and point out the special nsiderations about its implementation on our machine. It hoped that this guide will be of value to all who are terested, from the serious programmer to those who are nsidering purchasing the p-System.

Overview of the p-System

The p-System is an operating system -- a set of ograms to run the computer and allow us to write and run ograms. It is not a language; any language might be ailable under the p-System. While only PASCAL is rrently available to us, other users of the p-System can e BASIC, FORTRAN, and MODULA-2.

This concept is confusing to many TI 99/4 users because Console and Extended BASIC have the operating system ilt in -- they include text editors, disk management ograms, etc -- all in the ROM containing BASIC. In the System one adresses all these functions separately, and st are on disk.

The essential part of the p-System for the TI99/4a is e p-Code card (or the old p-Code peripheral). The p-Cod rd contains some of the system programs in ROM. You must so have the 32K memory expansion. There is currently no y to use the extra 96K RAM available in the 128K card nufactured by Foundation. A disk is not absolutely essary, as programs can be loaded and run from cassette. wever, I do not know of any cassette programs, nor do I ow of anyone who has used a cassette with the p-System.

The p-Code card includes, in ROM, the p-Code terpreter. This program accepts the user's and system ograms in p-Code and executes them by interpreting them to 9900 machine instructions.

Anyone serious about the p-System will have to have at ast one and preferably two disk drives. The p-System will pport double sided disk drives. If you is going to make e of the p-System for writing programs, then you will find it in a single drive system that your drive had better be uble sided; two drives are better and two double sided rives are best. I see little use for a third drive. ZZZ is anyone know if the Corcomp double density controller lows double density disks on the p-System (360 single ded, 720blocks double sided)?ZZZ

[illegible] experiences with other [illegible] peripherals, etc -- by 3rd party manufacturers?ZZZ

There are several programs that are important parts of the p-System. One is the PASCAL compiler, the program that takes PASCAL programs and translates them to p-Code, so the p-Code card can run them. This program comes on disk and is necessary to write your own programs but not to run someone else' programs.

Another disk contains two important programs: the Editor and Filer. The Editor is used to edit documents and programs; you only need this to write your own programs although it can even, to some extent, be used for word processing. The Filer manages the disks; virtually any user of the p-System needs this program. Also on this disk are a number of utility programs which are of much value to most users.

The last disk contains the Assembler and Linker, both of which are necessary to do assembly language programming.

Learning to use the p-System

The TI manuals that come with the system components are pretty good as manuals go, but are not designed to teach the basic use of the system. There are many books about the U.C.S.D. p-System and PASCAL. Two very useful books are: "Introduction to the UCSD p-System" by Grant and Butah (Sybex), and "The UCSD PASCAL Handbook" by Clark and Koehler (Reward Books). The former is an excellent introduction to the use of the p-System. This guide will describe the unique aspects of our implementation by comparing the TI 99/4a system to that described in this book. The latter book is a relatively technical guide to UCSD PASCAL with very good examples. There are a number of good books that teach PASCAL; a discussion of them is beyond the scope of this guide.

MUSUS (the U.C.S.D. p-System users' Group) is a good source of help in the use of the system. Many of its members can easily be reached via the MUSUS SIG on Compuserve (PCS-55). The beauty of the p-System is that it, at least in theory, acts exactly the same on all systems; thus, you can get help from any knowledgeable p-System user; MUSUS is a good place to find such people.

The Filer

The Filer works pretty much as described by Grant and Butah. The major exceptions are that some of the prompts are shorter and the TI 99/4 only displays 40 columns at once. To see the whole 80 column display one uses the FCTN-7 and FCTN-8 keys (screen left and right).

It should be noted that disks may be physically in the drives but cannot be addressed by their volume names unless they were present at startup of the p-System, or reinitialization (I command) or by invoking the V (Volumes) command of the Filer. This command doesn't just list the on-line diskettes, but formally recognizes them so that they can be accessed by volume name in user programs. If you want to use the FILER but didn't have it in a disk drive at start up, then you have a problem. You cannot invoke the FILER V command to get the system to recognize the disk - because you can't run the FILER. The solution to this catch-22 is to use the system I command and restart the system. On restart, the p-System looks anew at the disk drives, and will then recognize the diskettes loaded in them, and know where to find the FILER, EDITOR, etc.

The Z (zero) command sets up a new (or zeros an old disk) after it has been formatted by the DFORMAT utility program (or the Disk Manager Cartridge). Single sided disks have 180 (512 byte) blocks, double sided disks are twice that size. Using duplicate directories consumes 4 extra blocks but will often avert disaster when disk failure

The Editor

The Editor for the 99/4 is that which Grant and Butah all the "Screen Oriented Editor" — the Editor that is iscussed at length in their book. In general the Editor orks just like that of Grant and Butah with a few ifferences that could probably be best described as bugs. he "S" (Same) option on the "F" (Find) command does not ork. The copy buffer seems to be emptied at times for easons that are not obvious. After repeated insertions and eletions of a given page on screen, often the screen image s not correct. The actual text is different. As a result, t is best to use the "V" (Verify) command when in doubt. t re-displays the current screen page.

The 99/4a's small RAM size allows only a little more han 12k characters in the text buffer. If you need to reak down a larger file the AUTOPSY program in the USUS ibrary can be of help. Programs can be written in several art's and strung together with the PASCAL compiler $I (include file) command.

Compiler

The TI 99/4a PASCAL compiler is a full Version IV.0 mplementation. It seems to be identical to the compiler escribed by Clark and Koehler. In its TI version, the ompiler appears to be in four parts: 1) the start-up code; 2) the code to process the Declarations, 3) the code to rocess the body of procedures, and 4) the code to produce -Codes. Parts 2 and 3 are swaped in and out for each rocedure in the program, making compilation a good time or a coffee break. The swap parameters described by Grant nd Butah seem to be inoperative on the TI; I suspect that he compiler is set for maximum swapping to accomidate the I's limited RAM. Making a listing causes the already slow ompiler to run much slower still , so one should be sparing n using this function. Also, if errors are found by the ompiler it may not make a listing, even if the user keeps ontinuing compilation to the end. It may even hang with he error:

STACK OVERFLOW*REBOOT

At this point, one must turn the console off and on gain and to restart.

The TI 99/4 version of PASCAL includes a number of xtensions that allow the use of the special capabilities of he machine. These extensions are in the form of procedures ncluded in the "TEXAS INSTRUMENTS UNITS", which are in the ystem LIBRARY. These inclue the routines to change screen olor, use graphic modes (bit map, multi-color, etc), prites, sound and speech. The use of these routines will llow the sophisticated user to do in a high level language hat otherwise was previously only possible in Assembly. he new TI FORTH posesses these features as well. Please ote, however, that the use of these procedures will make he program machine dependent, and thus not portable. One annot, for example expect a program using sprites to run roperly on an APPLE, because that machine lacks the ardware to produce them. To keep a program portable, one hould avoid using the facilities in the TEXAS INSTRUMENTS NITS.

Among the routines included in these UNITS, I have only sed the procedures for random number generation and screen olor definition. These procedures work as specified. I elcome input from others who have used the other acilities.

For program development, it is desirable to be able to uickly go from compiler to editor, so it is good to have oth programs on one disk or both on different disks on ifferent drives (in a multi-drive system). Also, put the ile SYSTEM.PASCAL on the BOOT volume so that meaningful rror messages are displayed.

Utilities

There are a number of valuable programs included with the Editor and Filer disk. This guide will discuss the most useful of these programs.

DFORMAT formats new disks (or reformats old ones). It serves the same function as the Disk Manager module in this function. It can format double sided disks but usually fails (I was told by TI that they knew of this bug) but works fine on single sided disks. For double sided disks, one can always use the Disk Manager 2 module if available. The Zero command in the FILER must still be used in either case before a disk is usable.

MARKDUPDIR and COPYDUPDIR are useful in dealing with duplicate directories on disk, whose use is highly recomended.

MODRS232 alters the device definition associated with REMIN: and REMOUT: (these use the same definition) and PRINTER:. TI has kindly included the source code for these programs. If you use a printer whose definition is different from the default (RS232/2.BA=9600.DA=7.PA=0.EC — this probably isn't what you're using), you can alter the source code to make the printer definition without user prompting and run it as SYSTEM.STARTUP. The rules for describing the RS232 or PIO communication setup are the same as used by BASIC.

RECOVER is useful if an both directories are blown (or you foolishly used only single directories). It can help restore lost files. PATCH will allow repair of individual files. You have to be a very expert user to use PATCH, however.

SETLTYPE allows one to force a program to reside in RAM rather than VDPRAM. Its stated purpose is to declare whether a program is in p-Codes or native code (machine language). Machine language programs must reside in RAM; p-Code programs may be in RAM or VDPRAM as the p-System selects (usually in VDPRAM). The system views both VDPRAM and RAM as one large space to be filled with programs and data. It will store data down from the high end of RAM and programs up from the low end of VDPRAM until RAM or VDPRAM is filled; then it will overflow into the other space (i.e. programs will overflow in RAM or data into VDPRAM). One can use SETLTYPE to force a program or segment to be in RAM. If this is done it will run somewhat faster, but may impinge on (and thereby limit) the area for data. As mentioned above, assembly language programs must be put in RAM with SETLTYPE.

Using the p-System

Several notes can be made about using the p-System on the TI99/4a. First, you cannot use the TI LOGO cartridge on a system with the p-Code card in place. The p-System will work fine; LOGO will, in certain circumstances, loose control to the p-Code card, destroying what you were doing with LOGO. If your p-Code card has an on/off switch, all will be well if you turn the card off before using LOGO.

When a 99/4 is powered up with the p-Code card is in place (and turned on) the p-System initializes and then displays the p-System greeting. The p-System remains active until the user stops it; thereafter it will not restart without using the technique below (or turning off the system). To get from the the p-System to the BASIC environment is simple - use the p-System 'H' command. To go from BASIC to the p-System, you need to have one of the following cartridges in place on the console:

1. Extended BASIC
2. Editor/Assembler
3. Mini Memory

and then type reset (quit) of the p-System will then operate as if you had turned the system off and then on again. Some programs and cartridges seem to have the same effect as the CALL LOAD listed above. Exiting TI-WRITER, MULTIPLAN, or the COMPANION word processor will lead to restart of the p-System. Likewise, some of the commands in the DISK MANAGER will reset the p-System so that it will become active again upon leaving this module.

Speed is an issue of interest to all potential users of the p-System. If computing speed is the reason for the TI BASIC programmer to consider the p-System, the following facts apply: Because the p-System is interpreted, it is relatively slow compared to compiled PASCALs (such as TURBO PASCAL on MsDOS and CP/M systems). Compared to other machines, the p-System on the 99/4 is quite slow - I believe it is the slowest implementation of the p-System around. TI BASIC, however, can be benchmarked with a sundial, so it is easy for the p-system to beat. Number crunching programs PASCAL can run up to 60 times faster than their BASIC counterparts, especially if put in RAM (see SETLTYPE). Programs doing mostly screen I/O can be slower than BASIC, however. Disk I/O seems a little faster with the p-System -- perhaps because of the simpler file system, leading to less searches for the next block, etc. I have no experience with FORTH, but know that it too is faster than BASIC. The programmer desiring speed above all else should compare the p-System to FORTH.

On running the V command in the FILER program, you will notice that there is a device #14 called 'OS:'. If you do not have a disk in drive one (the usual root device), one will find that the 'PREFIX' and 'ROOT' devices are defined as 'OS'. This device includes files that are in ROM and are used for booting the system. You can read and examine these files but not, of course, modify them. To change the PREFIX definition, use the FILER P command; to change the ROOT definition you must use the system I (warm start) command.

Compatibility with other Machines

The p-System on the 99/4 is very compatible with other p-System implementations. Source code developed on the 99/4 can and does run without modification on most other machines if you avoid using the TEXAS INSTRUMENTS UNITS included with the PASCAL Compiler. The 99/4 is less able to run programs from other machines due to it's relatively small Ram size. If the program is properly segmented, however, there will be no problem. The PASCAL compiler, for instance, is 99 blocks long -- 49.5K. It runs on the 99/4 (with a lot of disk swapping). The only other problem in running program from other machines on the TI is the presence of a 40 (rather than 80) column display. The TI will display all 80 columns with the SCREEN RIGHT and LEFT keys, but you will probably want to modify the display (if possible) to fit the screen. In the file SCREENOPS.CODE on device #14 (see above) is a description of the TI screen size. This file can be read by a program. This file tells the program in the case of the TI that the screen is 40 columns. Clever programs will adapt themselves to the size of the screen.

A larger problem is physically transferring programs to and from the 99/4a. The TI p-System lacks any functional communications program at this time. Files can be exported from the 99/4a easily by using the FILER to Transfer to REMOUT:.

There is good but not complete compatibility with other PASCALs. I have programs that I wrote on the TI p-System that I ported to an MSDOS (IBM PC or compatible) machines and converted to TURBO PASCAL. Little modification was necessary. ZZZanybody have experience with other pascals?ZZZZZ

Software for the TI p-System

I know of no commercial software for the TI p-System. I suspect that if enough interest could be shown, then the

[illegible] members for a small fee - [illegible] most programs have not been modified to run on the TI. If you modify a USUS program to run on the 99/4, you are requested to send a copy back to the USUS TI distributor.

Wish List

The p-System on the TI 99/4a is powerful but remains largely undeveloped. What follows are my sugestions for software development which would greatly aid the use of the system.

FORTRAN: could someone port this compiler to the TI? There would surely be great interest in this language on the 99/4. Likewise, many would welcome MODULA-2.

Terminal Emulation: There have been programmers who have had TE programs "almost working". This is probably the highest priority for the p-System. A TE program would allow porting of other programs to the TI without re-keying the source code.

A good Data Base: there are no high quality data base programs available for the 99 in any form. Several excellent programs are available on the p-System. Could one be ported to the 99?

HELP!

The information in this guide is essentially all from my own meager experiences. Like most p-System users on the 99/4a, I doubt that I have ever laid eyes on anther user. This guide has been written, therefore, in a vacuum. Its purpose is to catalyze discussion, debate, and the creation of a better guide to the TI p-System.

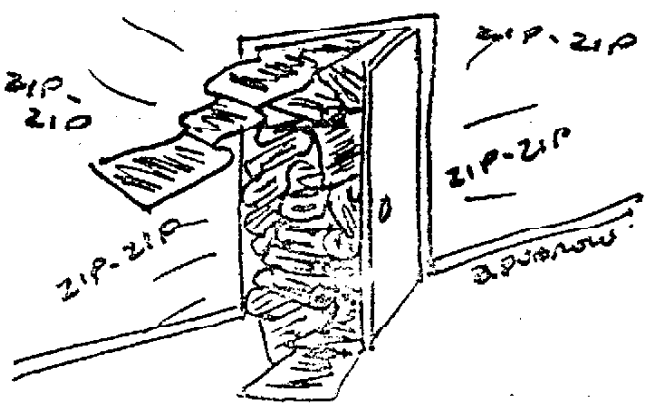
To improve on this guide, I need the experiences of other users, both the sophisticated and the novice.

This guide could be improved with the following information: Is there anybody out there who can help with machine language programming on the system? How about how to change character sets? Anybody have any suggestions or other tricks for use of the p-System? I solict your comments, suggestions, and criticisms. If you feel the urge, trash this guide and write your own. We all will benefit by the interchange of information and ideas.

_____By Steve Jacobs

D/L from CompuServe by Mel Gary

HEY MOM, THIS IS GREAT!!!!
WITH DAD GONE TWO WEEKS
HE'S SURE GOING TO BE SURPRIZED



A Review By Scott Anderson

Like father like son, while he designed shoes for Genesco, I was taking every art class availible in school. So it wasn't unusal that the first thing I wanted to do when I got a computer was to draw pictures. I soon found that the coding for char string could be very time consuming. A map of the United States took 32 hours to redefine. Right away I began to think that there had to be a better way. Now after waiting three years Graphx seems to be that better way. This program allows you to easily draw, save, and print bit map graphics. Clear instructions on every screen and menues make Graphx easy to use. Some of the important features are:

Adjustable Cursor Speed

Zoom - it allows you to enlarge a portion of the drawing area so that detail work can be done.

Automatic Lines - in this mode the joystick will control the movement of a line until it is positioned in the desired location and then will draw it.

Automatic Circle - the same as Automatic Lines but with circles.

Copy / Move - this allows a desired section of the screen to be moved or copied at any location.

Typewriter Mode - this function lets combinations of words and numbers be used on screen with drawings.

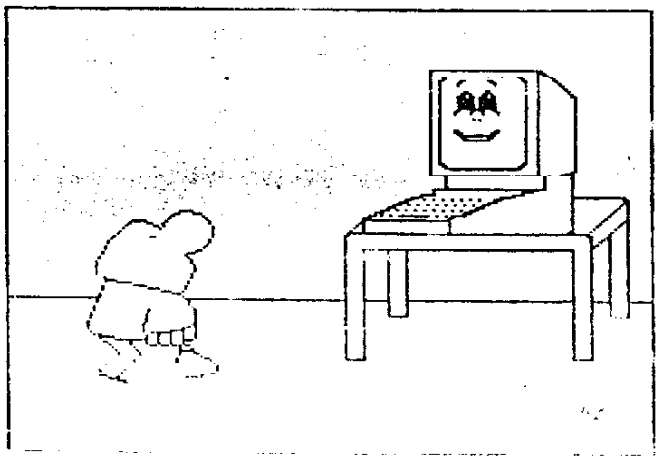
Clipboard - this will run a series of screens to produce animation.

Fill - this will color in any object without spilling out of a group of enclosed lines.

Read / Save - disc options.

Printer - any screen can be printed single or double sized.

All of these functions are accomplished by a combination of joystick and eleven keys on the console. After looking at many different graphic programs this one lacks two options that would be very useful, an automatic square and rotate function. Below is an examples of Graphx's capabilities.



his hard earned dollars wherever he can find the program with the utility he requires at the price he can afford. Quality will always have the edge over inferior products and if the value is there, its price will be paid.

Roger Wilkerson

FREEWARE

From

TOPICS - LA 99ers

1. DM1000 Bruce Caron ... POB 460 Route 9, Florence, AL 35620. A marvelous disk-based Disk Manager which rivals CorComp's manager.
2. MASSCOPY Steve Lawless ... 2514 Maple Avenue, Wilmington, Delaware 19808. EXCELLENT disk cloner; features ability to copy to 2 drives at once and uses the Foundation 128K card to copy a disk in ONE PASS!
3. X_DISASM Fred Hawkins ... 1020 North 6th Street, Allentown, PA. 18102. An XB disassembler with many unique features and terrific documentation for those that PAY!
4. SUPER DISK DUPLICATOR Tom Knight ... 7266 Bunion Drive, Jacksonville, FL. 32222. Allows inputting start and stop sector number for copying disks.
5. TK WRITER Tom Knight (See Above) Loads TI WRITER from XB or E/A. No cartridge needed!
6. NEATLIST Danny Michaels ... Route 9, Box 460, Florence, AL. 35630. XB utility to list multi-statement lines to printer or disk for easy reading and references program variables to line number used.
7. SCREENDUMP Danny Michaels (See Above). Screen dump to EPSON compatible printer with double or single size and verticle or horizontal page printout.
8. The DIRECTOR Ron Rutledge ... 1020 3rd Street, Waukee, IA 50363. XB program database that allows cataloging disk-based programs.
9. FAST TERM Paul Charlton ... 1110 Pinehurst Court, Charlottesville, VA 22901. Simply, THE BEST TERMINAL EMULATOR IN THE WORLD!
10. SPRITE BUILDER John Taylor ... 2170 Estaline Drive, Florence, AL. 35630. XB graphics generating program with assembly language routines for speed at crucial places. Includes a full disk of preformed graphics.
11. PILOT 99 Thomas Weithofer ... 1000 Harbury Drive, Cincinnati, OH 45220. An ENTIRE lanugage for the TI that is the simplest programming language known to us (or anyone else!).
12. MASTER CATALOG Mack McCormick ... 215 A Yorktown, Ft. Lee, Virginia 23801. A 100% asembly language disk catalog program that is super fast; handles up to 2000 different disk files.
13. EASYSPRITE Tom Freeman ... 515 Alma Real Dr., Pacific Palisades, CA 90272. An extremely fast XB program with assembly routines to create graphics sprites with easy cursor control saving for program insertion.
14. DISASSEMBLER Marty Kroll ... 218 Kaplan Avenue, Pittsburg, PA 15227. Superfast disassembler, 100% assembly and full featured.
15. TECHIE BBS Monty Schmidt ... 121 N. Blair, Madison, WI. 53703. Freeware BBS system for the 99/4A.
16. COMPACTOR Monty Schmidt (See Above). Assembly language program that takes an uncompressed D/F80 AL program and will compress to about 2/3 the disk space and yield faster load times.
17. UNCOMPACTOR Monty Schmidt (See Above). Opposite of above.
18. PRO 99er BBS Mark Hoogendoorne ... 21 Long Street, Burlington, MA 01803. TI BBS system with TRUE TE2 transfer capabilities.
19. DISK MANAGER Todd Kaplan ... 5802 N. Western Apt. 3S, Chicago, IL. 60659 INCREDIBLE Disk Manager on disk; forget TI's DM2.

GRAM KRACKER™

The Gram Kracker is a 4 inch by 6.5 inch by 1 inch unit that plugs into the Module port on the 99/4A and includes a port to plug modules into. It comes configured as a 56K unit but it is user expandable to 80K. It also includes an additional 8K of Preprogramming that displays the following menu:

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

1. Load Module: Loads any module or Basic Program saved to Disk or Cassette with selection 2 (Save Module). This selection will also allow you to load any memory image file that was generated with the Editor Assembler Save utility. (Like UTIL1 or the RUN PROGRAM FILE loaders). The module loading and UTIL1 type file loading can automatically be chained together with a built in Gram Kracker option. This means that you can load All of the Module space (56K) plus all of Memory Expansion (32K) with this loader by typing in 1 file name!

2. Save Module: Saves the contents of any module plugged into the Gram Kracker module port to Disk or Cassette. This selection will also save the contents of the Gram Kracker's Ram and Gram or YOUR Basic Program that has been set up in Gram with our utility. To save a module, simply plug it into the Module port, select 2 and type in the filename to save it to. If you are saving it to disk, the Gram Kracker will automatically check the diskette to make sure there enough is free space on it for the module.

3. Init Module Space: Clears out the Gram Kracker's Module Ram and Gram.

4. Load/Save Console: Pages in the following menu:

1. Load Console
2. Save Console
3. Grom/Gram 0
4. Grom/Gram 1
5. Grom/Gram 2

Which allows you to load and save the Optional Console Grams, that can be switched in and out to override Groms 0, 1 & 2, which are built into the console.

5. Edit Memory: Brings up a full screen memory editor similar to the Explorer's editor. This editor also allows you to Move Blocks of memory, from anywhere to anywhere, Fill any block of memory with a designated byte or dump any block of memory to a selected output device in Hex, AscII and AscII with the Basic Bias. (Please note: The Memory Editor requires Expansion Memory)

Besides the Built In software the Gram Kracker also comes with a diskette that includes the following:

1. A utility that allows you to write BASIC programs that can be stored and executed in the Module space. This means that all of Vdp Ram is left free for strings and numeric variables. It also means that YOUR Basic Program will appear on the normal TI menu as a selection and it can be directly executed with the press of a key!!!
2. A utility to add some new CALLs to Extended Basic.
3. A utility to move the Editor/Assembler and TI-Writer programs to different GRAMs to allow you to load more than 1 module at a time into the Gram Kracker.
4. A utility that will modify the Editor Assembler in the Gram Kracker so that it functions as a GRAM DISK for the EDIT1, ASSM1 and ASSM2 files. Then, when you select Edit or Assembler they are instantly loaded into Memory Expansion. No more waiting for them to load from Drive 1. No need to keep these files on your work disks since they are stored in the Gram Kracker!!
5. A utility that will modify the TI Writer in the Gram Kracker so that it functions as a GRAM DISK for the EDITA1, EDITA2, FORMA1, FORMA2 and CHARA1 (optional) files. Then when you select Editor or Formatter they are instantly loaded into Memory Expansion. No more waiting for them to load from drive 1. No need to keep these files on your work disks since they are loaded from the Gram Kracker!!

NOTE: With the 3 Optional Console Gram chips installed you can:

1. Override the operating system with the flip of a switch and make modifications to it, such as:
   Install a character set with true ascenders and descenders and a slashed zero. Put your name on the title screen. Change the title screen colors. Make a V2.2 console a non 2.2 console. And much more!
2. Override TI Basic's space and put something else there. This will allow you to have a menu with Disk Manager II, TI Extended Basic and either TI-Writer or Editor Assembler on it. Or TI-Writer, Editor/Assembler, Extended Basic and still have 8K of Gram free for an additional module or for additional enhancements to Extended Basic
   (Please note: To Load and Save software to the Optional Console Gram chips you must have Memory Expansion and a Disk System.)

Around the first of the year Millers Graphics will be releasing a GPL Assembler, GPL Disassembler and a GPL Programmers Guide to allow you to program in GPL Code (which is easier than Assembly Language but you still have full access to the entire computer and its peripherals!!)

This is the peripheral that TI should have released to show the REAL POTENTIAL of the 99/4A Computer!! Expected release date is around Mid November. Minimum system requirements are a Console, Monitor or TV and a Cassette Recorder (Memory Expansion and Disk System are optional except as noted). The Gram Kracker and the Utility Disk are fully compatible with any of the Disk Controllers and Peripheral Expansion Systems and Memory Expansions currently available. The price is 174.95 plus 4.00 for shipping and handling for U.S. and Canada. Order number GK01.

For Additional information please write:

Millers Graphics
1475 W. Cypress Ave.
San Dimas CA 91773